

Article

# Tailored Quantum Alternating Operator Ansatzes for Circuit Fault Diagnostics

Hannes Leipold<sup>1,2,3,4,\*</sup> , Federico M. Spedalieri<sup>1,5</sup> and Eleanor Rieffel<sup>3</sup><sup>1</sup> Information Sciences Institute, University of Southern California, Marina del Rey, CA 90292, USA<sup>2</sup> Department of Computer Science, University of Southern California, Los Angeles, CA 90089, USA<sup>3</sup> Quantum Artificial Intelligence Laboratory (QuAIL), NASA Ames Research Center, Moffett Field, CA 94035, USA<sup>4</sup> USRA Research Institute for Advanced Computer Science (RIACS), Mountain View, CA 94043, USA<sup>5</sup> Department of Electrical and Computer Engineering, University of Southern California, Los Angeles, CA 90089, USA

\* Correspondence: leipold@usc.edu

**Abstract:** The quantum alternating operator ansatz (QAOA) and constrained quantum annealing (CQA) restrict the evolution of a quantum system to remain in a constrained space, often with a dimension much smaller than the whole Hilbert space. A natural question when using quantum annealing or a QAOA protocol to solve an optimization problem is to select an initial state for the wavefunction and what operators to use to evolve it into a solution state. In this work, we construct several ansatzes tailored to solve the combinatorial circuit fault diagnostic (CCFD) problem in different subspaces related to the structure of the problem, including superpolynomially smaller subspaces than the whole Hilbert space. We introduce a family of dense and highly connected circuits that include small instances but can be scaled to larger sizes as a useful collection of circuits for comparing different quantum algorithms. We compare the different ansatzes on instances randomly generated from this family under different parameter selection methods. The results support that ansatzes more closely tailored to exploiting the structure of the underlying optimization problems can have better performance than more generic ansatzes.

**Keywords:** quantum algorithms; quantum computation; combinatorial optimization; circuit fault diagnostics



**Citation:** Leipold, H.; Spedalieri, F.M.; Rieffel, E. Tailored Quantum Alternating Operator Ansatzes for Circuit Fault Diagnostics. *Algorithms* **2022**, *15*, 356. <https://doi.org/10.3390/a15100356>

Academic Editors: Ying Bi, Mengjie Zhang, Bing Xue and Bo Peng

Received: 4 August 2022

Accepted: 24 September 2022

Published: 28 September 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

We introduce and characterize several different quantum alternating operator ansatz (QAOA) [1] approaches to solving the combinatorial circuit fault diagnostic (CCFD) problem, a combinatorial optimization problem of importance for diagnosing faults in circuits. In particular, we identify different constrained spaces of interest and explore various constructions of mixing and cost operators that allow us to evolve within each constrained subspace, allowing the protocol to focus on bringing the support of the system's wavefunction to an optimal or approximately optimal state within this subspace that can be significantly smaller than the space of the physical qubits needed to run the system. There are many such relevant subspaces, and through a different selection of mixing and phase separation operators, even the same subspaces can be explored in very different ways.

After describing QAOAs for optimization problems, we delineate the stuck-at-fault model of CCFD problems considered in this paper and how they can be cast as optimization problems. We then develop several different ansatzes for solving this optimization problem, beginning with the least constrained (measured as the resulting size of the constrained space maintained) and ending with the most constrained. The most constrained ansatz has a superpolynomially smaller constrained space. For circuits with a logarithmic or lower minimum fault explanation, it can be modified to have a constrained space that

grows subexponentially in the number of wires while the most generic approach grows exponentially.

The methods used for constructing ansätze in this domain can be useful for constructing more tailored approaches for other domains, which is an interesting avenue for future quantum algorithms. The results from running each ansatz under several parameter selections indicate that ansätze more closely tailored to a problem can have better performance than more generic ansätze. The relative success of the simple parameter selection methods used in this manuscript are of interest to researchers focusing on QAOAs for other domains.

For many classes of combinatorial optimization problems that are NP-hard, the features of specific instances can make them, in practice, more or less accessible to different algorithms. The features of typical CCFD instances, such as a planar-like topology that is often amendable to embedding on local two-dimensional lattice quantum architectures, make them a strong candidate for quantum algorithms utilized on NISQ devices. Given the real-life applications of CCFD problems, they have become problems of interest for leveraging the potential quantum advantage for industrial purposes [2]. Our results in this manuscript support that tailored ansätze should be explored on NISQ quantum computers to solve CCFD problems. The family of scalable, dense, and highly connected circuits we introduce is a useful testbed for such devices.

Even between different QAOA algorithms, the features of particular instances can affect which approaches are more or less promising. For example, we consider an ansatz that is similar to a recently studied CQA approach [3] that shows promise in the regime where an instance has higher degeneracy in the solution space and higher minimum fault explanations, while another approach maintains a subexponential constrained space when there is an upper bound on the minimum fault explanation. This suggests that the scaling of the minimum fault explanation for the instances of interest is important for determining the best ansatz to utilize.

However, there are still more considerations for implementing these approaches on quantum systems, including the preparation of the initial state and considering the best implementation of the required unitaries through a specific gate set. Certainly, more than the size of the constraint is important for determining the best ansatz for a problem, and it remains an open challenge to identify what specific features of the mixing operators and phase-separating operators would lead to the best performance.

In Section 2, we introduce a QAOA for combinatorial optimization problems and develop the principles around constrained quantum evolution within this context. In Section 3, we describe the single-pair input/output circuit fault diagnostic problem over the stuck-at fault (SAF) model, where circuit wires are either healthy or permanently stuck at one or zero. In Section 4, we delineate different QAOA protocols, with different resulting constrained spaces. In Section 5, we introduce a family of circuit instances to explore the suitability of our QAOA approaches for solving this problem for small sizes. In Section 6, we benchmark the approaches on the distribution of small-sized circuit instances introduced in Section 5. Our results support that tailored ansätze can be beneficial for obtaining better performance in optimization problems such as CCFD problems, where there is structure in the underlying problem that can be exploited.

## 2. Constrained Evolution in Quantum Alternating Operator Ansätze

For combinatorial optimization problems, there have been recent advances in the research of applying quantum algorithms to solve such problems that are centered around the quantum approximate optimization algorithm (QAOA1) [4], which is a very general protocol to approximately (or exactly) solve combinatorial optimization problems, such as the Max Cut problem [4–7]. However, further generalizations of this concept, such as the RQAOA [8] and quantum alternating operator ansatz (QAOA) [1,5,9], have been developed in the hope to better tailor protocols to particular problems and thereby exploit their specific structures. For the purpose of this paper, we focus on using QAOA protocols to find optimal solutions to a combinatorial optimization problem. For example, the au-

thors of [5] demonstrated a near-optimal quantum unstructured search algorithm based on using coherent phase-separation operators in the QAOA setting. Under reasonable complexity-theoretic assumptions, the QAOA cannot be efficiently simulated by any classical computer [10]. For a small number of qubits, this has been realized on an NISQ [11] device [12].

Given a general binary optimization problem with  $N$  bits, the quantum computer works with  $N$  qubits over a space  $\mathbb{C}^{2^N}$ . The wavefunction of the system  $|\psi\rangle$  is a normalized, complex-valued vector in this space, and potential actions on this state (i.e., algorithmic steps) are represented by unitary operators.

The QAOA works by starting the quantum system in a prepared state and then applying a series of angle-parameterized unitaries to evolve the system into a state that is then measured over a predetermined basis. The result can then be interpreted as a bit-string solution to the optimization problem. The QAOA divides the task of evolving the system into  $P$  rounds, each involving the application of two sets of operators. The first is a set of mixing operators, which are discussed in greater detail later. The second is a diagonal unitary operator associated with a classical cost function that acts to essentially evaluate the quality of different configurations.

Given a classical cost function  $C(x)$ , in both the QAOA and QAOA1, Hamiltonian form phase separation operators are used such that  $U_c(\alpha) = \sum_{x \in \{0,1\}^N} e^{i\alpha C(x)} |x\rangle\langle x|$ , where  $|x\rangle$  are the orthonormal computational basis states in  $\mathbb{C}^{2^N}$ . Any such state  $|x\rangle$  (a ket-state) can be written as an outerproduct state of individual qubit states  $|x\rangle = |x_1, \dots, x_N\rangle = |x_1\rangle \dots |x_N\rangle$  with  $x \in \{0,1\}^N$ ,  $|x_i\rangle \in \mathbb{C}^2$  and  $x_i \in \{0,1\}$ . As such, the classical cost function is interpreted as a classical Hamiltonian  $H_c = \sum_{x \in \{0,1\}^n} C(x) |x\rangle\langle x|$  (a real-valued diagonal matrix in  $\mathbb{C}^{2^N}$ ). The complex exponentiation of a Hermitian operator is a unitary operator, and so  $U_c(\alpha)$  is a diagonal unitary operator.

For example, a marked state cost function associated with a single solution state  $x^* = (1, \dots, 1)$  could be  $C(x) = -\prod_{i=1}^n x_i$ . Then,  $U_c(\alpha) = \mathbb{1} - (1 - e^{-i\alpha})|x^*\rangle\langle x^*|$ . The wave function associated with the quantum system  $|\psi\rangle = \sum_{x \in \{0,1\}^n} a_x |x\rangle$  will therefore evolve to  $|\psi\rangle = a_{x^*} e^{-i\alpha} |x^*\rangle + \sum_{x \in \{0,1\}^n / \{x^*\}} a_x |x\rangle$ , and so the behavior of  $U_c(\alpha)$  is to add the same phases to computational basis states that have the same energy evaluations according to the cost function  $C(x)$ .

QAOA1 is a type of QAOA approach with a specific mixing operator and a fixed starting state. Both of these conditions are replaced with a more general condition in the QAOA. In QAOA1, we begin in the uniform superposition state  $|+\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle$  so that every computational basis state  $|x\rangle$  has the same amplitude. The mixing operators are single local Pauli X operators:

$$U_d(\beta) = \prod_{j=1}^n e^{-i\beta\sigma_j^x}.$$

The action of  $\sigma_j^x$  on qubit  $j$  is to flip the bit, mapping  $|1\rangle$  to  $|0\rangle$  and vice versa. Any QAOA approach that utilizes  $U_d$  as mentioned above is considered a QAOA1 approach for the purpose of our descriptions. As such, QAOA1 proposes a very specific type of ansatz, in which the mixing operator takes this form for any problem. The QAOA will generalize this to allow for more structured ansatzes.

Then, for a single round  $t_i$  of a QAOA (including QAOA1), we pick two degrees  $\alpha_i$  and  $\beta_i$  and evolve the wave function as  $|\psi(t_i)\rangle = U_d(\beta_i)U_c(\alpha_i)|\psi(t_{i-1})\rangle$ .  $U_c$  changes the phase of states based on their evaluation from the classical cost function in the computational basis, while  $U_d$  acts as a mixer, leading to interference between the states in the superposition and potentially leading to concentration on low-lying energy states based on the cost function  $C(x)$  after several rounds. Indeed, as the number of rounds  $P$  goes to infinity, one can select  $\alpha_i$  and  $\beta_i$  such that  $|\psi(t_P)\rangle$  is guaranteed to minimize the cost function  $C(x)$  [4].

For a given finite value  $P$ , there are  $2P$  angles to select for running the algorithm which can therefore be cast as an optimization task where one wishes to find the angles that lead to a minimization of the cost function  $C(x)$ . QAOA1, as well as the QAOA, is a type of variational quantum algorithm (VQA) [13] for which finding the optimum parameters is typically an NP-hard problem [14]. The landscape of the final cost given by the quantum algorithm is known to suffer from a barren plateau [13,15–17], and optimization with random starts can lead to convergence to the local minima [6,13]. Nonetheless, the optimum or near-optimum solutions can follow similar patterns across instances [6,18]. For a more detailed discussion, we refer the reader to [13]. The techniques utilized in this paper will be discussed Section 6. Once the algorithm has run for  $P$  rounds, the wave function of the system is measured in the computational basis  $\{|x\rangle | x \in \{0,1\}^n\}$  such that the wave function collapses to a single state in that basis according to the Born rule, where  $\Pr(x) = |\langle x | \psi(t_P) \rangle|^2$ .

For many classes of problems, including certain class of linear and quadratic constraint problems, it is possible to find alternative mixing operators that allow us to limit the evolution of the wave function to a feasible subspace of a collection of those constraints [1,19–22]. The general problem is NP-hard, but there is a simple polynomial algorithm for bounded operators [22]. In QAOA1, the mixing operator utilized can be associated with there being no constraint placed on the evolution, since every state is reachable under its action (although it is clearly not the only such mixing operator). The two essential requirements for the mixing operators are that they must take actions moving states in the constrained space to potentially any other state in that constrained space, but they cannot move such states outside of the constrained space.

As such, the QAOA describes a more general approach than QAOA1, in which the mixing operator and phase-separating operators can be much more general. In particular, they are usually tailored to the specific type of symmetries of the underlying problem (such as the ansatz to the algorithm). Notice that the QAOA shares the same acronym as QAOA1 in the literature. For example, ref. [1] lists a compendium of mixing and phase-separating operators that can be utilized for different combinatorial optimization problems. Specific types of mixers have been studied more extensively. For example, XY mixers are associated with a very simple kind of equality constraint that makes them useful for many types of combinatorial optimization problems [23] as well as quantum chemistry [24].

### 3. Circuit Fault Diagnostics

Diagnosing errors and faults for gate-based digital circuits is an area of intense research, as large-scale integrated circuits and specialized circuit designs have become abundant in many scientific and engineering disciplines. Increasingly sophisticated automation techniques used to check and correct errors in the circuit design and fabrication are increasingly relied upon in practice. Because of the underlying combinatorial nature of this problem as well as the local design of many modern circuits, this is a class of problems that is well suited to typical NISQ [2,11] architectures.

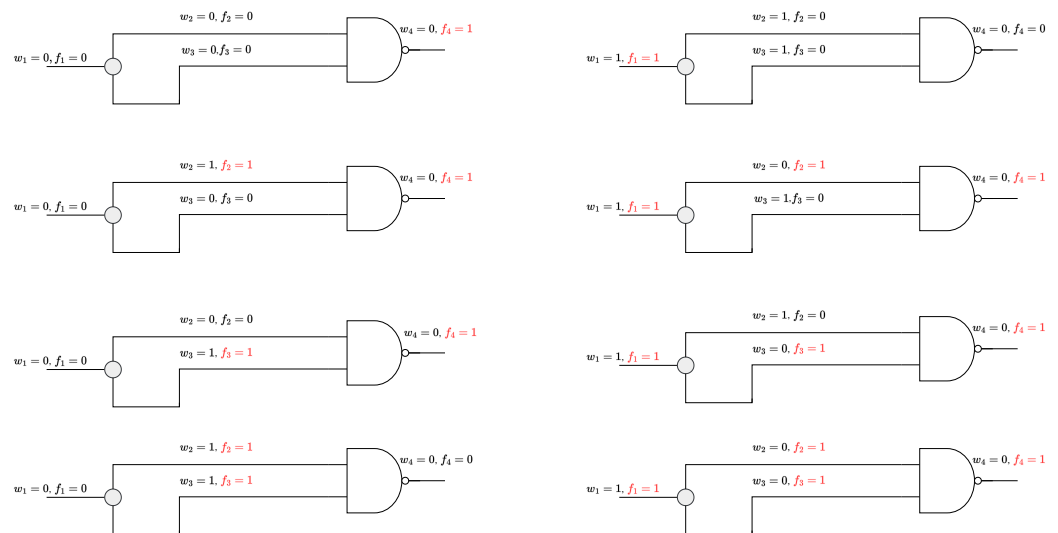
We employ a simple stuck-at-fault model for analyzing circuits with a given string of inputs and empirically found string of outputs [25]. Let  $n$  be the number of wires in the circuit and  $n_o$  be the number of output wires. Each wire in the circuit is either healthy or stuck at either zero (SA0) or one (SA1). Under the assumption that faults are equally likely to occur on every wire, the task of circuit fault diagnostics (CFD) is an optimization problem for finding the minimum number of faults needed to explain the input-output pair [2].

For the purpose of our discussion in this paper, we consider gates that are one-input/one-output, two-input/one-output, or one-input/two-output. One-input/one-output gates are an identity gate ID or an inverter gate INV. Two-input/one-output gates are OR, AND, XOR, NOR, NAND, or XNR. One-input/two-output gates are a fan-out gate FAN, a fan-out gate with an inverter on the first output F10, a fan-out gate with an inverter on the second output F01, or a fan-out gate with an inverter on both F11.

A configuration for the problem is a  $2n$ -bit string, where over bits  $1, \dots, n$  are the wire bits and over bits  $n + 1, \dots, 2n$  are the fault bits. A valid configuration  $(w_1, \dots, w_n, f_1, \dots, f_n)$  has an erroneous value of a wire  $w_i$  if the corresponding  $f_i$  is nonzero.

The inputs to a circuit are considered to also be potentially faulty such that the input wires themselves could be SA0 or SA1, but the output wires have to correspond to the empirical value they have (otherwise, this diagnosis would be invalid), but that value may come from error propagation in the circuit or from the output value itself being SA0 or SA1.

For whatever faults are present in the circuit, the output values can always be made to match the empirical value by taking the nonconforming output values and flipping their values (as well as their associated fault bit). As such, the size of the valid configuration space is  $2^{n-n_o}$  [3], since we have precisely  $n - n_o$  wire locations where the fault flag can be 0 or 1. (The fault flag on the  $n_o$  output wires is then forced to be such that the configuration is indeed valid.) Figure 1 shows all possible valid fault explanations for a small circuit. Note that there are 3 non-output wires and  $2^3$  valid fault configurations.



**Figure 1.** All possible valid fault configurations for a small circuit with one FAN gate and one NAND gate that has a faulty output. The diagrams on the top left and right show minimum fault explanations for this instance.

### 3.1. Valid and Invalid Configurations Around Gates

Enforcing the logic of the circuit on our configurations is essentially enforcing the logic of each individual gate on the part of the configuration that corresponds to that gate’s inputs and outputs.

Valid configurations around gates follow the same logical delineation as valid configurations for the entire circuit. For any circuit or subcircuit, including a single gate, we can list the potential valid configurations by considering the different input/output pairs. If a configuration is valid for a particular input/output pair, it cannot be valid for any other, and so the valid configurations always make disjointed sets. For that circuit or subcircuit, as discussed before, the inputs can potentially be faulty, but the outputs must match their empirical values. However, the size of the different valid configuration sets is always the same.

As such, the valid configurations around a gate are split by what the inputs before considering the fault status are (If they differ from this value, there must be an associated fault on this input.) and what the output after considering the fault status is (i.e., the same situation as for the circuit at large). Table 1 gives a full description of all the valid subconfiguration spaces for an NAND gate, where the first element in each entry is the trivial valid configuration that exists for every subcircuit (or the whole circuit) by applying

the faults on the outputs only (i.e., all wires are healthy, except the outputs that disagree with the logical value they should have).

If a configuration is valid and maintains a state status as described by an entry in Table 1, then swapping this subconfiguration with a state status in the same entry will also be valid, since we did not change the values that came from previous gates or were set as global inputs to the circuit, nor did we change the value of the output (which is then used as an input to a gate or is a global output for the circuit).

For a circuit with a collection of gates  $G$ , we consider the input and output wires ( $I_g$  and  $O_g$ ) for that gate. The collection of valid configuration sets around this gate

$$V_g = \{V_g^{(S_{I_g}; S_{O_g})} | S_{I_g} \in \{0, 1\}^{|I_g|}, S_{O_g} \in \{0, 1\}^{|O_g|}\},$$

is indexed by what the potential values the wire inputs  $S_{I_g}$  are set to without faults and what the potential wire outputs  $S_{O_g}$  are set to. For example,  $S_{I_g}$  for an NAND gate in the fault model is a collection of the possible inputs  $\{(0, 0), (0, 1), (1, 0), (1, 1)\}$ , and  $S_{O_g}$  is a collection of the possible outputs  $\{(0), (1)\}$ .

**Table 1.** Each kind of input/output pair for a faulty NAND is described. Each entry has different valid configurations, and configurations in the same entry form a subspace such that the action of the driver terms we construct connects the configurations of this subspace.

Entry	IO Pair	Valid Configurations
1	(0,0;0)	(0,0,0;0,0,1), (1,1,0;1,1,0), (1,0,0;1,0,1), (0,1,0;0,1,1)
2	(1,0;0)	(1,0,0;0,0,1), (1,1,0;0,1,0), (0,0,0;1,0,1), (0,1,0;1,1,1)
3	(0,1;0)	(0,1,0;0,0,1), (1,1,0;1,0,0), (0,0,0;0,1,1), (1,0,0;1,1,1)
4	(1,1;0)	(1,1,0;0,0,0), (0,0,0;1,1,1), (1,0,0;0,1,1), (0,1,0;1,0,1)
5	(0,0;1)	(0,0,1;0,0,0), (1,1,1;1,1,1), (1,0,1;1,0,0), (0,1,1;0,1,0)
6	(1,0;1)	(1,0,1;0,0,0), (1,1,1;0,1,1), (0,0,1;1,0,0), (0,1,1;1,1,0)
7	(0,1;1)	(0,1,1;0,0,0), (1,1,1;1,0,1), (0,0,1;0,1,0), (1,0,1;1,1,0)
8	(1,1;1)	(1,1,1;0,0,1), (0,0,1;1,1,0), (1,0,1;0,1,0), (0,1,1;1,0,0)

#### 4. QAOA Approaches to CFD

In this section, we give details on the construction for several QAOA approaches to solving the CFD problem, with each constraining the evolution of the wave function to a specific subspace and using mixing terms which connect all feasible states within this space to one another in different ways. We map each bit in the  $2n$ -bit string to a qubit such that the wave function of our system  $|\psi\rangle$  is a normalized vector in  $\mathbb{C}^{2^{2n}}$ . At the end of the QAOA algorithm, we measure  $|\psi\rangle$  in the computational basis to extract a solution to our problem.

##### 4.1. Approach 1: Transverse Field with a QUBO

This approach is closely related to a Hamiltonian description of this problem, which is similar to, for example, the approach described in [2] for using a transverse field and an Ising Hamiltonian with explicit fault mappings to represent this problem. As such, our approach is in the spirit of QAOA1 for Ising spin problems.

The mixing operators are the Pauli X operators:

$$U_d(\beta_t) = \prod_{i=1}^{n-n_o} e^{-i\beta_t\sigma_i^x}.$$

Since the output values of the circuit have to match the given empirical values, they are not allowed to change during the problem. As such, these qubits can be integrated out of the problem. The phase-separating operators are associated with an Ising Hamiltonian:

$$U_c(\alpha_t) = e^{i\alpha_t H_f},$$

where the Hamiltonian counts the number of faults and enforces the logic of the gates:

$$H_f = \left( \sum_{i=n-n_o+1}^{2n-n_o} (\mathbb{1} - \sigma_i^z) / 2 \right) + \kappa \left( \sum_{g \in \mathcal{G}} H_g \right),$$

where  $H_g = \sum_{V_g^v \in V_g} (\mathbb{1} - \sum_{v \in V_g^v} |v\rangle\langle v|)$  associates a cost to the state being in an invalid set for that particular gate. If a configuration  $x$  fails to be a valid configuration in the fault around a gate  $g$ , then  $H_g$  will associate a cost to the state  $x$ .

To ensure consistency with the model, we can set  $\kappa = n_o + 1$ , the number of outputs, so that it is never advantageous to break a gate over placing the faults at the very end of the circuit. The number of output bits is an upper bound on the number of faults of the minimal fault diagnosis. For individual instances, we can set  $\kappa$  to any value greater than the number of faulty output bits, which is less than or equal to  $n_o$ .

As in QAOA1, our initial state is the uniform superposition over  $2n - n_o$  qubits:  $|\psi\rangle = |+\rangle^{\otimes 2n-n_o} = \frac{1}{\sqrt{2^{2n-n_o}}} \sum_{x \in \{0,1\}^{2n-n_o}} |x\rangle$ .

#### 4.2. Approach 2: Transverse and XY Mixer with a QUBO

Any solution to the single input/output CFD problem has at most  $n_o$  faults. To exploit this bound, we can employ the XY mixer such that over the fault bits  $(f_1, \dots, f_n)$ , we impose the constraint  $\sum_{i=1}^n f_i \leq n_o$ . This can be achieved in different ways, for example, by introducing  $n_o$  ancilla qubits and employing the standard ring XY mixer over the  $n$  fault bits and  $n_o$  ancilla bits.

Note that the XY mixer operating on a computational basis state of  $N$  qubits keeps the number of ones fixed, so if the initial wave function only has support for the computational basis states with  $k$  ones ( $|\psi\rangle = \sum_{x \in \{0,1\}^N \text{ s.t. } |x|=k} a_x |x\rangle$ ), then the wave function after applying any collection of XY mixers will still have support only for computational basis states with  $k$  ones.

To achieve states with a number of ones between 0 and  $k$ , we can add  $k$  ancillas and then apply the XY mixer on  $N + k$  qubits such that when those mixers are applied to a wave function  $|\psi^{N+k}\rangle = \sum_{x \in \{0,1\}^N \text{ s.t. } |x| \leq k} \sum_{y \in \{0,1\}^k \text{ s.t. } |y|=k-|x|} a_{x+y} |x\rangle |y\rangle$ , the constraint on the number of ones in the wave function is maintained. Then, at the time of measurement, we simply discard the ancilla qubits and consider the resulting state over  $N$  qubits.

Since the outputs, as in the previous approach, can be integrated out, and so the wire bits correspond to the qubits between 1 and  $n - n_o$ , the fault bits correspond to qubits between  $n - n_o$  and  $2n - n_o$ , and the ancillas correspond to qubits between  $2n - n_o + 1$  and  $2n$ . Utilizing the XY mixer over qubits between  $2n - n_o + 1$  and  $2n$  and the transverse field applied on the wire bits leads to

$$H_d = \left( \sum_{i=1}^{n-n_o} \sigma_i^x \right) + \left( \sigma_n^x \sigma_{n-n_o+1}^x + \sigma_n^y \sigma_{n-n_o+1}^y \right) + \sum_{i=n-n_o+1}^{2n-1} \left( \sigma_i^x \sigma_{i+1}^x + \sigma_i^y \sigma_{i+1}^y \right),$$

which can be split into two noncommuting Hamiltonians:

$$H_1 = \left( \sum_{i=1}^{n-n_0} \sigma_i^x \right) + \left( \sigma_n^x \sigma_{n-n_0+1}^x + \sigma_n^y \sigma_{n-n_0+1}^y \right) + \sum_{i=1}^{\lfloor (n+n_0)/2 \rfloor} \left( \sigma_{n+2i}^x \sigma_{n+2i+1}^x + \sigma_{n+2i}^y \sigma_{n+2i+1}^y \right),$$

and

$$H_2 = \sum_{i=1}^{\lfloor (n+n_0)/2 \rfloor} \left( \sigma_{n+2i-1}^x \sigma_{n+2i}^x + \sigma_{n+2i-1}^y \sigma_{n+2i}^y \right).$$

When  $n + n_0$  is odd,  $H_1$  also has a noncommuting term associated with qubits  $n - n_0 + 1, n - n_0 + 2,$  and  $2n$ . As for the XY mixing operators [23], we can apply commuting mixing operators associated with  $H_1$  and then  $H_2$  as follows:

$$U_d(\beta_t) = \left( \prod_{i=1}^{n-n_0} e^{-i\beta_t \sigma_i^x} \right) \left( \prod_{i=1}^{\lfloor (n+n_0)/2 \rfloor} e^{-i\beta_t (\sigma_{n+2i}^x \sigma_{n+2i+1}^x + \sigma_{n+2i}^y \sigma_{n+2i+1}^y)} \right) \\ \left( e^{-i\beta_t (\sigma_n^x \sigma_{n-n_0+1}^x + \sigma_n^y \sigma_{n-n_0+1}^y)} \right) \left( \prod_{i=1}^{\lfloor (n+n_0)/2 \rfloor} e^{-i\beta_t (\sigma_{n+2i-1}^x \sigma_{n+2i}^x + \sigma_{n+2i-1}^y \sigma_{n+2i}^y)} \right).$$

The phase-separating operators are then the same as in the previous section. The initial state is an outer product (concatenation) of the uniform superposition over the wire bits and a state with a uniform superposition over  $n + n_0$  states with precisely  $n_0$  bits set to one, where  $|\psi\rangle = |\psi_1\rangle |\psi_2\rangle$  with  $|\psi_1\rangle = \frac{1}{\sqrt{2^{n-n_0}}} \sum_{x \in \{0,1\}^{n-n_0}} |x\rangle$  and  $|\psi_2\rangle = \binom{n+n_0}{n_0}^{-1/2} \sum_{x \in \{0,1\}^n \text{ s.t. } |x|_1 = n_0} |x\rangle$ . We discuss the preparation of the well-known entangled state  $|\psi_2\rangle$ , studied in several contexts with relation to the XY mixer, in Section 4.6.

### 4.3. Approach 3: Graph Diffusors with a Linear Field on the Fault Bits

In this approach, we tailor the mixing operators to maintain the valid fault configuration space. Around each gate, a mixing operator is associated with swapping valid fault configurations around that gate, utilizing the descriptions given in Section 3.1 such that, given a valid configuration, the mixing operators populate states that are also valid configurations.

Given a set of logical gates  $G = \{g_1, \dots, g_m\}$  in the fault model over  $w_1, \dots, w_n$  wires and  $f_1, \dots, f_n$  fault bits, define  $V_g^{(S_I; S_O)}$  as detailed in Section 3.1. We wish to construct the collection of unitaries  $\mathcal{U} = \{U_{g_1}^{(1)}, \dots, U_{g_1}^{(r)}, \dots, U_{g_m}^{(r)}\}$  associated with the mixing operator of each gate and the respective tuple of input/output pairs for each gate. (Note that the index  $(r)$  runs over a set of input/output pairs to the gate, so it is typically an  $n$ -tuple and not a single integer.) For example, as in the circuit considered in Section 3 and illustrated in Figure 1, if  $g_2$  is an NAND gate, then  $\{U_{g_2}^{(1)}, \dots, U_{g_2}^{(8)}\}$  are associated with the eight input/output pairs for an NAND gate. We define the uniform superposition over the collections  $V_{g_i}^{(j)}$  as  $|V_{g_i}^{(j)}\rangle = \frac{1}{\sqrt{|V_{g_i}^{(j)}|}} \sum_{v \in V_{g_i}^{(j)}} |v\rangle$ . (See Table 1 for such a description for an NAND gate) Then, we define each mixing operator associated with a specific input/output pair for a specific gate:

$$U_{g_i}^{(j)}(\beta_t) = \mathbb{1} - \left(1 - e^{-i\beta_t}\right) |V_{g_i}^{(j)}\rangle \langle V_{g_i}^{(j)}| \\ = \mathbb{1} - \left(1 - e^{-i\beta_t}\right) / |V_{g_i}^{(j)}| \sum_{v, u \in V_{g_i}^{(j)}} |v\rangle \langle u|.$$



Then, we define the mixing operator associated with each gate:

$$U_{g_i}(\beta_t) = \prod_{j=(1)}^{(r)} U_{g_i}^{(j)}(\beta_t).$$

Here,  $U_{g_i}(\beta_t)$  are unitary because  $U_{g_i}^{(j)}(\beta_t)$  are mutually commutative over the input/output pair index  $j$ .  $U_{g_i}^{(j)}$ , which has an eigenvalue  $e^{i\alpha}$  for an eigenvector  $|V_{g_i}^{(j)}\rangle$  and an eigenvalue of one for any vector in the range of the projector  $\mathbb{1} - |V_{g_i}^{(j)}\rangle\langle V_{g_i}^{(j)}|$ . Then, we define

$$U_d(\beta_t) = \prod_{i=1}^m U_{g_i}(\beta_t),$$

as the mixing operator for this ansatz, where we apply each gate one after another. However, for circuits with a highly regular structure (such as those introduced in Section 3), we can group many commuting unitaries together and apply all  $U_i(\beta_t)$  with in two steps.

The cost function (also one-local) simply counts the number of faults of the configurations:

$$U_c(\alpha_t) = \prod_{i=n}^{2n} \left( e^{i\alpha_t(\mathbb{1}-\sigma_i^z)/2} \right).$$

The feasibility space and how it is connected through the action of  $U_d$  is more complicated for this approach than previous approaches such that it cannot be as easily described by a uniform superposition over the states. We begin with a known feasible configuration and apply  $U_d(\beta_t)$  to explore the feasible configuration from there. A simple starting state used for this purpose is the state with only faults placed on the faulty output states.

We also consider a modified protocol which includes a second cost function that has this state as its minimum, in clear inspiration from a similar CQA [3] approach. A sufficient (and one-local) cost function puts a penalty on every bit that does not conform with the initial state chosen. As such, the cost associated with any state in the space is given by its Hamming distance from the initial state. Let  $x^0 = \{x_1^0, \dots, x_{2n}^0\}$  be the computational basis string associated with the initial state  $|\psi\rangle = |x^0\rangle$ . Then, the initial state cost function operator can be implemented as

$$U_s(\gamma_t) = \prod_{i=1}^{2n} e^{i\gamma_t(\mathbb{1} - (1-2x_i^0)\sigma_i^z)/2}.$$

For this modified protocol, we have three sets of operators: the initial state cost function operator, the mixing operators, and the phase-separating operator. As such, we generalize the QAOA protocol to have three angles to select for every round such that the state evolves under  $(\alpha_1, \beta_1, \gamma_1, \alpha_2, \beta_2, \gamma_2, \dots, \alpha_p, \beta_p, \gamma_p)$ . For step  $t_i$ ,  $|\psi(t_i)\rangle = U_s(\gamma_i)U_d(\beta_i)U_c(\alpha_i)|\psi(t_{i-1})\rangle$ . In Section 6, we consider the performance of both approaches for small circuit instances and different parameter selection methods. Since  $U_s$  and  $U_c$  are both one-local operators, it is also straightforward to apply them together in a device.

#### 4.4. Approach 4: Transverse Field on Faults with an Oracle Circuit Simulator

Rather than representing the wire and fault bits explicitly, in our next approach, we focus on the space of valid fault configurations (of dimensions  $2^{n-n_o}$ ). Notice that for any invalid fault configuration, it can be made a valid configuration by simply flipping the fault bit for the output bits which are incorrect.

Consider a fault configuration  $f = (f_1, \dots, f_{n-n_o})$  over non-output wires. The energy of the state is dependent on the number of nonzero fault bits  $P(f) = \sum_{i=1}^{n-n_o} f_i$  as well as

the number of implied faults needed on the outputs to make this a valid configuration  $Q(f) = \sum_{i=n-n_o}^n (w_i - S(f, i))^2$ , where  $S(f, i)$  is the value placed on the output wire  $w_i$  when simulating the circuit with the given inputs and the fault configuration  $f$  over the non-output wires.

Then,  $R(f) = P(f) + Q(f)$  counts the total number of faults implied by the fault configuration  $f$  in this constrained space. By constructing an oracle that simulates the circuit and then uses this information to find the proper fault count, we have a phase separation operator with the same cost function as in Section 4.3:

$$U_c(\alpha_t) = \sum_{x \in \{0,1\}^{n-n_o}} e^{i\alpha_t R(x)} |x\rangle \langle x|.$$

As in QAOA1, our starting state will be  $|\psi\rangle = |+\rangle^{\otimes n-n_o} = \frac{1}{\sqrt{2^{n-n_o}}} \sum_{f \in \{0,1\}^{n-n_o}} |f\rangle$ . and we use the one-local Pauli X mixing operators:

$$U_d(\beta_t) = \prod_{i=1}^n e^{-i\beta_t (\sigma_i^x)}.$$

To see how  $U_c(t)$  can be implemented in practice, we consider a protocol where ancilla qubits are used for the computation of the faults needed on the outputs. Let  $|\psi(r_{i-1})\rangle \in \mathbb{C}^{2^{n-n_o}}$  be the wave function at the beginning of round  $r_i$ , with the initial wave function defined as shown above. We show how  $|\psi(r_i)\rangle$  is then generated over the current round. For  $r_i \in [r_1, r_t]$ , we have

$$\begin{aligned} |\nu(r_i)\rangle &= U_{circ} |0\rangle_n |\psi(r_{i-1})\rangle |0\rangle_{n_o} \\ &= \sum_{x \in \{0,1\}^{n-n_o}} \langle x | \psi \rangle |f(x)\rangle_n |x\rangle_{n-n_o} |g(x)\rangle_{n_o} && \text{(Simulate Circuit)} \\ |\mu(r_i)\rangle &= U_c(\alpha_i) |\nu(r_i)\rangle && \text{(Apply Phases)} \\ &= \prod_{j=n+1}^{2n} e^{i\alpha_i (\mathbb{1} - \sigma_j^z)/2} \sum_{x \in \{0,1\}^n} |f(x)\rangle_n |x\rangle_{n-n_o} |g(x)\rangle_{n_o} \\ &= \sum_{x \in \{0,1\}^{n-n_o}} \left( \sum_{j=n+1}^{2n} e^{i\alpha_i x_j} \right) |f(x)\rangle_n |x\rangle_{n-n_o} |g(x)\rangle_{n_o} \\ |0\rangle_n |\phi(r_i)\rangle |0\rangle_{n_o} &= U_{circ}^\dagger |\mu(r_i)\rangle && \text{(Undo Simulation)} \\ |\psi(r_i)\rangle &= \prod_{j=1}^{n-n_o} e^{-i\beta_i \sigma_j^x} |\phi(r_i)\rangle && \text{(Apply Mixing)} \end{aligned}$$

Here,  $U_{circ}$  implements the simulation of the faulty circuit such that  $f(x)$  is the valid configuration over the wire bits that  $|x\rangle$  specifies from the non-output fault bits, while  $g(x)$  are the required faults on the output bits such that  $x \oplus g(x) \oplus f(x)$  is a valid configuration over the  $2n$  bits.

#### 4.5. Approach 5: Bounded Fault Count with Oracle Circuit Simulator

In this section, rather than restricting ourselves to the subspace of all fault configurations, we wish to restrict ourselves to the fault configurations up to a maximum number of faults. To accomplish this task, we can use the XY mixer also discussed in Section 4.2 by utilizing ancilla qubits in a complimentary way. Here, we have  $n - n_o$  non-output fault flag bits (the same as in Section 4.4) and  $n_o$  ancilla qubits to allow for representation of all state configurations with faults less than or equal to  $n_o$ .

The cost function remains the same as that used in Section 4.4, and the same procedure with the simulation of the circuit to find the faults on the end bits can be used, except for the aforementioned replacement of the mixing operator.

The initial state is then  $|\psi\rangle = \binom{n}{n_o}^{-1/2} \sum_{x \in \{0,1\}^{n-n_o} \text{ s.t. } |x|_1 \leq n_o} \sum_{y \in \{0,1\}^{n_o} \text{ s.t. } |y|_1 = n_o - |x|_1} |x\rangle|y\rangle$ , similar to the state mentioned in Section 4.2. Note that since we also have implied faults, the maximum number of faults that can be expressed is between  $n_o$  and  $2n_o$ . The initialization of this state is discussed in more detail in Section 4.6.

4.6. Size of the Relevant Constrained Space and Summaries for the Approaches

In each of the approaches delineated in Section 3, the size of the constrained space of interest which the evolution of the wave function is limited to differs. In Table 2 and Figure 2, each approach is summarized by the type of mixing operator, the type of cost function, the size of the constrained space, and the complexity of each round.

Table 2. Depending on our selection of mixing and phase-separating operators, we can constrain the evolution of the wave function to constrained spaces of differing sizes.

Approach	Mixing Operator	Cost Function	Size of Constrained Space	Round Complexity
1	Pauli X	Ising	$\mathcal{O}(2^{2n})$	$\mathcal{O}(1)$
2	XY mixer and X	Ising	$\mathcal{O}(2^{n-n_o}(n+n_o)^{n_o})$	$\mathcal{O}(1)$
3	Gate-based diffusors	Pauli Z	$\mathcal{O}(2^{n-n_o})$	$\mathcal{O}(1)$
4	Pauli X	Circuit Oracle	$\mathcal{O}(2^{n-n_o})$	$\mathcal{O}(C_p)$
5	XY mixer	Circuit Oracle	$\mathcal{O}(n^{n_o})$	$\mathcal{O}(C_p)$

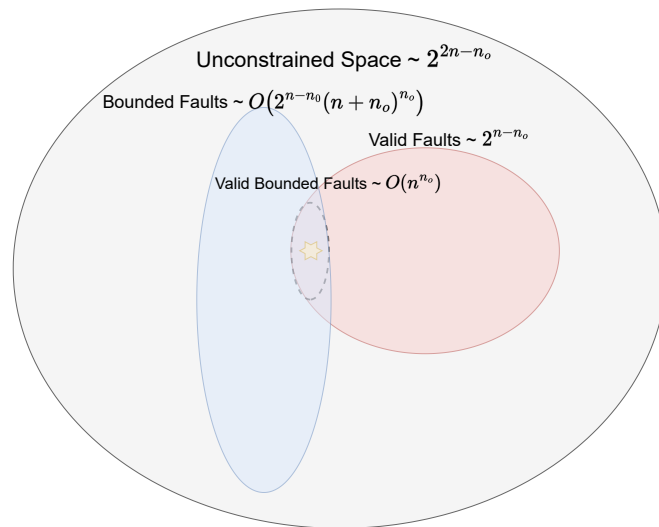


Figure 2. A visual representation of the subspaces that the wave function is kept within during evolution through each QAOA approach.

While the approaches in Sections 4.3 and 4.4 differ greatly in the type of mixing and phase separation operators used, they constrain the system to the same subspace. Consider the exploration of this space under these two different approaches. The latter begins with a uniform superposition over all valid fault configurations, and the neighborhood of each computational basis state under the action of each Pauli X operator in the mixer is a state that is a bit flip away, similar to a high-dimensional hypercube. The former begins with a single feasible state, and the neighborhood of each state under the action of the mixing operator is dependent on the available transformations around each gate to move it to a new fault configuration.

Moreover, while the cost functions associated with these two approaches are *equivalent*, the resulting phase separation operators for implementing these cost functions are very different. For example, while each call to the cost function runs in  $\mathcal{O}(1)$  for the fault counting in the former, each call to the cost function in the latter will require simulating the corresponding circuit, which has a particular depth  $C_p$ . For example, for the distribution

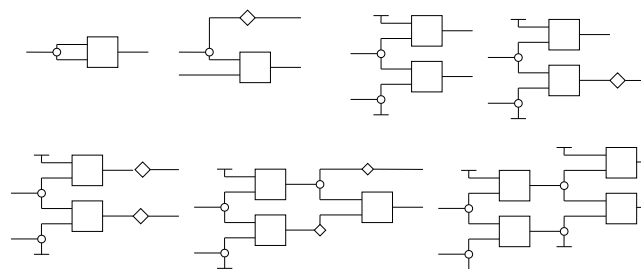
of the circuits introduced in Section 5, the depth of the classical circuit grows as  $\mathcal{O}(\sqrt{n})$  if we choose the diameter and depth to be equal. (This relationship holds for the family considered for the experiments in this paper.)

Approaches 2 and 5 require the preparation of a particle number-conserving state studied in many areas of quantum computation and quantum chemistry [26–29]. We refer the reader to the appendix of [23] for a discussion of this state in the context of a QAOA with the XY mixer. In general, for a uniform superposition over  $n$  qubits preserving a number  $k$ , the state can be constructed with  $\binom{n}{k} = \mathcal{O}(n^k)$  CNOT gates [29], which can be prohibitive for a circuit with a large bound on the number of faults. However, the authors of [26] provided a projective measurement method that grows polynomially with  $n$  and independent of  $k$  for large  $n$  values. Alternatively, one can use an approach similar to Approach 3, where the initial state is a computational basis state in the feasible space. The initial state selection for an QAOA remains an important area of current research [23], including warm starts [30,31]. Moreover, the number of minimum faults for many circuits of interest can be much less than  $n_o$ , and one could adapt the approaches to utilize this smaller space.

### 5. Random CFD Instances with Balanced Width and Depth

To demonstrate how these different approaches practically perform in the CFD problem, we introduce a family of random instances that scale with a single size parameter and allow for detailed analysis at small sizes. Figure 3 shows the first few such circuits with increasing sizes. The central feature is that both the depth of the circuit and the number of inputs or outputs (width) scale together with the size and number of wires of the circuit. We randomly select the inputs for the whole circuit and apply faults to the outputs such that every output is faulty. We randomly select the inputs and gates at each spot from the relevant gate set for every size to generate the instances used in Section 6.

The valid fault configurations for an instance of the smallest circuit in the family are in Figure 1. Unlike all the other circuits, this has only one output, and so we modified the model for this small circuit for our experiments such that placing a fault on the output wire required two faults, whereas every other location required one.



**Figure 3.** Diagrams for circuit instances of the CFD problem considered in Section 6, depending on the number of qubits, from these small, local, and dense circuits. Every box is a two input/one output gate, every dot is a one input/two output gate, and every diamond is a one input/one output gate. If a top wire has a  $\top$  and a bottom wire has a  $\perp$  at the same depth, they refer to the same wire (which has been wrapped around).

While at large scales we expected the performance of any specific ansatz strategy to decrease with the problem size, at the small sizes used for our simulations, the performance could vary by problem size in a more complicated fashion. For all instances, the maximum number of faults needed was two, since there were two outputs. We generated 100 instances that were filtered such that the minimum fault explanation required a single fault.

Approach 3, unlike all the other approaches, would be initialized in a solution state without this filtering. For larger-sized circuits of this family, the chance of a randomly selected instance having a minimum fault explanation that saturates this bound is increas-

ingly diminished, but this filtering is an important step for the small-sized circuits of this family.

For the filtered data sets of different sizes, the specific topology of the circuits can still lead to important differences in fault configurations. For example, the circuit depicted at the bottom middle of Figure 3 has much more diversity in the typical size (or degeneracy) of the solution space compared with the smaller sizes. Solution degeneracy has been seen to have a beneficial impact on a variety of optimization problems for QAOAs.

## 6. Performance with Different Parameter Optimizations

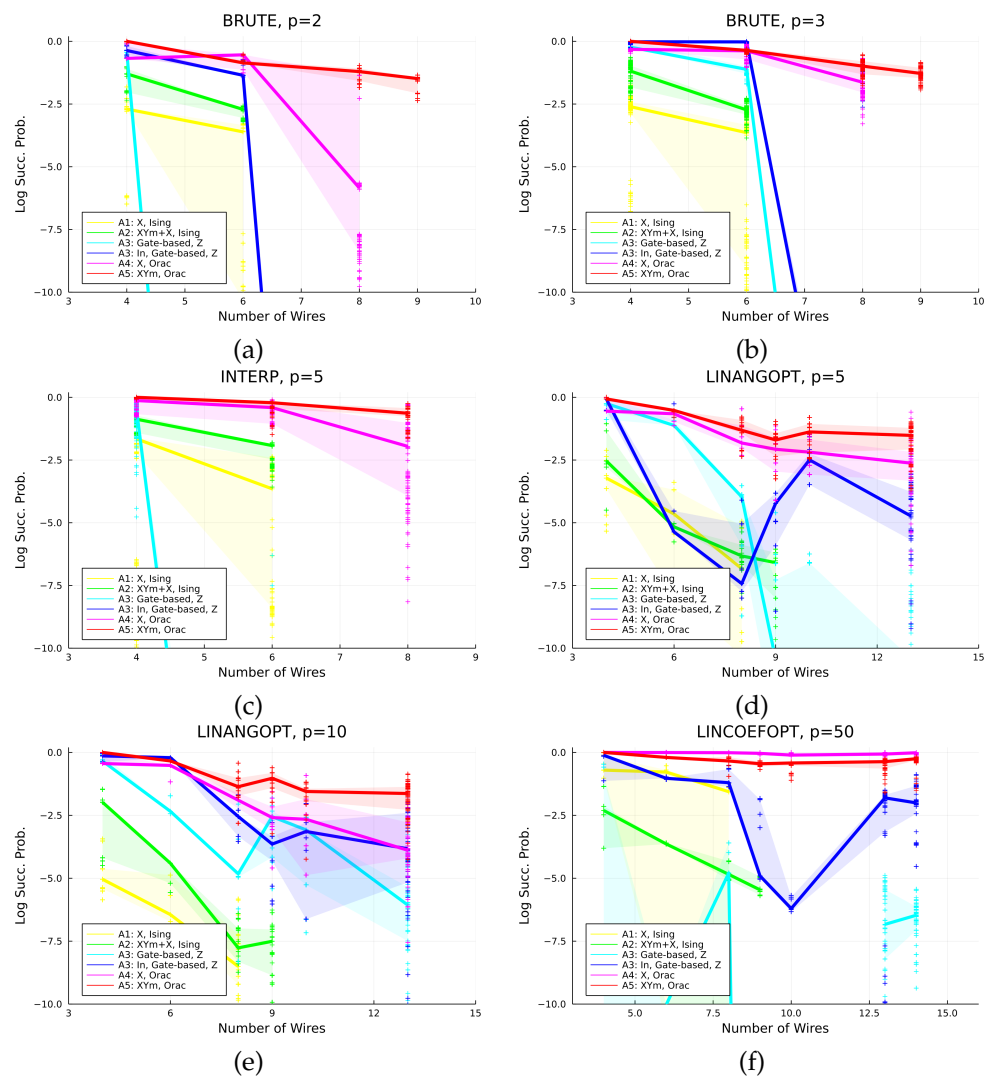
Using the family of CFD instances from the previous section, we consider the performance of QAOA strategies with different methods of parameter optimization: BRUTE, INTERP, LINANGOPT, and LINCOEFOPT. Each of the first three requires the expected cost of the final wave function to update the angle parameters. For implementation on a quantum computer, the expected cost of the final wave function has to be sampled with repeated measurements, and there will be a trade-off between repeated runs to accurately compute this value and run the protocol with updated parameters based on the approximation [6]. The angles were selected from the interval  $[-\pi, \pi]$ .

For BRUTE, we chose 100 random seeds and ran the Nelder–Mead algorithm with 100 iterations to optimize each choice, using the final Hamiltonian as the cost function and selecting the choice that minimized the cost function the most. For INTERP, we used the interpolation function called INTERP in [6], which showed good performance from this method in their results. We used the Nelder–Mead algorithm (with 100 random seeds and 100 iterations each) on  $p = 2$  and then for  $p > 2$ , beginning with the linear interpolation from  $p - 1$  and optimizing with 100 iterations for this result. For LINANGOPT, we started the protocol with a linear ramp for each angle (i.e., for round  $k \in [1, p]$ ,  $\alpha_k = k\pi/p$  and  $\beta_k = \pi - k\pi/p$ ) and then optimized the angles (with 200 iterations). For LINCOEFOPT, we used a coefficient  $\Gamma \in \{9/10, 1/2, 1/4, 1/8, 1/16\}$  and used the same linear ramp but replaced  $\alpha_k, \beta_k$  with  $\alpha_k\Gamma, \beta_k\Gamma$  and then optimized  $\Gamma$  over 20 iterations before selecting the best performing  $\Gamma$ .

In each situation described, Approach 3 (with an initial state cost function) is different from the other approaches, since it has three angles to select for a single round  $k$ . As the initial starting point for LINANGOPT and LINCOEFOPT (with  $\Gamma$ ), it begins with  $\gamma_k = \pi - k\pi/p$ ,  $\alpha_k = \Theta(k \leq p)(2k\pi/p) + \Theta(k > p)(\pi - 2k\pi/p)$  (reaching a maximum around  $p/2$  and minimums at 0 and  $p$ ), and  $\beta_k = k\pi/p$ . Note that Approach 3 was also considered without an initial state cost function, and the results support that this seemed to benefit its performance.

The results from [6] suggest that the Nelder–Mead algorithm can perform as well as other optimization algorithms, such as the Broyden–Fletcher–Goldfarb–Shanno algorithm. Utilizing other optimizers for these tailored ansätze is an interesting area of future study [21,32,33]. We report the results with  $p$  set to 2 and 3 for BRUTE, 5 for INTERP, 5 and 10 for LINANGOPT, and 50 for LINCOEFOPT.

The relative simulation range capable on a workstation grows depending on the size of the relevant constrained space for the problem and so the more constrained approaches have an increased simulation range, as such we report results on larger instances for these approaches for each parameter selection. Figure 4 details the performance of each ansatz with each parameter selection strategy.



**Figure 4.** Performance of ansätze with different parameter selections. The solid lines indicate the performance on the median of the instances, while the ribbons correspond to the lower and upper quantiles. Markers correspond to the success probabilities of individual instances.

The results support that ansätze which exploit a more constrained space by tailoring the operators used to the problem structure generally perform better in the CFD problem and therefore are important methods for improving the practical applicability of QAOAs and other variational quantum algorithms. Indeed, Approaches 4 and 5 consistently performed the best, and Approach 3 performed comparably with some parameter selections for certain sizes. Surprisingly though, Approach 1 performed quite strongly for LINCOEFOPT with  $p = 50$ . The performance across these different metrics generally supports that while a linear tramp function might be well suited for certain approaches, such as Approaches 1 and 4, it is interesting to consider different functions to parameterize the angles for other approaches. Utilizing an initial state cost function showed advantages for Approach 3 in each parameter selection method except for LINANGOPT, where the benefit was not as clear.

The performance showed high variance under BRUTE and INTERP for most approaches. Approach 3 in particular seemed to struggle more at very low  $p$  values, perhaps because the initial state had full support for a particular solution and had to use mixing operators to explore the space, while the other approaches began with a uniform superposition over all states in the particular subspace. Since INTERP utilizes the results from BRUTE at  $p = 2$ , this difficulty can translate to this approach as well. While INTERP can

alleviate the high cost associated with brute force evaluation to find good parameters with stepwise optimization as  $p$  is scaled higher, the parameters found at low  $p$  values can potentially be associated with a local minimum that may be the best cost minimization the algorithm can accomplish at this depth, and then the algorithm is subsequently stuck around this choice at higher  $p$  values.

Approach 3 also showed striking performance differences between instance sizes, especially for LINANGOPT with  $p = 5$  and LINCOEFOPT with  $p = 50$ . Approach 3 is initialized to a computational basis state with faults on the outputs, and so the distance, in terms of the number of mixing operators, to find a minimum state as well as the degeneracy of the solution space may play a more acute role compared with other approaches. For example, as noted in Section 5, the instances with 13 wires had instances with higher degeneracy than that at lower sizes. This typical degeneracy growth with the problem size for the circuits of interest is one reason leveraging quantum algorithms may be beneficial.

Note that the results were based on the number of rounds  $p$ , but the actual depth of a quantum circuit to implement each of the different approaches will differ, especially for larger circuit instances. Moreover, Approaches 2 and 5 used a starting state that required more involved preparation before the procedure could begin.

## 7. Conclusions

One of the most compelling areas for utilizing variational quantum algorithms on NISQ devices arises from solving optimization problems, including those with hard constraints. The quantum alternating operator ansatz (QAOA) and constrained quantum annealing (CQA) are methods that can enforce those constraints naturally throughout the protocol or anneal. Designing approaches that can usefully use the structures of different optimization problems remains an important task for developing state-of-the-art quantum algorithms.

Designing mixing operators and phasing operators for a constrained space can be more general than satisfying a global hard constraint, such as satisfying many local constraints (such as the valid configurations around a faulty gate [3], as in Section 4.3), a practical bound on the maximum value a counting problem can yield (Sections 4.2 and 4.5), or simply finding a way to represent a Hamiltonian that is difficult to implement without ancillas (Sections 4.4 and 4.5).

In this paper, we constructed several general approaches to exploit constrained quantum evolution to solve the CFD problem, some of which had superpolynomially smaller constraint subspaces. While some of these approaches may be related to protocols that could be implemented on a quantum annealer, others are more challenging and therefore better suited for QAOAs or digital adiabatic simulations. We introduced a family of random circuits that are parameterized by a single size parameter, which can be useful for benchmarking future NISQ devices. Simulations of QAOA protocols with optimized parameters, interpolated and optimized parameters, and linear interpolation suggest that more advanced ansätze can give better performance by utilizing the underlining structure of an optimization problem. As such, designing and experimenting with more novel operators for solving optimization problems remains an important area for future research. Nonetheless, the initial states of several approaches are more involved to prepare, and the unitaries necessitated for each ansatz require further analysis under specific quantum architectures.

There are many interesting areas of future work that arise from the constructions considered in this paper. It would be interesting to explore approaches such as Approach 5 with an initial state in the computational basis to relieve the cost of preparing a highly entangled initial state. Since the number of minimum faults needed for a circuit can be much smaller than the upper bound given, it would be of interest to explore several of the approaches where this is utilized to form a more constrained ansatz, especially in Approach 5. For example, if the minimum number of faults needed is known to scale logarithmically, the constrained space can then scale subexponentially for Approach 5. Given the rich and growing literature of approaches to find suitable parameters for QAOAs, it would be

interesting to utilize such approaches for the ansätze introduced here, especially those that are less cost prohibitive for intermediate  $p$  values, where the advantage of more tailored ansätze could be more pronounced compared with more generic ansätze using the same parameter selection approach. Given that near-term quantum devices are likely to be noisy and have inaccuracies in the application of gates, it would be interesting to compare these ansätze in this regime and consider modifications. The performance of each ansatz varied over the instances, and it would be interesting to analyze what kind of features of an instance can be predictive of the performance of an ansatz. Finally, it would be of interest to consider other optimization problems in which similar structures can be exploited to tailor ansätze.

**Author Contributions:** Conceptualization, H.L., F.M.S. and E.R.; Data curation, H.L.; Formal analysis, H.L., F.M.S. and E.R.; Funding acquisition, E.R.; Project administration, E.R.; Software, H.L.; Supervision, F.M.S. and E.R.; Visualization, H.L.; Writing—original draft, H.L., F.M.S. and E.R.; Writing—review & editing, H.L., F.M.S. and E.R. All authors have read and agreed to the published version of the manuscript.

**Funding:** We are grateful for the support from the NASA Ames Research Center and from DARPA under IAA 8839, Annex 128. The research is based upon work (partially) supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA) and the Defense Advanced Research Projects Agency (DARPA), via the U.S. Army Research Office contract W911NF-17-C-0050. H.L. was supported by the USRA Feynman Quantum Academy and funded by the NAMS R&D Student Program under contract no. NNA16BD14C.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author.

**Acknowledgments:** The authors are grateful for the support from the NASA Ames Research Center. The authors thank P. A. Lott and S. Grabbe for their helpful discussions.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Hadfield, S.; Wang, Z.; O’gorman, B.; Rieffel, E.G.; Venturelli, D.; Biswas, R. From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms* **2019**, *12*, 34. [\[CrossRef\]](#)
- Perdomo-Ortiz, A.; Feldman, A.; Ozaeta, A.; Isakov, S.V.; Zhu, Z.; O’Gorman, B.; Katzgraber, H.G.; Diedrich, A.; Neven, H.; de Kleer, J.; et al. Readiness of quantum optimization machines for industrial applications. *Phys. Rev. Appl.* **2019**, *12*, 014004. [\[CrossRef\]](#)
- Leipold, H.; Spedalieri, F.M. Quantum Annealing with Special Drivers for Circuit Fault Diagnostics. *Sci. Rep.* **2022**, *12*, 11691. [\[CrossRef\]](#) [\[PubMed\]](#)
- Farhi, E.; Goldstone, J.; Gutmann, S. A quantum approximate optimization algorithm. *arXiv* **2014**, arXiv:1411.4028.
- Wang, Z.; Hadfield, S.; Jiang, Z.; Rieffel, E.G. Quantum approximate optimization algorithm for maxcut: A fermionic view. *Phys. Rev. A* **2018**, *97*, 022304. [\[CrossRef\]](#)
- Zhou, L.; Wang, S.T.; Choi, S.; Pichler, H.; Lukin, M.D. Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices. *Phys. Rev. X* **2020**, *10*, 021067. [\[CrossRef\]](#)
- Hadfield, S.; Hogg, T.; Rieffel, E.G. Analytical Framework for Quantum Alternating Operator Ansatz. *arXiv* **2021**, arXiv:2105.06996.
- Bravyi, S.; Kliesch, A.; Koenig, R.; Tang, E. Hybrid quantum-classical algorithms for approximate graph coloring. *arXiv* **2020**, arXiv:2011.13420.
- Hadfield, S.; Wang, Z.; Rieffel, E.G.; O’Gorman, B.; Venturelli, D.; Biswas, R. Quantum approximate optimization with hard and soft constraints. In Proceedings of the Second International Workshop on Post Moores Era Supercomputing, Denver, CO, USA, 12–17 November 2017; pp. 15–21.
- Farhi, E.; Harrow, A.W. Quantum supremacy through the quantum approximate optimization algorithm. *arXiv* **2016**, arXiv:1602.07674.
- Preskill, J. Quantum computing in the NISQ era and beyond. *Quantum* **2018**, *2*, 79. [\[CrossRef\]](#)
- Harrigan, M.P.; Sung, K.J.; Neeley, M.; Satzinger, K.J.; Arute, F.; Arya, K.; Atalaya, J.; Bardin, J.C.; Barends, R.; Boixo, S.; et al. Quantum approximate optimization of non-planar graph problems on a planar superconducting processor. *Nat. Phys.* **2021**, *17*, 332–336. [\[CrossRef\]](#)
- Cerezo, M.; Arrasmith, A.; Babbush, R.; Benjamin, S.C.; Endo, S.; Fujii, K.; McClean, J.R.; Mitarai, K.; Yuan, X.; Cincio, L.; et al. Variational quantum algorithms. *Nat. Rev. Phys.* **2021**, *3*, 625–644. [\[CrossRef\]](#)



14. Bittel, L.; Kliesch, M. Training variational quantum algorithms is np-hard. *Phys. Rev. Lett.* **2021**, *127*, 120502. [[CrossRef](#)] [[PubMed](#)]
15. Arrasmith, A.; Cerezo, M.; Czarnik, P.; Cincio, L.; Coles, P.J. Effect of barren plateaus on gradient-free optimization. *Quantum* **2021**, *5*, 558. [[CrossRef](#)]
16. Cerezo, M.; Sone, A.; Volkoff, T.; Cincio, L.; Coles, P.J. Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nat. Commun.* **2021**, *12*, 1–12.
17. Holmes, Z.; Sharma, K.; Cerezo, M.; Coles, P.J. Connecting ansatz expressibility to gradient magnitudes and barren plateaus. *PRX Quantum* **2022**, *3*, 010313. [[CrossRef](#)]
18. Crooks, G.E. Performance of the quantum approximate optimization algorithm on the maximum cut problem. *arXiv* **2018**, arXiv:1811.08419.
19. Hen, I.; Spedalieri, F.M. Quantum annealing for constrained optimization. *Phys. Rev. Appl.* **2016**, *5*, 034007. [[CrossRef](#)]
20. Hen, I.; Sarandy, M.S. Driver Hamiltonians for constrained optimization in quantum annealing. *Phys. Rev. A* **2016**, *93*, 062312. [[CrossRef](#)]
21. Shaydulin, R.; Hadfield, S.; Hogg, T.; Safro, I. Classical symmetries and the Quantum Approximate Optimization Algorithm. *arXiv* **2020**, arXiv:2012.04713.
22. Leipold, H.; Spedalieri, F. Constructing driver Hamiltonians for optimization problems with linear constraints. *Quantum Sci. Technol.* **2021**. [[CrossRef](#)]
23. Wang, Z.; Rubin, N.C.; Dominy, J.M.; Rieffel, E.G. X y mixers: Analytical and numerical results for the quantum alternating operator ansatz. *Phys. Rev. A* **2020**, *101*, 012320. [[CrossRef](#)]
24. Kremenetski, V.; Hogg, T.; Hadfield, S.; Cotton, S.J.; Tubman, N.M. Quantum Alternating Operator Ansatz (QAOA) Phase Diagrams and Applications for Quantum Chemistry. *arXiv* **2021**, arXiv:2108.13056.
25. Jha, N.K.; Gupta, S. *Testing of Digital Systems*; Cambridge University Press: Cambridge, MA, USA, 2002.
26. Childs, A.M.; Farhi, E.; Goldstone, J.; Gutmann, S. Finding cliques by quantum adiabatic evolution. *arXiv* **2000**, arXiv:quant-ph/0012104.
27. Ortiz, G.; Gubernatis, J.E.; Knill, E.; Laflamme, R. Quantum algorithms for fermionic simulations. *Phys. Rev. A* **2001**, *64*, 022319. [[CrossRef](#)]
28. Bergholm, V.; Vartiainen, J.J.; Möttönen, M.; Salomaa, M.M. Quantum circuits with uniformly controlled one-qubit gates. *Phys. Rev. A* **2005**, *71*, 052330. [[CrossRef](#)]
29. Gard, B.T.; Zhu, L.; Barron, G.S.; Mayhall, N.J.; Economou, S.E.; Barnes, E. Efficient symmetry-preserving state preparation circuits for the variational quantum eigensolver algorithm. *NPJ Quantum Inf.* **2020**, *6*, 1–9. [[CrossRef](#)]
30. Egger, D.J.; Mareček, J.; Woerner, S. Warm-starting quantum optimization. *Quantum* **2021**, *5*, 479. [[CrossRef](#)]
31. Cain, M.; Farhi, E.; Gutmann, S.; Ranard, D.; Tang, E. The QAOA gets stuck starting from a good classical string. *arXiv* **2022**, arXiv:2207.05089.
32. Khairy, S.; Shaydulin, R.; Cincio, L.; Alexeev, Y.; Balaprakash, P. Learning to optimize variational quantum circuits to solve combinatorial problems. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 2367–2375.
33. Sung, K.J.; Yao, J.; Harrigan, M.P.; Rubin, N.C.; Jiang, Z.; Lin, L.; Babbush, R.; McClean, J.R. Using models to improve optimizers for variational quantum algorithms. *Quantum Sci. Technol.* **2020**, *5*, 044008. [[CrossRef](#)]