Quantum Information Processing (2023) 22:19
https://doi.org/10.1007/s11128-022-03769-2

Quantum approximate optimization algorithm for Bayesian network structure learning

Vicente P. Soloviev[1] · **Concha Bielza**[1] · **Pedro Larrañaga**[1]

Received: 7 June 2022 / Accepted: 30 November 2022 / Published online: 13 December 2022

Abstract
Bayesian network structure learning is an NP-hard problem that has been faced by a number of traditional approaches in recent decades. Currently, quantum technologies offer a wide range of advantages that can be exploited to solve optimization tasks that cannot be addressed in an efficient way when utilizing classic computing approaches. In this work, a specific type of variational quantum algorithm, the quantum approximate optimization algorithm, was used to solve the Bayesian network structure learning problem, by employing $3n(n-1)/2$ qubits, where n is the number of nodes in the Bayesian network to be learned. Our results showed that the quantum approximate optimization algorithm approach offers competitive results with state-of-the-art methods and quantitative resilience to quantum noise. The approach was applied to a cancer benchmark problem, and the results justified the use of variational quantum algorithms for solving the Bayesian network structure learning problem.

Keywords Quantum approximate optimization algorithm · Variational quantum algorithm · Quantum optimization · Bayesian network structure learning

1 Introduction

Bayesian networks (BNs) are a family of probabilistic graphical models that compactly represent the joint probability distribution of a set of random variables [1]. Some of the most important characteristics of this type of model are its capability of graphically

✉ Vicente P. Soloviev
vicente.perez.soloviev@fi.upm.es

Concha Bielza
mcbielza@fi.upm.es

Pedro Larrañaga
pedro.larranaga@fi.upm.es

[1] Computational Intelligence Group, Universidad Politécnica de Madrid, Campus de Montegancedo, Madrid, Madrid, Spain

representing the uncertain knowledge contained in data and the possibility of including expert knowledge in the model. For these reasons, BNs are widely used in machine learning [2] for different applications [3, 4].

Regarding the high computational demands associated with BNs, two main problems have been studied in the literature: inference, which involves calculating a posterior probability distribution for some system variables when observing the values of other variables; and structure learning, which involves finding the optimal BN graph that best fits some given data. This paper is focused on the latter type of problem.

The Bayesian network structure learning (BNSL) problem is known to be NP-hard [5] because the number of possible structures for a Bayesian network with n nodes $h(n)$ increases more than exponentially with the number of variables n in the given data [6]:

$$\begin{aligned} h(1) &= 1 \\ h(n) &= \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} 2^{i(n-i)} h(n-i), \end{aligned} \tag{1}$$

and thus, heuristic search algorithms are commonly used. In classic computing, a wide range of approaches, such as particle swarm [7, 8], evolutionary algorithms [9, 10], simulated annealing [11], and Tabu search [12], have been applied to solve the BNSL problem in recent decades [13].

More recently, the capabilities of quantum computers to reduce the required execution time when facing different optimization tasks and to solve very complex problems that may not be approachable with classic computing methods have attracted much interest. Quantum computing [14] is based on quantum mechanics principles such as quantum entanglement and quantum superposition, which allow quantum algorithms to explore areas of the search spaces of optimization problems in a parallel and more efficient way.

Quantum annealing [15] is a quantum heuristic that can solve certain optimization problems exponentially faster than classic approaches. The BNSL problem has been mapped to a quadratic unconstrained optimization problem (QUBO) to be solved by using quantum annealing [16, 17].

In recent years, quantum machine learning [18] has attracted much attention. It combines machine learning algorithms and quantum computing theory to construct new hybrid algorithms that exploit the benefits of both fields. An example of such a hybrid algorithm is a variational quantum algorithm (VQA) [19], which is composed of a classic optimization loop that embeds a quantum subroutine. This routine measures a quantum parametric circuit (variational ansatz), while the classic loop optimizes the parameters of the quantum circuit in each iteration of the algorithm used to minimize the cost function. VQAs, such as the variational quantum eigensolver [20] and the quantum approximate optimization algorithm (QAOA) [21], are widely used for different combinatorial optimization problems [22–26]. Some studies [27] have proven that some types of optimization problems have landscape dispositions that makes the quantum and simulated annealing methods converge to local optimal solutions, while the QAOA is able to overcome this limitation and provide better solutions. Quantum

and simulated annealing have already been applied to BNSL; however, to the best of our knowledge, the use of the QAOA has not been found in the literature. In this work, we address the BNSL problem with the QAOA and analyse the performance of different variants of the algorithm.

Currently, the state-of-the-art of quantum computers is the noisy intermediate-scale quantum (NISQ) era, which is characterized by quantum computers with hundreds of qubits and no error correction. Thus, there is a need to develop algorithms that do not require a large number of qubits and that offer resilience to the presence of quantum noise (which characterizes quantum devices). VQAs, and QAOAs in particular, are some of the most promising algorithms in the NISQ era, as their implementations optimize the number of utilized qubits, and moreover, the variational ansatzs are expected to offer resilience to quantum noise such as amplitude and phase damping errors [28, 29]. We also analyse the resilience of the algorithm to the presence of different types of quantum noise, in the particular case of the BNSL problem.

The paper is organized as follows. Section 2 describes the fundamental basis of BNs, the classic methods for learning the structures of these models, and the QUBO formulation in which this work is inspired. Section 3 provides an overview of the QAOA approach. Section 4 describes how the QAOA ansatz is built and the characteristics integrated in our approach. Section 5 analyses the performance of the QAOA approach, the resilience of the algorithm to quantum noise, and a real application of the algorithm for solving the BNSL problem. Finally, Sect. 6 rounds the paper off with the conclusions of our work.

2 Structure learning of Bayesian networks

A BN can be properly defined as a pair $(G, \boldsymbol{\Theta})$ that represents a joint probability distribution over a set of random variables $\boldsymbol{X} = \{X_1, X_2, \ldots, X_n\}$. Its representation is given by (i) a directed acyclic graph (DAG) $G = (V, A)$, whose nodes V correspond to the set of random variables, $X_1, \ldots, X_n$, and arcs A that represent the probabilistic dependencies among the variables; and (ii) a set $\boldsymbol{\Theta}$ of parameters that define the conditional probability distribution of any variable X_i given its parents Π_i in the graph, where the parents Π_i of a variable X_i are the nodes that have arcs that reach X_i.

Given this definition, the probability distribution $P(\boldsymbol{X})$ over the set of variables $\boldsymbol{X}$ is defined as the product of the conditional probability distributions of all variables:

$$P(\boldsymbol{X}) = P(X_1, \ldots, X_n) = P(X_1 \mid \Pi_1)P(X_2 \mid \Pi_2) \cdots P(X_n \mid \Pi_n).$$

The BNSL problem is a very complex NP-hard problem [5] that is well-known in the state-of-the-art research due to the combinatorial explosion of possible DAGs which can represent the relationships among the variables in $\boldsymbol{X}$. Given a dataset $\mathcal{D}$ with n columns and as many rows as variable observations, the objective is to determine the DAG that better reflects the relationships among the variables $\boldsymbol{X} = \{X_1, X_2, \ldots, X_n\}$ found in $\mathcal{D}$. Two variables A and B are said to be conditionally independent given C if $P(A \mid B, C) = P(A \mid C)$, and thus, the values that B takes contribute nothing

to the certainty of A. Three main BNSL approaches are available: (i) the score-based approach, whose objective is to optimize a function that evaluates the quality of the structure given the data; (ii) the constraint-based approach, which performs some statistical tests to check the conditional independences among the variables; and (iii) hybrid methods that combine both approaches. In this paper, we focus on implementing a score-based approach.

Some well-known scores have been used for the BNSL problem. The decomposability property is desirable for computational reasons. This means that the score of a structure given some data is computed as the sum of the local scores of the subgraphs formed by each variable X_i and its parents Π_i,

$$\text{score}(G, \mathcal{D}) = \sum_{i=1}^{n} score_i(\Pi_i, \mathcal{D}),$$

where G denotes a DAG. The objective of the optimization task is to maximize this score.

Relevant decomposable scores used for BNSL are the Bayesian information criterion (BIC) [30], K2 score [31] and Bayesian Dirichlet equivalent uniform (BDeu) [32], among others.

2.1 QUBO formulation of BNSL

In this section, we describe the original QUBO formulation introduced in [16], on which we base our approach. The formulation is based on four different Hamiltonians: H_{score}, which optimizes the likelihood of a structure given the input data; $H_{\max}$, which ensures the maximum in-degree of each node to limit the Hamiltonian complexity; and H_{trans} and $H_{consist}$ which guarantee that the adjacency matrix that represents the BN is acyclic. The computed QUBO expression is

$$H(\boldsymbol{A}, \boldsymbol{R}, \boldsymbol{Y}) = H_{\text{score}}(\boldsymbol{A}) + H_{\max}(\boldsymbol{A}, \boldsymbol{Y}) + H_{\text{trans}}(\boldsymbol{R}) + H_{\text{consist}}(\boldsymbol{A}, \boldsymbol{R}),$$

where $\boldsymbol{A}$, $\boldsymbol{R}$ and $\boldsymbol{Y}$ are the quantum bits associated to the adjacency matrix, topological order, and the maximum in-degree restriction variables, respectively, for variables $X_1, X_2, \ldots, X_n$.

The QUBO problem formulation for solving a BNSL problem with n nodes requires

$$v_{\text{size}} = n(n-1) + \frac{n(n-1)}{2} + 2n, \tag{2}$$

quantum bits, where $n(n-1)$, $\frac{n(n-1)}{2}$, and $2n$ are the number of bits associated with the adjacency matrix, the topological order, and the number of variables needed to restrict the maximum in-degree, respectively.

Each of the four different Hamiltonians that compose $H(\boldsymbol{A}, \boldsymbol{R}, \boldsymbol{Y})$ are deeply explained in this section.

2.1.1 $H_{score}(A)$

For H_{score}, we need to introduce the concept of an adjacency matrix ($\boldsymbol{A}$):

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}, \tag{3}$$

where $a_{ij} = 1$ if there exists an arc from X_i to X_j and $a_{ij} = 0$ otherwise.

In this case, as BNs are represented as DAGs, the diagonal of this matrix is equal to zero, and thus, the bits of the diagonal are not required for the QUBO formulation. Then, $n(n-1)$ qubits are needed for the H_{score} Hamiltonian to learn a BN of n nodes, and

$$H_{\text{score}}(\boldsymbol{A}) = \sum_{i=1}^{n} H^i_{\text{score}}(\boldsymbol{a}_i), \tag{4}$$

$$H^i_{\text{score}}(\boldsymbol{a}_i) = \sum_{\substack{J \subset \{1,\ldots,n\} \setminus \{i\} \\ |J| \leq m}} (w_i(J) \prod_{j \in J} a_{ji}), \tag{5}$$

where $\boldsymbol{a}_i = (a_{1i}, \ldots, a_{ni})$, is the i-th column of $\boldsymbol{A}$, $w_i(J) = \sum_{l=0}^{|J|} (-1)^{|J|-l} \sum_{\substack{K \subset J \\ |K|=l}} s_i(K)$, in which $s_i(K)$ is the score of node i given the parent set K, and m is the maximum in-degree allowed.

Note that the constant term is $w_i(\emptyset) = s_i(\emptyset)$, which refers to the score of node X_i without its parents. If X_i has a single parent X_j, then the above equation simplifies to

$$H^i_{\text{score}}(\boldsymbol{A}) = w_i(\emptyset) + w_i(\{j\}) = s_i(\emptyset) + s_i(\{X_j\}) - s_i(\emptyset) = s_i(\{X_j\}).$$

Similarly, if X_i has two parents X_j and X_k,

$$\begin{aligned} H^i_{\text{score}}(\boldsymbol{A}) &= w_i(\emptyset) + w_i(\{j\}) + w_i(\{k\}) + w_i(\{j,k\}) \\ &= s_i(\emptyset) + (s_i(\{X_j\}) - s_i(\emptyset)) + (s_i(\{X_j\}) - s_i(\emptyset)) + w_i(\{j,k\}) \\ &= s_i(\{X_j\}) + s_i(\{X_k\}) - s_i(\emptyset) + w_i(\{j,k\}) \\ &= s_i(\{X_j\}) + s_i(\{X_k\}) - s_i(\emptyset) + s_i(\{X_j, X_k\}) - s_i(\{X_j\}) - \\ &\quad - s_i(\{X_k\}) + s_i(\emptyset)) = s_i(\{X_j, X_k\}). \end{aligned}$$

2.1.2 $H_{max}(A, Y)$

To ensure that the quantum algorithm only considers the maximum in-degree $m = 2$ to restrict the search space, the H_{max} Hamiltonian is implemented in a way such that

$2n$ quantum bits are needed. These quantum bits are represented as a matrix:

$$\boldsymbol{Y} = \begin{bmatrix} y_{11} & y_{12} \\ \vdots & \vdots \\ y_{n1} & y_{n2} \end{bmatrix},$$

where $y_{ij} \in \{0, 1\}$ are random binary variables. $\boldsymbol{Y}$ represents a slack variable used to reduce the inequality constraint of the maximum in-degree to an equality constraint.

The corresponding Hamiltonian results in $H_{max}(\boldsymbol{A}, \boldsymbol{Y}) = 0$ if the restriction is met, and $H_{max}(\boldsymbol{A}, \boldsymbol{Y}) > 0$ otherwise. Thus,

$$H_{\max}(\boldsymbol{A}, \boldsymbol{Y}) = \sum_{i=1}^{n} H^i_{\max}(\boldsymbol{a}_i, y_i),$$

$$H^i_{\max}(\boldsymbol{a}_i, y_i) = \delta_{\max}(m - \sum_{j=1}^{n} a_{ij} - y_i)^2 =$$

$$= \begin{cases} 0, & d_i \leq m \\ \delta_{\max}(d_i - m)^2, & d_i > m \end{cases}$$

where $y_i = \sum_{l=1}^{2} 2^{l-1} y_{il}$, $i = (1, \ldots, n)$, and $\delta_{\max} \in \mathbb{R}^+$ is a prefixed penalization term.

2.1.3 $H_{trans}(R)$ and $H_{consist}(A, R)$

To ensure the acyclicity of the adjacency matrix, we need to implement two different Hamiltonians, $H_{trans}(\boldsymbol{R})$ and $H_{\text{consist}}(\boldsymbol{A}, \boldsymbol{R})$. The former uses the topological order to check the transitivity of the graph, and the latter checks the consistency between the topological order and the adjacency matrix.

A topological ordering of a directed graph is a linear ordering of its vertices such that for every arc $i \rightarrow j$ from vertex i to vertex j, i comes before j in the ordering ($i < j$). The topological order is represented as

$$\boldsymbol{R}_{top} = \begin{bmatrix} r_{11} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{n1} & \cdots & r_{nn} \end{bmatrix},$$

where r_{ij} can be equal to 1, only if $i \leq j$ and $r_{ij} = 0$ if $i > j$ for the given QUBO formulation. The lower triangular portion of $\boldsymbol{R}_{top}$ provides no additional information to the upper triangular portion of $\boldsymbol{R}_{top}$. Moreover, if a matrix is acyclic, then the trace of $\boldsymbol{R}_{top}$ is equal to zero. Considering this, the variables used for the QUBO formulation are represented as part of the matrix $\boldsymbol{R}_{top}$, where the diagonal of the matrix and all

the elements below it have been removed

$$\boldsymbol{R} = \begin{bmatrix} r_{12} & r_{13} & \cdots & r_{1n} \\ \ddots & r_{23} & \cdots & r_{2n} \\ \vdots & \ddots & \cdots & \vdots \\ \cdots & \cdots & \ddots & r_{n(n-1)} \end{bmatrix}. \tag{6}$$

Then, $H_{trans}(\boldsymbol{R})$ is zero if the relation encoded in the $\boldsymbol{R}$ matrix is transitive and δ_{trans} otherwise:

$$\begin{aligned} H_{\text{trans}}(\boldsymbol{R}) &= \sum_{1 \le i < j < k \le n} H^{ijk}_{\text{trans}}(r_{ij}, r_{ik}, r_{jk}), \\ H^{ijk}_{\text{trans}}(r_{ij}, r_{ik}, r_{jk}) &= \delta_{trans}(r_{ik} + r_{ij}r_{jk} - r_{ij}r_{ik} - r_{jk}r_{ik}) \\ &= \begin{cases} \delta_{\text{trans}}, & [(i \le j \le k \le i) \vee (i \ge j \ge k \ge i)] \\ 0, & \text{otherwise} \end{cases} \end{aligned}$$

$H_{consist}(\boldsymbol{A}, \boldsymbol{R})$ is zero if the order encoded in the $\boldsymbol{R}$ matrix is consistent with the structure encoded in $\boldsymbol{A}$, and $\delta_{consist}$ otherwise.

$$\begin{aligned} H_{\text{consist}}(\boldsymbol{A}, \boldsymbol{R}) &= \sum_{1 \le i < j \le n} H^{ij}_{\text{consist}}(a_{ij}, a_{ji}, r_{ij}), \\ H^{ij}_{\text{consist}}(a_{ij}, a_{ji}, r_{ij}) &= \delta_{\text{consist}}(a_{ji}r_{ij} + a_{ij}(1 - r_{ij})) = \\ &= \begin{cases} \delta_{\text{consist}}, & (a_{ji} = r_{ij} = 1) \vee (a_{ij} = 1 \wedge r_{ij} = 0) \\ 0, & \text{otherwise} \end{cases} \end{aligned}$$

where $\delta_{\text{trans}} \in \mathbb{R}^+$ and $\delta_{\text{consist}} \in \mathbb{R}^+$ are prefixed penalization terms.

3 Quantum approximate optimization algorithm

Many real optimization problems can be framed as combinatorial problems. The QAOA was proposed by Farhi, Goldstone and Gutmann [21] for solving combinatorial optimization problems.

A combinatorial optimization problem is formulated by n bits and m clauses. Each of the clauses affects a subset of bits and is satisfied when this subset is assigned to certain values. Satisfiability asks if a string that satisfies every clause is available. The objective is to maximize the following equation:

$$C(z) = \sum_{\alpha=1}^{m} C_\alpha(z), \tag{7}$$

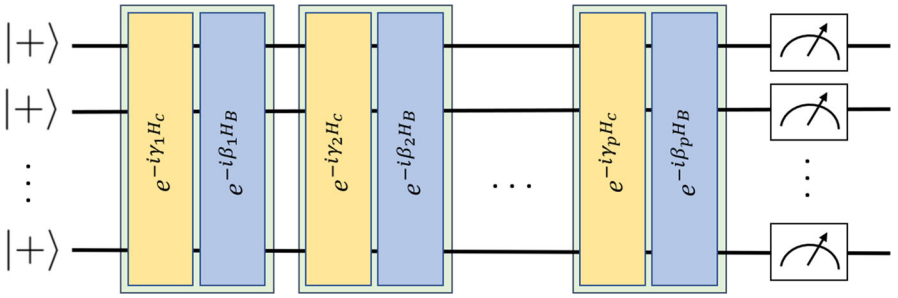

Fig. 1 A quantum parametric circuit with p layers and $2p$ parameters. The initial state is a superposition of all the possible computational states, and after applying the p layers, a measurement along the Z axis of all the qubits is performed

where $z = z_1 z_2 \cdots z_n$ is a bit string with n bits and $C_\alpha(z) = 0$ if clause α is not satisfied by the z string ($C_\alpha(z) = 1$ otherwise).

The QAOA [21] uses a quantum parametric circuit (variational ansatz) which is built for a specific combinatorial optimization problem and represents a quantum parametric state. In each iteration of the algorithm, the parameters are optimized to find the bit string z' for which $C(z')$ is the maximum of C. The quantum circuit consists of p layers, and each layer is formed by two different operators that encode the cost function to be optimized (Fig. 1): the cost operator $U(H_C, \gamma)$ parameterized by γ,

$$U(H_C, \gamma) = e^{-i\gamma H_C} = \prod_{\alpha=1}^{m} e^{-i\gamma C_\alpha}, \tag{8}$$

and the mixed operator $U(H_B, \beta)$ parameterized by β,

$$U(H_B, \beta) = e^{-i\beta H_B} = \prod_{j=1}^{n} e^{-i\beta\sigma_j^x}, \tag{9}$$

where $B = \sum_{j=1}^{n} \sigma_j^x$ and σ_j^x is the mapping of z_j from a binary variable to quantum spin {+1, -1}, which is rotated in the X-axis. The parameters γ and β are restricted to lie between 0 and 2π, as they represent the rotation angles (degrees) over the qubits.

In the literature [22, 26], it has been shown that by increasing the number of layers of a circuit, the algorithm improves its performance and better solutions are obtained. For $p \to \infty$, the QAOA approximates the adiabatic quantum evolution path [33], which is how the algorithm starts from the initial state and converges to a solution. The quantum adiabatic path is approximated in p steps. However, increasing p also increases the depth of the circuit, which entails other disadvantages, such as facing the quantum noise embedded in quantum computers. It is then necessary to find the ideal p so as not to drastically increase the depth of the circuit, while still being able to find the final solutions reached by the quantum adiabatic path.

The quantum parametric state is represented as:

$$|\psi(\boldsymbol{\gamma}, \boldsymbol{\beta})\rangle = U(H_B, \beta_p)U(H_C, \gamma_p)\ldots U(H_B, \beta_1)U(H_C, \gamma_1)|s\rangle,$$

where $p \geq 1$, $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_p)$, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$, and $|s\rangle$ is the uniform superposition state over all possible computational states. A quantum circuit with p layers and a total of $2p$ parameters $(\gamma_1, \beta_1, \ldots, \gamma_p, \beta_p)$ to be optimized is shown in Fig. 1.

The role of the optimizer is to find the optimal parameters $(\boldsymbol{\gamma}_{opt}, \boldsymbol{\beta}_{opt})$ such that the expectation value that encodes the cost function to be optimized,

$$f(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \langle\psi(\boldsymbol{\gamma}, \boldsymbol{\beta}) \mid C|\psi(\boldsymbol{\gamma}, \boldsymbol{\beta}),$$

is minimized. $f(\boldsymbol{\gamma}, \boldsymbol{\beta})$ encodes the total energy of the system, which should be minimized. Such an expectation can be obtained by performing measurements along the Z-axis of the quantum system and computing the following expression:

$$E = \frac{1}{t}\sum_{z\in Z} C(z)N_z, \tag{10}$$

where N_z is the number of times solution z is measured by executing the circuit t times and Z is the set of possible basis states obtained by the circuit. The total energy of the system, E, is expected to be minimized for increasing p.

The pseudocode of the QAOA is quite simple once the quantum parametric circuit is built. In each iteration of the algorithm, a new set of $(\boldsymbol{\gamma}, \boldsymbol{\beta})$ parameters is given to the circuit, which is run t times. Then, the optimizer computes the expectation value (Eq. (10)) and proposes a new set of $(\boldsymbol{\gamma}, \boldsymbol{\beta})$ parameters. The loop is repeated until a stopping criterion is met. In each iteration, when the quantum circuit is executed t times, a probability distribution is computed for the solutions. The expectation value (Eq. (10)) is also referred to in the machine learning literature as the uncertainty among the solutions, and is expected to be reduced as the algorithm runtime increases.

Thus, the quantum trial state is prepared, and we then optimize the parameters to bring the trial state as close as possible to the target state. The quality of the QAOA solutions heavily depends on the quality of the parameters (evaluated by Eq. (10)) obtained by the optimizer used during the runtime of the algorithm.

4 Method

This section explains how the variables are deployed in the QAOA approach and how the Hamiltonian is transformed to quantum circuits. All the implemented software is codified by using Qiskit-0.18.1 [34] and myQLM-1.5.1 [35] and is freely available in GitHub[1].

[1] Github repository with implemented code: https://github.com/VicentePerezSoloviev/QAOA_BNSL_IBM

4.1 QAOA variables

To make the QAOA able to manage the QUBO variables ($\boldsymbol{A}$, $\boldsymbol{R}$, $\boldsymbol{Y}$), it is necessary to arrange them in such a way that they are represented as a vector. Thus, the previous QUBO variables are disposed as a vector of qubits with a size of v_{size} (Eq. (2)).

As explained in Sect. 3, the QAOA is a hybrid approach in which the classic part of the algorithm computes the cost function of the obtained solutions and the expectation value of all the solutions of the corresponding iteration. Our proposal also computes the maximum in-degree ($m = 2$) of the solutions and penalizes those that do not meet the restriction in a classic manner. Upon doing so, v_{size} reduces to

$$v_{\text{size}-\text{QAOA}} = n(n-1) + \frac{n(n-1)}{2} = \frac{3n(n-1)}{2}, \tag{11}$$

where $v_{\text{size}-\text{QAOA}} < v_{size}\ \forall\ n$, because the $\boldsymbol{Y}$ variables in the Hamiltonian are not considered.

Thus, the vector $\boldsymbol{q}$ needed to solve the BNSL problem for a BN of n nodes by using the QAOA is an array of size $v_{size-QAOA}$,

$$\begin{aligned}\boldsymbol{q} &= (q_1, q_2, \ldots, q_{n*(n-1)}, \ldots, q_{v_{\text{size}-\text{QAOA}}}) \\ &= (a_{12}, a_{13}, a_{1n}, \ldots, a_{n(n-1)}, r_{12}, r_{13}, r_{1n}, r_{23}, \ldots, r_{(n-1)n}),\end{aligned}$$

where a_{ij} and r_{ij} are defined in Eq. (3) and Eq. (6), respectively.

4.2 QAOA circuit

In Sect. 3, we have seen that the process of preparing the quantum state during the operation of the QAOA is composed of three elements:

1. Preparing an initial state of superposition.
2. Applying the cost operator $U(H_C, \gamma)$ (Eq. (8)).
3. Applying the mixed operator $U(H_B, \beta)$ (Eq. (9)).

4.2.1 Initial state

The initial state used during the QAOA is usually the superposition of all the basis states, which is defined as:

$$|\psi_0\rangle = \left(\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)\right)^{\otimes v_{size-QAOA}},$$

where $\otimes v_{size-QAOA}$ refers to the number of qubits used in the quantum state (Eq. (11)).

To reach the superposition state of all the possible basis states, we apply Hadamard gates to each qubit ($\pi/2$ degrees in the qubit Y-axis).

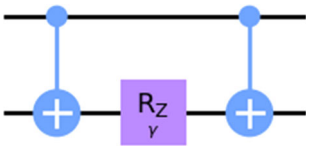

Fig. 2 A multiplication of two Pauli operators $Z_i Z_j$ is represented in the quantum circuit as a combination of two CNOT gates between qubits i and j and a rotation-Z gate in one of them

4.2.2 Applying the cost operator $U(H_C, \gamma)$

As the maximum in-degree verification is implemented in the classic part of the VQA, the Hamiltonian to be implemented is reduced to

$$H(\boldsymbol{D}, \boldsymbol{R}) = H_{score}(\boldsymbol{D}) + H_{trans}(\boldsymbol{R}) + H_{consist}(\boldsymbol{D}, \boldsymbol{R}). \tag{12}$$

The Hamiltonian described in Eq. (12) involves binary variables in $\{0, 1\}$, and the QAOA needs the Hamiltonian to be transformed into a spin Hamiltonian where all the variables are spins in $\{-1, 1\}$. Thus, each binary variable X_i in the QUBO formulation must be transformed as $X_i \rightarrow \frac{1-Z_i}{2}$, where Z_i is the Pauli Z operator that has eigenvalues of $\{-1, +1\}$ and acts on qubit i while ignoring all other qubits:

$$Z_i = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Thus, the QUBO formulation is transformed into a formula in which all the variables involved are $\boldsymbol{q}$. The QAOA is a circuit model-based approach, and thus, each Pauli operator Z_i is a quantum gate in the QAOA circuit. Each operator is a rotation-Z gate of qubit i, and each multiplication of two Pauli operators $Z_i Z_j$ is a sequence of three gates in qubits i and j (Fig. 2).

Each Z_i gate has a rotation angle that is parameterized by γ and influenced by the structure evaluation scores (Eq. (5)).

4.2.3 Applying the mixed operator $U(H_B, \beta)$

The last step of the QAOA circuit is the mixed operator. This operator consists of applying a rotation-X gate in all the qubits of the circuit with parameter β.

An example of the resultant circuit is shown in Fig. 3. For each extra layer, a cost and mixed operators $U(H_C, \boldsymbol{\gamma})$ and $U(H_B, \boldsymbol{\beta})$ should be added sequentially with their respective parameters to the actual circuit to increase p.

4.3 Conditional value at a risk

As explained in Sect. 3, in each iteration, the expectation value of the measurements along the Z-axis is computed for minimization. This value can be computed classically by using Eq. (10) after executing the QAOA circuit t times.

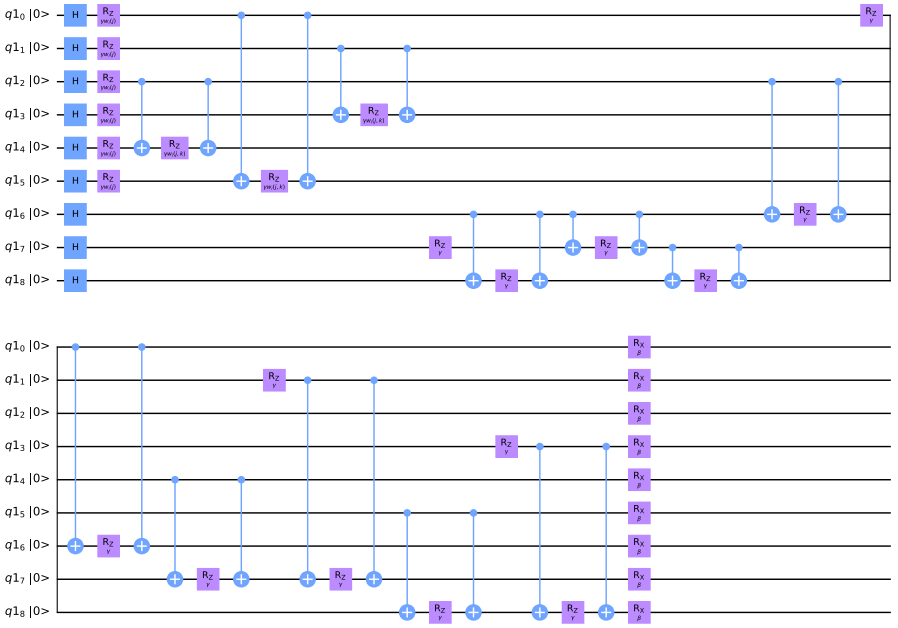

Fig. 3 A QAOA circuit example for a BNSL problem with three nodes and one layer. This variational ansatz has $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ parameters as the parameters of the first layer of the circuit

We add a modification to the standard QAOA baseline so that not all the solutions obtained after the measurement process are considered for the expectation value computation. Thus, instead of computing the expectation value, we compute the conditional value at a risk ($CVaR_\alpha$), which is a measure that takes only the tail of the distribution of the solutions obtained after measurement into account and is widely used in finance [36]. The $CVaR_\alpha$ is widely used in different VQAs, such as the QAOA and the variational quantum eigensolver [37], as it has been proven to lead to faster convergence to better solutions.

Evolutionary algorithms [38] have several characteristics in common with VQAs. A well-known type of evolutionary algorithm is the estimation of distribution algorithm [39], which in each iteration sample new solutions from a probability distribution learned from the best old solutions, and then selects the best solutions to update this probability distribution. The top solutions are a percentage of the total set of solutions. This is equivalent to the behaviour of the QAOA considering the $CVaR_\alpha$.

A new parameter α is added to the QAOA implementation, and its function is to select the best solutions from the set of solutions measured after executing the QAOA circuit t times. Given a cumulative density function F_K among all the basis states obtained after measuring the QAOA circuit t times, computing the $CVaR_\alpha$ implies computing the expectation value for the α-head of F_K assuming that samples are sorted in a decreasing order. This selection henceforth referred to as αK with size

$\lceil \alpha t \rceil$. Then, the $CVaR_\alpha$ is defined as

$$CVaR_\alpha = \frac{1}{\lceil \alpha t \rceil} \sum_{z \in \alpha K} C(z) N_{z_{\alpha K}}, \tag{13}$$

where $C(z)$ is defined in Eq. (7) and $N_{z_{\alpha K}}$ is the number of times that solution z is measured during selection αK. The α parameter is defined in the interval (0, 1], such that for $\alpha = 1$, the entire set of solutions is considered for the expectation value computation (Eq. (10)), and for a decreasing α, the number of solutions considered for computing the $CVaR_\alpha$ is reduced.

5 Results

In this section, some results are shown for the BNSL problem after applying the proposed QAOA. Some plots are first given to show how the QAOA performs in terms of cost function minimization (Sect. 5.1). Then, a performance evaluation of the QAOA considering different types of simulated noise is analysed (Sect. 5.2). Finally, a real example of BNSL is shown (Sect. 5.3).

Note that the real limitation of this algorithm has been the number of available qubits in the available architectures. In our experiments, the QAOA looks for the optimal BN structure in a search space containing 543 possible structures for $n = 4$ and 29281 structures for $n = 5$ (see Eq. (1)). This architecture restriction is imposed due to the number of qubits that we can access at the moment in Qiskit and myQLM. Despite the fact that these problem sizes are far from those examined by the classic BNSL approaches, the number of qubits that companies such as IBM and Google are offering is increasing rapidly, and thus, the use of VQAs is increasingly justified.

The optimizer used in the implementation is the constrained optimization by linear approximation algorithm [40], which is widely used in the state-of-the-art VQAs [41].

5.1 QAOA performance

The QAOA approach aims at minimizing the uncertainty among the solutions obtained after completing the QAOA circuit measurement process, which is optimized by the minimization of the expectation value; see Eq. (10). It is expected that by increasing the number of layers p of the circuit, the expectation value among the solutions must decrease. The task of the optimizer is to iteratively search the optimal parameters $(\boldsymbol{\gamma}_{opt}, \boldsymbol{\beta}_{opt})$ of the QAOA circuit to minimize the expectation value. When the optimizer converges to a solution, the parameters $(\boldsymbol{\gamma}_{opt}, \boldsymbol{\beta}_{opt})$ are set to those of the quantum circuit. Figure 4 shows an example histogram of the obtained solutions. This experiment is performed with different numbers of layers ($p = 2, 4, 6, 8$) to show the differences between the resultant histograms.

Figure 4 shows that increasing the number of layers, clearly minimizes the uncertainty. Note the existence of two clear optima with similar costs for $p = 8$; this is not as clear for $p = 2$. Moreover, a reduction in the number of solutions that are

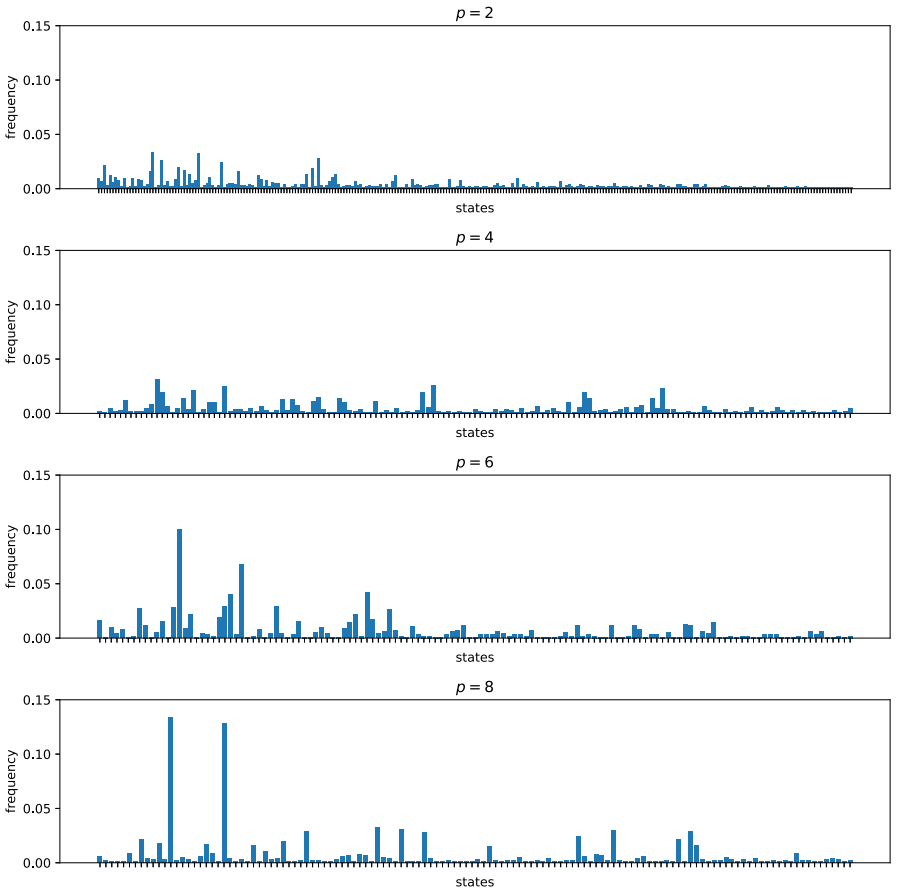

Fig. 4 Histograms for different numbers of layers p for the same BNSL problem. The Y- and X-axes represent the frequency, and the solutions, respectively. The names of the solutions have been removed from the X-axis for aesthetics, but how they are sorted is the same for each subplot

represented by the X-axis for increasing p is clearly visible. The solutions obtained with $p = 2$ have a density close to 0; for $p = 8$, they tend to have a density equal to 0 and thus become insignificant in the corresponding subplot.

As shown before, the QAOA approach is able to reduce the uncertainty among the solutions for the implemented Hamiltonian. However, the proposed approach heavily depends on the random pair of $(\boldsymbol{\gamma}, \boldsymbol{\beta})$ parameters from which the optimizer is initialized. Thus, depending on the initialization, different solutions might be proposed in different executions for the same Hamiltonian problem.

The minimization of the $CVaR_\alpha$ (Eq. (13)) is analysed next by using a random dataset of 4 variables. In Fig. 5, a comparison of the performances achieved by the QAOA for different values of the parameter α and the number of layers p in the QAOA circuit is shown. Note that increasing the number of layers decreases the mean best cost although it increases the depth of the circuit and the computing time.

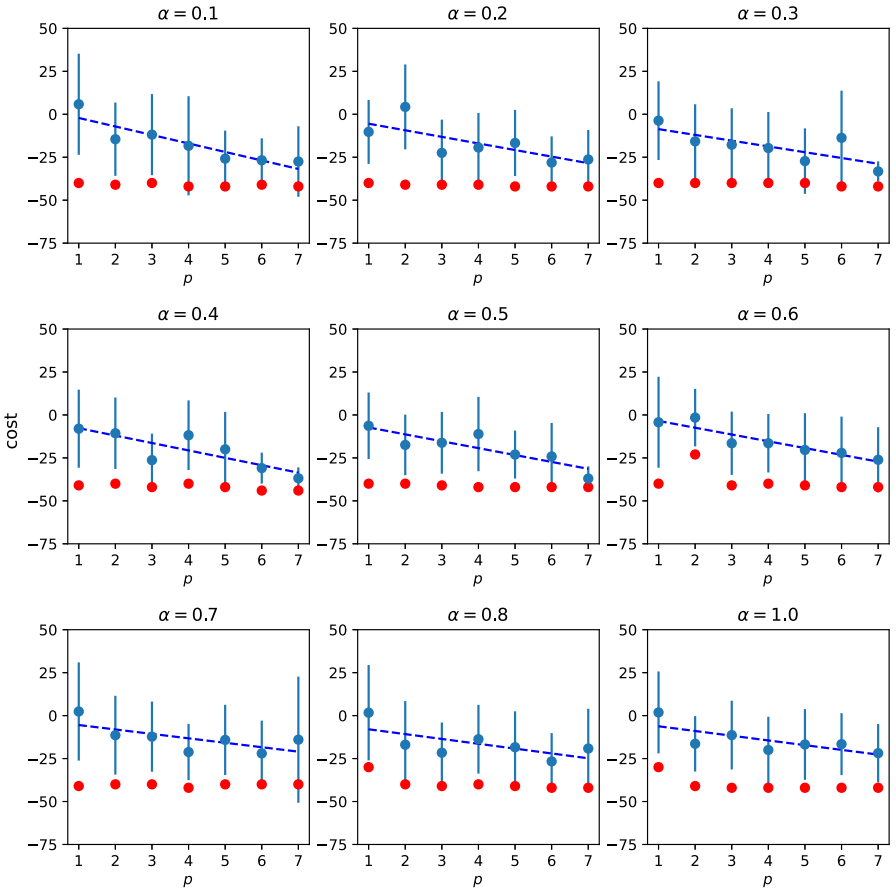

Fig. 5 Minimization of the $CVaR_\alpha$ (Eq. (13)) for different numbers of layers p (X-axis) in the QAOA circuit and different values of the parameter α. Blue dots and error bars correspond to the means and standard deviations of the best results found after executing the QAOA 50 times, respectively, and red denotes the minimum costs found in those executions. Dashed trend lines are plotted to guide the human eye

In Fig. 5, we observe an improvement as p increases and α takes intermediate values. The best solutions are found for intermediate values of α in the range $[0.3, 0.5]$ and $p = 7$. Note that this improvement with increasing p is not as noticeable for large values of α ($\alpha \to 1.0$) as it is for the lowest values ($\alpha \to 0$). Despite these results, we claim that it is not necessary to increase the number of layers in the QAOA circuit to find the best results. Figure 5 shows cases in which the same results are obtained, with fewer layers but different values of α.

In Sect. 4.3, a quantitative comparison of the QAOA approach with estimation of distribution algorithms is provided. This type of algorithm tends to converge to local optima when the percentage of solutions selected to update the probability distribution is too low. However, as shown in Fig. 5, the QAOA does converge to the same solution for any value of α. This is analysed in Fig. 6, where the mean numbers of iterations

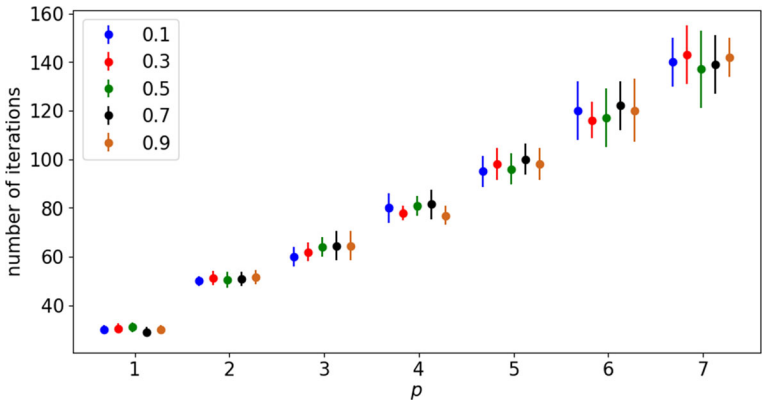

Fig. 6 Means and standard deviations of the number of iterations until convergence after executing the QAOA 50 times for different values of p (X-axis) and α (colours)

required until convergence are shown for different executions of the QAOA approach and different values of p. Independently of the value of α, the number of iterations remains approximately constant for the same value of p, whereas it increases with p.

5.2 Noise resilience

Two main disadvantages of NISQ computers are their limited numbers of qubits and the presence of quantum noise. Thus, there is a need to implement approaches that offer resilience to quantum noise and to optimize the number of qubits used to solve the given problem. It has been shown that VQAs can compensate for quantum errors such as over-/under-rotations [19]. However, a wide range of studies have analysed the QAOA in different applications to determine the hard limit of its resilience to quantum noise [24, 25, 42–45]. In other words, we analyse how much noise the QAOA can bear without worsening its optimization behaviour.

To perform this analysis, different noise channels ε have been constructed to try to simulate the different decoherence quantum noises. The quantum channels which define the quantum noises are described in Appendix 1 using Kraus operators formalism.

Firstly, the amplitude damping channel (ε_{AD}) is considered, which describes the energy dissipation of a quantum system. This channel involves a parameter $\omega \in [0, 1]$ that tunes the probability of a quantum state to decay from state $|1\rangle$ to $|0\rangle$. Then, $\omega = 0$ represents no amplitude damping error, while $\omega = 1$ is the maximal noise, which in this case means the state $|0\rangle\langle 0|$.

Secondly, the phase damping channel (ε_P) is considered, which describes the loss of information of a quantum state without loss of energy. It can be due to the iteration of the system with the environment, which can cause random phase shifts in the quantum states. This channel involves a parameter $\omega \in [0, 1]$ that tunes the probability of a quantum state to lose information. Then, $\omega = 0$ represents no phase damping error, while $\omega = 1$ is the maximal noise, which in this case means any linear combination

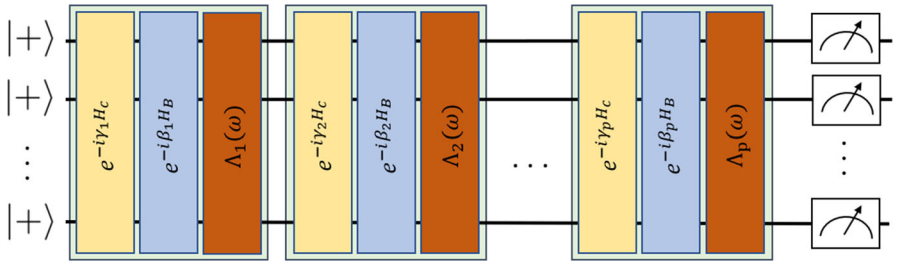

Fig. 7 A quantum parametric circuit with p layers and $2p$ parameters. The initial state is a superposition of all the possible computational states, and after applying the p layers, a measurement along the Z axis of all the qubits is performed. Each layer is composed by the noise channel and both the cost and mixed operators

of $|0\rangle\langle 0|$ and $|1\rangle\langle 1|$. The longitudinal relaxation time (T1) and the dephasing time (T2) are two quality metrics related to the amplitude and phase damping channels, respectively. After T1, the quantum behaviour is no longer that predictable, and T2 accounts for the phase loss of a qubit, after which the qubit can perform different phase rotations from those it was required to perform.

Thirdly, the depolarizing channel (ε_D) is considered to describe the probability of a qubit to be depolarized. The depolarization of a qubit is the replacement of the current state with the mixed state $I/2$. This event occurs with probability $\omega \in [0, 1]$, and the qubits leave untouched with probability $1-\omega$. Then, $\omega = 0$ represents no depolarizing error, while $\omega = 1$ is the maximal noise, which in this case means the mixed state $I/2$.

Other types of quantum noises exist and are studied in the literature, such as the cross-talk error, which is neglected in this study, as the main focus is to analyse the effects of decoherent noise channels. Similarly, the readout error is not considered as this type of noise is independent from the ansatz design.

For this study, the amplitude and phase damping noises are only applied to the one-qubit gates, while the depolarization noise is applied to the two-qubit gates, such as CNOT gates.

To carry out these simulations, a quantum circuit has been designed alternating the QAOA operators (Eq.(8) and Eq.(9)) with each noise channel operators parameterized by ω. By this way, each layer of the QAOA is accompanied by a noise operator, trying to replicate the behaviour of the processes in real quantum computers. Three different quantum circuits are used in order to analyse each quantum channel separately. The quantum parametric state is defined as

$$\varepsilon(|\psi(\boldsymbol{\gamma}, \boldsymbol{\beta})\rangle) = \Lambda_p(\omega)U(H_B, \beta_p)U(H_C, \gamma_p)\ldots\Lambda_1 U(H_B, \beta_1)U(H_C, \gamma_1)|s\rangle,$$

where $p \geq 1$, $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_p)$, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$, $|s\rangle$ is the uniform superposition state over all possible computational states, and $(\Lambda_1, \ldots, \Lambda_p)$ are the noisy operators that simulate each quantum channel. An example of a quantum circuit with p layers is shown in Fig. 7 where the orange operators represent the quantum channels.

The operators used in the circuit construction represent the same quantum channel and affect to all the qubits in the quantum system. Their application is defined onto a quantum state as:

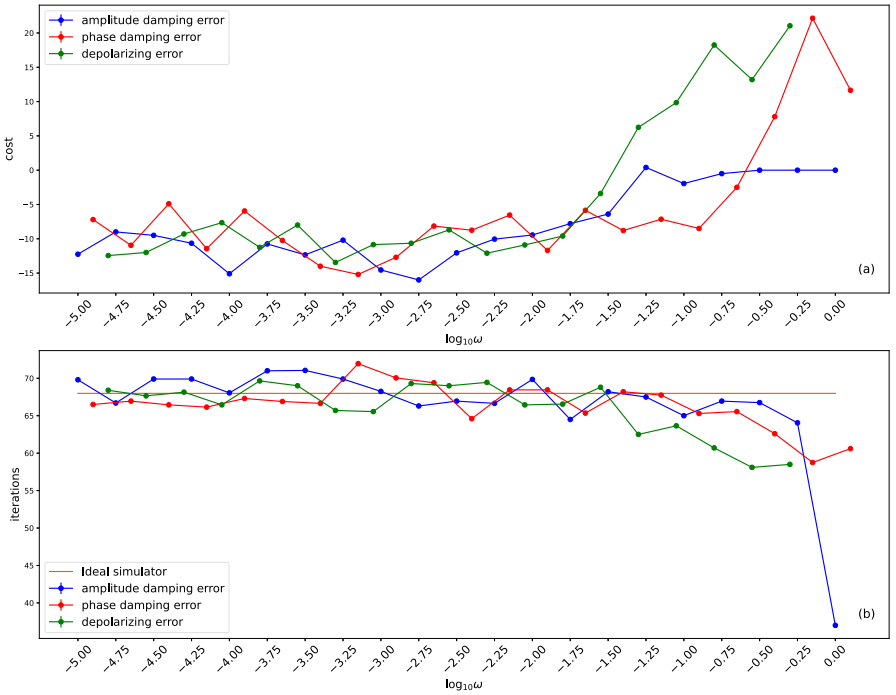

Fig. 8 Mean best costs (**a**) and mean numbers of iterations until convergence (**b**) as a function of the noise amplitude ω. Blue, red and green lines represent the simulated amplitude, phase damping and depolarizing noise models, respectively. For the detailed mean and standard deviation values of these experiments, see Appendix B. Fifty different executions of the QAOA algorithm were run for the four-node BNSL problem, and $p = 3$

$$\varepsilon(\psi) = \Lambda(\omega)\psi = \left(\bigotimes_{i=1}^{v_{size-QAOA}} \Lambda_i^{(1)}\right)\psi,$$

where $\varepsilon(\psi)$ is the resultant state after applying the quantum channel over the state ψ, ω is the parameter of the quantum channel and $\Lambda^{(1)}$ is the application of the one-qubit quantum channel in each qubit of the QAOA ansatz.

Figure 8a shows the resilience of the algorithm to the three types of previously explained noise channels. The figure shows the mean best costs for different values of ω. Supplementary numerical results are found in Appendix B.

To analyse the resilience of the simulated noises, we are interested in, fixing a range of mean cost values, knowing which is the value of ω that makes the mean best cost found to be out of this range. Since in Fig. 8a for the different noises when there is nearly no noise ($\omega \to 0$), the cost values are within the range $[-15, -5]$, we are interested in the values of ω at which the different noise channels make the found costs to leave this range. Thus, we set the values of $\omega = 10^{-2}$, $\omega = 10^{-1.5}$, $\omega = 10^{-0.75}$ for the depolarized, amplitude damping and phase damping noise channels, respectively.

The number of iterations needed by the algorithm to converge are shown in Fig. 8b (supplementary numerical results are at Appendix B). In the figure, we can generally

observe a decreasing trend in the number of average iterations correlated with the growth of ω. The orange line shows the average number of iterations required to run the algorithm with the ideal simulator without quantum noise. Analysing the values of ω identified in Fig. 8a, we can see how approximately at the same values, in Fig. 8b, there is a noticeable decrease tendency in the number of average iterations. Considering the worsening of the mean best cost found and the decrease in the number of iterations, we can conclude that the algorithm is falling into local optima solutions when $\omega \to 1$.

Considering the previous analysis, we can present the following conjectures which will be extended below: (i) the approach is less resilient to the depolarization noise channel than to the other channels; (ii) the amplitude damping noise channel makes the mean best cost to converge to solutions with costs close or equal to zero for values of $\omega \geq 10^{-1.25}$; (iii) the approach is resilient to the phase damping noise channel up to values of $\omega \geq 10^{-1}$.

Firstly, despite the fact that the depolarized noise channel has only been applied on the two-qubit gates, the design of the variational ansatz from the QUBO makes the number of CNOT gates high and therefore the depolarized noise channel has a high impact on the results shown in Fig. 8a. Consequently, we can say that the algorithm is resilient to this noise channel up to values of $\omega = 10^{-2}$. Higher values of ω rise noticeably the mean best cost found increasing also the standard deviation between solutions (Appendix B).

Secondly, it is worth noting that amplitude damping is the only noise channel that converges to mean best costs of zero with no standard deviation, while the others reach mean best cost values above 15 with a high standard deviation among the best results found when $\omega \to 1$ (Appendix B). Due to the effects that the amplitude damping noise channel has on the quantum states, we can affirm that the algorithm is converging to a partial solution, in which only some arcs are equal to those in the real BN, but all other values in the adjacency matrix are zero. The QUBO contemplates that those solutions such as $|00\cdots0\rangle$ should be penalized, and so it does not converge to them. Rather, in this case, it is converging to a solution in which a large majority of the values in the adjacency matrix are zeros, except for the correctly pointed arcs. Therefore, for maximum values of ω, the mean best cost of the solutions found is better than what we find for maximum values of ω with other noise channels.

Thirdly, the phase damping error channel is the most resilient if only the value of ω that causes the mean best cost to fall out of the range of values fixed in the analysis of Fig. 8a is taken as a reference. However, when $\omega \geq 10^{-0.75}$, the standard deviations among the solutions are increased (Appendix B).

After analysing Fig. 8, we conclude that our approach has better resilience to the amplitude damping error and phase damping error than to depolarizing error. However, we believe that it is of interest to analyse the convergence noted for the case of the amplitude damping noise channel.

In Fig. 9, a deeper analysis of the amplitude damping channel is shown for different values of p. The figure shows the influence of the quantum noise on the results based on the number of layers p of the QAOA circuit. As p increases, the depth of the circuit also increases, and thus, a greater part of the qubit lifespan will be executed outside the coherence times defined by T1 and T2. The results obtained by the QAOA for higher values of p are much worse than those obtained with low values of the parameter, and

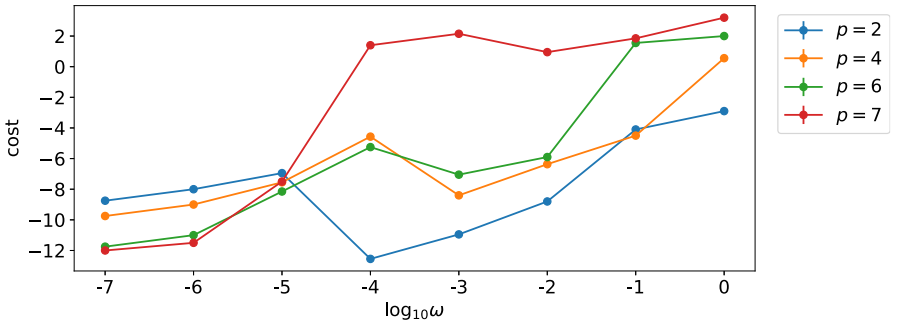

Fig. 9 Mean best cost obtained for different values of the parameters ω and p for 50 executions of the QAOA algorithm for the BNSL problem considering the amplitude damping error

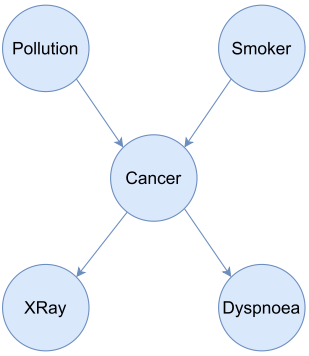

Fig. 10 Original `Cancer` BN structure composed of five nodes that represent five discrete variables and four arcs

in most cases a worsening of the mean best cost is observed as ω increases. From this analysis, we conjecture that as the number of layers p increases, the QAOA becomes less resilient to the amplitude damping error.

5.3 BNSL from real-world data

In this section, the QAOA approach is applied to a real BNSL problem by using the `Cancer`[2] benchmark. The `Cancer` BN (Fig. 10) has five discrete variables, from which we sampled three different datasets using probabilistic logic sampling [46] with 500, 1000 and 10000 instances. The structures provided by the QAOA are compared to the original BN structure through the structural Hamming distance metric, where a value of zero means that the QAOA approach fully recovers all the arcs of the original BN. We consider two different structure evaluation scores: the BIC and BDeu scores. This experiment is limited to five nodes due to the limit of qubits we can access with Qiskit and myQLM, for which the search space is composed of 29281 possible BN structures.

[2] https://www.bnlearn.com/bnrepository/discrete-small.html#cancer

Table 1 Comparison of the QAOA approach with three different classic approaches and simulated quantum annealing. The experiment is executed 10 times for each of the non-deterministic (SQA and QAOA) approaches (the best results are shown), and three different dataset sizes are simulated with 500, 1000 and 10000 instances. BIC and BDeu scores are used for the local BN structure evaluation. The structural Hamming distance metric is shown for each experiment

	α	500		1000		10000	
		BIC	BDeu	BIC	BDeu	BIC	BDeu
HC	–	4	4	4	4	**0**	**0**
Tabu	–	4	4	4	5	**0**	**0**
MMHC	–	4	4	4	4	**0**	**0**
SQA	–	5	5	4	5	3	2
QAOA	0.9	**0**	**0**	1	**0**	**0**	1
QAOA	0.7	1	1	1	**0**	**0**	**0**
QAOA	0.5	1	1	**0**	1	1	1
QAOA	0.3	**0**	**0**	1	**0**	**0**	1

Bold values indicate the best results

We compare the results obtained by the QAOA approach with those of two score-based algorithms: the hill climbing (HC) [47] and Tabu search [12] algorithms; with that of a hybrid algorithm: max–min hill climbing (MMHC) [48]; and with that of the simulated quantum annealing (SQA), which is a version of quantum annealing executed in the quantum learning machine [35]. The QAOA results are shown in Table 1 for different values of α, where the number of layers p is optimized for each case. The parameters of the algorithms are optimized, and the best results are shown in Table 1.

The QAOA approach obtains better results than those of classic approaches regardless of the α parameter for a low number of instances. As the number of instances increases, the α parameter seems to have a larger influence on the obtained results. The QAOA improves upon the results of the SQA in all the experiments.

6 Conclusions

In this work, the BNSL problem was approached by using the QAOA variational quantum algorithm. The problem was transformed into a Hamiltonian energy function, and then translated into a QAOA parametric circuit to be optimized into a classic loop.

A remarkable uncertainty reduction across all the possible solutions was shown for increasing number of QAOA circuit layers. We introduced the concept of the $CVaR_\alpha$ to reduce the number of solutions selected for computing the expectation value of each iteration. Considering this new parameter, we could observe that it was not necessary to increase the number of circuit layers to obtain the best results since the QAOA was able to converge to similar solutions by tuning the α parameter.

The NISQ-era quantum computers are characterized by the quantum noise embedded in these systems. We analysed the performance of our approach while simulating

different types of quantum noise, and our results show that the QAOA is resilient to quantum noise over a range of noise amplitudes. More specifically, our approach offers good performance when considering the amplitude damping error, which was more deeply analysed.

Our approach was also applied to the `Cancer` benchmark, and a comparison with other optimizers was shown with different structure evaluation scores and dataset sizes. The QAOA found the global optimum for every size-score combination and seemed to outperform classic and quantum approaches on any dataset.

Considering the results obtained in this work, we believe that the use of VQAs to solve the BNSL problem is justified. As future lines of research, we suggest considering the warm starting scenario [49] and a more in-depth analysis of other quantum noises with different quantum computers. Even though the sizes of the BNs that the QAOA has faced are small, we believe that this approach will be worthwhile for further research as the number of qubits that can be freely accessed increases.

Acknowledgements The authors would like to thank A. Gomez from Centro de Supercomputacion de Galicia (CESGA) for helpful discussions. We would like to thank Centro Singular de Investigación en Tecnoloxías Intelixentes (CITIUS) and CESGA for access to the computers where the experiments were carried out. We acknowledge the access to advanced services provided by the IBM Quantum Researchers Program. This work has been partially supported by the Spanish Ministry of Science and Innovation through the PID2019-109247GB-I00 and RTC2019-006871-7 projects, and by the BBVA Foundation (2019 Call) through the "Score-based nonstationary temporal Bayesian networks. Applications in climate and neuroscience" (BAYES-CLIMA-NEURO) project. Vicente P. Soloviev has been supported by the FPI PRE2020-094828 PhD grant from the Spanish Ministry of Science and Innovation.

Author Contributions VPS involved in conceptualization, investigation, software, and writing—original draft. PL involved in writing—review and editing. CB involved in writing—review and editing.

Data availability statement Different synthetic datasets have been generated using the available benchmarks in https://www.bnlearn.com/bnrepository/discrete-small.html#cancer, using own scripts available in GitHub (https://github.com/VicentePerezSoloviev/QAOA_BNSL_IBM).

Declarations

Conflict of interest The authors declare no competing financial or non-financial interests.

Code availability All the code used for implementing the approach and generating the results is available in https://github.com/VicentePerezSoloviev/QAOA_BNSL_IBM.

Appendix A: Kraus operators for noise simulation

Following the formalism of Kraus operators [14], each quantum channel ε is defined given a set of matrices E_i, which are applied to a quantum state, where i runs through all the operators considered for a given channel and k is the number f operators that define the noise channel. Thus, given a state ψ, the resulting state after applying a quantum channel is

$$\varepsilon(\psi) = \sum_k E_i \psi E_i^\dagger, \tag{14}$$

where the Kraus operators must meet the restriction $\sum_k E_i E_i^\dagger = 1$. Note that all the operators defined in this section are one-qubit operations ($E_i^{(1)}$), and that all the channels are parameterized by $\omega \in [0, 1]$, which regulates the probability of occurrence of the respective noise.

A.1 Amplitude damping channel

This quantum operation describes the energy dissipation of the quantum states. The operation over a one-qubit system is defined as

$$\varepsilon_{AD}(\psi) = E_1 \psi E_1^\dagger + E_2 \psi E_2^\dagger,$$

where E_1 and E_2 are defined as

$$E_1^{(1)} = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\omega} \end{pmatrix}, \quad E_2^{(1)} = \begin{pmatrix} 0 & \sqrt{\omega} \\ 0 & 0 \end{pmatrix}.$$

The E_2 operation converts a $|1\rangle$ to a $|0\rangle$, representing the physical process of the environment energy lost. E_1 leaves $|0\rangle$ unaltered but decreases the amplitude of a $|1\rangle$ state. Physically, this occurs because some energy was not lost to the environment, and thus the quantum state is more likely to be in the $|0\rangle$ state, rather than in the $|1\rangle$ state.

A.2 Phase damping channel

This quantum operation describes the loss of quantum information without loss of energy. The operation over a one-qubit system is defined as

$$\varepsilon_{PD}(\psi) = E_1 \psi E_1^\dagger + E_2 \psi E_2^\dagger,$$

where E_1 and E_2 are defined as

$$E_1^{(1)} = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\omega} \end{pmatrix}, \quad E_2^{(1)} = \begin{pmatrix} 0 & 0 \\ 0 & \sqrt{\omega} \end{pmatrix}.$$

In this case, the E_1 operator acts in the same way as in the case of the amplitude damping noise channel, leaving $|0\rangle$ unchanged, but reducing the amplitude of $|1\rangle$. However, in this case, E_2 also reduces the amplitude of the $|1\rangle$ state, but does not change it to $|0\rangle$.

A.3 Depolarizing channel

This quantum operation describes the depolarization of a qubit. That is, with certain probability ω a quantum state is replaced by the mixed state $I/2$. The operation over a one-qubit system is defined as

$$\varepsilon_D(\psi) = \omega\frac{I}{2} + (1-\omega)\psi. \tag{15}$$

Despite the fact that Eq. (15) does not involve any Kraus operators, it is possible to define the depolarizing channel with the following Kraus operators: [14],

$$\begin{aligned} E_1^{(1)} &= \sqrt{1-\frac{3\omega}{4}}I_2, \quad E_2^{(1)} = \sqrt{\frac{\omega}{4}}\sigma^x, \\ E_3^{(1)} &= \sqrt{\frac{\omega}{4}}\sigma^y, \qquad E_4^{(1)} = \sqrt{\frac{\omega}{4}}\sigma^z, \end{aligned} \tag{16}$$

where σ^x, σ^y and σ^z are the Pauli operators, and $\omega = 1$ implies the output state $\varepsilon_D(\psi)$ to be the mixed state $I/2$.

The application of the Kraus operators defined in Eq. (16) for two-qubits quantum system is defined as:

$$\begin{aligned} E_1^{(2)} &= E_1^{(1)} \bigotimes E_1^{(1)}, \quad E_2^{(2)} = E_2^{(1)} \bigotimes E_2^{(1)}, \\ E_3^{(2)} &= E_3^{(1)} \bigotimes E_3^{(1)}, \quad E_4^{(2)} = E_4^{(1)} \bigotimes E_4^{(1)}. \end{aligned}$$

Appendix B: Complementary results for noise simulation

Table 2 shows the detailed mean and standard deviation of the experimental results represented in Fig. 8.

Table 2 Mean best costs (μ) and standard deviations (σ) found for different values of the ω parameter over 50 executions of the QAOA approach for the BNSL problem, and the mean number of iterations (μ) and standard deviations (σ) until convergence for 50 executions. AD, PD and DE represent amplitude damping, phase damping, and depolarizing simulated errors, respectively

	cost						convergence					
	AD		PD		DE		AD		PD		DE	
$log_{10}\omega$	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
−5.00	−12.25	10.80	−7.20	15.61	−12.45	8.96	69.80	4.70	66.50	6.73	68.40	6.64
−4.75	−9.00	10.53	−10.95	10.80	−12.00	7.94	66.70	6.34	66.95	5.13	67.65	6.04
−4.50	−9.50	10.69	−4.90	11.49	−9.30	9.65	69.90	7.79	66.45	8.11	68.15	7.80
−4.25	−10.65	8.97	−11.45	10.95	−7.65	11.34	69.90	7.83	66.15	5.95	66.45	5.36
−4.00	−15.10	7.75	−5.95	9.58	−11.25	11.50	68.05	5.48	67.30	5.53	69.65	4.36
−3.75	−10.75	10.97	−10.25	11.54	−8.00	12.53	71.00	9.00	66.90	5.18	69.00	5.52
−3.50	−12.35	9.17	−14.00	8.47	−13.45	7.47	71.05	8.01	66.65	5.65	65.70	6.06
−3.25	−10.20	13.78	−15.20	8.79	−10.85	10.98	69.90	8.47	71.95	7.16	65.55	6.07
−3.00	−14.55	7.55	−12.70	8.86	−10.65	8.64	68.25	7.35	70.05	9.41	69.30	5.92
−2.75	−16.00	8.07	−8.15	13.69	−8.70	11.72	66.30	5.20	69.40	7.61	69.00	6.28
−2.50	−12.05	8.84	−8.75	13.18	−12.10	10.34	66.95	5.52	64.60	6.94	69.45	7.94
−2.25	−10.05	9.40	−6.55	12.15	−10.90	8.23	66.65	4.69	68.45	6.63	66.45	6.37
−2.00	−9.45	11.91	−11.70	9.59	−9.60	12.33	69.85	6.18	68.45	6.10	66.55	5.07
−1.75	−7.80	12.76	−5.85	12.78	−3.40	9.33	64.50	5.84	65.35	5.29	68.80	7.70
−1.50	−6.40	7.16	−8.80	11.62	6.25	15.34	68.20	5.52	68.20	6.96	62.50	4.88
−1.25	0.40	9.30	−7.15	10.16	9.85	16.68	67.50	6.06	67.75	6.22	63.65	5.58
−1.00	−1.95	5.67	−8.50	10.04	18.25	13.26	65.00	6.05	65.30	6.39	60.70	4.66
−0.75	−0.50	2.24	−2.50	7.25	13.20	12.89	66.95	6.48	65.55	5.86	58.10	4.59
−0.50	0.00	0.00	7.80	12.88	21.05	11.81	66.75	5.76	62.60	3.76	58.50	4.57
−0.25	0.00	0.00	22.15	14.53	20.01	10.30	64.05	6.73	58.75	4.02	59.10	4.59
0.00	0.00	0.00	11.65	13.35	21.50	11.20	37.00	0.00	60.60	5.86	58.60	4.80

References

1. Koller, D., Friedman, N.: Probabilistic graphical models: principles and techniques. The MIT Press, Cambridge (2009)
2. Murphy, K.P.: Machine learning: a probabilistic perspective. The MIT press, Cambridge (2012)
3. Bielza, C., Larrañaga, P.: Bayesian networks in neuroscience: a survey. Front. Comput. Neurosci. **8**, 131 (2014). https://doi.org/10.3389/fncom.2014.00131
4. Puerto-Santana, C., Larrañaga, P., Bielza, C.: Autoregressive asymmetric linear Gaussian hidden Markov models. IEEE Trans. Pattern Anal. Mach. Intell. (2021). https://doi.org/10.1109/TPAMI.2021.3068799
5. Chickering, D.M.: Learning bayesian networks is np-complete. Learning from Data, Springer, New York (1996)
6. Robinson, R.W.: Counting unlabeled acyclic digraphs. combinatorial mathematics. Springer, New York (1977)
7. Aouay, S., Jamoussi, S., Ayed, Y.B.: Particle swarm optimization based method for Bayesian network structure learning. In: 2013 5th International Conference on Modeling, Simulation and Applied Optimization, pp. 1–6 (2013). doi:https://doi.org/10.1109/ICMSAO.2013.6552569. IEEE
8. Quesada, D., Bielza, C., Larrañaga, P.: Structure learning of high-order dynamic Bayesian networks via particle swarm optimization with order invariant encoding. In: International Conference on Hybrid Artificial Intelligence Systems, pp. 158–171 (2021). doi:https://doi.org/10.1007/978-3-030-86271-8_14. Springer
9. Blanco, R., Inza, I., Larrañaga, P.: Learning Bayesian networks in the space of structures by estimation of distribution algorithms. Int. J. Intell. Syst. **18**(2), 205–220 (2003). https://doi.org/10.1002/int.10084
10. Larrañaga, P., Poza, M., Yurramendi, Y., Murga, R.H., Kuijpers, C.M.H.: Structure learning of Bayesian networks by genetic algorithms: a performance analysis of control parameters. IEEE Trans. Pattern Anal. Mach. Intell. **18**(9), 912–926 (1996). https://doi.org/10.1109/34.537345
11. Lee, S., Kim, S.B.: Parallel simulated annealing with a greedy algorithm for Bayesian network structure learning. IEEE Trans. Knowl. Data Eng. **32**(6), 1157–1166 (2019). https://doi.org/10.1109/TKDE.2019.2899096
12. Ji, J.-Z., Zhang, H.-X., Hu, R.-B., Liu, C.-N.: A tabu-search based Bayesian network structure learning algorithm. J. Beijing Univ. Technol. **37**, 1274–1280 (2011)
13. Scanagatta, M., Salmerón, A., Stella, F.: A survey on Bayesian network structure learning from data. Progr. Artif. Intell. **8**(4), 425–439 (2019). https://doi.org/10.1007/s13748-019-00194-y
14. Nielsen, M.A., Chuang, I.: Quantum computation and quantum information. American Association of Physics Teachers, Washington DC (2002)
15. Hauke, P., Katzgraber, H.G., Lechner, W., Nishimori, H., Oliver, W.D.: Perspectives of quantum annealing: methods and implementations. Rep. Progr. Phys. **83**(5), 054401 (2020). https://doi.org/10.1088/1361-6633/ab85b8
16. O'Gorman, B., Babbush, R., Perdomo-Ortiz, A., Aspuru-Guzik, A., Smelyanskiy, V.: Bayesian network structure learning using quantum annealing. Eur. Phys. J. Special Topics **224**(1), 163–188 (2015). https://doi.org/10.1140/epjst/e2015-02349-9
17. Shikuri, Y.: Efficient conversion of Bayesian network learning into quadratic unconstrained binary optimization. http://arxiv.org/abs/2006.06926 (2020). https://doi.org/10.48550/arXiv.2006.06926
18. Schuld, M., Petruccione, F.: Supervised learning with quantum computers. Springer, New York (2018)
19. McClean, J.R., Romero, J., Babbush, R., Aspuru-Guzik, A.: The theory of variational hybrid quantum-classical algorithms. J. Phys. **18**(2), 023023 (2016). https://doi.org/10.1088/1367-2630/18/2/023023
20. Peruzzo, A., McClean, J., Shadbolt, P., Yung, M.-H., Zhou, X.-Q., Love, P.J., Aspuru-Guzik, A., O'Brien, J.L.: A variational eigenvalue solver on a photonic quantum processor. Nat. Commun. **5**(1), 1–7 (2014). https://doi.org/10.1038/ncomms5213
21. Farhi, E., Goldstone, J., Gutmann, S.: A quantum approximate optimization algorithm. http://arxiv.org/abs/1411.4028 (2014). https://doi.org/10.48550/arXiv.1411.4028
22. Utkarsh, Behera, B.K., Panigrahi, P.K.: Solving vehicle routing problem using quantum approximate optimization algorithm. http://arxiv.org/abs/2002.01351 (2020). doi:https://doi.org/10.48550/arXiv.2002.01351
23. Choi, J., Kim, J.: A tutorial on quantum approximate optimization algorithm (QAOA): Fundamentals and Applications. In: 2019 International Conference on Information and Communication Technology Convergence, pp. 138–142 (2019). doi:https://doi.org/10.1109/ICTC46691.2019.8939749. IEEE

24. Shaydulin, R., Alexeev, Y.: Evaluating quantum approximate optimization algorithm: A case study. In: 2019 Tenth International Green and Sustainable Computing Conference, pp. 1–6 (2019). doi:https://doi.org/10.1109/IGSC48788.2019.8957201. IEEE
25. Fontana, E., Fitzpatrick, N., Ramo, D.M., Duncan, R., Rungger, I.: Evaluating the noise resilience of variational quantum algorithms. Phys. Rev. A **104**(2), 022403 (2021). https://doi.org/10.1103/PhysRevA.104.022403
26. Verdon, G., Broughton, M., Biamonte, J.: A quantum algorithm to train neural networks using low-depth circuits. http://arxiv.org/abs/1712.05304 (2017). https://doi.org/10.48550/arXiv.1712.05304
27. Streif, M., Leib, M.: Comparison of QAOA with quantum and simulated annealing. http://arxiv.org/abs/1901.01903 (2019). https://doi.org/10.48550/arXiv.1901.01903
28. Xue, C., Chen, Z.-Y., Wu, Y.-C., Guo, G.-P.: Effects of quantum noise on quantum approximate optimization algorithm. Chin. Phys. Lett. **38**(3), 030302 (2021). https://doi.org/10.1088/0256-307X/38/3/030302
29. Sharma, K., Khatri, S., Cerezo, M., Coles, P.J.: Noise resilience of variational quantum compiling. New J. Phys. **22**(4), 043006 (2020). https://doi.org/10.1088/1367-2630/ab784c
30. Schwarz, G.: Estimating the dimension of a model. Annal. Stat. **25**, 461–464 (1978). https://doi.org/10.1214/aos/1176344136
31. Cooper, G.F., Herskovits, E.: A Bayesian method for the induction of probabilistic networks from data. Mach. Learn. **9**(4), 309–347 (1992). https://doi.org/10.1007/BF00994110
32. Heckerman, D., Geiger, D., Chickering, D.M.: Learning Bayesian networks: the combination of knowledge and statistical data. Mach. Learn. **20**(3), 197–243 (1995). https://doi.org/10.1023/A:1022623210503
33. Farhi, E., Goldstone, J., Gutmann, S.: Quantum adiabatic evolution algorithms with different paths. quant-ph/0208135 (2002). https://doi.org/10.48550/arXiv.quant-ph/0208135
34. Aleksandrowicz, G., Alexander, T., Barkoutsos, P., Bello, L., Ben-Haim, Y., Bucher, D., Cabrera-Hernández, F.J., Carballo-Franquis, J., Chen, A., Chen, C.-F., et al.: Qiskit: An open-source framework for quantum computing (2021). https://doi.org/10.5281/zenodo.2573505
35. ATOS: Quantum learning machine. https://atos.net/en/solutions/quantum-learning-machine. [Online; Accessed 26-January-2022] (2021)
36. Acerbi, C., Tasche, D.: On the coherence of expected shortfall. J. Bank. Financ. **26**(7), 1487–1503 (2002). https://doi.org/10.1016/S0378-4266(02)00283-2
37. Barkoutsos, P.K., Nannicini, G., Robert, A., Tavernelli, I., Woerner, S.: Improving variational quantum optimization using CVaR. Quantum **4**, 256 (2020). https://doi.org/10.22331/q-2020-04-20-256
38. De Jong, K.: Evolutionary computation: A unified approach. In: Proceedings of the 2016 Genetic and Evolutionary Computation Conference Companion, pp. 185–199. The MIT Press, Cambridge (2016). doi:https://doi.org/10.1007/s10710-007-9035-9
39. Larrañaga, P., Lozano, J.A.: Estimation of distribution algorithms: a new tool for evolutionary computation, vol. 2. Springer, New York (2001)
40. Powell, M.J.: A direct search optimization method that models the objective and constraint functions by linear interpolation. in: advances in optimization and numerical analysis. Springer, New York (1994)
41. Bonet-Monroig, X., Wang, H., Vermetten, D., Senjean, B., Moussa, C., Bäck, T., Dunjko, V., O'Brien, T.E.: Performance comparison of optimization methods on variational quantum algorithms. http://arxiv.org/abs/2111.13454 (2021). https://doi.org/10.48550/arXiv.2111.13454
42. Urbanek, M., Nachman, B., Pascuzzi, V.R., He, A., Bauer, C.W., de Jong, W.A.: Mitigating depolarizing noise on quantum computers with noise-estimation circuits. http://arxiv.org/abs/2103.08591 (2021). https://doi.org/10.1103/PhysRevLett.127.270502
43. Kandala, A., Temme, K., Córcoles, A.D., Mezzacapo, A., Chow, J.M., Gambetta, J.M.: Error mitigation extends the computational reach of a noisy quantum processor. Nature **567**(7749), 491–495 (2019). https://doi.org/10.1038/s41586-019-1040-7
44. Sun, J., Yuan, X., Tsunoda, T., Vedral, V., Benjamin, S.C., Endo, S.: Mitigating realistic noise in practical noisy intermediate-scale quantum devices. Phys. Rev. Appl. **15**(3), 034026 (2021). https://doi.org/10.1103/PhysRevApplied.15.034026
45. Vovrosh, J., Khosla, K.E., Greenaway, S., Self, C., Kim, M., Knolle, J.: Simple mitigation of global depolarizing errors in quantum simulations. Phys. Rev. E **104**(3), 035309 (2021). https://doi.org/10.1103/PhysRevE.104.035309
46. Henrion, M.: Propagating uncertainty in Bayesian networks by probabilistic logic sampling. In: machine intelligence and pattern recognition. Elsevier, Amsterdam (1988)

47. Gámez, J.A., Mateo, J., Puerta, J.M.: Learning Bayesian networks by hill climbing: Efficient methods based on progressive restriction of the neighborhood. Data Min. Knowl. Discov. **22**, 106–148 (2011). https://doi.org/10.1007/s10618-010-0178-6
48. Tsamardinos, I., Brown, L.E., Aliferis, C.F.: The max-min hill-climbing Bayesian network structure learning algorithm. Mach. Learn. **65**(1), 31–78 (2006). https://doi.org/10.1007/s10994-006-6889-7
49. Egger, D.J., Mareček, J., Woerner, S.: Warm-starting quantum optimization. Quantum **5**, 479 (2021). https://doi.org/10.22331/q-2021-06-17-479