Annals of Operations Research
https://doi.org/10.1007/s10479-024-06253-5

ORIGINAL RESEARCH

An introduction to variational quantum algorithms for combinatorial optimization problems

Camille Grange[1,2] · Michael Poss[1] · Eric Bourreau[1]

Received: 17 June 2024 / Accepted: 23 August 2024

Abstract
Noisy intermediate-scale quantum computers are now readily available, motivating many researchers to experiment with Variational Quantum Algorithms. Among them, the Quantum Approximate Optimization Algorithm is one of the most popular one studied by the combinatorial optimization community. In this tutorial, we provide a mathematical description of the class of Variational Quantum Algorithms, assuming no previous knowledge of quantum physics from the readers. We introduce precisely the key aspects of these hybrid algorithms on the quantum side (parametrized quantum circuit) and the classical side (guiding function, optimizer). We devote a particular attention to QAOA, detailing the quantum circuits involved in that algorithm, as well as the properties satisfied by its possible guiding functions. Finally, we discuss the recent literature on QAOA, highlighting several research trends.

Keywords Variational quantum algorithm · QAOA · Combinatorial optimization · Metaheuristics

1 Introduction

This tutorial, which is an updated version of Grange et al. (2023), aims at providing the operational research community with some background knowledge of quantum computing and explore the particular branch of heuristic quantum algorithms for combinatorial optimization. Today, the quantum information theory community is excited by the emergence of quantum computers first theorized in the 1980 s. Indeed, since then, some theoretical advantages of quantum algorithms compared with classical algorithms have been proved for several problems. For instance, the Quantum Phase Estimation (Kitaev, 1995), which is the subroutine of Shor's algorithm (Shor, 1994), enables the latter to solve the integer factorization problem efficiently. Moreover, Grover Search (Grover, 1996) finds an element in an unstructured base with a quadratic speedup compared with classical algorithms. This algorithm is also used as a subroutine in dynamic programming algorithms to improve their exponential complexity for

This is an updated version of the paper that appeared in 4OR, 21 (3), 363-403 (2023).

✉ Camille Grange
camille.grange@lirmm.fr

[1] LIRMM, University of Montpellier, CNRS, Montpellier, France

[2] SNCF, Technology, Innovation and Group Projects Department, Saint-Denis, France

Published online: 13 September 2024

problems such as the Travelling Salesman Problem and the Minimum Set Cover (Ambainis et al., 2019). More recently, a polynomial speedup that relies on a Quantum Interior Point Method (Kerenidis & Prakash, 2020) applies to Linear Programming, Semi-Definite Programming, and Second-Order Cone Programming (Kerenidis et al., 2021). Furthermore, a quantum subroutine for the Simplex also provides a polynomial speedup (Nannicini, 2021). One can also mention the subroutine that quantum walks represent in backtracking algorithms (Montanaro, 2015), themselves used in quantum Branch-and-Bound achieving an almost quadratic speedup (Montanaro, 2020). Other problems or methods in optimization going beyond combinatorial optimization are studied with a quantum approach as presented in Abbas et al. (2023).

These algorithms prove quantum advantages on numerous problems, but their implementations require a lot of quantum resources with high quality. Specifically, they need quantum computers with many qubits that can interact two by two and quantum operations that can be applied in a row on qubits without generating noise, namely they can support deep-depth quantum circuits. Such quantum computers are believed to be built in the medium term but current devices are noisy intermediate-scale quantum computers, usually referred to as NISQ computers (Preskill, 2018). Therefore, NISQ computers cannot handle the implementations of such algorithms. These physical limitations encouraged quantum algorithm theory to look into *lighter* quantum algorithms to implement them on quantum computers today. It consists of hybrid algorithms that require both quantum and classical resources. Indeed, the classical part overcomes the limited and noisy quantum resources, whereas the quantum part still takes partial advantage of quantum information theory. This tutorial focuses on a particular type of hybrid algorithm, that is the class of Variational Quantum Algorithms (VQAs).

Variational Quantum Algorithms are heuristic algorithms that alternate between a quantum circuit and a classical optimizer. They tackle optimization problems of the form

$$\min_{x\in\{0,1\}^n} f(x), \tag{1}$$

where f is any function defined on $\{0, 1\}^n$. VQAs are of great interest to the quantum information theory community today because they have the convenient property of an adjustable quantum circuits' depth, making them implementable on the current NISQ computers. The variational approach of VQAs (Cerezo et al., 2021) consists of probing the initial search space $\{0, 1\}^n$ with a relatively small set of parameters optimized classically. Specifically, these parameters describe a probability distribution over the search space. This description results from a sequence of quantum logic gates: this is a quantum circuit. VQAs take advantage of the quantum computing principle (Nielsen & Chuang, 2022) that prepares a probability distribution over an exponential search space in a short sequence of quantum gates. The choice of the quantum circuit, and in particular the number of parameters, is a huge stake for VQAs (Nannicini, 2019). On the one hand, the more parameters, the more precise the search space probing. On the other hand, too many parameters can harden the classical optimization part. The function that drives this optimization, which depends on the probability distribution and f, is also a non-trivial choice and worth investigating (Barkoutsos et al., 2020).

In this tutorial, we provide a mathematical description of Variational Quantum Algorithms and focus on one of them, specifically the Quantum Approximate Optimization Algorithm (QAOA) (Farhi et al., 2014). We shall also devote particular attention to problems in which f is a polynomial function. In Sect. 2, we give a brief introduction to the basics of quantum computing for combinatorial optimization. Notice that the model of quantum computation considered is the gate-based model, also called the circuit model. Then, in Sect. 3, we describe the general class of VQAs, starting by defining the different parts that constitute and define a

VQA, namely the quantum circuit (also called *ansatz* in the literature), the classical optimizer, and the guiding function driving the classical optimization. Then, we characterize each part with properties that should be valid for potential theoretical guarantees. Eventually, in Sect. 4, we focus on a particular case of VQAs, the Quantum Approximate Optimization Algorithm. We describe the necessary reformulation of the initial problem (1) into a Hermitian matrix (also called Hamiltonian in the literature) to be solved by QAOA and analyze this algorithm in light of the previous properties of Sect. 3. We also provide a universal decomposition of the QAOA quantum circuit for the general case where f is polynomial while, as far as we know, only the quadratic f case had been treated in the literature so far. Finally, we give a condensed overview of empirical trends and theoretical limitations of QAOA. We remind some helpful basic notions of linear algebra in Appendix A. Furthermore, the proofs of technical properties of the function optimized in VQAs and specific constructions of the quantum circuit in QAOA are deferred to Appendices B.1 and B.2, respectively.

2 Basics of quantum computing for combinatorial optimization

This section aims at providing the basic notions of quantum computing necessary for the understanding of the quantum resolution of combinatorial problems.

2.1 Quantum bits

Let $|0\rangle$ and $|1\rangle$ denote the basic states of our quantum computer (the counterpart of states 0 and 1 in classical computers). The first building block of quantum algorithms is the quantum bit, also called qubit.

Definition 1 (Qubit) We define a qubit as

$$|q\rangle = q_0|0\rangle + q_1|1\rangle, \tag{2}$$

where $(q_0, q_1) \in \mathbb{C}^2$ is a pair of complex numbers that satisfies the normalizing condition

$$|q_0|^2 + |q_1|^2 = 1.$$

We say that q_0 and q_1 are the *coordinates* of $|q\rangle$ in the basis $(|0\rangle, |1\rangle)$.

It is often convenient to use the matrix representation of $|0\rangle$ and $|1\rangle$, namely

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ and } |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

With this matrix representation, the qubit $|q\rangle$ defined in (2) is equal to

$$|q\rangle = \begin{pmatrix} q_0 \\ q_1 \end{pmatrix}.$$

Example 2 Important examples of one-qubit states are $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and $\frac{|0\rangle-|1\rangle}{\sqrt{2}}$. These are usually denoted as $|+\rangle$ and $|-\rangle$, respectively.

The algorithms studied in this paper typically manipulate quantum states of larger dimension. Specifically, let n denote the number of qubits that our quantum computing device is able to manipulate simultaneously. An n-qubit state is defined by 2^n complex numbers that satisfy the normalizing condition and represent the normal decomposition in the canonical basis.

Definition 3 (Canonical basis) The canonical basis of an n-qubit state is the set

$$\mathcal{CB}_n = \left(\bigotimes_{k=1}^{n} |i^{(k)}\rangle, (i^{(1)}, \ldots, i^{(n)}) \in \{0,1\}^n \right),$$

where $i^{(k)}$ represents the state of the k-th qubit and $\otimes$ is the tensor product defined in Appendix A. The canonical basis is the set of all possible combinations of tensor products of n one-qubit basis states, $|0\rangle$ and $|1\rangle$. Thus, in the matrix representation, each canonical basis state is a column vector of 2^n components with one component equal to 1 and the rest equal to 0. It directly results that the size of the canonical basis of an n-qubit state is 2^n. For more readability, we omit to write the tensor product between qubits, i.e., we refer to the canonical basis as

$$\mathcal{CB}_n = (|i\rangle, i \in \{0,1\}^n).$$

The matrix representation of canonical basis states results from the definition above. We illustrate it for the canonical basis of two qubits:

$$|00\rangle = \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}1\\0\\0\\0\end{pmatrix}, \quad |01\rangle = \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}0\\1\\0\\0\end{pmatrix},$$

$$|10\rangle = \begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}0\\0\\1\\0\end{pmatrix}, \quad |11\rangle = \begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}0\\0\\0\\1\end{pmatrix}.$$

Definition 4 (n-qubit state) An n-qubit state $|\psi\rangle$ is a normalized linear combination of the basis states in $\mathcal{CB}_n$,

$$|\psi\rangle = \sum_{i \in \{0,1\}^n} \psi_i |i\rangle,$$

where $(\psi_i)_{i\in\{0,1\}^n} \in \mathbb{C}^{2^n}$ are its coordinates, which satisfy the normalizing condition

$$\langle\psi\rangle = \sum_{i \in \{0,1\}^n} |\psi_i|^2 = 1. \tag{3}$$

For $n = 1$, we find the definition of a qubit (Definition 1), as expected.

Example 5 For instance, $\frac{|00\rangle+|11\rangle}{\sqrt{2}}$, called $|\Phi^+\rangle$, is a two-qubit state. Its matrix representation is:

$$\begin{aligned}
|\Phi^+\rangle &= \frac{|00\rangle + |11\rangle}{\sqrt{2}} \\
&= \frac{1}{\sqrt{2}} \left(\begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} + \begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix} \right) \\
&= \frac{1}{\sqrt{2}} \left(\begin{pmatrix}1\\0\\0\\0\end{pmatrix} + \begin{pmatrix}0\\0\\0\\1\end{pmatrix} \right) \\
&= \frac{1}{\sqrt{2}} \begin{pmatrix}1\\0\\0\\1\end{pmatrix}.
\end{aligned}$$

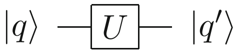


Fig. 1 Circuit of gate U applying to $|q\rangle$

Remark 6 Our notation $(\psi_i)_{i\in\{0,1\}^n} \in \mathbb{C}^{2^n}$ for the coordinates of $|\psi\rangle$ in basis $\mathcal{CB}_n$ is used to simplify the presentation throughout. It is, however, uncommon in the quantum computing literature where different bases may be used.

2.2 Quantum gates

In the model of gate-based quantum computation, qubits are manipulated with quantum gates. Mathematically speaking, these quantum gates are modeled by unitary matrices. More precisely, a quantum gate that manipulates n-qubit states is a matrix in $\mathcal{M}_{2^n}(\mathbb{C})$ that modifies the 2^n complex coefficients of a quantum state such that they still satisfy the normalizing condition (3).

Definition 7 (Unitary matrix) A matrix $U \in \mathcal{M}_{2^n}(\mathbb{C})$ is unitary if its inverse is equal to its conjugate transpose (denoted by the dagger symbol †):

$$UU^\dagger = U^\dagger U = I.$$

Notice that the application of a quantum gate is logically reversible.

We easily see that a unitary matrix U preserves the normalizing condition since, denoting $|\psi'\rangle = U|\psi\rangle$, we have

$$\langle\psi'\rangle = \langle\psi|U^\dagger U|\psi\rangle = \langle\psi\rangle.$$

As an illustration, we describe below all the possible one-qubit gates, i.e., all the unitary matrices in $\mathcal{M}_2(\mathbb{C})$.

Example 8 (Generic one-qubit gate) A generic one-qubit gate $U = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathcal{M}_2(\mathbb{C})$ satisfies

$$\begin{cases} |a|^2 + |b|^2 = 1 \\ a\bar{c} + b\bar{d} = 0 \\ \bar{a}c + \bar{b}d = 0 \\ |c|^2 + |d|^2 = 1 \end{cases}$$

Its application on a qubit $|q\rangle = q_0|0\rangle + q_1|1\rangle$ is:

$$U|q\rangle = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} q_0 \\ q_1 \end{pmatrix} = \begin{pmatrix} aq_0 + bq_1 \\ cq_0 + dq_1 \end{pmatrix} = (aq_0 + bq_1)|0\rangle + (cq_0 + dq_1)|1\rangle = |q'\rangle \quad (4)$$

We introduce next the circuit representation of a quantum gate, which is a useful formalism to represent unitary matrices. Specifically, Fig. 1 represents the application of the quantum gate U to the one-qubit state $|q\rangle$, as in (4), by:

This representation easily generalizes to n-qubit quantum gates, where each *horizontal line* is associated with a qubit.

Fig. 2 Composition of U_1 and U_2

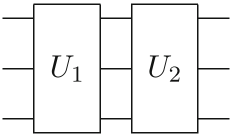


Fig. 3 Tensor product of U_1 and U_2

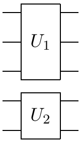
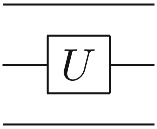


Fig. 4 Application of U to qubit number 2



2.3 Quantum circuits

We can manipulate quantum gates to build other quantum gates in two different ways. The first one is the *composition*, and the other one is the *tensor product*.

Definition 9 (Composition of quantum gates) The composition of gates only operates between gates acting on the same qubits. Let k be the number of involved qubits. The composition of $U_1 \in \mathcal{M}_{2^k}(\mathbb{C})$ and $U_2 \in \mathcal{M}_{2^k}(\mathbb{C})$ consists of the application of U_1 followed by the application of U_2. The matrix representation of this composition is the product U_2U_1. The circuit representation of this composition is illustrated on Fig. 2 for $k = 3$.

It can be seen as a series sequence of gates.

Notice that we read from right to left in the matrix representation, and from left to right in the circuit representation. Moreover, in the circuit representation, qubits are numbered in ascending order from top to bottom which is important for the tensor product circuit representation that follows.

Definition 10 (Tensor product of quantum gates) The tensor product of gates only operates between gates acting on different qubits. Suppose $U_1 \in \mathcal{M}_{2^k}(\mathbb{C})$ applies on the first k qubits and $U_2 \in \mathcal{M}_{2^{k'}}(\mathbb{C})$ applies on the k' following ones. Their tensor product is the application of U_1 and U_2 on respective qubits in parallel. The matrix representation of this tensor product is $U_1 \otimes U_2$ (Definition 40). The circuit representation is depicted on Fig. 3 for $k = 3$ and $k' = 2$.

Notice that when we apply a quantum gate on k qubits of an n-qubit system, it supposes that we apply identity gate $I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ on the qubits not concerned. For instance, let us consider a 3-qubit system on which we apply $U \in \mathcal{M}_2(\mathbb{C})$ on qubit number 2. The matrix representation of the resulting 3-qubit gate is $I \otimes U \otimes I$, and its circuit representation is illustrated on Fig. 4, where the application of I is usually replaced by a simple wire.

One readily verifies that both composition and tensor product transform unitary matrices into a resulting unitary matrix.

Throughout, we consider that a quantum algorithm is a quantum circuit acting on n qubits, that is, a sequence of quantum gates' compositions and/or tensor products. These quantum gates can be k-qubit gates, for $k \in [n]$. However, quantum gates involving many qubits are typically not implementable natively on quantum computers and need to be decomposed into smaller and simpler gates. This set of small gates can be considered as the quantum counterpart of the elementary logic gates used in classical circuit computing to assess the circuit complexity of a classical algorithm. Thus, an n-qubit quantum algorithm is described by a unitary matrix in $\mathcal{M}_{2^n}(\mathbb{C})$, and we decompose it as a sequence of universal gates (Definition 11) to obtain the complexity of the quantum algorithm.

Definition 11 (Set of universal gates) A set of quantum gates PU is *universal* if we can decompose any n-qubit quantum gate through a circuit composed solely of the gates in PU.

Fortunately, there exist different universal sets of quantum gates. We introduce below such a set formed of four types of gates. First, we consider the three following families of one-qubit gates, each of which is parametrized by a real number $\theta \in \mathbb{R}$:

$$R_X(\theta) = \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix} \tag{5}$$

$$R_Y(\theta) = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix} \tag{6}$$

$$R_Z(\theta) = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix}. \tag{7}$$

Notice that these gates are often referred to as rotation gates because they correspond to rotations in a certain representation of qubits, known as the Bloch sphere (Mosseri & Dandoloff, 2001). Second, we consider the two-qubit gate CX:

$$CX = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{8}$$

CX applies gate $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ on the second qubit if and only if the first qubit is in state $|1\rangle$. Indeed, for instance, we have

$$CX|01\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = |01\rangle\,,$$

and

$$CX|11\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = |10\rangle\,.$$

In other words, we can define CX on the canonical basis as

$$CX|a,b\rangle = |a, b \oplus a\rangle,$$

for $a, b \in \{0, 1\}$, where $\oplus$ is the addition modulo 2.

Remark 12 (Notation on gates indexes) Henceforth, we use the notation $R_{X,i}$ for the application of R_X on qubit i (and the application of identity matrix on the remaining qubits). We do the same with $R_{Y,i}$ and $R_{Z,i}$. We note $CX_{i,j}$ the application of CX gate to qubits i and j: X is applied to qubit j if and only if qubit i is in state $|1\rangle$.

Theorem 13 *(Universal gates (Nielsen & Chuang, 2022)) The set of one-qubit gates and the CX gate (8) is universal. Thus, because any one-qubit gate is the composition of rotation gates (5)–(7), the set $PU = \{R_X(\alpha), R_Y(\beta), R_Z(\gamma), CX : \alpha, \beta, \gamma \in \mathbb{R}\}$ is universal.*

In comparison, the sets of gates {NAND}, {NOR}, {NOT, AND} and {NOT, OR} are universal for classical computation. Indeed, we can compute any arbitrary classical function with them. In view of the above, we typically consider that the quantum counterpart of the classical number of elementary operations is the number of universal gates used to decompose the circuit. Of course, this decomposition depends on the set of universal gates PU considered, but the number of gates required is the same with any set, modulo a multiplicative constant (Nielsen & Chuang, 2022). Thereafter, we only consider the set PU defined in Theorem 13. This choice is motivated by the algorithms we study in this paper, as it will be convenient to express them with this set. Furthermore, if we consider the family of circuits $(\mathcal{C}_n)_{n\in\mathbb{N}}$ where $\mathcal{C}_n$ is a circuit on n qubits and is decomposed on $\mathcal{O}(poly(n))$ universal quantum gates, then this family is said to be *efficient*.

2.4 Non-classical behaviors

The quantum algorithms we present in this paper rely on three characteristics of quantum states with no classical equivalent: measurement, superposition, and entanglement. Let us present these notions through the bare minimum mathematical background.

2.4.1 Measurement

We need to measure a quantum state $|\psi\rangle$ to get information from it. Otherwise, no information is accessible. The peculiar property of measurement is that it only extracts partial information from the quantum state: the single measurement output of $|\psi\rangle$ is a bitstring.

Definition 14 (Measurement) In the gate-based quantum model, the measurement $\mathcal{M}$ of an n-qubit state $|\psi\rangle = \sum_{i\in\{0,1\}^n} \psi_i |i\rangle$ outputs the n-bitstring i with probability $|\psi_i|^2$. After having been measured, state $|\psi\rangle$ no longer exists: it has been replaced by the state $|i\rangle$.

For example, measuring qubit $|q\rangle = q_0|0\rangle + q_1|1\rangle$ outputs 0 with probability $|q_0|^2$ and 1 with probability $|q_1|^2$, and changes the state $|q\rangle$ to $|0\rangle$ and $|1\rangle$, respectively. A measurement appears as a loss of information. Indeed, we describe an n-qubit state by 2^n normalized complex coefficients, but we only extract an n-bitstring after measuring it. The perfect knowledge of the probabilities representing a given quantum state $|\psi\rangle$, namely the square module of each of its coordinates $(|\psi_i|^2)_{i\in\{0,1\}^n} \in [0, 1]^{2^n}$, can be obtained only if we measure $|\psi\rangle$ an infinite number of times. Notice that it requires resetting state $|\psi\rangle$ after each measurement.

Remark 15 (Sampling of quantum states) In reality we are limited to approximating a given quantum state through sampling. In particular, if $|\psi\rangle$ is the result of an algorithm, this means we have to repeat the same algorithm for every measurement of $|\psi\rangle$ we wish to perform.

2.4.2 Superposition

Classically, the state of an n-bit computer is given by a bitstring in $\{0, 1\}^n$. We have seen so far that the state of an n-qubit quantum computer is given by its coordinates $(\psi_i)_{i\in\{0,1\}^n} \in \mathbb{C}^{2^n}$, which satisfy $\sum_{i\in\{0,1\}^n} |\psi_i|^2 = 1$. In general, more than one of the coordinates are different from 0, meaning that measuring $|\psi\rangle$ may result in different bitstring $i \in \{0, 1\}^n$.

Definition 16 (Superposition) A quantum state $|\psi\rangle$ is said in superposition if $|\psi\rangle = \sum_{i\in\{0,1\}^n} \psi_i |i\rangle$ where there are at least two terms with non-zero coefficients in the sum. A quantum state that is not a basis state is in superposition.

The following Hadamard gate is the usual one-qubit gate that produces superposition starting from a canonical basis state.

Example 17 (Hadamard gate) The Hadamard gate $H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$ is essential in quantum computing because it creates superposition starting from a basis state. We obtain the state $|+\rangle$, respectively $|-\rangle$, of Example 2 by applying H on $|0\rangle$, respectively $|1\rangle$:

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} = |+\rangle\,,$$
$$H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} = |-\rangle\,.$$

Example 18 (Uniform superposition) The two states $|+\rangle$ and $|-\rangle$ are in uniform superposition because both have the same probability of being measured as 0 or 1. In general, an n-qubit state uniformly superposed is equal to $\frac{1}{\sqrt{2^n}} \sum_{j\in\{0,1\}^n} e^{i\alpha_j}|j\rangle$, with $\alpha_j \in [0, 2\pi[, \forall j \in \{0, 1\}^n$. In what follows, we shall often use the uniformly superposed n-qubit state $|+\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{i\in\{0,1\}^n} |i\rangle$.

Notice that applying an n-qubit quantum gate U to $|\psi\rangle$ possibly modifies the 2^n coordinates of $|\psi\rangle$ since

$$U|\psi\rangle = \sum_{i\in\{0,1\}^n} \psi_i'|i\rangle,$$

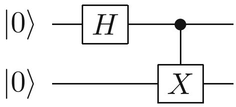
where possibly each ψ_i' differs from ψ_i. This is the case, for instance, when applying the tensor product of n Hadamard gates, each applied to a qubit initially in state $|0\rangle$, specifically

$$H^{\otimes n}|0\rangle^{\otimes n} = \bigotimes_{i=1}^{n} H|0\rangle = \bigotimes_{i=1}^{n} \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) = \frac{1}{\sqrt{2^n}} \sum_{i\in\{0,1\}^n} |i\rangle = |+\rangle^{\otimes n}. \tag{9}$$

Equation (9) illustrates the potential benefit of quantum circuits: applying $\mathcal{O}(n)$ universal one-qubit gates impacts the exponentially many coefficients of $|\psi\rangle$. Indeed, one readily verifies that $H = R_X(\pi)R_Y(\frac{\pi}{2})$ modulo a global phase[1], so $H^{\otimes n}$ amounts to applying $2n$ universal one-qubit gates.

[1] Two quantum states $|\psi\rangle$ and $|\psi'\rangle = e^{i\alpha}|\psi\rangle$, with $\alpha \in [0, 2\pi[$, that only differ by a global phase are indiscernible by measurement. Thus, we do not consider global phase of quantum states nor quantum gates.

Fig. 5 Entangling circuit



2.4.3 Entanglement

Each quantum state is either a product state or an entangled state. Entanglement has the peculiar and helpful property that we can apply a circuit only on a part of the n-qubit system, and as a result, the whole system is affected.

Definition 19 (Product state) An n-qubit state is a product state if it is the tensor product of n one-qubit states. In other words, an n-qubit state $|\psi\rangle$ is a product state if it exists $2n$ complex coefficients $(q_0^{(j)}, q_1^{(j)})_{j\in[n]}$ such that

$$|\psi\rangle = \bigotimes_{j=1}^{n}(q_0^{(j)}|0\rangle + q_1^{(j)}|1\rangle), \text{ with } |q_0^{(j)}|^2 + |q_1^{(j)}|^2 = 1, \forall j \in [n].$$

Thus, each state of a qubit that composes $|\psi\rangle$ can be described independently of the states of the other.

If an n-qubit state is not a product state, it is an entangled state.

Definition 20 (Entangled state) An n-qubit state $|\psi\rangle = \sum_{i\in\{0,1\}^n} \psi_i |i\rangle$ is entangled if the numerical values of its coordinates $(\psi_i)_{i\in\{0,1\}^n} \in \mathbb{C}^{2^n}$ admit no solution $(q_0^{(j)}, q_1^{(j)})_{j\in[n]} \in \mathbb{C}^{2n}$ to the system

$$\begin{cases} \sum_{i\in\{0,1\}^n} \psi_i |i\rangle = \bigotimes_{j=1}^{n}(q_0^{(j)}|0\rangle + q_1^{(j)}|1\rangle) \\ |q_0^{(j)}|^2 + |q_1^{(j)}|^2 = 1, \forall j \in [\![1, n]\!] \end{cases}$$

It means that operations performed on some coordinates of the entangled state can affect the other coordinates without direct operations on them.

Notice that when an n-qubit system is entangled, it makes no sense to speak about qubit number $k \in [n]$ because isolated qubits are not defined. Notice also that there is a difference between superposition and entanglement. A state is in superposition if at least two non-zero coefficients are in its basis decomposition. A state is entangled if it cannot be written as a tensor product of independent qubits, implying that it is in superposition.

To illustrate the notion of entanglement, we consider the following quantum circuit on Fig. 5. Starting from a two-qubit product state $|00\rangle$, it results an entangled state.

This circuit is the application of Hadamard gate on qubit 1, followed by $CX_{1,2}$. The quantum state that results is $|\Phi^+\rangle$ of Example 5:

$$\begin{aligned} CX_{1,2}(H \otimes I)|00\rangle &= CX_{1,2}\left(\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)\right) \\ &= \frac{1}{\sqrt{2}}\left(CX_{1,2}|00\rangle + CX_{1,2}|10\rangle\right) \\ &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\ &= |\Phi^+\rangle. \end{aligned}$$

We prove now by contradiction that $|\Phi^+\rangle$ is entangled. Suppose that $|\Phi^+\rangle$ is a product state. Thus, there exists $(q_0^{(0)}, q_1^{(0)}) \in \mathbb{C}^2$ such as $|q_0^{(0)}|^2 + |q_1^{(0)}|^2 = 1$ and $(q_0^{(1)}, q_1^{(1)}) \in \mathbb{C}^2$ such as $|q_0^{(1)}|^2 + |q_1^{(1)}|^2 = 1$, satisfying:

$$\begin{aligned} |\Phi^+\rangle &= (q_0^{(0)}|0\rangle + q_1^{(0)}|1\rangle) \otimes (q_0^{(1)}|0\rangle + q_1^{(1)}|1\rangle) \\ &= q_0^{(0)}q_0^{(1)}|00\rangle + q_0^{(0)}q_1^{(1)}|01\rangle + q_1^{(0)}q_0^{(1)}|10\rangle + q_1^{(0)}q_1^{(1)}|11\rangle\,. \end{aligned}$$

Because the decomposition of a quantum state's coordinates is unique in the canonical basis, it follows by identification:

$$\begin{cases} q_0^{(0)}q_0^{(1)} &= \frac{1}{\sqrt{2}} \\ q_0^{(0)}q_1^{(1)} &= 0 \\ q_1^{(0)}q_0^{(1)} &= 0 \\ q_1^{(0)}q_1^{(1)} &= \frac{1}{\sqrt{2}} \end{cases}$$

This equation system admits no solution. We deduce that $|\Phi^+\rangle = \frac{|00\rangle+|11\rangle}{\sqrt{2}}$ is not a product state, hence it is entangled.

3 Variational Quantum Algorithms for optimization

Some of the quantum algorithms that tackle combinatorial optimization problems are Variational Quantum Algorithms (VQAs) (Cerezo et al., 2021). They are hybrid algorithms because they require both quantum and classical computations. VQAs are studied today because they represent an alternative approach that reduces the quality and quantity of the quantum resources needed (Peruzzo et al., 2014). Specifically, they are designed to run on the NISQ era (Preskill, 2018) where quantum computers are noisy with few qubits: for instance, they harness low-depth quantum circuits.

In this section, we consider optimization problems of the form

$$\min_{x \in \{0,1\}^n} f(x), \tag{1}$$

where f is any function defined on $\{0, 1\}^n$. We note $\mathcal{F}$ the set of optimal solutions.

3.1 General description

Variational Quantum Algorithms (VQAs) are hybrid algorithms that, given an input $|0\rangle^{\otimes n}$, alternate between a quantum and a classical part. Henceforth, we note $|0_n\rangle$ the state $|0\rangle^{\otimes n}$ to ease the reading. Let us first provide a high-level description of the key elements of VQAs that are detailed in this section. Let $d \in \mathbb{N}$, the three key elements of VQAs are:

- A *parametrized quantum circuit*, $U : \mathbb{R}^d \to \mathcal{M}_{2^n}(\mathbb{C})$,
- A *guiding function*, $g : \mathbb{R}^d \to \mathbb{R}$,
- And a *classical optimizer*, which is an algorithm $\mathcal{A}$ that optimizes g over space $\mathbb{R}^d$.

These elements, the parametrized quantum circuit and the two others, are detailed and illustrated in Subsections 3.2 and 3.3, respectively. Before that, we provide a general overview of VQAs in Algorithm 1.

The main idea of a VQA is as follows. The main loop of the algorithm executes the following steps until the classical optimizer stops according to a given stopping criterion. First, given $\theta \in \mathbb{R}^d$, the *quantum part* executes the parametrized quantum circuit $U(\theta)$ for several times to sample the quantum state (see Remark 15). It results in a distribution probability over $\{0, 1\}^n$ that we note $\{p_\theta(x) : x \in \{0, 1\}^n\}$. Second, the *classical part* computes the cost of this state through the evaluation of the guiding function according to the sampling results, specifically, $g(\theta) = G(p_\theta, f)$ for a given function G that is constructed from f and p_θ. More precisely, for a given θ, the computation of G requires only the values $f(x)$ for x such that $p_\theta(x) > 0$. Eventually, this cost value is given to the classical optimizer $\mathcal{A}$, which outputs a new parameter in order to minimize g. Notice that, between the two parts the best-found solution is possibly updated by a classical computer.

In the following subsections, we detail each part of VQAs and show how specific choices of g and U ensure that VQAs optimize f. Let us begin with the quantum part.

3.2 Quantum part

The quantum part of VQAs applies a quantum circuit on the n-qubit system that constitutes the quantum computer. Importantly, *variational* in VQAs stands for the parametrization of the quantum circuit. Let $d \in \mathbb{N}$ be the number of parameters.

Definition 21 A parametrized quantum circuit is a continuous function $U : \mathbb{R}^d \to \mathcal{M}_{2^n}(\mathbb{C})$ mapping any $\theta \in \mathbb{R}^d$ to unitary matrix $U(\theta)$.

As defined in Subsection 2.3, a quantum circuit is a sequence of universal quantum gates' compositions and/or tensor products. Thus, all coefficients of matrix $U(\theta)$ are continuous functions on $\mathbb{R}^d$.

Example 22 A simple example of a parametrized quantum circuit for $n = 3$ and $d = 3$ is depicted in Fig. 6.

The expression of $U(\theta)$ for this circuit is as follows: $\forall \theta = (\theta_1, \theta_2, \theta_3) \in \mathbb{R}^3$,

```
**Input**: guiding function $g$, parametrized quantum circuit $U$, classical optimizer $\mathcal{A}$, initial parameter $\theta_0$
**Output**: approximate minimum $f^*$ and the corresponding minimizer $x^*$
$\theta \leftarrow \theta_0$;
$f^* \leftarrow +\infty$;
$x^* \leftarrow 0^n$;
while stopping criterion of the classical optimizer is false do
    begin quantum part
        for size of the sampling do
            measure the state $U(\theta)|0\rangle^{\otimes n}$, outputting $x$ for some $x \in \{0, 1\}^n$;
        end
        let $p_\theta(x)$ be the frequency of $x \in \{0, 1\}^n$ in the above sampling;
    end
    begin solution update
        for $x$ such that $p_\theta(x) > 0$ do
            compute $f(x)$;
            if $f(x) < f^*$ then
                $f^* \leftarrow f(x)$;
                $x^* \leftarrow x$;
            end
        end
    end
    begin classical part
        compute $g(\theta) = G(p_\theta, f)$;
        given $g(\theta)$ (and possibly its derivatives), $\mathcal{A}$ outputs $\theta'$;
        $\theta \leftarrow \theta'$;
    end
end
```

Algorithm 1: Variational Quantum Algorithm

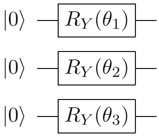


Fig. 6 Quantum circuit parametrized by $(\theta_1, \theta_2, \theta_3) \in \mathbb{R}^3$

$$
\begin{aligned}
U(\theta) &= R_Y(\theta_1) \otimes R_Y(\theta_2) \otimes R_Y(\theta_3) \\
&= \begin{pmatrix} \cos\frac{\theta_1}{2} & -\sin\frac{\theta_1}{2} \\ \sin\frac{\theta_1}{2} & \cos\frac{\theta_1}{2} \end{pmatrix} \otimes \begin{pmatrix} \cos\frac{\theta_2}{2} & -\sin\frac{\theta_2}{2} \\ \sin\frac{\theta_2}{2} & \cos\frac{\theta_2}{2} \end{pmatrix} \otimes \begin{pmatrix} \cos\frac{\theta_3}{2} & -\sin\frac{\theta_3}{2} \\ \sin\frac{\theta_3}{2} & \cos\frac{\theta_3}{2} \end{pmatrix} \\
&= \begin{pmatrix}
c_1c_2c_3 & -c_1c_2s_3 & -c_1s_2c_3 & c_1s_2s_3 & -s_1c_2c_3 & s_1c_2s_3 & s_1s_2c_3 & -s_1s_2s_3 \\
c_1c_2s_3 & c_1c_2c_3 & -c_1s_2s_3 & -c_1s_2c_3 & -s_1c_2s_3 & -s_1c_2c_3 & s_1s_2s_3 & s_1s_2c_3 \\
c_1s_2c_3 & -c_1s_2s_3 & c_1c_2c_3 & -c_1c_2s_3 & -s_1s_2c_3 & s_1s_2s_3 & -s_1c_2c_3 & s_1c_2s_3 \\
c_1s_2s_3 & c_1s_2c_3 & c_1c_2s_3 & c_1c_2c_3 & -s_1s_2s_3 & -s_1s_2c_3 & -s_1c_2s_3 & -s_1c_2c_3 \\
s_1c_2c_3 & -s_1c_2s_3 & -s_1s_2c_3 & s_1s_2s_3 & c_1c_2c_3 & -c_1c_2s_3 & -c_1s_2c_3 & c_1s_2s_3 \\
s_1c_2s_3 & s_1c_2c_3 & -s_1s_2s_3 & -s_1s_2c_3 & c_1c_2s_3 & c_1c_2c_3 & -c_1s_2s_3 & -c_1s_2c_3 \\
s_1s_2c_3 & -s_1s_2s_3 & s_1c_2c_3 & -s_1c_2s_3 & c_1s_2c_3 & -c_1s_2s_3 & c_1c_2c_3 & -c_1c_2s_3 \\
s_1s_2s_3 & s_1s_2c_3 & s_1c_2s_3 & s_1c_2c_3 & c_1s_2s_3 & c_1s_2c_3 & c_1c_2s_3 & c_1c_2c_3
\end{pmatrix},
\end{aligned}
$$

where $c_i = \cos\frac{\theta_i}{2}$ and $s_i = \sin\frac{\theta_i}{2}$ for $i \in [3]$.

Remark 23 The use of the generalized circuit to n qubits $\bigotimes_{i=1}^{n} R_{Y,i}(\theta_i)$ amounts to a continuous relaxation of the $\{0,1\}$-problem (1), where each decision variable $x_i \in [0,1]$ is represented by a rotation angle $\theta_i \in [0, 2\pi]$ as follows:

$$x_i = \left(\cos\frac{\theta_i}{2}\right)^2 .$$

3.3 Classical part

The classical part of VQAs consists of a classical optimization over the parameters $\theta \in \mathbb{R}^d$. The classical optimizer essentially aims at finding the optimal parameters θ^* that lead to optimal solutions of the initial problem (1) with high probability, specifically, such that

$$\sum_{s \in \mathcal{F}} |\langle s|U(\theta^*)|0_n\rangle|^2 \geq 1 - \epsilon, \tag{10}$$

for small $\epsilon > 0$. Henceforth, we use notation $|x\rangle$ instead of $|i\rangle$ to efficiently recall that we deal with solutions of optimization problems.

The classical part is characterized by two aspects: the function that guides the optimization and the optimizer itself.

3.3.1 Guiding function

Let $g : \mathbb{R}^d \to \mathbb{R}$ be the guiding function, formally defined in Definition 24 below, that the classical optimizer minimizes. The function g acts as a link between the quantum and classical parts. For a given $\theta \in \mathbb{R}^d$, we evaluate $U(\theta)|0_n\rangle$ *according to* f as we will exemplify below. Notice that f and g are distinct since g is defined on $\mathbb{R}^d$ and outputs a quality measure of an n-qubit quantum state whereas f is defined on $\{0,1\}^n$. Let us denote $\mathcal{F}_{\text{quant}} = \left\{\sum_{s\in\mathcal{F}} \psi_s|s\rangle : \sum_{s\in\mathcal{F}} |\psi_s|^2 = 1\right\}$ as the set of quantum states that are superpositions of optimal solutions of problem (1). Naturally, we would like to define g such that minimizing g tends to minimize f.

Definition 24 (Guiding function) Let $g : \mathbb{R}^d \to \mathbb{R}$ be a function and $\mathcal{G}$ be its set of minimizers. We call g a guiding function for f with respect to U if g is continuous and

$$\{U(\theta)|0_n\rangle : \theta \in \mathcal{G}\} \subseteq \mathcal{F}_{\text{quant}}. \tag{11}$$

In other words, optima of a guiding function g must lead to optima of f or superpositions of optima of f. Indeed, Equation (11) implies that measuring the quantum state $U(\theta^*)|0\rangle^{\otimes n}$ for $\theta^* \in \mathcal{G}$ outputs with probability 1 an optimal solution of the initial problem $s \in \mathcal{F}$. Thus, minimizing g amounts to minimizing f, and finding $\mathcal{F}$ is done by finding $\mathcal{G}$. The latter is found with the classical optimizer. Without any information on $\mathcal{F}$, we need to choose a quantum circuit U such that any optimal solution $s \in \mathcal{F}$ is reachable, specifically,

$$\{U(\theta)|0_n\rangle : \theta \in \mathbb{R}^d\} \supseteq \mathcal{CB}_n. \tag{12}$$

Notice that this condition is weak and easily satisfied. For instance, the circuit depicted in Fig. 6 satisfies this condition. If ever one is interested in finding all optimal solutions of problem (1), the circuit and the guiding function should satisfy instead the stronger condition

$$\{U(\theta)|0_n\rangle : \theta \in \mathcal{G}\} = \mathcal{F}_{\text{quant}}. \tag{13}$$

In that case, U satisfying (12) is not enough. Without any information on $\mathcal{F}$, we need to choose U that can reach any n-qubit quantum states, specifically,

$$\{U(\theta)|0_n\rangle : \theta \in \mathbb{R}^d\} = \left\{ \sum_{x\in\{0,1\}^n} \psi_x |x\rangle : \sum_{x\in\{0,1\}^n} |\psi_x|^2 = 1 \right\}.$$

A popular choice for the guiding function in the literature is the mean function

$$g_{\text{mean}}(\theta) = \sum_{x\in\{0,1\}^n} p_\theta(x) f(x), \tag{14}$$

where $p_\theta(x) = |\langle x|U(\theta)|0_n\rangle|^2$ is the probability of finding x when $U(\theta)|0_n\rangle$ is measured. We show next that g_{mean} is indeed a guiding function according to Definition 24, see Appendix B.1.1 for a proof.

Proposition 25 *Function g_{mean} is a guiding function.*

Example 26 We illustrate the mean function on the 3-qubit quantum circuit $U(\theta)$ depicted in Fig. 6. The generalization of its computation for n qubits is trivial since it needs to replace 3 by n. The single application of rotation gate R_Y (6) of angle θ_i on a qubit initially on state $|0\rangle$ is

$$\begin{aligned} R_Y(\theta_i)|0\rangle &= \cos\frac{\theta_i}{2}|0\rangle + \sin\frac{\theta_i}{2}|1\rangle \\ &= \sum_{j\in\{0,1\}} \cos\frac{\theta_i - j\pi}{2}|j\rangle\,, \end{aligned}$$

since $\sin(\phi) = \cos(\phi - \frac{\pi}{2})$ for $\phi \in \mathbb{R}$. Eventually, the quantum state resulting from $U(\theta)$ is

$$\begin{aligned} U(\theta)|0_3\rangle &= \bigotimes_{i=1}^{3} R_{Y,i}(\theta_i)|0\rangle \\ &= \sum_{j_1,j_2,j_3\in\{0,1\}} \left(\prod_{i=1}^{3} \cos\frac{\theta_i - j_i\pi}{2}\right)|j_1 j_2 j_3\rangle\,. \end{aligned}$$

Thus, the probability to measure $x = (x_1, x_2, x_3) \in \{0, 1\}^3$ is

$$p_\theta(x) = \left(\prod_{i=1}^{3} \cos\frac{\theta_i - x_i\pi}{2}\right)^2, \tag{15}$$

and the expression of g_{mean} of equation (14) directly results from it.

Other functions are compatible with Definition 24. One of these functions encountered in the literature is the Gibbs function (Li et al., 2020) which stems from statistical mechanics. Let $\eta > 0$ be a parameter to be set. The Gibbs function is defined as

$$g_{G,\eta}(\theta) = -\ln\left(\sum_{x\in\{0,1\}^n} p_\theta(x)e^{-\eta f(x)}\right). \tag{16}$$

The choice of this function is motivated by the exponential shape that highly rewards the increase of probabilities of low-cost states. Notice that for small η, minimizing the Gibbs function is essentially equivalent to minimizing the mean function in the sense that the Taylor series of $g_{G,\eta}$ at first order in $\eta = 0$ gives $g_{G,\eta} = \eta g_{\text{mean}}$. We show next that $g_{G,\eta}$ is indeed a guiding function according to Definition 24, see Appendix B.1.2 for a proof.

Proposition 27 *Let $\eta > 0$. Function $g_{G,\eta}$ is a guiding function.*

One might suggest other guiding functions, such as the minimum function

$$g_{\min}(\theta) = \min_{x\,:\,p_\theta(x)>0} f(x). \tag{17}$$

However, $g_{\min}$ does not verify (11), and is not even continuous, hardening its optimization and excluding its choice for the guiding function.

Example 28 Let us illustrate that $g_{\min}$ is not a guiding function. For that, we consider the circuit of Fig. 6 and the following function $f : \{0,1\}^3 \mapsto \mathbb{R}$ to minimize,

$$\begin{cases} f(0,0,0) & = 1 \\ f(x) & = 0, \quad \forall x \neq (0,0,0) \end{cases}$$

where $f^* = 0$ is the optimal value. Function $g_{\min}$ reaches its optimal value $g^*_{\min} = 0$ on the set of its optimizers

$$\mathcal{G} = \mathbb{R}^3 \setminus \{(2k\pi, 2k\pi, 2k\pi) : k \in \mathbb{Z}\}.$$

However,

$$\forall \theta \in \mathcal{G} \setminus \{((2k+1)\pi, (2k+1)\pi, (2k+1)\pi) : k \in \mathbb{Z}\}, U(\theta)|0_n\rangle \notin \mathcal{F}_{\text{quant}},$$

because there is a non-zero probability of sampling $(0,0,0)$. Thus, $g_{\min}$ violates (11). Moreover, $g_{\min}$ is not continuous. Indeed, $g_{\min}(0,0,0) = 1$, whereas $\forall \epsilon > 0$, $g_{\min}(\epsilon, 0, 0) = 0$.

The above observation motivated (Barkoutsos et al., 2020) to suggest another function, the CVaR (Conditional Value-at-Risk) function. The CVaR function is the average on the lower α-tail of values of f encountered, where $\alpha \in]0,1]$ is a parameter to be set. Let $(x_1, \ldots, x_{2^n})$ be the n-bitstrings sorted in non decreasing order, namely $f(x_i) \leq f(x_{i+1})$ for any $i \in [2^n - 1]$. Let N_α be the index that delimits the α-tail elements of the distribution, specifically,

$$N_\alpha = \min\left\{N \geq 1 : \sum_{i=1}^{N} p_\theta(x_i) \geq \alpha\right\}.$$

Then, the CVaR function is

$$g_{C,\alpha}(\theta) = \frac{1}{\sum_{i=1}^{N_\alpha} p_\theta(x_i)} \sum_{i=1}^{N_\alpha} p_\theta(x_i) f(x_i). \tag{18}$$

The special case $\alpha = 1$ implies $g_\alpha = g_{\text{mean}}$, whereas when α approaches zero, we find $g_{\min}$.

The CVaR function is an alternative to the non-smooth minimum function. While CVaR does not verify (11) either, it keeps continuity and still focuses on the best solutions that appear on the probability distribution.

Example 29 Let us illustrate the violation of (11) by the CVaR function, for any $\alpha \in]0,1[$. For that, we consider function f of Example 28 with the same quantum circuit of Fig. 6. Let $\alpha \in]0,1[$. Let us find $\theta \in \mathcal{G}$ such that $U(\theta)|0_n\rangle \notin \mathcal{F}_{\text{quant}}$. In other words, we search $\theta \in \mathbb{R}^3$ such that

$$g_{C,\alpha}(\theta) = 0 \text{ and } p_\theta(0,0,0) > 0.$$

Let us look at $\theta = (\pi - \epsilon, 0, 0)$, where $\epsilon \in]0, \pi[$. According to (15), we have

$$p_\theta(0,0,0) = \cos^2\left(\frac{\pi - \epsilon}{2}\right) > 0.$$

It remains to choose ϵ to ensure $g_{C,\alpha}(\theta) = 0$. Namely, we want ϵ such that

$$\sum_{x \neq (0,0,0)} p_\theta(x) \geq \alpha,$$

meaning

$$1 - \cos^2\left(\frac{\pi - \epsilon}{2}\right) \geq \alpha.$$

This holds for any $\epsilon \leq \pi - 2\arccos(\sqrt{1-\alpha})$.

Even if CVaR does not verify (11), it can be seen as a pseudo-guiding function, defined just below. Notice that in practice, using the CVaR function seems appropriate because it accepts a probability of measuring optimal solutions after optimizing to be lower than one, such as expressed in (10).

Definition 30 (Pseudo-guiding function) Let $g : \mathbb{R}^d \to \mathbb{R}$ be a function and $\mathcal{G}$ be its set of minimizers. We call g a pseudo-guiding function for f with respect to U if g is continuous and if there exists $\alpha \in]0,1[$ such that optima of g can lead to non-optimal solutions of f with a probability strictly lower than $1-\alpha$. Specifically, let $\theta \in \mathcal{G}$. Thus, either $U(\theta)|0_n\rangle \in \mathcal{F}_{\text{quant}}$ or $\sum_{x \notin \mathcal{F}} |\langle x|U(\theta)|0_n\rangle|^2 < 1 - \alpha$.

We show next that $g_{C,\alpha}$ is indeed a pseudo-guiding function according to Definition 30, see Appendix B.1.3 for a proof.

Proposition 31 *Let $\alpha \in]0,1[$. Function $g_{C,\alpha}$ is a pseudo-guiding function.*

3.3.2 Classical optimizer

The role of the classical optimizer is to minimize the guiding function. The function g is continuous and is usually differentiable but not convex. Any unconstrained optimization algorithm can be used to minimize g, such as local search, descent gradient method, or any black-box optimization algorithm.

To speak in terms of stochastic optimization, the classical optimizer aims at solving the stochastic programming model under endogenous uncertainty

$$\min_\theta \{g(\theta) = \mathbb{E}[G(\theta, \xi_\theta)]\}, \tag{19}$$

where the definition of G depends on the choice of a specific guiding function, and ξ_θ is an endogenous vector that depends on θ. Specifically, ξ_θ is a discrete random variable, with the set of possible outcomes $\{0, 1\}^n$ and the following distribution probability:

$$\mathbb{P}(\xi_\theta = x) = p_\theta(x), \quad \forall x \in \{0, 1\}^n.$$

Thus, this problem falls into the class of stochastic dependent-decision probabilities problems (Hellemo et al., 2018). In practice, the classical optimizer approximates the function by a Monte Carlo estimation as follows:

$$\hat{g}_N(\theta) = \frac{1}{N} \sum_{j=1}^{N} G(\theta, \xi_\theta^j),$$

where $\{\xi_\theta^j\}_{j \in [N]}$ is a sample of size N from the distribution of ξ_θ. Notice that for a given θ, the quantity $\hat{g}_N(\theta)$ itself is a random variable since its value depends on the sample that has been generated, which is random. In contrast, the value of $g(\theta)$ is deterministic. In practice, the classical optimizer iterates the loop that consists of, given a sampling distribution of size N of the quantum state $U(\theta)|0_n\rangle$ (Remark 15), outputs a θ'. This θ' is then transmitted to the quantum part. Eventually, the aim is to output θ^* such that $U(\theta)|0_n\rangle \subseteq \mathcal{F}_{\text{quant}}$. In each iteration, the value of $\hat{g}_N(\theta)$ is computed. Notice that according to the Law of Large Numbers (Shapiro, 2003), $\hat{g}_N(\theta)$ converges with probability one to $g(\theta)$ as $N \to \infty$.

For instance, for the case of $g = g_{\text{mean}}$, we have

$$G(\theta, \xi_\theta) = f(\xi_\theta).$$

Hence, for each $\xi_\theta^j \in \{0, 1\}^n$ sampled, either we compute classically $f(\xi_\theta^j)$ and store it if not already computed, or we get its value. Thus, we can compute $\hat{g}_N(\theta)$. Notice that for the CVaR function, we compute the empirical mean only on the best $\lceil \alpha N \rceil$ values $f(\xi_\theta)$ found by sampling.

The solution returned by the VQA is the minimum value of f and the associated minimizer x encountered while the algorithm runs.

4 Quantum approximate optimization algorithm

We assume throughout this section that the function f to be minimized is polynomial. First, we reformulate problem (1) to a more suitable form for quantum optimization. This reformulation is motivated by the quantum adiabatic evolution (Farhi et al., 2000) that, for a given Hermitian matrix,[2] approximates the eigenvector with the lowest eigenvalue under certain conditions. For that, we interpret the objective function of problem (1) as an Hermitian matrix H_f such that each eigenvector $|u_x\rangle$ is matching a classical solution $x \in \{0, 1\}^n$ with an eigenvalue equal to $f(x)$, specifically,

$$H_f |u_x\rangle = f(x)|u_x\rangle.$$

Thus, the solutions of problem (1) are the solutions corresponding to the lowest eigenvalues of H_f. The Quantum Approximate Optimization Algorithm (QAOA) (Farhi et al., 2014) presented in this section aims at finding the lowest eigenvalue of H_f.

[2] A complex square matrix is Hermitian if it is equal to its conjugate transpose.

4.1 Problem reformulation

The construction of H_f is as follows. First, we transform the $\{0, 1\}$-problem (1) into a $\{-1, 1\}$-problem. For that, we apply the following linear transformation: for $x = (x_1, \dots, x_n) \in \{0, 1\}^n$, we define $z = (z_1, \dots, z_n) \in \{-1, 1\}^n$ where

$$z_i = 1 - 2x_i, \forall i \in [n]. \tag{20}$$

This leads to the problem

$$\min_{z \in \{-1,1\}^n} f_{\pm}(z),$$

where, for $z \in \{-1, 1\}^n$,

$$f_{\pm}(z) = \sum_{\alpha=(\alpha_1,\dots\alpha_n)\in\{0,1\}^n} h_\alpha \prod_{i=1}^{n} z_i^{\alpha_i},$$

where $h_\alpha \in \mathbb{R}, \forall \alpha \in \{0, 1\}^n$. Second, we define H_f as

$$H_f = \sum_{\alpha=(\alpha_1,\dots\alpha_n)\in\{0,1\}^n} h_\alpha \bigotimes_{i=1}^{n} Z_i^{\alpha_i},$$

where $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$, $Z^0 = I$ and $Z^1 = Z$. We note Z_i the application of Z to qubit i. Notice that Z is equal to the universal gate $R_Z(\pi)$ modulo a global phase (see (7)). This construction of H_f leads to the following property.

Proposition 32 *The eigenvectors of H_f are the canonical basis $|x\rangle \in \mathcal{CB}_n$ with eigenvalues that are the cost of the solutions $f(x)$, specifically,*

$$\forall |x\rangle \in \mathcal{CB}_n, \quad H_f|x\rangle = f(x)|x\rangle. \tag{21}$$

Proof First, the eigenvectors of H_f are the canonical basis states. Indeed, each term of the sum that constitutes H_f is a tensor product of n matrices I or Z, both diagonal. Thus, H_f is a 2^n diagonal matrix. Second, let us find the eigenvalues associated with the eigenvectors. Let $|x\rangle = |x_1 \dots x_n\rangle$ be in $\mathcal{CB}_n$. Let $z = (z_1, \dots, z_n)$ be the result of transformation (20). Thus, we can easily show that $Z|x_i\rangle = z_i|x_i\rangle, \forall i \in [n]$ and

$$\begin{aligned}
H_f|x\rangle &= \sum_{\alpha=(\alpha_1,\dots,\alpha_n)\in\{0,1\}^n} h_\alpha \bigotimes_{i=1}^{n} Z_i^{\alpha_i}|x_i\rangle \\
&= \sum_{\alpha=(\alpha_1,\dots,\alpha_n)\in\{0,1\}^n} h_\alpha \bigotimes_{i=1}^{n} z_i^{\alpha_i}|x_i\rangle \\
&= \left(\sum_{\alpha=(\alpha_1,\dots,\alpha_n)\in\{0,1\}^n} h_\alpha \prod_{i=1}^{n} z_i^{\alpha_i} \right) |x\rangle \\
&= f_{\pm}(z)|x\rangle \\
&= f(x)|x\rangle\,.
\end{aligned}$$

□

Example 33 We illustrate this transformation on a small example with $n = 2$. Let us consider the problem

$$\min_{x\in\{0,1\}^2} f(x) = x_1 + 2x_2 - 3x_1x_2.$$

Using (20), the equivalent $\{-1, 1\}$-problem is

$$\min_{z\in\{-1,1\}^2} f_{\pm}(z) = \frac{1}{4}z_1 - \frac{1}{4}z_2 - \frac{3}{4}z_1z_2 + \frac{3}{4}.$$

Thus, the Hermitian matrix associated with the problem is

$$H_f = \frac{1}{4}Z \otimes I - \frac{1}{4}I \otimes Z - \frac{3}{4}Z \otimes Z + \frac{3}{4}I \otimes I.$$

To illustrate (21), we compute the eigenvalue of the canonical basis state $|10\rangle$.

$$\begin{aligned} H_f|10\rangle &= \frac{1}{4}(Z \otimes I)|10\rangle - \frac{1}{4}(I \otimes Z)|10\rangle - \frac{3}{4}(Z \otimes Z)|10\rangle + \frac{3}{4}(I \otimes I)|10\rangle \\ &= -\frac{1}{4}|10\rangle - \frac{1}{4}|10\rangle + \frac{3}{4}|10\rangle + \frac{3}{4}|10\rangle \\ &= |10\rangle \\ &= f(1,0)|10\rangle\,, \end{aligned}$$

because $f(1, 0) = 1$.

Notice that most of the problems solved with QAOA in the literature are QUBO (Quadratic Unconstrained Binary Optimization) problems. Thus, H_f has the specific form

$$H_f = \sum_i h_{ii}Z_i + \sum_{i<j} h_{ij}Z_i \otimes Z_j,$$

where $h_{ij} \in \mathbb{R}$, $\forall i \leq j$. It is justified by the fact that solving QUBO problems to optimality is *already* NP-hard, and the quantum gates of the circuit are easier to implement on hardware in that case.

4.2 Quantum part

QAOA is a Variational Quantum Algorithm where the quantum part derives from the Hamiltonian H_f. This quantum part consists of a quantum circuit with $2p$ parameters $(\gamma, \beta) = (\gamma_1, \ldots, \gamma_p, \beta_1, \ldots, \beta_p) \in \mathbb{R}^{2p}$, where p is called *depth*. The quantum circuit $U(\gamma, \beta)$ is the sequence of p layers of two blocks, initially applied to the uniform superposition $|+\rangle^{\otimes n}$ (see Example 18). The first block is of the form $\mathrm{Exp}(H_f, \gamma)$, for $\gamma \in \mathbb{R}$, which is the unitary operator (see Definition 43) associated with the Hamiltonian H_f. The second block is of the form $\mathrm{Exp}(H_B, \beta)$, for $\beta \in \mathbb{R}$, which is the unitary operator associated with the Hamiltonian $H_B = \sum_{i=1}^{n} R_{X,i}(\pi)$. Thus, the quantum circuit is

$$U(\gamma, \beta) = \mathrm{Exp}(H_B, \beta_p)\mathrm{Exp}(H_f, \gamma_p)\ldots\mathrm{Exp}(H_B, \beta_1)\mathrm{Exp}(H_f, \gamma_1)H^{\otimes n}, \tag{22}$$

where the first gates applied in this circuit are $H^{\otimes n}$ to then apply the p layers to state $|+\rangle^{\otimes n}$. The three propositions that follow express the quantum circuit of QAOA with the set of universal gates (see Theorem 13). We give first the general decomposition of the QAOA quantum circuit defined in (22), see Appendix B.2.1 for a proof.

Proposition 34 *The first block* $\mathrm{Exp}(H_f, \gamma)$ *parametrized by* $\gamma \in \mathbb{R}$ *is*

$$\mathrm{Exp}(H_f, \gamma) = \prod_{\alpha \in \{0,1\}^n} \mathrm{Exp}(\bigotimes_{i=1}^{n} Z_i^{\alpha_i}, h_\alpha \gamma).$$

The second block $\mathrm{Exp}(H_B, \beta)$ *parametrized by* $\beta \in \mathbb{R}$ *is*

$$\mathrm{Exp}(H_B, \beta) = \bigotimes_{i=1}^{n} R_{X,i}(2\beta).$$

The particular case of QUBO is mainly considered in the literature. Thus, we propose next a decomposition of the QAOA quantum circuit for this specific case, see Appendix B.2.2 for a proof.

Proposition 35 *For the case of QUBO, the expression of* $\mathrm{Exp}(H_f, \gamma)$ *simplifies in*

$$\mathrm{Exp}(H_f, \gamma) = \left(\bigotimes_{i=1}^{n} R_{Z,i}(2h_{ii}\gamma)\right) \prod_{i<j} CX_{i,j} R_{Z,j}(2h_{i,j}\gamma) CX_{i,j},$$

and is rather easily implemented with universal quantum gates.

The decomposition in universal quantum gates of the term $\mathrm{Exp}(Z_i \otimes Z_j, t)$ for the specific case of QUBO (see Proposition 35) is mainly used in the literature. We propose in Proposition 37 a generalization of such a decomposition for the term $\mathrm{Exp}(\bigotimes_{i=1}^n Z^{\alpha_i}, t)$, where the number of Z gates in effect can be bigger than two, namely, $|\{\alpha_i,\ i \in [n] : \alpha_i = 1\}| \geq 2$. This proposition enables us to overtake the QUBO problems, namely, to deal with a polynomial function f with a degree strictly larger than two. We introduce next a technical result that will be necessary to derive the subsequent proposition, see Appendix B.2.3 for a proof.

Lemma 36 $\forall n \in \mathbb{N}^*$,

$$(I^{\otimes n-1} \otimes X) e^{-itZ^{\otimes n}} (I^{\otimes n-1} \otimes X) = e^{itZ^{\otimes n}}.$$

The proposition that follows enables a decomposition in universal quantum gates of any quantum circuit of QAOA, see Appendix B.2.4 for a proof.

Proposition 37 *Let us consider the subsystem composed of the* N *qubits to which the Z gate is applied. Specifically,* $N = |\{\alpha_i\ i \in [n] : \alpha_i = 1\}|$, *and we renumber the qubits in question in* $[N]$. *Thus, for* $N \geq 2$, *the term* $\mathrm{Exp}(\bigotimes_{i=1}^n Z^{\alpha_i}, t)$ *on this subsystem simplifies in*

$$\mathrm{Exp}(Z^{\otimes N}, t) = \prod_{j=0}^{N-2} CX_{1,N-j} R_{Z,N}(2t) \prod_{j=0}^{N-2} CX_{1,N-j}.$$

We represent this decomposition on Fig. 7.

Example 38 We illustrate the construction of the quantum circuit with the problem of Example 33 where $n = 2$, for the case $p = 1$. Thus,

$$U(\gamma, \beta) = \mathrm{Exp}(H_B, \beta)\mathrm{Exp}(H_f, \gamma)|+\rangle^{\otimes 2}, \quad \forall \gamma, \beta \in \mathbb{R}.$$

The circuit is detailed in Fig. 8.

Notice that we do not take into account the term $\frac{3}{4} I \otimes I$ of H_f in this circuit. More generally, the term $h_{0\ldots0} I^{\otimes n}$ of H_f never appears on the quantum circuit because it represents a constant term and does not influence the optimization.

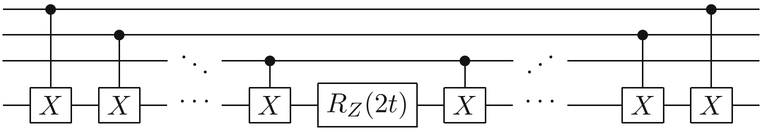


Fig. 7 Decomposition of $\mathrm{Exp}(Z^{\otimes N}, t)$ on the N-qubit subsystem

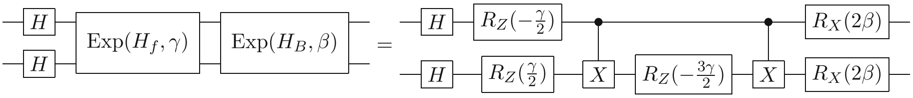


Fig. 8 QAOA circuit of Example 33 for $p = 1$

Notice that the choice of QAOA circuit does not ensure (12). For example, one can show that the probability of measuring 00 at the end of the circuit depicted in Fig. 8 never reaches 1. Specifically,

$$p_{(\gamma,\beta)}(00) = |\langle 00|\mathrm{Exp}(H_B, \beta)\mathrm{Exp}(H_f, \gamma)|+\rangle^{\otimes 2}|^2 < \frac{1}{2}, \ \ \forall \gamma, \beta \in \mathbb{R}.$$

However, QAOA satisfies another important property: it uses entangling gates. Each gate $\mathrm{Exp}(Z_i \otimes Z_j, t)$, for $t \in \mathbb{R} \setminus \{k\pi : k \in \mathbb{Z}\}$, entangles the qubits i and j. Indeed, $\mathrm{Exp}(Z_i \otimes Z_j, t) = CX_{i,j} R_{Z,j}(2t) CX_{i,j}$, and unless $R_{Z,j}(2t) = I$, namely $t \in \{k\pi : k \in \mathbb{Z}\}$, the CNOT gate operates[3] and creates entanglement as mentioned in Subsection 2.4.3. The other gates, which are one-qubit gates, do not have this power. Entanglement is not necessary to (12). However, Remark 23 can justify the use of entanglement gate. Indeed, without it, it seems unlikely to achieve better results than pure classical optimization because the optimizer essentially solves a classical continuous relaxation.

In fact, the popularity of QAOA originates essentially from the fact that it mimics the adiabatic schedule (Wurtz & Love, 2022). The n-qubit system verifies the adiabatic condition when $p \to \infty$ and ensures that for a particular set of parameters, the quantum circuit gives the exact solution. However, quantum computers' quality today makes implementations on large instances impossible. But because the number of gates of the quantum circuit is $\mathcal{O}(pn^2)$, for small depth d, QAOA, and more generally VQAs, can already be implemented on current NISQ computers.

5 Literature review for QAOA

Many papers have recently addressed the empirical evaluation of QAOA, some also comparing it with specific implementations of VQA. We present below a non-exhaustive list of these trends, we mention several theoretical limitations for specific cases known up to now and we end with the different leverages that are at stake to improve QAOA performances. This section does not provide a complete, up-to-date overview of QAOA performances, but rather aims at illustrating trends on combinatorial problems of interest to the operations research community.

[3] For $t \in \{k\pi : k \in \mathbb{Z}\}$, $R_{Z,j}(2t) = I$. Thus, $CX_{i,j} R_{Z,j}(2t) CX_{i,j} = CX_{i,j} CX_{i,j} = I$ because CNOT is its own inverse.

5.1 Empirical and theoretical trends on QAOA

Let us begin with the numerical trends of QAOA performances. All the empirical experiments are presented on small instances because quantum computers' quality today makes implementations on large instances impossible, leading to difficult conclusions. Thus, many experiments are done on classical simulators of quantum computers. Most of the empirical results of QAOA apply to the MAX-CUT problem because it was initially the first application of QAOA (Farhi et al., 2014).

Definition 39 (MAX-CUT problem) Let $G = (V, E)$ be a undirected graph. A cut in G is a subset $S \subseteq V$. We define its *cost* as the number of edges with one node in S and one node in $V \setminus S$. The MAX-CUT problem aims at finding a cut with maximum *cost*. A version with weighted edges can also be defined.

Notice that for the MAX-CUT problem, with the notations of Sect. 4, the objective function is

$$f(x) = - \sum_{(i,j)\in E} \left(x_i(1 - x_j) + x_j(1 - x_i) \right),$$

where x_i is 1 if node i is in the cut, 0 otherwise. The Hermitian matrix that corresponds to this problem is

$$H_f = -\frac{1}{2} \sum_{(i,j)\in E} (1 - Z_i Z_j).$$

The approximation ratio r mainly quantifies the performance of QAOA as follows:

$$r = \frac{f^{\text{QAOA}}}{f^*},$$

where f^{QAOA} is the value returned by QAOA, and f^* is the optimal value. Papers often compare this ratio with the best-known guaranteed ratio of Goemans-Williamson algorithm (Goemans & Williamson, 1995), specifically, $r = 0.87856$. The seminal paper of QAOA (Farhi et al., 2014) provides a lower bound of r for the specific class of 3-regular graphs for $p = 1$ that is $r = 0.6924$. More precise analytical expressions of the lower bound of the ratio for $p = 1$ and for some other typical cases are given in Wang et al. (2018). In addition, the authors of Wurtz and Love (2021) provide lower bounds of the ratio for larger depth for uniform 3-regular graphs (under specific assumptions implying the absence of large cycles in the graph): $r \geq 0.7559$ for $p = 2$ and $r \geq 0.7924$ for $p = 3$.

Several empirical results on MAX-CUT spotlight patterns of optimal parameters and enable QAOA to exceed Goemans-Williamson bound for some specific instances. Some classes of MAX-CUT instances studied reveal patterns of optimal parameters. Thus, it seems to offer efficient heuristics for parameter selection and initialization. For example, the authors of Crooks (2018) look at the class of Erdös-Rényi graphs (random graphs where an edge appears between two nodes with probability 0.5) of size up to 17 nodes, where the classical optimizer is an automatic differentiation with stochastic gradient descent. The authors of Lotshaw et al. (2021) examine the exhaustive set of graphs with $n \leq 9$ nodes, with the gradient-based search BFGS (Broyden-Fletcher-Goldfarb-Shanno algorithm (Fortran et al., 1992)) for the classical optimizer. Both exhibit instances that exceed the bound of Goemans-Williamson for small depth, $p \leq 8$ and $p \leq 3$, respectively. The sets of unweighted and weighted 3-regular graphs also lead to patterns in Zhou et al. (2020) for graphs of a maximum size of 22 nodes, also detected in the parameter space of the Job Shop Scheduling problem (Kurowski et al., 2023). But even if the performance of QAOA sometimes exceeds the Goemans-Williamson bound for the low-depth circuits, it is believed that p must grow with the instance size to have a chance to outperform the best classical algorithms. Indeed, for random large-girth d-regular graphs, QAOA with depth $p = 11$ presents better performances than any known classical algorithms, in the case where the optimal parameters are found and where d goes to infinity (Basso et al., 2021). Notice that the authors of the nominal paper of QAOA (Farhi et al., 2014) tackle also the Sherrington-Kirkpatrick model, well-known in the spin glass theory. The Hamiltonian of this model represents the energy of n spins with random coefficients, specifically,

$$H_f = \frac{1}{\sqrt{n}} \sum_{i<j} h_{ij} Z_i Z_j,$$

where the coefficients h_{ij} are chosen independently from a distribution centered in 0 and with a variance equal to 1. In Farhi et al. (2022), it is shown that, for $p = 11$, QAOA outperforms asymptotically with n the standard semidefinite programming algorithm and the spectral relaxation.

Some theoretical limits of QAOA are displayed, where the shape of the quantum state produced by the quantum circuit is at stake. The authors of Bravyi et al. (2020) point out that the symmetry and locality of this resulting variational state fund amentally limit the performances of QAOA. Indeed, they show that Goemans-Williamson outperforms QAOA for several instances of the MAX-CUT problem for any fixed depth p. Consequently, this paper suggests a non-local version of QAOA to overcome these limitations. The limits of the locality also appear when solving the problem of the Maximum Independent Set (MIS). In Farhi et al. (2020), MIS instances are random graphs of n vertices, with a fixed average degree $\bar{d}$. Thus, it proves that for depth $p \leq C \log(n)$, where C is a constant depending on $\bar{d}$, QAOA cannot return an independent set better than 0.854 times the optimal for $\bar{d}$ large. Due to this locality issue, the author of Hastings (2019) compares QAOA with *local* classical algorithms which also have this locality notion: at each step, the value of a variable is updated depending on the values of its *neighbors*, i.e., the variables that share the same term within the objective function. Namely, after t steps, the value of a variable depends on all information gathered in its t*-neighborhood*. Yet, these classical algorithms still outperform QAOA. A single step of these algorithms outperforms, resp. achieves the same performance as, a single step of QAOA ($p = 1$) for MAX-CUT instances, resp. MAX-3-LIN-2 instances. Notice that given binary variables and a set of linear equations modulo 2 with exactly three variables, the MAX-3-LIN-2 problem aims at finding a variable value assignment that maximizes the number of satisfied equations. Several other canonical combinatorial optimization problems have been tackled with QAOA in recent years. While these examples are limited to numerical tests on toy instances and do not include comparisons of performance with classical resolutions, we mention some of them to illustrate the growing interest in solving reference combinatorial problems with Variational Quantum Algorithms. For instance, in Radzihovsky et al. (2019), the authors reformulate the Traveling Salesman Problem (TSP) as an appropriate Hamiltonian matrix and test it on small instances of three and four cities. The authors of Tabi et al. (2020) reformulate and solve the Graph Coloring problem for a dozen of nodes. The authors of Kurowski et al. (2023) study the Job Shop Scheduling problem, find a suitable formulation, and implement it on an artificial instance with three machines and three jobs, each of them containing one or two operations. We end by citing a benchmark provided in Khumalo et al. (2022), that compares classical techniques, among them the Simulated Annealing, and quantum techniques including VQAs on NISQ quantum computers. It applies to the TSP and the Quadratic Assignment problem and shows that in terms of running time and quality solution, the classical methods significantly outperform the quantum ones. Notice that besides the latter paper, the mention of the computation time for VQAs in literature is rare. It could be explained by the fact that most of the experiments are tested on trivial instances and solved with quantum algorithms simulated on classical high-performance computers due to the noise on current quantum hardware (De Palma et al., 2023).

5.2 Improvements and adaptations of QAOA

Despite the theoretical limitations displayed above, QAOA has leverages (guiding function, parametrized quantum circuit, classical optimizer, etc.) that are still of interest in the literature. We display some of these studies on different choices of leverages that empirically improve QAOA performances.

First, the guiding function is mainly the mean function (14) as in the seminal paper of QAOA. However, both the CVaR function (18) and the Gibbs function (16) give an alternative to the mean function and show empirical improvements. For the former, several optimization problems, such as MAX-CUT, Maximum Stable Set, MAX-3SAT, etc., are solved with QAOA in Barkoutsos et al. (2020) and show better results with faster convergence. For the latter, the authors of Li et al. (2020) display better results solving MAX-CUT with this guiding function. Notice that comparing these improvements is hard because they use different classical optimizers. A different method to guide the optimization, presented in Amaro et al. (2022) as a Filtering Variational Quantum Algorithm, substitutes the guiding function for *filtering functions*. A *filtering function* is a function associated with a *filtering operator* that, given a Hamiltonian matrix and a quantum state, essentially modifies the latter by increasing the probability of eigenvectors with low eigenvalues and decreasing the probability of eigenvectors with high eigenvalues. Empirical results show improvements in the quality solution and the speed of convergence for the weighted Max-Cut problem.

Second, the choice of the quantum circuit is challenged in the literature. The underlying question is whether the quantum circuit of QAOA (see Subsection 4.2) is a good choice for finite depth p. Several papers suggest better circuits based on empirical results for small depth and small instances. For example, VQA with the circuit proposed in Barkoutsos et al. (2020) has better performances than QAOA, or with the *Bang-Bang* circuit described in Yang et al. (2017). The benefit of entanglement gates in the circuit is also discussed in Nannicini (2019) without giving a clear advantage. One can refer to the survey (Blekos et al., 2024) for other variants of quantum circuits. Notice that the authors of Egger et al. (2021) suggest to warm-start QAOA with either a continuous relaxation or a randomized rounding. This consists of initializing the quantum circuit with a solution of a continuous relaxation (Quadratic Programming or Semi Definite Programming), respectively with a randomly rounded solution of a continuous relaxation, instead of the state $|+\rangle^{\otimes n}$. In both cases, QAOA performances for small p are better with a warm-start. Other initialization techniques are proposed in the literature. Among them, the annealing-inspired method (Sack & Serbyn, 2021) shows that a random parameters initialization on random graphs for the MAX-CUT problem is outperformed by the use of a Trotterized Quantum Annealing initialization. Another proposition is to initialize the parameters by using Machine Learning models (Alam et al., 2020). Specifically, looking for Machine Learning models that enable the prediction of parameters close to the optimal is relevant because there is an observed correlation between parameters of low-depth and high-depth circuits for some problems. For instance, this type of initialization empirically reduces the number of iterations of the classical optimizer of 44.9% on average for MAX-CUT problems.

The choice of the classical optimizer represents another leverage, where both gradient-based and gradient-free optimizers can be used. The most encountered gradient-based methods used in VQAs in the literature are the Gradient Descent (Ruder, 2016), the Broyden-Fletcher-Goldfarb-Shanno algorithm (Fortran et al., 1992), and the stochastic optimization algorithm ADAM (Kingma & Ba, 2014). As gradient-free methods, we can find the constrained optimization with linear approximation algorithm COBYLA (Powell, 1994), the simplex-based algorithm Nelder-Mead (Nelder & Mead, 1965), and the simultaneous perturbation stochastic approximation algorithm SPSA (Spall, 1992). The authors of Nannicini (2019) advise the choice of a global optimizer rather than a local optimizer to avoid numerous local optima. In Soloviev et al. (2022), the authors choose the gradient-free evolutionary algorithm called Estimation of Distribution Algorithm as the classical optimizer. This algorithm generates new solutions from a probabilistic model that depends on the best solutions of the previous iterations. They empirically show that it improves the results compared to traditional optimizers as mentioned above. The expression of the quantum circuit can also produce barren plateaus, hardening the optimization (McClean et al., 2018; Holmes et al., 2022). However, the authors of Mastropietro et al. (2023) indicate that re-starting the optimization when reaching barren plateaus, where the new parameters are chosen using a stochastic process, improves QAOA performances on MAX-CUT problems. Notice that the classical optimization loop is not always required. Indeed, the authors of Brandao et al. (2018) express that for some classes of QUBO problems, the (near) optimal parameters do not depend on the instance, allowing to train parameters before executing the quantum circuit only once. This is called the *concentration of parameters* and is numerically illustrated on random 3-regular graphs of 20 nodes for low-depth circuits regarding the MAX-CUT objective function. We do not list all the possible choices of VQA leverages. The surveys (Cerezo et al., 2021; Blekos et al., 2024) propose other possibilities.

In parallel, several algorithms derived from QAOA are appearing in the literature. An adaptation of QAOA is the Recursive-QAOA (Bravyi et al., 2022), also called RQAOA. It applies first several times QAOA to the problem in order to reduce its size, namely the number of variables, according to the correlation that appears between some variables. Then, it solves with classical brute force the resulting smaller problem. This algorithm seems to be competitive compared to QAOA for problems such as MAX-k-CUT. The authors of Zhu et al. (2022) propose another algorithm, the Adaptive-QAOA, that converges faster than QAOA on some instances of MAX-CUT. It consists of applying recursively QAOA, increasing step by step the depth of the quantum circuit.

Eventually, there is ongoing work on the formulation and implementation of the QAOA circuit to ease and lighten QAOA implementation. For instance, the authors of Nüßlein et al. (2022) present an algorithmic method to reduce the growth of the QUBO matrix size with the problem size for the k-SAT problem and the Hamiltonian Cycles problem. Besides that, the authors of Herrman et al. (2021) expose a *global variable substitution method* that, given an initial linear formulation of a 3-SAT problem, exploits the advantage of product representation in QAOA and, thus, minimizes the circuit depth.[4] The shallower depth, the more efficient the implementation of the quantum circuit for NISQ computers. A different approach is proposed in Nagarajan et al. (2021), that is, given a quantum circuit, minimizes the circuit depth by solving a Mixed-Integer Program, with an optimality guarantee on the

[4] Notice that here we talk about the general depth, not p, which is the longest path in the circuit, namely the maximum number of gates executed on a qubit.

quantum circuit produced (it can find up to 57% reduction of the number of gates). It applies to any gate-based quantum algorithm and thus, can be applied to QAOA and, more generally, to VQAs.

Notice that solving combinatorial optimization problems using QAOA involves first formulating them into unconstrained problems, and more precisely into QUBO for easier circuits. There are essentially two types of formulation. The first and most common one is to integrate constraints as suitable penalty terms into the objective function. For instance, in Lucas (2014) we find formulations of Karp's 21 NP-complete problems, and the authors of Oh et al. (2019) tackle the k-coloring graph problem with the same method. The tutorial (Glover et al., 2019) addresses more general cases for this formulation problem methodology. The second type of formulation is to change the expression of the *mixing* Hamiltonian (referred to as H_B in Sect. 4). Initially presented in Hadfield et al. (2019), the main idea is to make this Hamiltonian varying the quantum state only in the feasible search space. Some problems have been specifically studied with this method, such as the Traveling Salesman problem (Ruan et al., 2020; Radzihovsky et al., 2019). Notice that there exists a third way to deal with constrained problems, which relies on the quantum Zeno dynamics. This consists of a generic method that essentially projects the quantum states into the feasible space (Herman et al., 2023). This requires auxiliary qubits which are measured in the middle of VQA or QAOA circuit. This shows improvements of QAOA's performances, compared to the method with penalty terms, for the portfolio optimization problem numerically assessed on instances with less than ten assets.

6 Conclusion

In this tutorial, we have provided a mathematical description of VQAs and how they are translated into quantum algorithms. There has been a growing interest in VQAs, and more particularly QAOA, mostly because of their compatibility with NISQ computers. We have described how VQAs raise interesting questions regarding the choice of guiding functions and quantum circuits, among others. They also raise many technical questions that we did not address in this work, such as the noise at the implementation level (Lao et al., 2019).

Since its inception, researchers have argued about the potential advantages of QAOA over classical optimization algorithms (Farhi & Harrow, 2016; Lotshaw et al., 2021; Marwaha & Hadfield, 2022). This research seems to indicate that there is currently no scientific evidence that QAOA will soon beat classical heuristics. The available results even suggest the opposite, such as Hastings (2019) showing that local search algorithms can do better at low depth. Moreover, the current numerical results are hard to assess because they are run on small instances for which optimal or near-optimal solutions can be obtained easily with classical algorithms. Hopefully, the quickly growing capabilities of quantum computers will soon lead to a better understanding of the numerical efficiency of such algorithms (Guerreschi & Matsuura, 2019).

A Basics of linear algebra

This section provides some basic notions of linear algebra used throughout the tutorial. Notice that some specific definitions, such as unitary and Hermitian matrices, are introduced together with their use in quantum computing.

Definition 40 (Tensor product) Let $A = \begin{pmatrix} a_{11} & \dots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nm} \end{pmatrix} \in \mathcal{M}_{n,m}(\mathbb{C})$ and $B = \begin{pmatrix} b_{11} & \dots & b_{1q} \\ \vdots & \ddots & \vdots \\ b_{p1} & \dots & b_{pq} \end{pmatrix} \in \mathcal{M}_{p,q}(\mathbb{C})$ be two complex matrices. Then, we define their tensor product, a bilinear operation, as

$$A \otimes B = \begin{pmatrix} a_{11}B & \dots & a_{1m}B \\ \vdots & \ddots & \vdots \\ a_{n1}B & \dots & a_{nm}B \end{pmatrix} \in \mathcal{M}_{np,mq}(\mathbb{C}).$$

where $a_{ij}B = \begin{pmatrix} a_{ij}b_{11} & \dots & a_{ij}b_{1q} \\ \vdots & \ddots & \vdots \\ a_{ij}b_{p1} & \dots & a_{ij}b_{pq} \end{pmatrix} \in \mathcal{M}_{p,q}(\mathbb{C}).$

Proposition 41 *(Tensor product properties) We draw attention to some useful properties of the tensor product:*

- $A \otimes B \neq B \otimes A$
- $(A \otimes B) \otimes C = A \otimes (B \otimes C) = A \otimes B \otimes C$
- $A \otimes (cB) = (cA) \otimes B = c(A \otimes B)$, *for* $c \in \mathbb{C}$

Definition 42 (Braket notation) We also introduce the *braket* notation used in quantum computing. Let $\psi = \begin{pmatrix} \psi_1 \\ \vdots \\ \psi_{2^n} \end{pmatrix} \in \mathbb{C}^{2^n}$ be a column vector. We note $|\psi\rangle$, said *ket* ψ, the vector ψ itself. Thus, we define $\langle\psi|$, said *bra* ψ, the conjugate transpose of $|\psi\rangle$. Specifically,

$$|\psi\rangle = \psi = \begin{pmatrix} \psi_1 \\ \vdots \\ \psi_{2^n} \end{pmatrix},$$
$$\langle\psi| = \bar{\psi}^T = \begin{pmatrix} \bar{\psi_1} & \dots & \bar{\psi_{2^n}} \end{pmatrix}.$$

Thus, *ket* vectors are always column vectors whereas *bra* vectors are row vectors. With this notation, the norm of the vector ψ is

$$\langle\psi\rangle = \sum_{i=1}^{2^n} |\psi_i|^2.$$

We also consider the complex inner product

$$\psi, \phi \mapsto \langle\psi\rangle\phi,$$

where we use the notation $\langle \psi | \cdot | \phi \rangle = \langle \psi \rangle \phi$. Notice that it is not commutative. Indeed, for $\psi = \begin{pmatrix} \psi_1 \\ \vdots \\ \psi_{2^n} \end{pmatrix} \in \mathbb{C}^{2^n}$ and $\phi = \begin{pmatrix} \phi_1 \\ \vdots \\ \phi_{2^n} \end{pmatrix} \in \mathbb{C}^{2^n}$, we have:

$$\begin{aligned} \langle \phi \rangle \psi &= \begin{pmatrix} \bar{\phi_1} & \dots & \bar{\phi_{2^n}} \end{pmatrix} \begin{pmatrix} \psi_1 \\ \vdots \\ \psi_{2^n} \end{pmatrix} \\ &= \sum_{i=1}^{2^n} \bar{\phi_i} \psi_i \\ &\neq \sum_{i=1}^{2^n} \phi_i \bar{\psi_i} = \langle \psi \rangle \phi . \end{aligned}$$

Definition 43 (Unitary operator associated with Hermitian matrix) Given a Hermitian matrix A, we define its associated quantum gate $\mathrm{Exp}(A, t)$ parametrized by the parameter $t \in \mathbb{R}$ as:

$$\begin{aligned} \mathrm{Exp}(A, t) &= e^{-iAt} \\ &= \sum_{k=0}^{\infty} \frac{1}{k!} (-i)^k t^k A^k . \end{aligned}$$

Because A is Hermitian, $\mathrm{Exp}(A, t)$ is a unitary matrix.

B Omitted proofs

B.1 Guiding function properties

B.1.1 Proof of Proposition 25

Proposition 25 *Function g_{mean} is a guiding function.*

Proof Let us prove that g_{mean} is continuous. Let $x \in \{0, 1\}^n$. The function $\theta \mapsto p_\theta(x) = |\langle x | U(\theta) | 0_n \rangle|^2$ is continuous, because each coefficient of $U(\theta)$ is continuous. Thus, because multiplication and addition preserve continuity, g_{mean} is continuous.

We prove by contradiction that (11) holds. Let $\theta \in \mathcal{G}$ and let us consider the quantum state $|\psi(\theta)\rangle = U(\theta)|0_n\rangle$. Its decomposition in the canonical basis is $|\psi(\theta)\rangle = \sum_{x \in \{0,1\}^n} \psi_x |x\rangle$.

Assume that $|\psi(\theta)\rangle \notin \mathcal{F}_{\text{quant}}$. By definition, there exists $x_0 \in \{0, 1\}^n$ such that $x_0 \notin \mathcal{F}$ and $|\psi_{x_0}| \neq 0$. Thus,

$$\begin{aligned} g_{\text{mean}}(\theta) &= \sum_{x \in \{0,1\}^n} |\psi_x|^2 f(x) \\ &= \sum_{x \in \mathcal{F}} |\psi_x|^2 f(x) + \sum_{x \notin \mathcal{F}} |\psi_x|^2 f(x) \\ &= \left(\sum_{x \in \mathcal{F}} |\psi_x|^2 \right) f^* + \sum_{x \notin \mathcal{F}} |\psi_x|^2 f(x) , \end{aligned}$$

where f^* is the optimal value of f, reached on $\mathcal{F}$. By definition, $f(x) > f^*, \forall x \notin \mathcal{F}$, and because we assume that $|\psi_{x_0}| \neq 0$, thus the second term of the sum is bounded below as follows: $\sum_{x \notin \mathcal{F}} |\psi_x|^2 f(x) > \left(\sum_{x \notin \mathcal{F}} |\psi_x|^2\right) f^*$, where $\sum_{x \notin \mathcal{F}} |\psi_x|^2 \geq |\psi_{x_0}|^2 > 0$. Thus,

$$g_{\text{mean}}(\theta) > \left(\sum_{x \in \mathcal{F}} |\psi_x|^2\right) f^* + \left(\sum_{x \notin \mathcal{F}} |\psi_x|^2\right) f^* = f^* .$$

This contradicts the previous statement that $\theta \in \mathcal{G}$, as one readily verifies that the minimum of g_{mean} is $g^*_{\text{mean}} = f^*$. □

B.1.2 Proof of Proposition 27

Proposition 27 *Let $\eta > 0$. Function $g_{G,\eta}$ is a guiding function.*

Proof Let us prove that $g_{G,\eta}$ is continuous. Let $x \in \{0, 1\}^n$. The function $\theta \mapsto p_\theta(x) = |\langle x|U(\theta)|0_n\rangle|^2$ is continuous, because each coefficient of $U(\theta)$ is continuous. Thus, because multiplication, addition, and composition preserve continuity, $g_{G,\eta}$ is continuous.

Let $\eta > 0$. Let us prove by contradiction that (11) holds. As before, let $\theta \in \mathcal{G}$ and let us consider the quantum state $|\psi(\theta)\rangle = U(\theta)|0_n\rangle$. Its decomposition in the canonical basis is $|\psi(\theta)\rangle = \sum_{x \in \{0,1\}^n} \psi_x |x\rangle$. Assume that $|\psi(\theta)\rangle \notin \mathcal{F}_{\text{quant}}$. Thus, there exists $x_0 \in \{0, 1\}^n$ such that $x_0 \notin \mathcal{F}$ and $|\psi_{x_0}| \neq 0$. Thus,

$$\begin{aligned} g_{G,\eta}(\theta) &= -\ln\left(\sum_{x \in \{0,1\}^n} |\psi_x|^2 e^{-\eta f(x)}\right) \\ &= -\ln\left(\sum_{x \in \mathcal{F}} |\psi_x|^2 e^{-\eta f(x)} + \sum_{x \notin \mathcal{F}} |\psi_x|^2 e^{-\eta f(x)}\right) \\ &= -\ln\left(\left(\sum_{x \in \mathcal{F}} |\psi_x|^2\right) e^{-\eta f^*} + \sum_{x \notin \mathcal{F}} |\psi_x|^2 e^{-\eta f(x)}\right) , \end{aligned}$$

where f^* is the optimal value of f, reached on $\mathcal{F}$. By definition, $f(x) > f^*, \forall x \notin \mathcal{F}$, and because $\eta > 0$, we have $e^{-\eta f(x)} < e^{-\eta f^*}$. Moreover, we assume that $|\psi_{x_0}| \neq 0$, thus the second term of the sum in the logarithm is bounded above as follows: $\sum_{x \notin \mathcal{F}} |\psi_x|^2 e^{-\eta f(x)} < \left(\sum_{x \notin \mathcal{F}} |\psi_x|^2\right) e^{-\eta f^*}$, where $\sum_{x \notin \mathcal{F}} |\psi_x|^2 \geq |\psi_{x_0}|^2 > 0$. Thus, because $y \mapsto -\ln(y)$ is a decreasing function,

$$g_{G,\eta}(\theta) > -\ln\left(\left(\sum_{x \in \mathcal{F}} |\psi_x|^2\right) e^{-\eta f^*} + \left(\sum_{x \notin \mathcal{F}} |\psi_x|^2\right) e^{-\eta f^*}\right) = \eta f^* .$$

This contradicts the previous statement that $\theta \in \mathcal{G}$. Indeed, we can easily verify that the minimum of $g_{G,\eta}$ is $g^*_{G,\eta} = \eta f^*$. □

B.1.3 Proof of Proposition 31

Proposition 31 *Let $\alpha \in]0, 1[$. Function $g_{C,\alpha}$ is a pseudo-guiding function.*

Proof For the same reason as for g_{mean}, $g_{C,\alpha}$ is continuous.

Let $\alpha \in]0, 1[, \theta \in \mathcal{G}$ and let us consider the quantum state $|\psi(\theta)\rangle = U(\theta)|0_n\rangle$. Its decomposition in the canonical basis is $|\psi(\theta)\rangle = \sum_{x \in \{0,1\}^n} \psi_x |x\rangle$.

Let us prove that either $|\psi(\theta)\rangle \in \mathcal{F}_{\text{quant}}$ or that $\sum_{x \notin \mathcal{F}} |\psi_x|^2 = \sum_{x=N_{\mathcal{F}}+1}^{2^n} |\psi_{x_i}|^2 < 1 - \alpha$, where $N_{\mathcal{F}}$ is the index that delimits $\mathcal{F}$, specifically,

$$\mathcal{F} = \{x_i, i \in [N_{\mathcal{F}}]\}.$$

There are two cases for N_α (we recall that N_α depends on θ):

- If $N_\alpha \leq N_{\mathcal{F}}$, thus

  $$\begin{aligned} g_{C,\alpha}(\theta) &= \frac{1}{\sum\limits_{i \in [N_\alpha] \subseteq [N_{\mathcal{F}}]} |\psi_{x_i}|^2} \sum_{i \in [N_\alpha] \subseteq [N_{\mathcal{F}}]} |\psi_{x_i}|^2 f(x_i) \\ &= f^*, \end{aligned}$$

  where f^* is the optimal value of f, reached on $\mathcal{F}$. This contradicts (11) but the probability of sampling non-optimal solutions when measuring $|\psi(\theta)\rangle$ is strictly lower than $1 - \alpha$. Indeed, by definition of N_α,

  $$\sum_{i=1}^{N_{\mathcal{F}}} |\psi_{x_i}|^2 \geq \sum_{i=1}^{N_\alpha} |\psi_{x_i}|^2 \geq \alpha.$$

- Otherwise, $N_\alpha > N_{\mathcal{F}}$. Let us prove by contradiction that $|\psi(\theta)\rangle \in \mathcal{F}_{\text{quant}}$. Assume that $|\psi(\theta)\rangle \notin \mathcal{F}_{\text{quant}}$. Thus, there exists $k > N_{\mathcal{F}}$ such that $|\psi_k| \neq 0$. We can show that $g_{C,\alpha}(\theta) > f^*$ using essentially the same proof as that of Proposition 25 for $x_0 = \min\{k > N_{\mathcal{F}} : |\psi_k| \neq 0\}$. This contradicts the statement that $\theta \in \mathcal{G}$, because the minimum of $g_{C,\alpha}$ is $g^*_{C,\alpha} = f^*$. □

B.2 Quantum circuit of QAOA

B.2.1 Proof of Proposition 34

Proposition 34 *The first block* $\text{Exp}(H_f, \gamma)$ *parametrized by* $\gamma \in \mathbb{R}$ *is*

$$\text{Exp}(H_f, \gamma) = \prod_{\alpha \in \{0,1\}^n} \text{Exp}(\bigotimes_{i=1}^{n} Z_i^{\alpha_i}, h_\alpha \gamma).$$

The second block $\text{Exp}(H_B, \beta)$ *parametrized by* $\beta \in \mathbb{R}$ *is*

$$\text{Exp}(H_B, \beta) = \bigotimes_{i=1}^{n} R_{X,i}(2\beta).$$

Proof Let $\gamma \in \mathbb{R}$ and let us consider the first block $\text{Exp}(H_f, \gamma)$. By Definition 43, and because each pair of matrices of the family $\{\bigotimes_{i=1}^n Z_i^{\alpha_i} : \alpha = (\alpha_1, \ldots, \alpha_n) \in \{0,1\}^n\}$ commutes two by two,

$$\begin{aligned} \text{Exp}(H_f, \gamma) &= e^{-i(\sum_{\alpha=(\alpha_1,\ldots,\alpha_n)\in\{0,1\}^n} h_\alpha \bigotimes_{i=1}^n Z_i^{\alpha_i})\gamma} && (23) \\ &= \prod_{\alpha=(\alpha_1,\ldots,\alpha_n)\in\{0,1\}^n} e^{-ih_\alpha \bigotimes_{i=1}^n Z_i^{\alpha_i} \gamma} && (24) \\ &= \prod_{\alpha=(\alpha_1,\ldots,\alpha_n)\in\{0,1\}^n} \text{Exp}(\bigotimes_{i=1}^{n} Z_i^{\alpha_i}, h_\alpha \gamma)\,. && (25) \end{aligned}$$

Let $\beta \in \mathbb{R}$ and let us consider the second block $\operatorname{Exp}(H_B, \beta)$. For more readability, we write $X_i = R_{X,i}(\pi)$ the application of matrix $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ on qubit i. With the same development as above, and because each pair of matrices of the family $\{X_i : i \in [n]\}$ commutes two by two,

$$\operatorname{Exp}(H_B, \beta) = \prod_{i=1}^{n} \operatorname{Exp}(X_i, \beta).$$

Let $i \in [n]$. Thus,

$$\operatorname{Exp}(X_i, \beta) = \sum_{k=0}^{\infty} \frac{1}{k!} (-i)^k \beta^k X_i^k \tag{26}$$

$$= \sum_{k=0}^{\infty} \frac{1}{(2k)!} (-i)^{2k} \beta^{2k} X_i^{2k} + \sum_{k=0}^{\infty} \frac{1}{(2k+1)!} (-i)^{(2k+1)} \beta^{(2k+1)} X_i^{(2k+1)} \tag{27}$$

$$= \sum_{k=0}^{\infty} \frac{1}{(2k)!} (-i)^{2k} \beta^{2k} I + \sum_{k=0}^{\infty} \frac{1}{(2k+1)!} (-i)^{(2k+1)} \beta^{(2k+1)} X_i \tag{28}$$

$$= \sum_{k=0}^{\infty} \frac{1}{(2k)!} (-1)^{k} \beta^{2k} I - i \sum_{k=0}^{\infty} \frac{1}{(2k+1)!} (-1)^{k} \beta^{(2k+1)} X_i \tag{29}$$

$$= \cos(t) I - i \sin(t) X_i \tag{30}$$

$$= R_{X,i}(2\beta)\,, \tag{31}$$

where line (27) exploits the fact that $X^2 = I$, implying $X^{2k} = I$ and $X^{2k+1} = X$. Moreover, line (28) applies the definition of the complex number i, and one can recognize the power series of cosinus and sinus functions. □

B.2.2 Proof of Proposition 35

Proposition 35 *For the case of QUBO, the expression of* $\operatorname{Exp}(H_f, \gamma)$ *simplifies in*

$$\operatorname{Exp}(H_f, \gamma) = \left(\bigotimes_{i=1}^{n} R_{Z,i}(2 h_{ii} \gamma) \right) \prod_{i<j} CX_{i,j} R_{Z,j}(2 h_{i,j} \gamma) CX_{i,j},$$

and is rather easily implemented with universal quantum gates.

Proof Let $\gamma \in \mathbb{R}$. The application of (23)–(25) to the case of QUBO gives

$$\begin{aligned} \operatorname{Exp}(H_f, \gamma) &= e^{-i(\sum_{i=1}^{n} h_{ii} Z_i + \sum_{i<j} h_{ij} Z_i \otimes Z_j)\gamma} \\ &= \prod_{i=1}^{n} e^{-i Z_i h_{ii} \gamma} \prod_{i<j} e^{-i Z_i \otimes Z_j h_{ij} \gamma} \\ &= \prod_{i=1}^{n} \operatorname{Exp}(Z_i, h_{ii}\gamma) \prod_{i<j} \operatorname{Exp}(Z_i \otimes Z_j, h_{ij}\gamma)\,. \end{aligned}$$

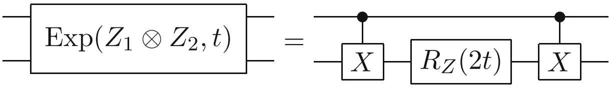


Fig. 9 Decomposition of Exp($Z_1 \otimes Z_2, t$) into universal gates

Then, let us prove that, for $i \in [n]$ and $t \in \mathbb{R}$, $\operatorname{Exp}(Z_i, t) = R_{Z,i}(2t)$. The same development as above (27)–(30), replacing X by Z that have the same property $Z^2 = I$, gives

$$\begin{aligned} \operatorname{Exp}(Z_i, t) &= \cos(t)I - i\sin(t)Z_i && (32)\\ &= R_{Z,i}(2t)\,. && (33) \end{aligned}$$

Eventually, line (32) is the application of the gate $\begin{pmatrix} e^{-it} & 0 \\ 0 & e^{it} \end{pmatrix} = R_Z(2t)$ on qubit i, and the identity on the others.

It remains to prove that, for $i < j \in [n]$ and $t \in \mathbb{R}$, $\operatorname{Exp}(Z_i \otimes Z_j, t) = CX_{i,j}R_{Z,j}(2t)CX_{i,j}$. Following the same developments as above, we have

$$\operatorname{Exp}(Z_i \otimes Z_j, t) = \cos(t)I - i\sin(t)Z_i \otimes Z_j\,. \qquad (34)$$

We consider the two-qubit system that corresponds to the qubit i as the first qubit and the qubit j as the second qubit (the others are unchanged by the transformation). Thus, it remains to prove the equality of the two circuits depicted on Fig. 9.

On the one hand, (34) is the application of the gate $= \begin{pmatrix} R_Z(2t) & 0 \\ 0 & R_Z(-2t) \end{pmatrix}$ to this system. Indeed,

$$\begin{aligned} \cos(t)I - i\sin(t)Z_1 \otimes Z_2 &= \cos(t)\begin{pmatrix} 1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1 \end{pmatrix} - i\sin(t)\begin{pmatrix} 1&0&0&0\\0&-1&0&0\\0&0&-1&0\\0&0&0&1 \end{pmatrix}\\ &= \begin{pmatrix} e^{-it}&0&0&0\\0&e^{it}&0&0\\0&0&e^{it}&0\\0&0&0&e^{-it} \end{pmatrix}\\ &= \begin{pmatrix} R_Z(2t) & 0\\ 0 & R_Z(-2t) \end{pmatrix}. \end{aligned}$$

On the other hand, the composition of gates $CX_{1,2}R_{Z,2}(2t)CX_{1,2}$ on this system amounts to

$$\begin{aligned} CX_{1,2}R_{Z,2}(2t)CX_{1,2} &= \begin{pmatrix} I & 0\\ 0 & X \end{pmatrix}\begin{pmatrix} R_Z(2t) & 0\\ 0 & R_Z(2t) \end{pmatrix}\begin{pmatrix} I & 0\\ 0 & X \end{pmatrix}\\ &= \begin{pmatrix} R_Z(2t) & 0\\ 0 & XR_Z(2t)X \end{pmatrix}\\ &= \begin{pmatrix} R_Z(2t) & 0\\ 0 & R_Z(-2t) \end{pmatrix}. \end{aligned}$$

Thus, the proof results from replacing t by appropriate values $h_{ii}\gamma$ for $i \in [n]$ and $h_{ij}\gamma$ for $i < j$. □

B.2.3 Proof of Lemma 36

Lemma 36 $\forall n \in \mathbb{N}^*$,

$$(I^{\otimes n-1} \otimes X)e^{-itZ^{\otimes n}}(I^{\otimes n-1} \otimes X) = e^{itZ^{\otimes n}}.$$

Proof Let P_n be the statement

$$(I^{\otimes n-1} \otimes X)e^{-itZ^{\otimes n}}(I^{\otimes n-1} \otimes X) = e^{itZ^{\otimes n}}.$$

Let us prove by induction that P_n holds for all integer $n \in \mathbb{N}^*$.
Base case: Let us prove P_1. According to (33), $e^{-itZ} = R_Z(2t)$, thus,

$$\begin{aligned} Xe^{-itZ}X &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} e^{-it} & 0 \\ 0 & e^{it} \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\ &= \begin{pmatrix} e^{it} & 0 \\ 0 & e^{-it} \end{pmatrix} \\ &= e^{itZ}. \end{aligned}$$

Induction step: Let $n \geq 1$ be given and suppose P_n. Let us prove P_{n+1}. According to (35), we have $e^{-itZ^{\otimes n+1}} = \begin{pmatrix} e^{-itZ^{\otimes n}} & 0 \\ 0 & e^{itZ^{\otimes n}} \end{pmatrix}$. Thus,

$$\begin{aligned} &(I^{\otimes n} \otimes X)e^{-itZ^{\otimes n+1}}(I^{\otimes n} \otimes X) = (I \otimes I^{\otimes n-1} \otimes X)e^{-itZ^{\otimes n+1}}(I \otimes I^{\otimes n-1} \otimes X) \\ &= \begin{pmatrix} I^{\otimes n-1} \otimes X & 0 \\ 0 & I^{\otimes n-1} \otimes X \end{pmatrix} \begin{pmatrix} e^{-itZ^{\otimes n}} & 0 \\ 0 & e^{itZ^{\otimes n}} \end{pmatrix} \begin{pmatrix} I^{\otimes n-1} \otimes X & 0 \\ 0 & I^{\otimes n-1} \otimes X \end{pmatrix} \\ &= \begin{pmatrix} (I^{\otimes n-1} \otimes X)e^{-itZ^{\otimes n}}(I^{\otimes n-1} \otimes X) & 0 \\ 0 & (I^{\otimes n-1} \otimes X)e^{itZ^{\otimes n}}(I^{\otimes n-1} \otimes X) \end{pmatrix}. \end{aligned}$$

By induction hypothesis P_n,

$$\begin{aligned} (I^{\otimes n} \otimes X)e^{-itZ^{\otimes n+1}}(I^{\otimes n} \otimes X) &= \begin{pmatrix} e^{itZ^{\otimes n}} & 0 \\ 0 & e^{-itZ^{\otimes n}} \end{pmatrix} \\ &= e^{itZ^{\otimes n+1}}. \end{aligned}$$

□

B.2.4 Proof of Proposition 37

Proposition 37 *Let us consider the subsystem composed of the N qubits to which the Z gate is applied. Specifically, $N = |\{\alpha_i\ i \in [n] : \alpha_i = 1\}|$, and we renumber the qubits in question in $[N]$. Thus, for $N \geq 2$, the term $\mathrm{Exp}(\bigotimes_{i=1}^{n} Z^{\alpha_i}, t)$ on this subsystem simplifies in*

$$\mathrm{Exp}(Z^{\otimes N}, t) = \prod_{j=0}^{N-2} CX_{1,N-j} R_{Z,N}(2t) \prod_{j=0}^{N-2} CX_{1,N-j}.$$

We represent this decomposition on Fig. 7.

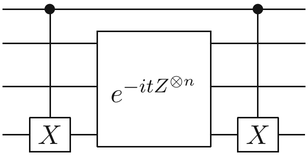


Fig. 10 Circuit representation of $\prod_{j=0}^{n-1} CX_{1,n+1-j} R_{Z,n+1}(2t) \prod_{j=0}^{n-1} CX_{1,n+1-j}$

Proof Let P_n be the statement

$$\mathrm{Exp}(Z^{\otimes n}, t) = \prod_{j=0}^{n-2} CX_{1,n-j} R_{Z,n}(2t) \prod_{j=0}^{n-2} CX_{1,n-j}.$$

Let us prove by induction that P_n holds for all integer $n \geq 2$.
Base case: Proposition 35 proves P_2.
Induction step: Let $n \geq 2$ be given and suppose P_n. Let us prove P_{n+1}.
On the one hand,

$$\mathrm{Exp}(Z^{\otimes n+1}, t) = e^{-itZ^{\otimes n+1}} = e^{-itZ\otimes Z^{\otimes n}}$$

where $Z \otimes Z^{\otimes n} = \begin{pmatrix} Z^{\otimes n} & 0 \\ 0 & -Z^{\otimes n} \end{pmatrix}$. This latter matrix is diagonal, thus,

$$\mathrm{Exp}(Z^{\otimes n+1}, t) = \begin{pmatrix} e^{-itZ^{\otimes n}} & 0 \\ 0 & e^{itZ^{\otimes n}} \end{pmatrix}. \tag{35}$$

On the other hand, we compute the term

$$\prod_{j=0}^{n-1} CX_{1,n+1-j} R_{Z,n+1}(2t) \prod_{j=0}^{n-1} CX_{1,n+1-j}$$
$$= CX_{1,n+1} \left(\prod_{j=1}^{n-2} CX_{1,n+1-j} R_{Z,n+1}(2t) \prod_{j=1}^{n-2} CX_{1,n+1-j} \right) CX_{1,n+1}$$

that is represented on Fig. 10, where $e^{-itZ^{\otimes n}}$ applies on the qubits 2 to n by induction hypothesis.

Thus,

$$\prod_{j=0}^{n-1} CX_{1,n+1-j} R_{Z,n+1}(2t) \prod_{j=0}^{n-1} CX_{1,n+1-j} = CX_{1,n+1}(I \otimes e^{-itZ^{\otimes n}})CX_{1,n+1}$$
$$= \begin{pmatrix} I^{\otimes n} & 0 \\ 0 & I^{\otimes n-1} \otimes X \end{pmatrix} \begin{pmatrix} e^{-itZ^{\otimes n}} & 0 \\ 0 & e^{-itZ^{\otimes n}} \end{pmatrix} \begin{pmatrix} I^{\otimes n} & 0 \\ 0 & I^{\otimes n-1} \otimes X \end{pmatrix}$$
$$= \begin{pmatrix} e^{-itZ^{\otimes n}} & 0 \\ 0 & (I^{\otimes n-1} \otimes X)e^{-itZ^{\otimes n}}(I^{\otimes n-1} \otimes X) \end{pmatrix}$$
$$= \begin{pmatrix} e^{-itZ^{\otimes n}} & 0 \\ 0 & e^{itZ^{\otimes n}} \end{pmatrix},$$

where the last line comes from the application of Lemma 36. Thus,

$$\mathrm{Exp}(Z^{\otimes n+1}, t) = \prod_{j=0}^{n-1} CX_{1,n+1-j} R_{Z,n+1}(2t) \prod_{j=0}^{n-1} CX_{1,n+1-j},$$

proving P_{n+1}. □

Funding This study was funded by the ANRT (Association Nationale de la Recherche et de la Technologie) through the PhD number 2021/0281 with CIFRE funds.

References

Abbas, A., Ambainis, A., Augustino, B., Bärtschi, A., Buhrman, H., Coffrin, C., Cortiana, G., Dunjko, V., Egger, D. J., Elmegreen, B. G., et al. (2023). Quantum optimization: Potential, challenges, and the path forward. arXiv preprint arXiv:2312.02279.

Alam, M., Ash-Saki, A., & Ghosh, S. (2020). Accelerating quantum approximate optimization algorithm using machine learning. In *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 686–689. IEEE .

Amaro, D., Modica, C., Rosenkranz, M., Fiorentini, M., Benedetti, M., & Lubasch, M. (2022). Filtering variational quantum algorithms for combinatorial optimization. *Quantum Science and Technology, 7*(1), 015021.

Ambainis, A., Balodis, K., Iraids, J., Kokainis, M., Prūsis, K., & Vihrovs, J. (2019). Quantum speedups for exponential-time dynamic programming algorithms. *In Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms,* 1783–1793, SIAM .

Barkoutsos, P. K., Nannicini, G., Robert, A., Tavernelli, I., & Woerner, S. (2020). Improving variational quantum optimization using CVaR. *Quantum, 4*, 256.

Basso, J., Farhi, E., Marwaha, K., Villalonga, B., & Zhou, L. (2021). The quantum approximate optimization algorithm at high depth for maxcut on large-girth regular graphs and the sherrington-kirkpatrick model. arXiv preprint arXiv:2110.14206.

Blekos, K., Brand, D., Ceschini, A., Chou, C.-H., Li, R.-H., Pandya, K., & Summer, A. (2024). A review on quantum approximate optimization algorithm and its variants. *Physics Reports, 1068*, 1–66.

Brandao, F. G., Broughton, M., Farhi, E., Gutmann, S., Neven, H. (2018). For fixed control parameters the quantum approximate optimization algorithm's objective function value concentrates for typical instances. arXiv preprint arXiv:1812.04170.

Bravyi, S., Kliesch, A., Koenig, R., & Tang, E. (2020). Obstacles to variational quantum optimization from symmetry protection. *Physical Review Letters, 125*(26), 260505.

Bravyi, S., Kliesch, A., Koenig, R., & Tang, E. (2022). Hybrid quantum-classical algorithms for approximate graph coloring. *Quantum, 6*, 678.

Cerezo, M., Arrasmith, A., Babbush, R., Benjamin, S. C., Endo, S., Fujii, K., McClean, J. R., Mitarai, K., Yuan, X., Cincio, L., et al. (2021). Variational quantum algorithms. Nature Reviews. *Physics, 3*(9), 625–644.

Crooks, G. E. (2018). Performance of the quantum approximate optimization algorithm on the maximum cut problem. arXiv preprint arXiv:1811.08419.

De Palma, G., Marvian, M., Rouzé, C., & França, D. S. (2023). Limitations of variational quantum algorithms: a quantum optimal transport approach. *PRX Quantum, 4*(1), 010309.

Egger, D. J., Marecek, J., & Woerner, S. (2021). Warm-starting quantum optimization. *Quantum, 5*, 479.

Farhi, E., Gamarnik, D., & Gutmann, S. (2020). The quantum approximate optimization algorithm needs to see the whole graph: A typical case. arXiv preprint arXiv:2004.09002.

Farhi, E., Goldstone, J., Gutmann, S. (2014). A quantum approximate optimization algorithm. arXiv preprint arXiv:1411.4028.

Farhi, E., Goldstone, J., Gutmann, S., & Sipser, M. (2000). Quantum computation by adiabatic evolution. arXiv preprint quant-ph/0001106.

Farhi, E., Goldstone, J., Gutmann, S., & Zhou, L. (2022). The quantum approximate optimization algorithm and the sherrington-kirkpatrick model at infinite size. *Quantum, 6*, 759.

Farhi, E., & Harrow, A. W. (2016). Quantum supremacy through the quantum approximate optimization algorithm. arXiv preprint arXiv:1602.07674.

Fortran, I., Press, W., Teukolsky, S., Vetterling, W., & Flannery, B. (1992). Numerical recipes. *Cambridge, UK, Cambridge University Press*, 01.

Glover, F., Kochenberger, G., & Du, Y. (2019). Quantum Bridge Analytics I a tutorial on formulating and using QUBO models. *4or, 17*, 335–371.

Goemans, M. X., & Williamson, D. P. (1995). Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM), 42*(6), 1115–1145.

Grange, C., Poss, M., & Bourreau, E. (2023). An introduction to variational quantum algorithms for combinatorial optimization problems. *4or, 21*(3), 363–403.

Grover, L. K. (1996) A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219.

Guerreschi, G. G., & Matsuura, A. Y. (2019). QAOA for Max-Cut requires hundreds of qubits for quantum speed-up. *Scientific Reports, 9*(1), 1–7.

Hadfield, S., Wang, Z., Ogorman, B., Rieffel, E. G., Venturelli, D., & Biswas, R. (2019). From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms, 12*(2), 34.

Hastings, M. B. (2019). Classical and quantum bounded depth approximation algorithms. arXiv preprint arXiv:1905.07047, .

Hellemo, L., Barton, P. I., & Tomasgard, A. (2018). Decision-dependent probabilities in stochastic programs with recourse. *Computational Management Science, 15*(3), 369–395.

Herman, D., Shaydulin, R., Sun, Y., Chakrabarti, S., Hu, S., Minssen, P., Rattew, A., Yalovetzky, R., & Pistoia, M. (2023). Constrained optimization via quantum zeno dynamics. *Communications Physics, 6*(1), 219.

Herrman, R., Treffert, L., Ostrowski, J., Lotshaw, P. C., Humble, T. S., & Siopsis, G. (2021). Globally optimizing QAOA circuit depth for constrained optimization problems. *Algorithms, 14*(10), 294.

Holmes, Z., Sharma, K., Cerezo, M., & Coles, P. J. (2022). Connecting ansatz expressibility to gradient magnitudes and barren plateaus. *PRX Quantum, 3*(1), 010313.

Kerenidis, I., & Prakash, A. (2020). A quantum interior point method for LPs and SDPs. *ACM Transactions on Quantum Computing, 1*(1), 1–32.

Kerenidis, I., Prakash, A., & Szilágyi, D. (2021). Quantum algorithms for second-order cone programming and support vector machines. *Quantum, 5*, 427.

Khumalo, M. T., Chieza, H. A., Prag, K., & Woolway, M. (2022). An investigation of IBM quantum computing device performance on combinatorial optimisation problems. *Neural Computing and Applications,* 1–16.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kitaev, A. Y. (1995). Quantum measurements and the Abelian stabilizer problem. arXiv preprint quant-ph/9511026, .

Kurowski, K., Pecyna, T., Slysz, M., Różycki, R., Waligóra, G., & Węglarz, J. (2023). Application of quantum approximate optimization algorithm to job shop scheduling problem. *European Journal of Operational Research, 310*(2), 518–28.

Lao, L., Manzano, D., van Someren, H., & Ashraf, I. (2019). *and C*. G. Almudever. Mapping of quantum circuits onto NISQ superconducting processors. arXiv: Quantum Physics.

Li, L., Fan, M., Coram, M., Riley, P., Leichenauer, S., et al. (2020). Quantum optimization with a novel Gibbs objective function and ansatz architecture search. *Physical Review Research, 2*(2), 023074.

Lotshaw, P. C., Humble, T. S., Herrman, R., Ostrowski, J., & Siopsis, G. (2021). Empirical performance bounds for quantum approximate optimization. *Quantum Information Processing, 20*(12), 1–32.

Lucas, A. (2014). Ising formulations of many NP problems. *Frontiers in physics, 2*, 5.

Marwaha, K., & Hadfield, S. (2022). Bounds on approximating Max kXOR with quantum and classical local algorithms. *Quantum, 6*, 757.

Mastropietro, D., Korpas, G., Kungurtsev, V., & Marecek, J. (2023). Fleming-viot helps speed up variational quantum algorithms in the presence of barren plateaus. arXiv preprint arXiv:2311.18090.

McClean, J. R., Boixo, S., Smelyanskiy, V. N., Babbush, R., & Neven, H. (2018). Barren plateaus in quantum neural network training landscapes. *Nature Communications, 9*(1), 1–6.

Montanaro, A. (2015). Quantum walk speedup of backtracking algorithms. arXiv preprint arXiv:1509.02374.

Montanaro, A. (2020). Quantum speedup of branch-and-bound algorithms. *Physical Review Research, 2*(1), 013056.

Mosseri, R., & Dandoloff, R. (2001). Geometry of entangled states, Bloch spheres and Hopf fibrations. *Journal of Physics A: Mathematical and General, 34*(47), 10243.

Nagarajan, H., Lockwood, O., & Coffrin, C. (2021). QuantumCircuitOpt: An open-source framework for provably optimal quantum circuit design. In *2021 IEEE/ACM Second International Workshop on Quantum Computing Software (QCS)*, pages 55–63. IEEE.

Nannicini, G. (2019). Performance of hybrid quantum-classical variational heuristics for combinatorial optimization. *Physical Review E, 99*(1), 013304.

Nannicini, G. (2021). Fast Quantum Subroutines for the Simplex Method. In M. Singh and D. P. Williamson, editors, *Integer Programming and Combinatorial Optimization - 22nd International Conference, IPCO 2021, Atlanta, GA, USA, May 19-21, 2021, Proceedings*, volume 12707 of *Lecture Notes in Computer Science*, pages 311–325. Springer.

Nelder, J. A., & Mead, R. (1965). A simplex method for function minimization. *The Computer Journal, 7*(4), 308–313.

Nielsen, M. A., & Chuang, I. (2002). Quantum computation and quantum information.

Nüßlein, J., Gabor, T., Linnhoff-Popien, C., & Feld, S. (2022). Algorithmic QUBO formulations for k-SAT and hamiltonian cycles. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 2240–2246.

Oh, Y.-H., Mohammadbagherpoor, H., Dreher, P., Singh, A., Yu, X., & Rindos, A. J. (2019). Solving multi-coloring combinatorial optimization problems using hybrid quantum algorithms. arXiv preprint arXiv:1911.00595.

Peruzzo, A., McClean, J., Shadbolt, P., Yung, M. H., Zhou, X. Q., Love, P. J., Aspuru-Guzik, A., & Obrien, J. L. (2014). A variational eigenvalue solver on a photonic quantum processor. *Nature communications, 5*(1), 4213.

Powell, M. J. (1994). *A direct search optimization method that models the objective and constraint functions by linear interpolation*. Springer.

Preskill, J. (2018). Quantum computing in the NISQ era and beyond. *Quantum, 2*, 79.

Radzihovsky, M., Murphy, J., & Mason, S. (2019). A QAOA solution to the traveling salesman problem using pyQuil.

Ruan, Y., Marsh, S., Xue, X., Liu, Z., Wang, J., et al. (2020). The quantum approximate algorithm for solving traveling salesman problem. *Computers, Materials & Continua, 63*(3), 1237–1247.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747.

Sack, S. H., & Serbyn, M. (2021). Quantum annealing initialization of the quantum approximate optimization algorithm. *Quantum, 5*, 491.

Shapiro, A. (2003). Monte Carlo sampling methods. *Handbooks in Operations Research and Management Science, 10*, 353–425.

Shor, P. W. (1994). Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. Ieee, .

Soloviev, V. P., Larrañaga, P.,& Bielza, C. (2022). Quantum parametric circuit optimization with estimation of distribution algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 2247–2250.

Spall, J. C. (1992). Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control, 37*(3), 332–341.

Tabi, Z., El-Safty, K. H., Kallus, Z., Hága, P., Kozsik, T., Glos, A., & Zimborás, Z. (2020). Quantum optimization for the graph coloring problem with space-efficient embedding. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 56–62. IEEE.

Wang, Z., Hadfield, S., Jiang, Z., & Rieffel, E. G. (2018). Quantum approximate optimization algorithm for MaxCut: A fermionic view. *Physical Review A, 97*(2), 022304.

Wurtz, J., & Love, P. (2021). Maxcut quantum approximate optimization algorithm performance guarantees for p> 1. *Physical Review A, 103*(4), 042612.

Wurtz, J., & Love, P. J. (2022). Counterdiabaticity and the quantum approximate optimization algorithm. *Quantum, 6*, 635.

Yang, Z.-C., Rahmani, A., Shabani, A., Neven, H., & Chamon, C. (2017). Optimizing variational quantum algorithms using Pontryagin's minimum principle. *Physical Review X, 7*(2), 021027.

Zhou, L., Wang, S.-T., Choi, S., Pichler, H., & Lukin, M. D. (2020). Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices. *Physical Review X, 10*(2), 021067.

Zhu, L., Tang, H. L., Barron, G. S., Calderon-Vargas, F., Mayhall, N. J., Barnes, E., & Economou, S. E. (2022). Adaptive quantum approximate optimization algorithm for solving combinatorial problems on a quantum computer. *Physical Review Research, 4*(3), 033029.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.