IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 29 December 2024, accepted 14 January 2025, date of publication 17 January 2025, date of current version 28 January 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3530952

PERSPECTIVE

Evolutionary Algorithms and Quantum Computing: Recent Advances, Opportunities, and Challenges

JUNAID UR REHMAN[1,2,3], MUHAMMAD SHOHIBUL ULUM[4], ABDURRAHMAN WACHID SHAFFAR[4], AMIRUL ADLIL HAKIM[4], MUJIRIN[4], ZAID ABDULLAH[2], (Member, IEEE), HAYDER AL-HRAISHAWI[2,5], (Senior Member, IEEE), SYMEON CHATZINOTAS[2], (Fellow, IEEE), AND HYUNDONG SHIN[4], (Fellow, IEEE)

[1]Department of Electrical Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia
[2]Interdisciplinary Centre for Security, Reliability and Trust (SnT), 1855 Luxembourg City, Luxembourg
[3]Interdisciplinary Research Center for Intelligent Secure Systems, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia
[4]Department of Electronics and Information Convergence Engineering, Kyung Hee University, Giheung-gu, Yongin-si, Gyeonggi-do 17104, South Korea
[5]Department of Electrical Engineering, University of South Florida, Tampa, FL 33620, USA

Corresponding author: Hyundong Shin (hshin@khu.ac.kr)

The work of Hyundong Shin was supported in part by the National Research Foundation of Korea (NRF) grant funded by Korean Government (MSIT) under Grant NRF-2022R1A4A3033401; and in part by the Ministry of Science and ICT (MSIT), South Korea, under the Information Technology Research Center (ITRC) Support Program supervised by the Institute for Information & Communications Technology Planning & Evaluation (IITP) under Grant IITP-2024-2021-0-02046.

ABSTRACT Quantum computers have made significant progress in the last two decades showing great potential in tackling some of the most challenging problems in computing. This ongoing progress creates an opportunity to implement and evaluate quantum-inspired metaheuristics on real quantum devices, with the aim of uncovering potential computational advantages. Additionally, the practical constraints associated with current quantum computers have highlighted a critical need for classical heuristic methods to optimize the tunable parameters of quantum circuits. Nature-inspired metaheuristics have emerged as promising candidates for fulfilling this optimization role. In this paper, we discuss both of these potential directions at the intersection of evolutionary computing and quantum computing while surveying some of the most promising advancements in these directions. We start with the review of quantum-inspired metaheuristics and then explore implementations of some of these quantum-inspired algorithms on physical quantum devices, capitalizing on the progress in quantum computing technology. Furthermore, we investigate the role of nature-inspired metaheuristics in enhancing the performance of noisy intermediate-scale quantum computers by fine-tuning their parameters. Finally, we discuss some of the recent progress at the intersection of both computing frameworks to highlight the current status and potential of the currently available quantum computing hardware. Synergies between these two computing frameworks demonstrate the potential of a strongly symbiotic relation that can contribute to the simultaneous advancements in both of these computing paradigms.

INDEX TERMS Evolutionary algorithms, genetic algorithm, quantum-inspired algorithms, quantum computing.

I. INTRODUCTION

The idea of a computing device based on quantum mechanics began to emerge in early 1980s [1], [2], [3]. Early proposals of quantum computers were mainly motivated by the difficulty of simulating nature at quantum mechanical level. It was theorized that a quantum mechanical device would have advantage in these simulations due to its inherent quantum mechanical nature. Soon it was found that quantum computing devices have a strong potential even beyond simulating

The associate editor coordinating the review of this manuscript and approving it for publication was Derek Abbott.

quantum-mechanics to general computing problems. For example, it was found that quantum computers can have a large computational advantage over classical counterparts in determining some global properties of functions [4], [5]. Another celebrated work is the Shor's factoring algorithm [6] that has the potential to break the modern cryptography [7]. In addition to these celebrated works, there is a plethora of other quantum computing algorithms that offer a speedup over classical counterparts for certain tasks [8]. Currently, these theoretical developments cannot translate to a practical quantum computational advantage due to limited size of available quantum hardware. Indeed, fault-tolerant implementations of these algorithms would require quantum computers consisting of a hundred thousands to millions of physical qubits [7], [9]. On the other hand, the quality and the number of qubits in the available quantum computing hardware are rather limited. Due to these limitations, currently available quantum computing devices are called noisy intermediate-scale quantum (NISQ) devices [10]. The computational capabilities of currently available NISQ-era devices seem to lie near the boundary of capabilities of classical computers as demonstrated on some benchmark computational tasks, yet these devices are not big enough to implement traditional quantum computing algorithms such as Shor's factoring algorithm [11], [12], [13], [14], [15], [16]. To harness the potential of these devices, special-purpose NISQ-era algorithms are actively being developed that incorporate active measures to mitigate the limitations caused by the imperfect NISQ hardware [10], [17].

The *formal* study of heuristics and metaheuristics began in 1940s and 1980s, respectively [18]. Due to their simplicity and intuitive nature for humans, it is difficult to trace back the exact moments in time at which heuristics started being used. Metaheuristics refers to a problem-independent framework that provides a set of guidelines to develop a heuristic algorithm. The same term is also used for a problem-specific implementation of a heuristic algorithm according the guidelines of such a framework [19]. A particularly active area of research within the field of metaheuristics is the development of nature-inspired methods. These methods draw inspiration from natural phenomena or physical systems to design the guidelines for developing heuristic algorithms. [20]

Nature-inspired heuristic algorithms appear in three different contexts in the discussion about quantum computing. Firstly, there exists a class of quantum-inspired evolutionary algorithms. This class consists of *classical* algorithms that take inspiration from quantum mechanical phenomenon such as superposition and inherent randomness of measurement in quantum phenomena. We emphasize that these algorithms are classical, i.e., they are designed for and implemented on classical digital computers. Accordingly, these algorithms are not quantum computing algorithms. Secondly, some nature-inspired heuristic algorithms are designed for and implemented on quantum computing devices. These are truly quantum computing algorithms that run on physical quantum computers. Some of the algorithms discussed in the first context can also be implemented on quantum devices where it is possible to obtain some runtime advantage. Finally, a trademark of NISQ-era quantum computing algorithms is the classical-hybrid nature, i.e., these algorithms utilize classical and quantum computers in tandem to solve some complex optimization/computation problems. Nature-inspired heuristic algorithms may be utilized to carry out the classical computation in these NISQ-era algorithms. The classical computation part in these algorithms involves finding the optimal parameters of a tunable quantum circuit, which is a highly nonconvex optimization problem. In some cases, a metaheuristic algorithm can act as a suitable classical optimizer.

TABLE 1. A comparison of existing literature dealing with the topics of quantum computing and evolutionary computation.[1]

Reference	Evolutionary Computation	Quantum-Inspired (classical)	Quantum Evolutionary Algorithms	C-Q Hybrid Computing	Quantum Implementations
[22]	✓	✓	✓	-	✓
[23]	-	✓	✓	✓	-
[24]	✓	✓	-	-	-
[25]	✓	-	-	-	-
[26]	✓	-	-	-	-
[27]	✓	✓	-	-	-
[24]	✓	✓	-	-	-
This work	✓	✓	✓	✓	✓

The distinction between the first two contexts can be somewhat confusing and there is no well-defined terminology to distinguish these two types. This lack of distinction becomes even more confusing since some of the quantum-inspired evolutionary algorithms (classical) can be implemented on quantum devices with little to no modification. To avoid this confusion, we adopt the following terminology. Quantum-inspired evolutionary algorithm (QiEA) refers to the *classical* algorithms that take inspiration from quantum mechanical phenomena. Whereas, quantum evolutionary algorithm (QEA) refers to the evolutionary algorithms that are designed for and implemented on physical quantum computers. A classification of purely classical, purely quantum, and interplay of classical and quantum algorithms is shown in Fig. 1. The main objective of this paper is to emphasize the intersection of quantum computing and evolutionary algorithms and how they can mutually benefit from one another. This paper serves as a valuable resource for researchers in both fields of quantum computing and evolutionary algorithms, enabling

[1] A checkmark ✓ denotes that the header of the corresponding column appears as one of the central themes in this reference. C-Q hybrid computing column points to specific implementations of classical-quantum hybrid algorithms where evolutionary algorithms take a central role in the classical or the quantum part.

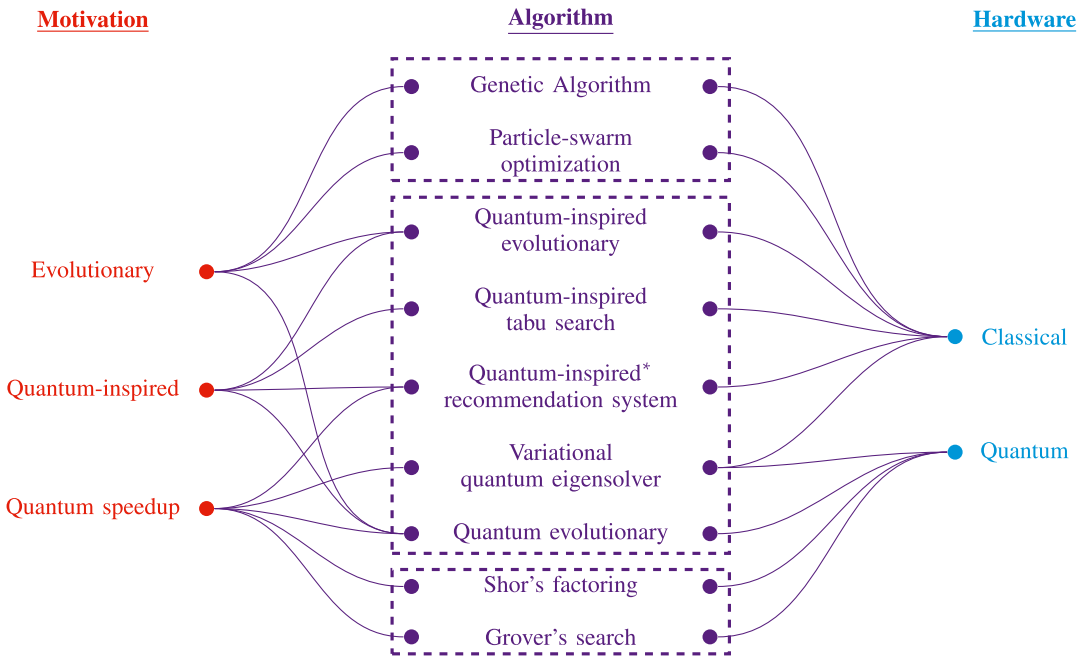


FIGURE 1. Evolutionary and quantum algorithms appear in different contexts based on the motivation and the hardware implementation. The algorithms within the top dashed box are purely classical algorithms with non-quantum motivations and developed for classical hardware. The algorithms in the bottom dashed box are purely quantum algorithms developed to obtain computational advantage on a quantum hardware without any nature-inspired motivations. The main discussion in this manuscript revolves around the algorithms in the middle box where a mix of quantum-related motivations and classical/quantum hardware are involved. *Quantum-inspired recommendation system [21] is inspired by an existing quantum algorithm for recommendation system, and not by quantum mechanics in general. Hence, it is quantum-inspired in a different sense and not a topic of present work.

them to grasp the most recent research advancements in quantum computing, and potentially inspiring further research endeavors. To this end, we discuss quantum-inspired metaheuristics that give rise to classical algorithms, quantum evolutionary algorithms that can be implemented on quantum computing devices, and the role of traditional metaheuristics in the era of NISQ computing. Throughout this paper, we present existing algorithms, e.g., quantum-inspired evolutionary algorithm (QiEA) of [28], quantum-inspired tabu-search (QiTS) of [29], and quantum genetic optimization algorithm (QGOA) of [30], as well as provide numerical examples that provide a benchmark for comparing these different classes of algorithms in practical scenarios. A comparison of this work with some of the existing literature is given in Table 1. Although we provide some preliminary information in the subsequent section, we assume a certain level of familiarity with quantum information sciences from the reader. For more information about this field from different perspectives, we refer the readers to some introductory papers [31], [32], [33] and/or standard textbooks [34], [35], [36], [37] for in-depth technical details.

The remainder of this paper is organized as follows. In Section II, we outline some preliminaries of metaheuristics and quantum computing. We discuss quantum-inspired metaheuristics in Section III, quantum evolutionary algorithms in Section IV and the role of evolutionary algorithms in NISQ computing in Section V. Finally, we briefly discus some recent experimental progress in Section VI and conclude our discussion in Section VII.

II. PRELIMINARIES

In this section we provide some preliminaries on evolutionary computation and quantum computing and set some notation. A list of acronyms in this manuscript is provided in Table 2.

A. EVOLUTIONARY COMPUTATION AND METAHEURISTICS

The origins of evolutionary computation can be traced back to 1950's, [38], [39], [40], [41]. The popularity of evolutionary computation algorithms stems from their generally problem-independent structure, low complexity, and tendency to find solutions that are sub-optimal but have a satisfactory performance [42], [43]. These properties make this field suitable for quickly finding good enough solutions of optimization problems of high complexity where structured algorithms cannot operate reasonably [44]. The evolutionary computation algorithms have applications in a number of fields including commercial applications in drug design, supply-chain management, portfolio optimization, as well as scientific applications in newly developing fields [45], [46]. Specific algorithms that fall under the framework include well-known algorithms like genetic algorithms and swarm intelligence algorithms [47], [48]. One recent highlight of successful implementation of evolutionary computation is the development of AlphaStar by DeepMind that beat a

TABLE 2. List of acronyms defined in the manusript.

Acronym	Definition
CTS	classical tabu search
DBGA	decomposition-based genetic algorithm
EA	evolutionary algorithm
GA	genetic algorithm
NISQ	noisy intermediate-scale quantum
PB	population-based
PQC	parameterized quantum circuit
QAOA	quantum approximate optimization algorithm
QCC	quantum circuit compilation
QEA	quantum evolutionary algorithm
QGOA	quantum genetic optimization algorithm
QiEA	quantum-inspired evolutionary algorithm
QiMA	quantum-inspired metaheuristic algorithm
QiTS	quantum-inspired tabu-search
SSB	single-solution-based
VQA	variational quantum algorithm
VQE	variational quantum eigensolver

professional player at the game of StarCraft II [49]. In recent times, the field of evolutionary computation is extended to evolutionary deep learning to offer promising results in the field of deep learning [26].

Metaheuristics, on the other hand, is an algorithmic framework that provides a set of guidelines to develop heuristic optimization algorithms [19]. Nature-inspired metaheuristics aim to mimic natural processes to develop a set of instructions that heuristically attempts to solve complex optimization problems.[2] The evolutionary computation algorithms and metaheuristics should not be considered as off-the-shelf methods but a general recipe to design an algorithm that can be tailored to the requirements of the problem at hand [41].

The QiEA falls under the broad category of nature-inspired evolutionary computation where the evolution of candidate solutions is inspired from the evolution of quantum state during a quantum computation, i.e., state preparation, evolution, and measurement [28]. Due to its inspiration, both the terminology and the notation of QiEA are similar to the standard notation of quantum computing, with some minor but key differences. In the next subsection, we provide the standard terminology and notation of quantum computing.

B. QUANTUM COMPUTING

1) QUANTUM STATES

In classical information sciences, the smallest unit of information is a bit that takes a binary value of either 0 or 1. In this article, we represent the state of a bit by $b \in \{0, 1\}$. A string of n bits is represented as $\vec{b} = b_{n-1} \cdots b_1 b_0$, where each bit $b_i \in \{0, 1\}$ takes a binary value. Then, the size of the state space of $\vec{b}$ is 2^n, i.e., the bitstring $\vec{b}$ is in any one of the 2^n possible states.

On the other hand, the smallest unit of quantum information is a quantum bit (qubit), which is characterized by a vector in the Hilbert space $\mathcal{H}$.[3] We utilize the Dirac's bra-ket notation to represent the state of a qubit. That is, the general state of a (pure, i.e., a noiseless) qubit is

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle , \tag{1}$$

where $|0\rangle , |1\rangle \in \mathcal{B}$ are the elements of some orthonormal basis $\mathcal{B}$ of the 2-dimensional Hilbert space $\mathcal{H}$; and $\alpha_0, \alpha_1 \in \mathbb{C}$ satisfy the normalization $|\alpha_0|^2 + |\alpha_1|^2 = 1$. Generally, the orthonormal basis $\{|0\rangle , |1\rangle\}$ is fixed to be the standard (also called computational) basis $\mathcal{B}_C$. Then, we can write the state of qubit $|\psi\rangle$ of (1) explicitly as

$$|\psi\rangle = \alpha \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} . \tag{2}$$

Another qubit basis that appear frequently is the Hadamard basis $\mathcal{B}_H = \{|+\rangle , |-\rangle\}$. The elements of the Hadamard basis are defined as

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \quad |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \tag{3}$$

It is simple to verify that Hadamard basis is also an orthonormal basis.

An n-qubit state is a vector in 2^n-dimensional Hilbert space. The general form of such quantum state can be written as

$$|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle , \tag{4}$$

where $\{|i\rangle\}$ is some orthonormal basis of the 2^n-dimensional Hilbert space, and α_i satisfy the normalization $\sum_{i=0}^{2^n} |\alpha_i|^2 = 1$. Similar to qubits, it is possible to use any orthonormal basis, but a typical choice is the n-qubit (or 2^n-dimensional) computational basis, whose elements are obtained by the tensor (Kronecker's) product of elements of computational bases of individual qubits. As an example, the general state of a (pure) two-qubit system can be written as

$$\begin{aligned} |\psi\rangle_{1,2} &= \alpha_{00} |0\rangle_1 \otimes |0\rangle_2 + \alpha_{01} |0\rangle_1 \otimes |1\rangle_2 \\ &\quad + \alpha_{10} |1\rangle_1 \otimes |0\rangle_2 + \alpha_{11} |1\rangle_1 \otimes |1\rangle_2 \qquad (5) \\ &= \alpha_{00} |00\rangle_{1,2} + \alpha_{01} |01\rangle_{1,2} + \alpha_{10} |10\rangle_{1,2} + \alpha_{11} |11\rangle_{1,2} , \qquad (6) \end{aligned}$$

where the subscript is used for indexing the two qubits and $\otimes$ represent the tensor product.

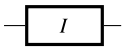
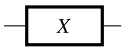
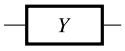
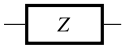
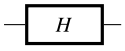
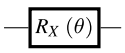
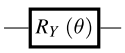
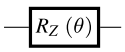
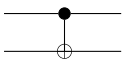
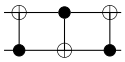
A two-qubit state that can be factored into two constituent qubits is called separable. If the state is not separable, it is called an entangled state. For example, we can write

$$\begin{aligned} |\psi\rangle_{1,2} &= \alpha |00\rangle_{1,2} + \beta |01\rangle_{1,2} \qquad (7) \\ &= \alpha |0\rangle_1 \otimes |0\rangle_2 + \beta |0\rangle_1 \otimes |1\rangle_2 \qquad (8) \\ &= |0\rangle_1 \otimes (\alpha |0\rangle_2 + \beta |1\rangle_2) \qquad (9) \\ &= |\phi\rangle_1 \otimes |\eta\rangle_2 . \qquad (10) \end{aligned}$$

[2] In addition to noting the power and wide applicability of resulting heuristic algorithms, we must also note that the scientific rigor of some of directions in this field is criticized [50], [51].

[3] In finite-dimensional complex vector spaces, the Hilbert space is exactly the same as the complex inner product space and all calculations can be carried out identically, e.g., in [36]. However, we follow the convention of defining quantum states and operations with respect to Hilbert space (see [37, Paragraph after (2.15)]).

TABLE 3. Some important quantum operators.

Classification	Gate	Short Description	Matrix Representation	Circuit
Fixed Single Qubit	Identity	No change in the state	$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	
	Pauli X	Bit-flip in computational basis	$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	
	Pauli Y	Bit- and phase-flip in computational basis	$Y = \begin{bmatrix} 0 & -\iota \\ \iota & 0 \end{bmatrix}$	
	Pauli Z	Phase-flip in computational basis	$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	
	Hadamard	Basis change between the computational and Hadamard basis	$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$	
Parameterized Single Qubit	Pauli X Rotation	Rotation around X axis	$R_X(\theta) = \begin{bmatrix} \cos(\theta/2) & -\iota\sin(\theta/2) \\ -\iota\sin(\theta/2) & \cos(\theta/2) \end{bmatrix}$	
	Pauli Y Rotation	Rotation around Y axis	$R_Y(\theta) = \begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{bmatrix}$	
	Pauli Z Rotation	Rotation around Y axis	$R_Z(\theta) = \begin{bmatrix} \exp(-\iota\theta/2) & 0 \\ 0 & \exp(\iota\theta/2) \end{bmatrix}$	
Fixed Two Qubit	CNOT	Controlled NOT operation from control to target qubit	(15)	
	SWAP	Swap the states of two qubits	CX (2, 1) CX (1, 2) CX (2, 1)	

Hence, $|\psi\rangle_{1,2}$ is a separable state for any α, β, where we have defined $|\phi\rangle_1 = |0\rangle_1$ and $|\eta\rangle_2 = \alpha\,|0\rangle_2 + \beta\,|1\rangle_2$. On the other hand

$$|\Phi\rangle_{1,2} = \alpha\,|00\rangle_{1,2} + \beta\,|11\rangle_{1,2} \tag{11}$$

$$\neq |\phi\rangle_1 \otimes |\eta\rangle_2 \tag{12}$$

for any $|\phi\rangle_1$ and $|\eta\rangle_2$ except for the trivial case $|\alpha| = 0$ or $|\alpha| = 1$. Therefore, $|\Phi\rangle_{1,2}$ is an entangled state for $|\alpha| \neq 0, 1$.

Alternatively, given the descriptions of two individual qubits $|\psi\rangle_1$ and $|\psi\rangle_2$, their joint state $|\psi\rangle_{1,2}$ can be written as the tensor product of the individual descriptions, i.e., $|\psi\rangle_{1,2} = |\psi\rangle_1 \otimes |\psi\rangle_2$. It is trivial that such a state is separable for any $|\psi\rangle_1$ and $|\psi\rangle_2$. However, it is possible to transform a separable state into an entangled one by joint quantum operations on both of the qubits.

2) QUANTUM OPERATIONS

The noiseless (ideal) evolution of a quantum system is characterized by unitary operators, i.e., the operators U that satisfy $UU^\dagger = U^\dagger U = I$, where I is the identity operator on the Hilbert space. Based on the context, different classifications of quantum operators can be defined. For example, depending on the number of qubits on which an operator is applied we can have single qubit, two-qubit or multi-qubit quantum operators. An important classification in the context of tunable quantum circuits and NISQ-era quantum computing is the fixed and parameterized quantum operators. Fixed quantum operators do not depend on any circuit parameter and provide a fixed transformation of a quantum state. Parameterized quantum operators, on the other hand, are parameterized with some variable(s) that can be varied according to application requirements. These parameterized quantum operators serve as the fundamental elements of tunable quantum circuits commonly used in NISQ-era quantum algorithms.

From single qubit operators, it is possible to compose simple multiqubit operators with tensor product. For example, an operator that represents Pauli X gate on the first qubit and Hadamard gate on the second qubit can be written as

$$A = X_1 \otimes H_2. \tag{13}$$

Despite being a two-qubit operator, A is incapable of generating entanglement between the operand qubits since both X and H operate independently on each qubit. A more interesting two-qubit operator that represents interaction between the operand qubits is the controlled-NOT (CNOT or CX (i, j) for i as the control and j as the target qubit) operator. The CNOT operator applies Pauli X on the target qubit if the controlled qubit is in the state $|1\rangle$. Otherwise, it leaves the target qubit unchanged. The effect of CNOT on the two-qubit

computational basis can be summarized as follows:

$$\mathrm{CX}(1,2): \begin{array}{l} |00\rangle_{1,2} \mapsto |00\rangle_{1,2} \\ |01\rangle_{1,2} \mapsto |01\rangle_{1,2} \\ |10\rangle_{1,2} \mapsto |11\rangle_{1,2} \\ |11\rangle_{1,2} \mapsto |10\rangle_{1,2} \end{array}, \tag{14}$$

where the first qubit is the control qubit and the second qubit is the target qubit. The CNOT gate has the following matrix representation

$$\mathrm{CX}(1,2) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{15}$$

A more general way of writing the expression for an arbitrary unitary U applied in controlled fashion is

$$\mathrm{CU}(1,2) = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U. \tag{16}$$

The CNOT gate is a fundamental building block in quantum circuits and is essential for various quantum algorithms and operations. Specifically, the CNOT operator not only itself is an important element in quantum information processing but also serves as a fundamental operator to synthesize several important quantum operators, e.g., SWAP operator, which can be written as three consecutive CNOT gates as shown in Table 3. Please note that since the unitary U in the above expression is arbitrary, it can even be a parameterized U, e.g., $R_X(\theta)$, $R_Y(\theta)$, and $R_Z(\theta)$. This results in parameterized two-qubit operations that represent interaction between two qubits that is tunable by changing the parameter values. Similar to single-qubit parameterized gates, multi-qubit parameterized gates also serve as important primitive in designing tunable quantum circuits that is an essential element of NISQ-era algorithms.

3) MEASUREMENT

Quantum gates take a quantum state as the input and output another quantum state. In order to extract/infer any information from this state, it is necessary to measure it. Quantum measurement is a nonlinear process that takes a quantum state as the input and gives some classical information (bits) as output. In its simplest form, quantum measurement is characterized with the projection operators based on some observable (a Hermitian operator). For example, recall that elements of $\mathcal{B}_C$ are the eigenstates of Pauli Z operator. Then, measuring Pauli Z component of a qubit is equivalent to asking the question whether the unknown given state $|\psi\rangle$ is in state $|0\rangle$ or $|1\rangle$. This amounts to using the projectors

$$\Pi_0 = |0\rangle\langle 0|, \quad \Pi_1 = |1\rangle\langle 1|. \tag{17}$$

The answer to this question is the binary digit 0 or 1 according to Born's rule, i.e., with the corresponding probabilities

$$p_0 = \mathrm{tr}(\Pi_0 |\psi\rangle\langle\psi|), \text{ and } p_1 = \mathrm{tr}(\Pi_1 |\psi\rangle\langle\psi|). \tag{18}$$

TABLE 4. A comparison of Qubit, Q-bit, and Q-bit individual.

Term	Description
Qubit	A short form of quantum bit. State of a two-level quantum system. Smallest unit of quantum information.
Q-bit	A term used in quantum-inspired metaheuristics literature. Similar to a qubit, a Q-bit is represented by a two-element complex vector. In contrast to a qubit, a Q-bit is an entirely classical variable designed to mimic the probabilistic nature of a qubit in classical heuristic algorithms.
Q-bit individual	An array of Q-bits. Each Q-bit individual represents a possible solution to the optimization problem, where the number of Q-bits in the Q-bit individual is equal to the number of variables to be optimized.

Furthermore, the quantum state collapses during the measurement process to the corresponding eigenstate of the observable being measured. Operationally, this means that we cannot extract more than one bit of information since any subsequent measurement of the same operator will return the same classical bit with unit probability as observed for the first time.

III. QUANTUM-INSPIRED METAHEURISTIC ALGORITHMS

Metaheuristic algorithms are low-complexity, though sub-optimal, iterative optimization methods that are widely adopted in practice for solving highly complex and nonlinear optimization problems [52], [53], [54]. One way to classify Metaheuristic algorithms is by categorizing them into two main categories: population-based (PB) and single-solution-based (SSB) algorithms [55]. The PB algorithms perform their operations on a population of candidate solutions, and the final solution is the one that achieves the highest cost-function gain in the population. If the population of a Metaheuristic algorithm applies principles of natural biological evolution to produce new candidate solutions, then the corresponding algorithm is also called an evolutionary algorithm (EA) [53]. In contrast, the SSB algorithms perform their operations to improve the cost-function gain of a single candidate solution.

For quantum-inspired metaheuristic algorithms (QiMA), a generic candidate (or potential) solution is often referred to in the literature as a Q-bit individual, which is a string of M Q-bits with M being the number of variables to be optimized by the algorithm [28]. Inspired by the concept of the *qubit* in Quantum computing, a single Q-bit in the Q-bit individual represents the smallest unit of information in a QiMA, and it is defined by a pair of real numbers (α, β), where $|\alpha|^2$ and $|\beta|^2$ are the probabilities that the corresponding Q-bit will be found in the "0" or "1" states, respectively. Thus, the parameters of a single Q-bit should satisfy $|\alpha|^2 + |\beta|^2 = 1$. A generic Q-bit individual can be represented as:

$$q = \left[\begin{array}{c|c|c|c} \alpha_1 & \alpha_2 & \cdots & \alpha_M \\ \beta_1 & \beta_2 & & \beta_M \end{array}\right], \tag{19}$$

The representation of a Q-bit individual in (19) can be interpreted as a superposition of states. For example, consider

a Q-bit individual $\bar{q}$ with 2 Q-bits in the following form

$$\bar{q} = \left[\begin{array}{c|c} \frac{1}{\sqrt{2}} & \frac{1}{2} \\ \frac{-1}{\sqrt{2}} & \frac{\sqrt{3}}{2} \end{array}\right], \tag{20}$$

the superposition of corresponding system states, denoted by $\left|\Phi_{\bar{q}}\right\rangle$, can be expressed as follows:

$$\left|\Phi_{\bar{q}}\right\rangle = \frac{1}{2\sqrt{2}}\left|00\right\rangle + \frac{\sqrt{3}}{2\sqrt{2}}\left|01\right\rangle - \frac{1}{2\sqrt{2}}\left|10\right\rangle - \frac{\sqrt{3}}{2\sqrt{2}}\left|11\right\rangle, \tag{21}$$

where the coefficients of component $|ij\rangle$ are obtained by multiplying the coefficients of i for the first Q-bit and j for the second Q-bit as given in (20). The above superposition reflects the probabilities of different system states, with states $|00\rangle$ and $|10\rangle$ having equal probabilities of $\frac{1}{8}$, while states $|01\rangle$ and $|11\rangle$ are each associated with a probability of $\frac{3}{8}$. Such a powerful representation of Q-bit individuals with a probabilistic sense means higher diversity with better characteristics in QiMAs compared to the conventional Metaheuristic schemes that adopt a binary representation of candidate solutions. In this context, it is important to highlight that QiMAs are in fact not quantum algorithms, but rather a special class of classical Metaheuristic optimization that takes inspiration from the principles and concepts of quantum computing.[4] Table 4 describes the key differences between quantum bits (qubits), Q-bits of classical algorithms, and Q-bit individuals.

In the following, we will explain the operational concept of PB and SSB QiMA, and highlight the main algorithms in each category.

A. POPULATION-BASED QIMAS

Since the start of the new millennium, a large number of novel PB QiMAs have emerged. Among the earliest and also most well-known methods in the field are the quantum-inspired evolutionary algorithm (QiEA) [28], and the *Quantum-inspired Genetic Algorithm* [58]. In particular, the QiEA was originally proposed to tackle the class of combinatorial optimization problems [28].

In the following, we will briefly explain the operational concept of the QiEA in [28] to help the reader to have a better understanding on how the PB QiMA operate to tackle different optimization problems.

1) THE QUANTUM-INSPIRED EVOLUTIONARY ALGORITHM

The algorithm was originally designed to tackle combinatorial optimization problems, and the term "evolutionary" was used to reflect the fact that only the fittest solution(s) in the population will survive. A flowchart of essential elements of QiEA is shown in Fig. 2.

[4]We refer the interested reader on QiMAs to the surveys in [22], [56], and [57] and the references therein.

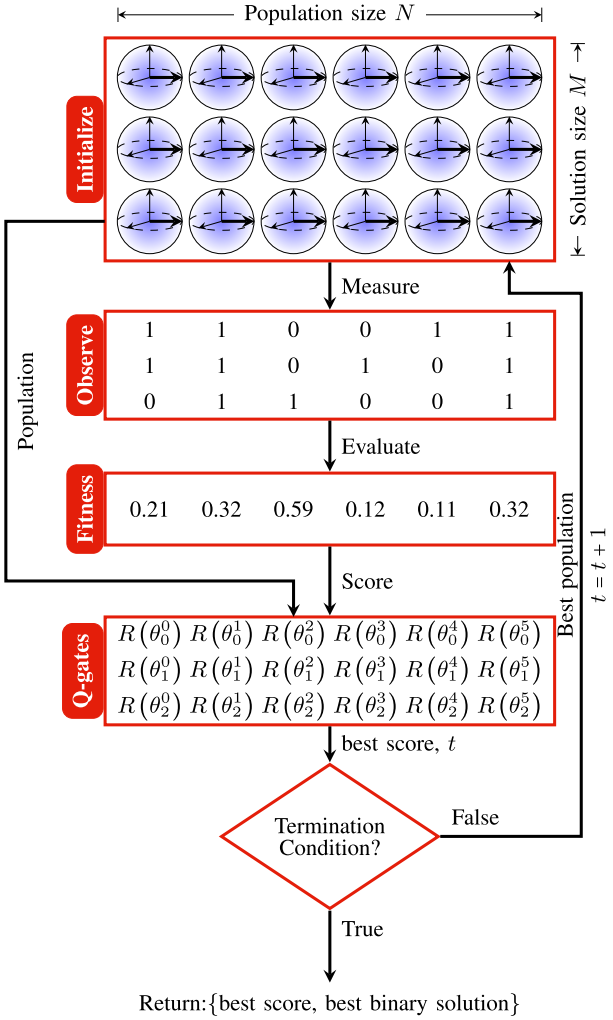


FIGURE 2. A workflow of QiEAs. The q-bit individuals are measured to obtain bit values that represent candidate solutions. Based on the fitness values, these candidate solutions are evolved. In some of the iterations, local or global migrations might be triggered (not depicted in the flowchart). The best individual and corresponding score are returned if the termination criterion is satisfied.

Like all other metaheuristic algorithms, the QiEA is an iterative optimization method, where the population of candidate solutions evolves by applying certain operations at each iteration. After the final iteration, the solution with the highest fitness (or quality) accross all different iterations will be declared as the final solution. Below, we summarize the steps of the QiEA.

a: INITIALIZE THE QUANTUM-INSPIRED POPULATION

The QiEA starts by initializing a population of N Q-bit individuals denoted by $Q^{(t)}|_{t=0}$. Each Q-bit individual in $Q^{(t)}$ has a string of M Q-bits. The population and the nth Q-bit individual ($n \in \{1, 2, \cdots, N\}$) can be expressed, respectively, as follows:

$$Q^{(t)} = \left\{q_1^{(t)}, q_2^{(t)}, \cdots, q_N^{(t)}\right\}, \tag{22}$$

$$q_n^{(t)} = \left[\begin{array}{c|c|c|c} \alpha_{n1}^{(t)} & \alpha_{n2}^{(t)} & \cdots & \alpha_{nM}^{(t)} \\ \beta_{n1}^{(t)} & \beta_{n2}^{(t)} & & \beta_{nM}^{(t)} \end{array}\right]. \tag{23}$$

At $t = 0$, all values of $\alpha_{nm}^{(0)}$ and $\beta_{nm}^{(0)}$ ($n \in \{1, 2, \cdots N\}$, $m \in \{1, 2, \cdots, M\}$) are set to $1/\sqrt{2}$, which means that each Q-bit individual is a superposition of all possible states with equal probabilities.

b: INCREASE ITERATION INDEX

Assign $t \leftarrow t + 1$

c: CREATE A BINARY-BASED POPULATION

In this step, a population of binary solutions $P^{(t)}$ is created by observing the state of the Quantum-inspired population $Q^{(t-1)}$. More specifically, $P^{(t)}$ can be expressed as

$$P^{(t)} = \left\{x_1^{(t)}, x_2^{(t)}, \cdots, x_N^{(t)}\right\}, \tag{24}$$

where $x_n^{(t)}$ ($n \in \{1, \cdots, N\}$) is an $M-$dimensional binary vector created by observing $q_n^{(t-1)}$. In particular, each element in $x_n^{(t)}$ is set to either 0 or 1 based on the probability of the corresponding Q-bit in $q_n^{(t-1)}$ being found in either the 0 or the 1 state, respectively.

d: FITNESS EVALUATION

In this step, the fitness (or the quality) of each binary solution in $P^{(t)}$ is evaluated.

e: APPLY THE Q-GATES

In this step, the Quantum-inspired population matrix from the previous iteration $Q^{(t-1)}$ will be updated using Q-gates to produce a new population $Q^{(t)}$. Examples of Q-gates include rotation gates, Hadamard gates and NOT gates.

f: UPDATE THE POPULATION OF BEST SOLUTIONS

Let $B^{(t)} = \{b_1^{(t)}, b_2^{(t)}, \cdots, b_N^{(t)}\}$ be the population of best solutions at the tth iteration, and b denote the best solution across all previous iterations. For iteration $t = 1$, set the initial solutions as the best solutions, i.e. $B^{(1)} = P^{(1)}$, and store the best solution among them as b. For all iterations of $t \geq 2$, store the solution with highest quality between $B^{(t-1)}$ and $P^{(t)}$ into $B^{(t)}$, and if the best solution in $B^{(t)}$ is fitter than b, replace b by the best solution in $B^{(t)}$.

g: LOCAL AND GLOBAL MIGRATION

If a migration-condition is met, then perform local or global migration on $B^{(t)}$. While the migration-condition is a design parameter, the global migration means replace all solutions in $B^{(t)}$ by b, and the local migration means replace some of the solutions in $B^{(t)}$ by the best one among them.

h: ITERATE

repeat steps *(b)* to *(g)* until reaching a maximum number of iterations.

2) OTHER QIMAS

In addition to the PB QiMAs mentioned above, there are many other approaches that apply quantum-inspired properties to various PB metahueristic methods to obtain improved performance and/or faster convergence. Some examples include the quantum-inspired particle swarm optimization algorithm [59], quantum-inspirIed social evolution algorithm [60], quantum inspired acromyrmex evolutionary algorithm [61], among many others.

B. SINGLE SOLUTION-BASED QIMA

SSB metaheuristic algorithms perform local optimization on a single candidate solution. Such methods start from an initial point in the multi-dimensional search space and then move toward a better solution by exploring the surrounding area (or neighbourhood) at each given iteration. Due to their local search process, the SSB metaheuristic algorithms are unlikely to reach the global optimal solution, and their performance can also be highly affected by the initial point they start from. However, the main advantage of these methods is the low computational complexity which can be cruicial for real-time applications.

The quantum-inspired SSB algorithms have received much less attention than the PB methods. Nonetheless, there have been few works that incorporated the properties of quantum computing into classical local search methods, including the QiTS algorithm [29] and the *quantum-inspired simulated annealing* algorithm [62]. Similar to QiEA, QiTS was originally proposed for solving the combinatorial optimization problems.

In the following, we will focus on the QiTS by briefly explaining the idea behind it and its concept, and we will also highlight how the QiTS is different from the classical tabu search (CTS) approach.

1) THE QITS ALGORITHM

As an SSB algorithm, the QiTS adopts a single Q-bit individual for its operation, and it applies the rotation gate $\mathcal{R}(\theta)$ to update its state from one iteration to another. The value of the rotation angle θ is a design parameter that depends on the considered problem.

The QiTS is different from the CTS in one key perspective. The CTS algorithm utilizes memory structure to avoid going back into local areas (or solutions) that have been recently visited and ensure an efficient exploration of the search space. The CTS stores the recently explored solutions in a special list known as ''tabu list'', and hence the name *tabu search*. In contrast, the QiTS does not rely on the memory, and it has a different view on what should be labeled as ''tabued''. The main steps of the QiTS algorithm can be summarized as follows:

a: Q-BIT INDIVIDUAL GENERATION

At iteration $t = 0$, the algorithm generates a single Q-bit individual ($q^{(0)}$) with M Q-bits. Each Q-bit is initialized with equal probabilities of α_m and β_m ($m \in \{1, 2, \cdots, M\}$).

b: INCREASE ITERATION INDEX

Assign $t \leftarrow t + 1$

Algorithm 1 Quantum-Inspired Tabu Search Algorithm [29]

procedure QiTS
Initialization:
  $t \leftarrow 0$ ▷ *Initialize iteration counter.*
  $M \leftarrow$ Number of Q-bits
  $N \leftarrow$ Number of measurements per iteration
  $\theta \leftarrow$ Rotation angle parameter
Generate initial Q-bit individual:
  $q^{(0)} \leftarrow$ new QbitIndividual(M)
  for $m = 1$ to M **do**
    $q^{(0)}[m].\alpha \leftarrow 1/\sqrt{2}$
    $q^{(0)}[m].\beta \leftarrow 1/\sqrt{2}$
  $b \leftarrow$ null
Loop through iterations:
  while $t <$ max_iterations **do**
    $t \leftarrow t + 1$
Perform measurements:
    solutions $\leftarrow \emptyset$
    for $i = 1$ to N **do**
      $s \leftarrow$ measure($q^{(t-1)}$)
      $f \leftarrow$ evaluate_fitness(s)
      solutions.append((s, f))
Find best and worst solutions:
    $s_b \leftarrow$ solutions.get_best()
    $s_w \leftarrow$ solutions.get_worst()
Update global best if necessary:
    if $b =$ null or fitness(s_b) $>$ fitness(b) **then**
      $b \leftarrow s_b$
Create tabu list:
    tabu $\leftarrow \emptyset$
    for $m = 1$ to M **do**
      if $s_b[m] = s_w[m]$ **then**
        tabu.append(m)
Update non-tabued Q-bits:
    $q^{(t)} \leftarrow q^{(t-1)}$
    for $m = 1$ to M **do**
      if $m \notin$ tabu **then**
        sign $\leftarrow$ determine_rotation_sign($q^{(t)}[m]$, $s_b[m]$)
        $q^{(t)}[m] \leftarrow$ apply_rotation($q^{(t)}[m]$, sign $\cdot\ \theta$)
  return b

c: MEASUREMENT AND FITNESS EVALUATION

The next step is performing N measurements on $q^{(t-1)}$ to generate N binary solutions and evaluate their fitness values.

d: BEST AND WORST SOLUTION

In this step, the best and worst solutions from the latest measurement will be stored and denoted by s^b and s^w, respectively, both of which are M-dimensional binary vectors. In addition, denote b as the binary vector with the highest fitness in all previous iterations, if the fitness value of s^b in the current iteration is higher than that of b, then replace b by s^b.

e: CREATE THE TABU LIST

In this step, a tabu list will be created. In particular, for all $m = \{1, 2, \cdots, M\}$, if the value of the mth bit in s^b and s^w is the same (i.e. either both ones or both zeros), then the mth Q-bit in $q^{(t-1)}$ will be marked as "tabu" and remains unchanged for the current iteration.

f: UPDATE Q

For all Q-bits which are not "tabued", apply the rotation gate on each Q-bit with either *a positive or a negative value of the rotation angle* θ. Specifically, the sign of θ is determined for each of the non-tabued Q-bits to maximize the probability of that Q-bit to be found in a state that is similar to the corresponding bit in s^b upon measurement in subsequent iterations.

g: ITERATE

repeat steps *(b)* to *(f)* until reaching a maximum number of iterations.

As explained in step (e) above, the QiTS differs from the CTS in the sense that its tabu list does not rely on memory, but rather on observation of measurements of the current iteration. As a result, the dimension of the tabu list in the QiTS algorithm varies from one iteration to another, which is not the case for the CTS approach. The pseudocode for QiTS is given as Algorithm 1.

IV. QUANTUM EVOLUTIONARY ALGORITHMS

A quantum evolutionary algorithm (QEA) is a specialized optimization technique that employs principles from quantum mechanics to tackle complex problems. When implemented on quantum computers, QEAs leverage the quantum properties of these machines to explore and exploit solution spaces more efficiently. On the other hand, QiEA are classical algorithms that incorporate quantum-inspired techniques to enhance their performance, even when executed on classical computers [63]. While QEAs offer the potential for quantum speedups on quantum hardware, QiEAs utilize quantum-inspired strategies on classical platforms to improve exploration and exploitation capabilities. The choice between QEA and QiEA depends on factors such as the availability of quantum hardware and the specific requirements of the problem at hand. This allows researchers to select the most suitable approach for their optimization tasks. In this section, we will discuss the QGOA of [30], one of the earliest known quantum evolutionary algorithms proposed in 2008. Subsequently, we will apply this algorithm to a quantum computer to address the Max-Cut problem.

A. QUANTUM GENETIC OPTIMIZATION ALGORITHM

The basic structure of the QGOA [30] is based on the classical structure of steady-state genetic algorithm. In the classical genetic algorithm, given a representation of the population and the fitness function, the first step is to evaluate the fitness of every element of the population. After that, a subpopulation containing a fraction p of the best elements of the population can be selected, then two element can be chosen randomly from this subpopulation. A crossover and mutation can be performed by exchanging two random substrings and mutate each allele, which is a bit of the strings, with probability P_M. Finally, two random elements from

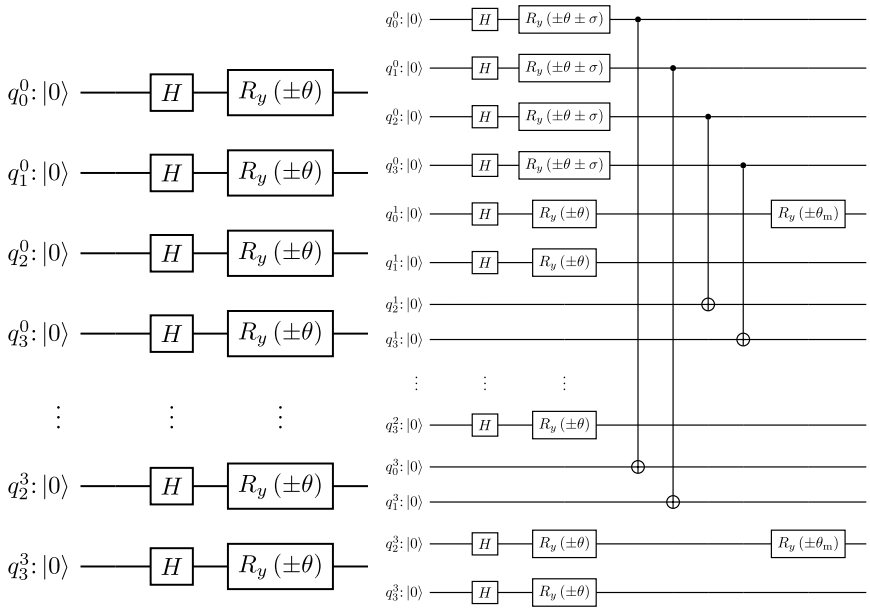


FIGURE 3. Schematic representation of the QGOA implementation on a superconducting quantum computer. (a) Quantum initialization circuit generating four quantum chromosomes (q_0, q_1, q_2, q_3) for the Max-Cut problem, each representing a classical genetic population with 16 binary strings. (b) Entangled crossover circuit for the initial evolutionary step, incorporating R_y mutation for the Max-Cut problem. Measurement gates are excluded for simplicity.

the population can be chosen to be replaced with the new offspring. These steps can be repeated for M times.

In QGOA, given a qubit representation of the population and a quantum evaluation unit, the first step is to use the quantum selection procedure to choose one element, and then the procedure can be run again to choose another element. The quantum selection procedure performs the creation of a superposition of all elements of the population and the application of the quantum fitness evaluation unit. The similarity of this algorithm with the classical one is the steps that involve crossover, mutation, and substitution. As detailed in [30], the QGOA can have a significant advantage over its classical counterpart when the fitness function is varying. In this scenario, QGOA will have a polynomial speedup in terms of oracle calls in the selection procedure.

QUANTUM FITNESS EVALUATION UNIT

The quantum fitness evaluation unit is an operator U_F which can be constructed from a classical reversible circuit that compute a fitness function $F(j) = F_j$, where $j \in \{0, \ldots, N-1\}$ are the elements of the population in binary representation. The classical circuit then can be converted into a quantum circuit that represents the operator U_F. The physical realization of U_F depends on the problem, so there is no general procedure to construct it.

The superposition of all elements of the population can be represented by using the quantum binary encoding as follows:

$$|\psi\rangle = \frac{1}{\sqrt{N}\sum_{j=0}^{N-1}} |j\rangle . \tag{25}$$

While the classical procedure requires N fitness evaluations, the operator U_F can be use to compute all the fitness values $\{F_j | j = 0, \ldots, N-1\}$ of the population at once as follows:

$$U_F |\psi\rangle |0\rangle = \frac{1}{\sqrt{n}} \sum_{J=0}^{n-1} |j\rangle \left|F_j\right\rangle . \tag{26}$$

However, the superposition would be destroyed by the measurement, leaving only one fitness value, possibly resulting the destruction of the useful information on the best elements of the population. An oracle circuit of the quantum selection procedure marks the elements of the population that fulfill the condition $F_j \geq F_y$, where y is a threshold index.

The oracle circuit has the properties as follows: It stays the same at every genetic step, its input is a superposition of all the N elements of the population at every genetic step, and therefore its capacity is N. However, external factors, such as environmental noise or statistical fluctuations can cause the variation of the fitness function between steps.

QUANTUM SELECTION PROCEDURE

The quantum selection procedure is based on the Dür-Høyer algorithm [64] to find the minimum of a list of N elements. In quantum selection procedure part of QGOA, the Dür-Høyer algorithm can be used to find the elements with a relatively high value of fitness. This procedure works as follows:

1) An index $y \in \{0, 1, \ldots, N-1\}$, that corresponds to the threshold F_y is randomly chosen. The threshold value can be computed classically, $F_y = F(y)$.

TABLE 5. QGOA evolutions for the max-cut problem.

Iteration	Quantum chromosome	Clasical chromosome	Fitness value	Average fitness value
0	$\mathbf{q_0}$	0011	-2	-2
	q_1	1001	-2	
	q_2	1101	-2	
1	$\mathbf{q_0}$	1000	-2	-1.333
	q_1	0000	0	
	q_2	1001	-2	
2	q_0	1011	-2	-1.333
	q_1	1001	-2	
	$\mathbf{q_2}$	1111	0	
3	$\mathbf{q_0}$	1010	-4	-4
	q_1	1010	-4	
	q_2	1010	-4	
4	q_0	1010	-4	-3.333
	$\mathbf{q_1}$	1010	-4	
	q_2	0010	-2	
5	q_0	1010	-4	-2.667
	$\mathbf{q_1}$	1110	-2	
	q_2	1001	-2	

2) These steps are to be performed n_h times:
   a) Memory initialization to $|0\rangle\,|y\rangle$.
   b) Transform $|0\rangle\,|y\rangle$ into $\frac{1}{\sqrt{N}}\sum_j |j\rangle\,|y\rangle$. The amplitude of the elements that satisfy $F_j \geq F_y$ are inverted by the oracle.
   c) The first ket is measured to obtain a new index y'. The new $F_{y'}$ can be computed classically, $F_{y'} = F(y')$. The index y is set to y' if $F_{y'} > F_y$.
3) Return the index y.

After n_h iterations, the probability for choosing the best element of the population is R/N where R is a fraction of the entire population. Meanwhile, a probabilistic algorithm would obtain $\log(R/N)$ probability for choosing the best element of the population [30].

B. MAX-CUT PROBLEM

The Max-Cut problem is a well-known combinatorial optimization problem that entails partitioning the nodes of a graph into two distinct sets to maximize the number of edges that cross between these sets. The problem is well-known as being NP-hard, and it has been found to have numerous applications across multiple domains. This task aims to determine a partition with the highest number of edges intersecting between the two sets. Numerous algorithms and heuristics have been implemented to address this issue, encompassing exact separation algorithms [65], greedy heuristics [66], and branch-and-bound algorithms [67]. Specific methodologies employ relaxation techniques, such as semidefinite relaxation, to approximate optimal solutions [67]. The Max-Cut problem has undergone substantial investigation, with researchers persistently exploring novel algorithms and strategies to improve the quality and efficiency of solutions.

In this section, we will apply the QGOA to tackle the Max-Cut problem. Consider an undirected graph comprising e edges and v vertices. Our objective is to determine a partition s of these vertices into two sets, denoted as X and Y, in a manner that minimize the expression:

$$C(s) = -\sum_{i=1}^{e} C_i(s). \tag{27}$$

Here, C represents the count of cut edges to find the configuration that maximizes this count. For each of the e edges, $C_i(s) = 1$ if the vertices from the i^{th} edge are distributed between sets X and Y in configuration s, and $C_i(s) = 0$ otherwise. We represent the partition of vertices into sets A or B using a bitstring, where $s_i = 0$ if the i^{th} vertex is in set A, and $s_i = 1$ if it is in set B. Equation (27) will serve as our fitness function for QGOA.

C. IMPLEMENTATION QGOA ON QUANTUM COMPUTER

When implementing the QGOA on a NISQ device, it is of utmost importance to consider the hardware's architecture and limitations. Quantum computing encompasses numerous hardware platforms, such as trapped ion qubits, superconducting qubits, silicon qubits, photonic qubits, and topological qubits. Of these, superconducting qubits have become a popular and readily available option for researchers and the general public [68], [69]. They utilize the quantum circuit representation to encode and execute quantum algorithms, with qubits, quantum gates, and measurements as the fundamental components [70].

The first step in developing a quantum circuit for the QGOA is qubit encoding, which establishes the groundwork for quantum chromosome development. Afterward, individuals are measured to establish the initial population. Quantum measurement transforms the cluster of quantum chromosomes into a set of classical binary strings. The fitness function of these binary strings is then evaluated using conventional computing, allowing for determining the most optimal solution.

The circuit depth is the primary limitation of implementing the QGOA in the NISQ device. If every generation applies

$R_y(\theta)$, the minimum depth for each individual would be $g+1$, where g represents the number of generations. The mutation gate is applied randomly and is rare compared to the entire circuit, meaning that the average depth of the circuit will not change because of it. One method of reducing the depth is storing the cumulative rotation angle in a matrix and applying the $R_y(\theta)$ with this cumulative angle [71].

In the subsequent sections, we will elaborate on the quantum implementation of an evolutionary algorithm based on [63] and [71]. The pseudocode for QGOA is provided as Algorithm 2.

1) THE QUANTUM CHROMOSOME

The quantum chromosome plays a crucial role in the algorithm by efficiently embedding a large classical genetic population onto a single quantum register. Traditional evolutionary algorithms use a population of candidate solutions, represented by chromosomes comprising bit strings encoding each solution. With the Hadamard gate applied to each qubit in the quantum register, a quantum chromosome can simultaneously represent all possible bit configurations.

To begin, the quantum genetic population is typically prepared in a superposition state where every individual qubit is set to $|0\rangle$ and a Hadamard gate is applied to each qubit at time $t = 0$. Qubits can be represented by their geometrical form in quantum circuits as

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle, \tag{28}$$

where θ, ϕ are real numbers, $0 \leq \theta \leq \pi$ and $0 \leq \phi \leq 2\pi$. The state as above is represented by a sphere called Bloch sphere. Using this representation, quantum individuals can be encoded as quantum registers with M qubits. To update the population, a rotation gate $R_y(\theta)$ is applied to change the probability $\left|\alpha^{(t)}\right|^2$ and $\left|\beta^{(t)}\right|^2$ as the following

$$\begin{bmatrix}\alpha' \\ \beta'\end{bmatrix} = \begin{bmatrix}\cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta)\end{bmatrix}\begin{bmatrix}\alpha \\ \beta\end{bmatrix}, \tag{29}$$

where θ is designed according to the problem. Next, the entangled crossover gates are applied to the quantum circuit to modify the probability amplitudes of the qubits and generate new quantum individuals. For our Max-Cut problem with e edges, we must generate chromosomes containing e qubits each.

2) THE ENTANGLED CROSSOVER

Quantum evolutionary algorithms are designed to evolve quantum chromosomes on quantum computers. The entangled crossover method is a specialized technique employed within this algorithm, which combines distinct elements from different solutions without duplicating individual components. This approach establishes a robust and distinctive interdependence among qubits by entangling specific qubits, which has been observed to positively impact solution generation. By facilitating combining key elements from the ideal solution, the entangled crossover method leads to an improvement in the overall quality of solutions. In addition, this technique introduces a stochastic linkage between the qubits of the optimal solution and those of alternative solutions, which effectively mitigates the risk of the algorithm becoming trapped in local minima. The unique crossover technique described in this study is a significant advancement in quantum evolutionary algorithms and has the potential to improve the efficiency of quantum computing.

Consider a scenario in which a quantum genetic population is composed of k n-qubit quantum chromosomes, denoted as $Q_p = \{c_1, \ldots, c_k\}$. Each chromosome is represented as $\left|q_0^i \ldots q_{n-1}^i\right\rangle$, where $0 \leqslant i \leqslant n-1$. In this scenario, we assume that $c^* = c_l$ is the current optimal solution, where $1 \leqslant l \leqslant k$.

The entangled crossover operation is a crucial aspect of this scenario. It partitions the optimal chromosome c^* into $(k-1)$ quantum states, with each state consisting of $\lfloor \frac{n}{k-1} \rfloor$ qubits. Specifically, the entangled crossover operation is defined by the following $(k-1)$-tuple:

$$G = (c_1^*, c_2^*, \ldots, c_{k-1}^*) \tag{30}$$

where

$$c_1^* = \left|q_0^l \ldots q_{\lfloor \frac{n}{k-1} \rfloor - 1}^l\right\rangle$$

Algorithm 2 Quantum Genetic Optimization Algorithm [30]

procedure QGOA
Input:
  $b \leftarrow$ problem dimension (4)
  $k \leftarrow$ population size parameter (3)
  $max\_gen \leftarrow$ maximum generations (5)
  $\delta \leftarrow$ rotation angle ($\pi/8$)
  $\mu \leftarrow$ mutation probability (0.3)
  $\rho \leftarrow$ reinforcement angle ($\pi/16$)
  $elitism\_type \leftarrow$ type of elitism
Initialize:
  Initialize quantum simulator backend
  Create MaxCut problem with dimension b
  Setup quantum circuit with population size and dimensions
Loop:
  for *generation* $= 1$ to *max_gen* **do**
    Measure quantum states to generate population
    Evaluate fitness of each solution
    if *elitism_type* $=$ *quantum* **then**
      Update quantum angles based on best solutions
    else if *elitism_type* $=$ *deterministic* **then**
      Preserve best solutions directly
    else if *elitism_type* $=$ *reinforcement* **then**
      Apply reinforcement learning update
    Apply mutation with probability μ
    Update best solution (gBest)
    Store generation statistic
  return gBest, evolution_history, fitness_history.

$$c_2^* = \left| q^l_{\lfloor \frac{n}{k-1} \rfloor} \cdots q^l_{(2\cdot\lfloor \frac{n}{k-1} \rfloor - 1)} \right\rangle \quad (31)$$
$$\cdots$$
$$c_{k-1}^* = \left| q^l_{(k-2)\lfloor \frac{n}{k-1} \rfloor} \cdots q^l_{(n-1)} \right\rangle$$

During this phase, the genetic information residing in chromosome c^* is transmitted to other chromosomes within the population $Q_p - \{c^*\}$. This transfer is achieved by entangling the qubits in every quantum state present in G with certain qubits belonging to the remaining quantum chromosomes.

3) THE MUTATION

Mutation is an essential technique in evolutionary algorithms to ensure that the population does not converge prematurely to suboptimal solutions. It involves utilizing the Pauli gate to rotate a single qubit not used in the entangled crossover at a specific θ around an axis. The mutation probability ($0 \leq \mu \leq 1$) determines the likelihood of applying the mutation operator to each qubit. It is usually set at a conservative value, such as 0.01, to avoid excessive disturbances to the quantum population. When implemented, the mutation operator flips the probability amplitude of a qubit, causing a switch from $|0\rangle$ to $|1\rangle$ or vice versa. This small perturbation generates new quantum entities that allow the QEA to explore new regions within the search space and avoid becoming trapped in local minima.

4) MEASUREMENT AND FITNESS EVALUATION

In order to compute the fitness function, it is necessary to transform quantum chromosomes into classical binary strings through a process known as quantum measurement. This is an important step, as quantum states alone cannot facilitate the computation of the fitness function, which requires classical inputs. Once the quantum chromosomes have collapsed, a classical computer conducts fitness function evaluations on the resulting set of binary strings. The fitness function assesses the effectiveness of each solution within the population by assigning a fitness score to every binary string. This score quantifies how well the solution satisfies the constraints and objectives of the optimization problem. A direct relationship exists between the fitness score and solution quality, with a higher fitness score indicating a superior solution. The fitness function's design is problem-specific and typically provided by the user. After computing the fitness scores, the algorithm proceeds to identify the most optimal option within the population.

5) THE QUANTUM ELITISM

The concept of elitist selection was originally introduced in classical genetic algorithms to enhance the formation of new genetic populations. This is accomplished by transferring the best individuals from the current generation to the next, guaranteeing that the algorithm generates high-quality solutions in subsequent iterations. The mentioned concept is also integrated into the QEA. Nevertheless, the probabilistic nature of quantum chromosomes calls for a transition in the elitism concept, aligning it with the principles of quantum computing. In this context, there are three primary approaches for transferring the best quantum chromosome from the current population to the next: pure quantum elitism, quantum elitism with reinforcement, and deterministic quantum elitism [63]. As the name suggests, pure quantum elitism generates the exact quantum state with best objective function value from the current iteration to the next iteration. Quantum elitism with reinforcement, on the other hand, modifies the current best quantum state to another state in the next generation such that the probability of the new state to collapse to the same bit string is increased. Finally, the deterministic quantum elitism generates the computational basis state with the best bit string from the previous iteration. That is, this new state has the unit probability of collapsing into the previously known best bitstring [63].

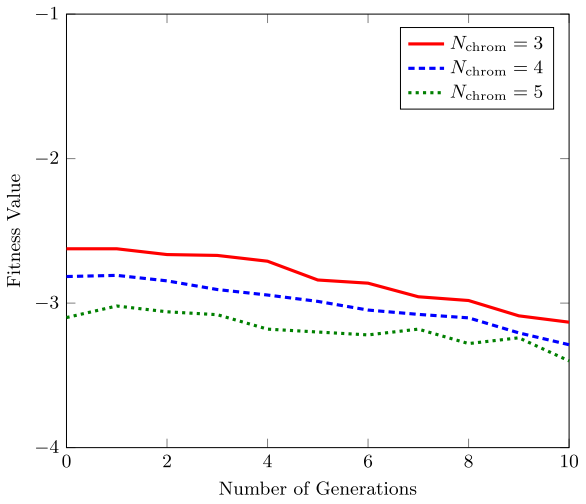


FIGURE 4. Solving Max-Cut problem with QGOA. For each generation j, we plot $Y_j = \frac{1}{M}\sum_{i=1}^{M} f\left(x^j_{\text{best}}\right)$, where $M(= 10^3)$ is the number of iterations for averaging and $f\left(x^j_{\text{best}}\right)$ is the fitness value of the objective function for the fittest chromosome in the jth generation.

D. SIMULATION RESULT OF QGOA FOR MAX-CUT PROBLEM

We have simulated a simplified Max-Cut problem using the QGOA. In this simulation, we define a graph with only 4 vertices and 4 edges. The graph is encoded in a binary string, where the two adjoining bits in the bitstring represent two adjoining vertices in the graph. The vertex that is represented by the last bit adjoins the vertex that is represented by the first bit. The cutting of the graph divided the graph into two regions. Vertices in the first (second) region can be assigned to 0 (1) and vice versa. We use the QGOA to find the best configuration of 1 and 0 in the quantum chromosome bit strings. The QGOA evolution for the max-cut problem is provided in Table 5. We can see from the fitness function defined above, the best configuration will result in fitness

value equals to −4. We have run the algorithm in the IBM Quantum simulator and present the result in Figure 4.

The result that is presented in the Figure 4 was obtained as follows: We set the generation step to 10. Including the initial generation, each generation was iterated 10^3 times. To plot a point in the figure, we took an average of the best fitness values among all chromosomes from all 10^3 iterations. These steps were repeated for each number of chromosomes, $k \in \{3, 4, 5\}$. As we can see, the higher the chromosomes number, the closer the initial generation average fitness value to the target fitness value. This is also the case for the final generation, more chromosomes produce better average fitness value at the final generation. This is because more chromosomes give more diversity in the gene pool so there are more choices of solutions to be explored. However, high number of chromosomes requires high computational resources. In the context of real quantum computers, each gene is encoded with one qubit. That means, if a chromosome contains 4 genes, that chromosome is encoded with 4 qubits. Therefore, as in the example that we have implemented, 5 chromosomes with 4 genes each were encoded with 20 qubits. This could become a trade-off when this algorithm is implemented in the NISQ quantum computers, as a higher number of chromosomes will produce a better fitness value but will require a higher number of qubits.

V. EVOLUTIONARY ALGORITHMS IN NISQ COMPUTING

In this section we outline the potential use cases of evolutionary computing for NISQ-era computing. Due to specific characteristics of NISQ devices, evolutionary computing algorithms have a great potential to contribute to NISQ-era quantum computing. We outline these specific characteristics and opportunities below.

A. NISQ COMPUTER

The common approach to perform computational task in quantum computer is using gate-based model where the computational operation is done via sequence of quantum gates similar with classical computer. As classical computing has sets of universal gates that can construct any classical circuit with the combination of the element of the sets, quantum computing also has such set of universal gate which can construct any quantum circuit. Such set of universal gates consist of single qubit and two-qubit gates. An example of such set consist of Hadamard, CNOT, $\sqrt[2]{Z}$, and $\sqrt[4]{Z}$. Here, universal refer to the ability to approximate with arbitrary precision any n-qubit gate using finite number of gates taken from the set of universal gates [72].

Harnessing the potential advantage from quantum computing for many practical applications is hindered by limited scale quantum devices that are prone to external disturbance and errors due to the complexities of fabricating, controlling and maintaining quantum system as well as the susceptibility of quantum system to various types of noise, such as, thermal noise and electromagnetic interference. The sensitivity of quantum system to noises and errors leads to inaccuracies in the computed results, which impedes its implementation in various practical problems. The current era of quantum computing is marked by development of quantum devices with moderate number of qubits which have short coherence times called the NISQ era. In this NISQ era, the gate-based quantum computer meets some challenges such as, qubits decoherence, gate fidelity, and scalability [10].

Quantum state preparation is a fundamental problem in quantum computing since it is important in quantum algorithms and experiment. Some quantum algorithms need to prepare the initial state into a specific quantum state (typically quantum state that has entanglement or superposition) to provide quantum advantage, e.g., quantum phase estimation algorithm, Grover's algorith, and Deutsch-Jozsa algorithm require equal superposition initial states [5], [73], [74]. Another interesting family of entangled quantum states includes the W-state

$$|\psi\rangle = \frac{1}{\sqrt{n}} \left(|10\ldots0\rangle + |01\ldots0\rangle + \cdots + |00\ldots1\rangle\right). \quad (32)$$

Preparing W-state using set of universal quantum gate is not trivial. Considering 6-qubit system and $\{R_X, R_Y, R_Z, \text{CNOT}\}$ set of gates using Qiskit's initialize function that utilize Shende, Bullock, and Markov (SBM) method results in 125 gates with a depth of 120 [75]. Since the more gates and depth used in quantum circuit the more noise is accumulated, finding an optimal way of preparing a quantum state using a fewer number of gates and smaller circuit depth becomes an important problem in NISQ era.

Limitation on NISQ hardware produce algorithms that are suited for the noisy and limited scale quantum devices. The common algorithms are the variational quantum algorithms (VQAs) which use parameterized quantum circuits (PQCs) to perform a computational task and leverage classical computer to optimize the parameters of the PQC making it a hybrid quantum classical algorithm. VQA is a suitable algorithm for NISQ devices as it optimize the parameters of the quantum circuit instead of relying on a fixed noise-free quantum circuit. VQAs aim to pave the way towards achieving quantum advantage in the current NISQ era. VQAs have been applied to various problems, such as quantum chemistry, machine learning, and optimization, using currently limited number of qubits [76].

After designing quantum circuit, the circuit is compiled to match the physical hardware of the quantum system. One of the important part of the compilation is to match the circuit with the physical qubits connectivity. Physically, two-qubit gates can only be applied to qubits that are physically connected with each other. For unconnected qubits, the SWAP gates need to be applied to bring the state from unconnected qubits into the connected ones. Then, the two-qubit gates can be applied into the quantum state [77].

B. QUANTUM STATE PREPARATION

The problem of preparing arbitrary state of quantum system is an important problem in quantum algorithms. The limitation

of NISQ era quantum hardware requires the optimal design of quantum circuit to prepare such arbitrary quantum state. Algorithm that can prepare arbitrary quantum state with high fidelity with few number of quantum gates and depth become important due to the exponential grow of error rate as the circuit depth increases.

Using a smaller number of two-qubit gates is better, as physical hardware rarely provides full connectivity for all the qubits. The fewer two-qubit gates used, the fewer SWAP operators are needed to bring the quantum states onto the physically connected qubits. The single-qubit gates have an order of magnitude smaller error rate compared to the CNOT gate, which emphasizes the importance of the number of CNOT gates used in the state preparation circuit. Many attempts has been done to bound the number of CNOT gates used to prepare arbitrary quantum state, for example SBM method can prepare arbitrary quantum state of n-qubit system with number of CNOT gates bounded above by $2^{n+1} - 2n$ [75].

We follow the formal definition of the state preparation problem and its evolutionary algorithm solution in [75] and [78]. Given a set of universal quantum gates S, the quantum state preparation problem is defined as to find a sequence of quantum gates $\{s_m\}_{m=1}^{M}$ taken from set of universal quantum gates, $s_m \in S$, to prepare the specific initial state of a quantum hardware (usually $|0\rangle$ state) to a state $|\psi\rangle$ given in equation (4) with arbitrary $\{\alpha_i\}$ values such that,

$$s_M s_{M-1} \cdots s_1 |0\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle \tag{33}$$

while the performance of $\{s_m\}$ is measured by how close the resulted state $s_M s_{M-1} \cdots s_1 |0\rangle$ to the target state using fidelity given by,

$$F(\{s_m\}) = \left| \sum_{i=0}^{2^n-1} \alpha_i^* \langle i| s_M s_{M-1} \cdots s_1 |0\rangle \right|^2. \tag{34}$$

The genome in the genetic algorithm can be represented by the quantum circuit that consist of a list of genes that are constructed by information about the quantum gate such as the gate, the control qubit, the target qubit, and the angle of the parameterized gate. An example of a gene is,

$$g_i = (s_i, t_i, c_i, \theta_i), \tag{35}$$

where $s_i \in S$, t_i is the target qubit, c_i is the control qubit, and θ_i is the angle of the parameterized gate. For a single qubit gate, c_i is not used and for the fixed gate like CNOT or Hadamard θ_i is not used. The unused variable can be defined as a specific value like -999 or None. Having the genes, the genome is then defined as,

$$G_k = (g_1, g_2, \cdots, g_n) \tag{36}$$

The mutation can be performed in multiple ways such as randomly changing the target or control qubit, adjusting the angle on the parameterized gate can also be done to perform the mutation, changing the position of the genes, inserting a new gene in random position, deleting a specific gene, replacing a gene, swapping a pair of genes and scrambling the genes by a permutation.

The crossover operation can be implemented by picking a random number genes in random position in the genome from one parent and pick the remaining genes from another parent to produce an offspring. The unselected genes from both parents are combined as well to produce the second offspring.

The fitness function used for evaluating the genomes can have multiple objective such as the accuracy of preparing the target state and the complexity of the circuit which will have trade-off within each other. The accuracy measure can be quantified using the fidelity function while the circuit complexity can be quantified by the number of gates used in the circuit. The number of single qubit and two-qubit gates can have different weight in the circuit complexity metrics where the two-qubit gate can have more weight since they have more error as compared to the single qubit gates.

The genetic algorithm starts by specifying the size of the population, the number of qubits, the set of universal gates, and the target state. Each generation is evaluated by the fitness function and the high quality genomes are used to build the next generation by means of the aforementioned evolutionary operations. The evolutionary operations are performed in probabilistic fashion. This process is performed for the specified number of generation or a termination condition is met.

C. VARIATIONAL QUANTUM ALGORITHM

Finding a suitable algorithm for currently available NISQ devices is challenging. VQA answers this challenge by introducing PQCs that can be optimize to absorb the effect of noise in the quantum gates. Being a hybrid quantum classical algorithm, VQA uses PQC to perform computational task while guided by classical optimizer to improve the quality of the computation via a specified cost function. Thus, there are at least three main components of VQA: the PQC, the cost function, and the classical optimizer [76]. A typical workflow of a VQA is shown in Fig. 5.

In this subsection, we will focus on the evolutionary algorithm for parameter optimization in VQA, as this particular problem presents significant challenges. These challenges include issues such as local optima and barren plateaus, which become increasingly problematic, especially when dealing with a large number of qubits [79].

The first introduction of VQA is the proposal of finding the lowest energy eigenstate of a Hamiltonian termed variational quantum eigensolver (VQE). VQE uses PQC to create a trial quantum state that will be measured by the Hamiltonian of the system. The parameters in the PQC will be optimized based on the expected value of the measurement. VQE aims to minimize this expected value and the trial state that minimize the expected value is seen as the approximation of the lowest energy eigenstate [80].

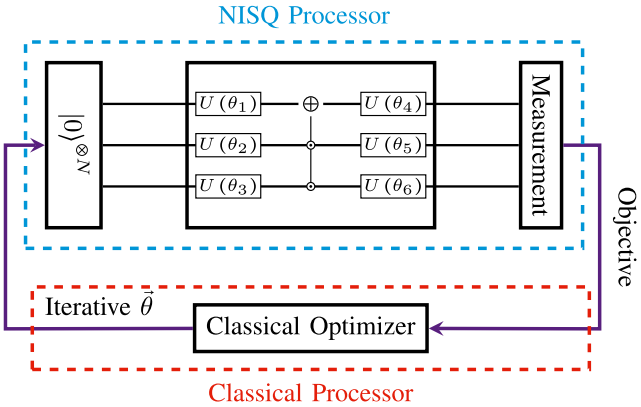


FIGURE 5. Variational quantum algorithms is a representative class of hybrid classical-quantum algorithms. A parameteric quantum state is prepared and measured on the NISQ processor. A classical processor, based on the measured objective function value, attempts to optimize the state parameters. This iterative process is continued until a convergence is achieved or a termination criterion is satisfied.

In the optimization frontier, the QAOA is highly recognized to heuristically solve combinatorial optimization problems [81], [82]. QAOA is inspired by adiabatic quantum computing where it can find the lowest energy eigenstate of a given Hamiltonian (in QAOA is called the cost Hamiltonian) by evolving a lowest energy eigenstate of a known Hamiltonian (in QAOA is called the mixer Hamiltonian) adiabatically. Differ with general VQA, QAOA has particular circuit structure where it has alternating Hamiltonian structure between the mixer and cost Hamiltonians. Instead of adiabatically evolving the lowest energy eigenstate of the mixer Hamiltonian, QAOA takes the parameterized approach putting it as a VQA. Based on this parameter optimization, QAOA tries to find the lowest energy eigenstate of a cost Hamiltonian that corresponds to the potential solution of the combinatorial optimization problem [83].

Quantum reinforcement learning algorithm mainly depends on gradient based optimization algorithm. The main different structure of quantum reinforcement learning or quantum machine learning in general as compared to other variational quantum algorithm is the encoding process of classical data into quantum system. Typically, this encoding structure does not include trainable parameters. However, there are encoding structures that introduce trainable parameters in the encoding process, as seen in the data reuploading approach [84]. In this case, this additional trainable parameters can also be optimized with the parameters of PQC.

Optimizing the parameters of VQA using evolutionary algorithm starts by initializing a population of parameters randomly. Next, each individual within the population is evaluated via a fitness function. The fitness function can be vary depends on a specific VQA. For VQE and QAOA, the fitness function is the expected value of the Hamiltonian for which the lowest energy eigenstate to be found. Meanwhile for quantum reinforcement learning or quantum machine learning, the fitness function can be the cost function, typically an expected value of an observable such as, Pauli Z, or the probability of finding a specific state [85]. From the evaluation, high quality individuals will be picked to create a new generation. The new generation is created from the selected high quality solution with mutation operation. The mutation operation can be performed by adding a random value with a specified scaling factor or can be a linear combination of the high quality individuals. This process is repeated for a number of specified iteration or a termination criteria is met [86], [87], [88].

D. QUANTUM CIRCUIT COMPILATION

The goal of quantum circuit compilation (QCC) is to modify the initial quantum circuit so it can be run on quantum hardware while minimizing the additional quantum gates required for the compilation and the overall circuit depth. This process includes incorporating swap gates into the optimal circuit, determining the sequence of gate operations, and relocating quantum states to enable the execution of quantum gates in the subsequent layer. The QCC can be expressed as a tuple
$C = \langle C_{\text{in}}, \pi_{\text{in}}, G_{\text{hw}} \rangle$, where C_{in} denotes the initial quantum circuit, which symbolizes the quantum algorithm designed to address the specific problem, π_{in} represents the initial allocation of quantum states to qubits, and G_{hw} characterizes the quantum hardware as a multigraph [77].

Various approaches exist for addressing the QCC problem, including the Rollout Heuristic (RH) [89], genetic algorithm (GA) [90], and decomposition-based genetic algorithm (DBGA) [77]. The RH employs a heuristic single-pass strategy, which implies that it tackles the problem in a single iteration without revisiting any portion of the solution. This contrasts with the DBGA, which utilizes a single-pass strategy at the decoding stage but leverages a genetic-based optimization strategy to more effectively navigate the search space and identify higher quality solutions. The GA is an optimization technique influenced by the principles of natural selection and genetics, used to identify approximate solutions to optimization and search challenges. Unlike the DBGA, which employs a decomposition approach, the GA aims to address the entire problem in one iteration.

The DBGA is a step-by-step process designed to address the QCC problem [77]. It constructs solutions progressively over several rounds. In each round, a genetic algorithm is applied to the sub-problem presented by the current round, beginning with solutions from preceding rounds. This method is distinct from other GA variants for the same issue in multiple ways, including chromosome encoding, decoding algorithm, heuristic population initialization, diversification operator, and decomposition strategy. Based on [77], key aspects of DBGA include:

1) Coding/Decoding Schema: During each round r, a chromosome consists of the quantum gates sequence in round r, along with a partial solution to the sub-problem of rounds $1, \ldots, r-1$. Each gate interacts

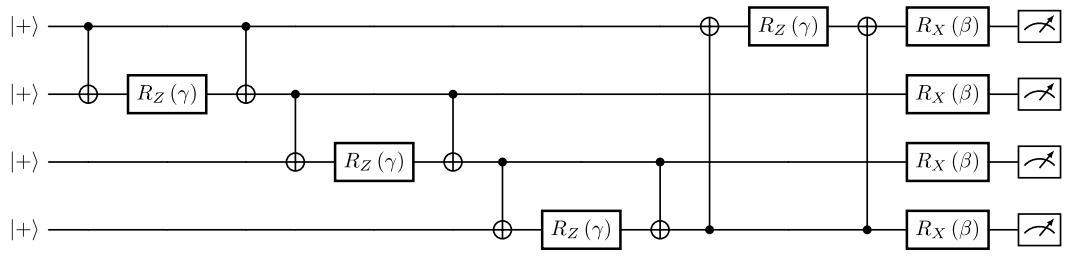


FIGURE 6. A single layer of quantum approximate optimization algorithm (QAOA) circuit. Addition of each layer introduces two new tunable parameters. Hence, a *p*-layered QAOA circuit requires tuning 2*p* parameters.

with a pair of quantum states, and the chromosome specifies the adjacent qubits pair where it must be executed. The decoding algorithm incorporates the fewest swap gates necessary to relocate the quantum states to the appropriate qubits.

2) Decomposition Approach: The DBGA employs a decomposition based method and aims to optimize the p-s gates scheduling in round r, given several solutions to the sub-problem defined by rounds $1, \ldots, r-1$. This strategy contrasts with others that address the entire problem simultaneously and may result in sub-optimal solutions.
3) Genetic Algorithm: In each round, the DBGA utilizes solutions from the preceding round and "expands" them by incorporating the new round's corresponding gates. This is achieved through a genetic algorithm, which entails generating an initial solutions population, evolving them using selection, recombination, replacement, and diversification operators until a termination criterion is satisfied.

E. SIMULATION RESULTS

We consider Max-Cut problem with 4 nodes, $V = \{0, 1, 2, 3\}$, and 4 edges, $E = \{(0, 1), (1, 2), (2, 3), (3, 0)\}$ with all the weights are set to 1. The problem is encoded in the cost Hamiltonian as,

$$H_C = \sum_{i,j \in E} Z_i Z_j, \tag{37}$$

where the subscript denotes in which qubit the operator acts. While the mixer Hamiltonian is

$$H_M = \sum_{i \in V} X_i. \tag{38}$$

Depending on the number of layer p, the output state is given as,

$$|\psi\rangle = \exp\left(-\iota H_M \beta_p\right) \exp\left(-\iota H_C \gamma_p\right) \cdots \exp\left(-\iota H_M \beta_1\right) \exp\left(-\iota H_C \gamma_1\right) |+\rangle^{\otimes 4}, \tag{39}$$

where β, γ are the parameters that will be optimized using the genetic algorithm. The genetic algorithm can outperform other optimization algorithms (such as COBYLA, Powell, Nelder-Mead, and SPSA) in optimizing the QAOA parameters in solving the max-cut problem as shown in [88]. The genetic algorithm performs better as the problem size becomes larger and the number of QAOA layers increases which shows the advantage of using genetic algorithm in complex optimization problems.

Algorithm 3 QAOA With Evolutionary Algorithm [88]

procedure QAOA-EA
  Input:
  $p \leftarrow$ QAOA layers $\{1, 2, 3, 4\}$
  $N_{\text{pop}} \leftarrow$ population sizes $\{10, 20, 30\}$
  $N_{\text{run}} \leftarrow$ number of runs (10)
  $N_{\text{gen}} \leftarrow$ number of generations (100)
  shoots $\leftarrow 10,000$
  $k \leftarrow$ tournament size (5)
  $N_{\text{par}} \leftarrow$ number of parents (3)
  r_cross $\leftarrow$ crossover probability (0.5)
  r_mut $\leftarrow$ mutation probability (0.3)
  mut_strength $\leftarrow$ mutation strength (0.001)
  Loop:
  for $p \in \{1, 2, 3, 4\}$ **do**
    for $N_{\text{pop}} \in \{10, 20, 30\}$ **do**
      for run $= 1$ to N_{run} **do**
        Initialize random population $\mathcal{P}_0$ of size N_{pop}
        for gen $= 1$ to N_{gen} **do**
          Evaluate Fitness:
          Compute fitness of each individual using 10, 000 shots
          Elitist Selection:
          Retain the best individual
          Tournament Selection:
          Select N_{par} parents with size k
          Generate Next Population:
          Retain best and parents
          for remaining individuals **do**
            Apply crossover and mutation
        Save results for p and N_{pop} to CSV
  return Average and plot cost values, highlight best results

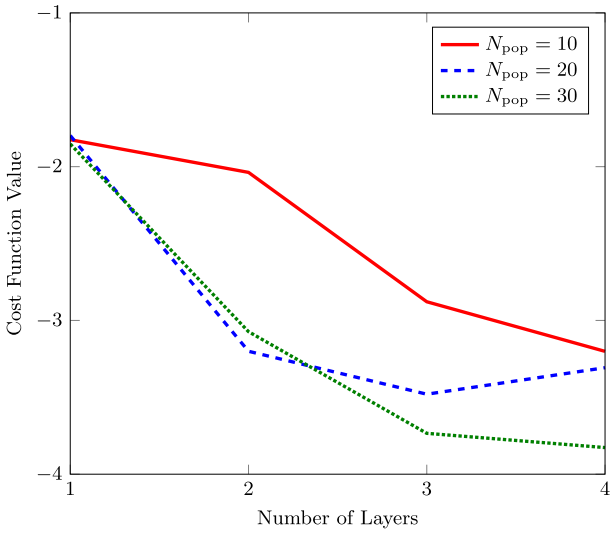


FIGURE 7. Solving Max-Cut with QAOA. We plot the average of 10 runs for each pair of parameters (p, N_{pop}). A clear improvement in the approximation ratio can be seen by increasing the number of QAOA layers (p).

The evolutionary algorithm starts by randomly generating N_{pop} individual where each individual is in $[0, 2\pi)^{2p}$. For each generation, the individual is evaluated by the fitness function which is the average of H_C with respect to the output state $|\psi\rangle$ using 10000 shots. To generate the next population, the elitist approach is used where the best individual is kept for the next generation. Furthermore, the tournament selection is performed to obtained the parents for the next generation. We fix the tournament size to 5 and the number of parents to $N_{\text{par}} = 3$. This N_{par} parents are included for the next generation. The rest of the individuals for the next generation are obtained by the uniform crossover where each gene of the child will be inherited from the first or second parent with probability of 0.5 and Gaussian mutation where a random number from Gaussian distribution will be added to each gene with scalar factor of 0.001 with probability of 0.3.

The pseudo-code for our QAOA simulation with genetic algorithm is provided as Algorithm 3. The simulation is run for QAOA layers $p \in \{1, 2, 3, 4\}$ and population size $N_{\text{pop}} \in \{10, 20, 30\}$. For each p, N_{pop} the simulation is run 10 times and for each run the genetic algorithm is run for 100 iteration. The cost function values are then averaged. The result shows that as p and N_{pop} increase the cost function value tends to decrease as depicted in Figure 7. When the population set is 30 and the number of layers is 4, the average cost function value approaches the best solution with $\approx$ 0.97 approximation ratio. With this approximation ratio, starting from the equal probability of all possible quantum states (solutions for max-cut problem), the QAOA will produce the state $|0101\rangle$ or $|1010\rangle$, the states that correspond to the best solutions of this particular max-cut problem, after measurement with nearly 0.5 probability for each state. Hence the best solution will be found with nearly unit probability.

Before concluding this section, we make several remarks on the current state of NISQ computing that highlights the challenges and opportunities for this era of quantum computing. Although VQA serves as one of the primary avenues in NISQ computing, it faces a critical challenge: the trainability of the PQC. The main cause of this issue is the potential presence of barren plateaus (can be circuit-, measurement-, and noise-induced) where the gradient of the cost function exponentially vanished as the system size grows [91], [92], [93], [94]. Common approaches to remove the existence of barren plateaus lead to the use of classically simulable subspaces instead of using the whole Hilbert space which renders the nonclassicality of the VQA questionable [95]. This leaves us with the need to design sophisticated optimizer that can overcome this issue to realize the success of VQA. In such gradient-related problems, gradient-based optimizers can face tremendous hurdle in training the PQC. Hence, gradient-free optimizers present opportunities in such scenario without relying on the gradient to update the PQC parameter and strategically maneuvering the cost function landscape. Specifically, evolutionary computing algorithms offer potential solutions to heuristically navigate at the barren plateau region by applying series of genetic operators to successfully train the PQC as demonstrated by the recent work [96].

VI. EXPERIMENTAL PROGRESS

This section is devoted to present the recent experimental studies and implementations of quantum evolutionary algorithms, emphasizing the feasibility of these algorithms and their real-world performance. In particular, the study referenced in [63] has introduced a novel evolutionary algorithm leveraging the capabilities of a real quantum processor, a computing device exploiting quantum mechanics for accelerated computation. The approach utilizes quantum principles like superposition and entanglement to implement quantum genetic concepts, achieving efficient genetic evolution on quantum devices. The algorithm was implemented using a hybrid hardware architecture, combining classical processors with the family of quantum processors provided by the IBM Q Experience initiative. The experimental results of [63] have shown that this quantum genetic algorithm offers a promising and innovative bio-inspired optimization strategy.

Further, variational quantum circuits were designed and implemented in [97]. These circuits involve adjusting the parameters until their empirical outcomes closely match the desired results. Numerical first-order methods are used to optimize the parameters while keeping the circuit's gate composition fixed. Specifically, an evolutionary strategy is considered to balance both finding suitable circuit architectures and tuning parameters simultaneously. The study assessed this approach through simulations and actual quantum hardware, tackling benchmark problems like the transverse field Ising Hamiltonian and the Sherrington-Kirkpatrick spin model. Despite the limitations of NISQ hardware, the slowdown on real quantum machines compared to simulations is found to be minimal. Additionally, the research

investigates the mutation operations that play a significant role in the optimization process. The findings of [97] offer insights into how randomized search heuristics perform on quantum hardware and suggest directions for further refinement of evolutionary quantum gate circuits.

A genetic algorithm for state preparation (GASP) has been implemented and evaluated in [75] to generate relatively low-depth quantum circuits for initializing a quantum computer in a specific quantum state. GASP has efficiently generated low-depth quantum circuits to initialize quantum computers in specific states. Through employing a basis set of gates and a genetic algorithm, GASP has produced circuits with fewer gates and lower depth compared to other methods, enhancing overall accuracy by avoiding error accumulation in high-depth circuits. In comparisons with the state initialization technique based on exact synthesis in IBM Qiskit, tested under both noisy simulations and physical IBM Quantum devices, GASP has outperformed Qiskit's exact general circuit synthesis method. The significant reduction in the overall gate count achieved by GASP can be valuable when considering its application for quantum state initialization within the constraints of feasible circuit lengths on NISQ-era hardware.

The work in [79] has developed an evolutionary variational quantum eigensolver (EVQE), a new algorithm that dynamically generates and optimizes ansatzes using evolutionary programming techniques. Unlike traditional variational quantum algorithms like VQE, EVQE is versatile and applicable across various domains, delivering precise energy evaluations with efficient ansatzes. Specifically, EVQE can adapt to the noise and connectivity constraints of specific quantum computers, reducing error in noisy simulations by at least three times compared to VQE with any tested ansatz configuration. The experiments in this study were conducted using the Qiskit Aer QASM Simulator, it was configured to mimic the noise profile and qubit connectivity of two IBMQ devices: the 20-qubit Tokyo and the 5-qubit Vigo. Experimental results, both in simulation and on real quantum hardware, affirm EVQE's effectiveness for general-purpose optimization in today's and near-future quantum computers.

A major challenge in quantum computing is to efficiently design control pulses for quantum state and gate preparation. While gradient-based methods are commonly used, they struggle when gradient information is hard to obtain due to system uncertainties. Gradient-free evolutionary algorithms are an alternative but often less efficient. As an alternative, the paper [98] has introduced an efficient mutation rule that considers both current and previous individuals, resulting in an enhanced differential evolution algorithm. This algorithm aims to solve the quantum state and gate preparation problems instead of the frequently-used gradient based optimal control methods. It has been demonstrated its effectiveness through numerical benchmarks in pulse optimization for quantum systems; namely, in a two-qubit nuclear magnetic resonance (NMR) system. Comparisons with traditional differential evolution algorithms highlight the proposed algorithms can achieve superior convergence speed and robustness, particularly in handling uncertainties like pulse imperfections and measurement errors.

VII. SUMMARY AND CONCLUSION

In this work we explored two distinct but related topics at the intersection of quantum computation and evolutionary computation.

First, the field of quantum-inspired metaheuristics began before the availability of programmable quantum computers. With the development and availability of modern quantum computers, it is possible to prototype these metaheuristics on actual quantum devices. A comparison of quantum hardware implementations with their classical counterparts is interesting due to the following reasons. 1) Classical implementations have the liberty of direct access to a quantum state, even after the measurement—since it is a simulation of quantum phenomenon. This makes the evolutionary process easier since we can maintain a record of the description of simulated states. Quantum computing implementations do not have this privilege as quantum state is not an observable, making the track of evolutionary process challenging. 2) Simulating quantum computations on a classical computer is expensive. In fact, this was the first motivation for the proposal of quantum computing device. In terms of time-complexity, QEAs outperform QiEAs exponentially, which serves as a motivation for porting quantum-inspired metaheuristics to quantum-hardware implementations.

Second, quantum computers hold the potential to solving some of the most challenging problems in computing. The current generation of quantum processors have limited number of qubits and fidelity of quantum operations, requiring frequent measurements and iterations with classical processors to optimize the quantum circuits. This scenario of requiring heuristic optimization of quantum circuits provides an excellent opportunity for researchers in the field of evolutionary computation to explore the possible role in NISQ-era quantum computing.

There exists a clear potential for the cooperation between the two computing frameworks to create impact in diverse fields requiring optimization of nonconvex problems. In future, it will be interesting to see a stronger bond between the two fields to possibly offer a practical advantage in the general field of computation.

REFERENCES

[1] R. P. Feynman, "Simulating physics with computers," *Int. J. Theor. Phys.*, vol. 21, nos. 6–7, pp. 467–488, Jun. 1982.
[2] P. Benioff, "The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines," *J. Stat. Phys.*, vol. 22, no. 5, pp. 563–591, May 1980.
[3] Y. Manin, "Computable and uncomputable," *Sovetskoye Radio*, vol. 128, p. 28, Apr. 1980.
[4] D. Deutsch, "Quantum theory, the church-turing principle and the universal quantum compute," *Proc. R. Soc. Lond. A*, vol. 400, no. 18, pp. 97–117, Dec. 1985.
[5] D. Deutsch and R. Jozsa, "Rapid solution of problems by quantum computation," *Proc. Roy. Soc. London A, Math. Phys. Sci.*, vol. 439, no. 1907, pp. 553–558, Dec. 1992.

[6] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proc. 35th Annu. Symp. Found. Comput. Sci.*, 1994, pp. 124–134.
[7] C. Gidney and M. Ekera, "How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits," *Quantum*, vol. 5, p. 433, Apr. 2021.
[8] *Quantum Algorithm Zoo*. Accessed: Nov. 20, 2024. [Online]. Available: https://quantumalgorithmzoo.org/
[9] M. Webber, V. Elfving, S. Weidt, and W. K. Hensinger, "The impact of hardware specifications on reaching quantum advantage in the fault tolerant regime," *AVS Quantum Sci.*, vol. 4, no. 1, Mar. 2022, Art. no. 013801.
[10] J. Preskill, "Quantum computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, Aug. 2018.
[11] F. Arute, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, no. 7779, pp. 505–510, Oct. 2019.
[12] L. S. Madsen, F. Laudenbach, M. F. Askarani, F. Rortais, T. Vincent, J. F. F. Bulmer, F. M. Miatto, L. Neuhaus, L. G. Helt, M. J. Collins, A. E. Lita, T. Gerrits, S. W. Nam, V. D. Vaidya, M. Menotti, I. Dhand, Z. Vernon, N. Quesada, and J. Lavoie, "Quantum computational advantage with a programmable photonic processor," *Nature*, vol. 606, no. 7912, pp. 75–81, Jun. 2022.
[13] H.-S. Zhong, "Quantum computational advantage using photons," *Science*, vol. 370, no. 6523, pp. 1460–1463, Dec. 2020.
[14] E. Pednault, J. A. Gunnels, G. Nannicini, L. Horesh, and R. Wisnieff, "Leveraging secondary storage to simulate deep 54-qubit sycamore circuits," 2019, *arXiv:1910.09534*.
[15] B. Villalonga, M. Yuezhen Niu, L. Li, H. Neven, J. C. Platt, V. N. Smelyanskiy, and S. Boixo, "Efficient approximation of experimental Gaussian boson sampling," 2021, *arXiv:2109.11525*.
[16] C. Oh, M. Liu, Y. Alexeev, B. Fefferman, and L. Jiang, "Classical algorithm for simulating experimental Gaussian boson sampling," 2023, *arXiv:2306.03709*.
[17] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, W. Mok, S. Sim, L. C. Kwek, and A. Aspuru-Guzik, "Noisy intermediate-scale quantum algorithms," *Rev. Mod. Phys.*, vol. 94, no. 1, Feb. 2022, Art. no. 015004.
[18] K. Sörensen, M. Sevaux, and F. Glover, "A history of metaheuristics," in *Handbook of Heuristics*. Cham, Switzerland: Springer, 2018, pp. 791–808.
[19] E. Talbi, "Metaheuristics," *Scholarpedia*, vol. 10, p. 6532, Jun. 2009.
[20] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*. Bristol, U.K.: Luniver Press, 2010.
[21] E. Tang, "A quantum-inspired classical algorithm for recommendation systems," in *Proc. 51st Annu. ACM SIGACT Symp. Theory Comput.*, Jun. 2019, pp. 217–228.
[22] O. H. M. Ross, "A review of quantum-inspired metaheuristics: Going from classical computers to real quantum computers," *IEEE Access*, vol. 8, pp. 814–838, 2020.
[23] L. Huynh, J. Hong, A. Mian, H. Suzuki, Y. Wu, and S. Camtepe, "Quantum-inspired machine learning: A survey," 2023, *arXiv:2308.11269*.
[24] F. S. Gharehchopogh, "Quantum-inspired Metaheuristic algorithms: Comprehensive survey and classification," *Artif. Intell. Rev.*, vol. 56, no. 6, pp. 5479–5543, Jun. 2023.
[25] L. Miguel Antonio and C. A. Coello Coello, "Coevolutionary multiobjective evolutionary algorithms: Survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 22, no. 6, pp. 851–865, Dec. 2018.
[26] Z.-H. Zhan, J.-Y. Li, and J. Zhang, "Evolutionary deep learning: A survey," *Neurocomputing*, vol. 483, pp. 42–58, Apr. 2022.
[27] S. Hakemi, M. Houshmand, E. Kheirkhah, and S. A. Hosseini, "A review of recent advances in quantum-inspired metaheuristics," *Evol. Intell.*, vol. 17, no. 2, pp. 627–642, Apr. 2024.
[28] K.-H. Han and J.-H. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *IEEE Trans. Evol. Comput.*, vol. 6, no. 6, pp. 580–593, Dec. 2002.
[29] H.-P. Chiang, Y.-H. Chou, C.-H. Chiu, S.-Y. Kuo, and Y.-M. Huang, "A quantum-inspired Tabu search algorithm for solving combinatorial optimization problems," *Soft Comput.*, vol. 18, no. 9, pp. 1771–1781, Sep. 2014.
[30] A. Malossini, E. Blanzieri, and T. Calarco, "Quantum genetic optimization," *IEEE Trans. Evol. Comput.*, vol. 12, no. 2, pp. 231–241, Apr. 2008.
[31] J. C. Bardin, D. Sank, O. Naaman, and E. Jeffrey, "Quantum computing: An introduction for microwave engineers," *IEEE Microw. Mag.*, vol. 21, no. 8, pp. 24–44, Aug. 2020.
[32] J. Ng and D. Abbott, "Introduction to solid-state quantum computation for engineers," *Microelectron. J.*, vol. 33, nos. 1–2, pp. 1–177, Jan. 2002.
[33] M. Bonilla-Licea, D. Bonilla-Licea, M. Bonilla-Estrada, A. Soria-López, and J. Carlos Martínez-García, "A primer on quantum mechanics for electrical engineers," *IEEE Access*, vol. 12, pp. 114658–114669, 2024.
[34] D. Ferry, *Quantum Mechanics: An Introduction for Device Physicists and Electrical Engineers*. Boca Raton, FL, USA: CRC Press, Jan. 2001.
[35] D. A. B. Miller, *Quantum Mechanics for Scientists and Engineers*. Cambridge, U.K.: Cambridge Univ. Press, Apr. 2008.
[36] J. Watrous, *The Theory of Quantum Information*, 1st ed., Cambridge, U.K.: Cambridge Univ. Press, Apr. 2018.
[37] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, 10th ed., Cambridge, U.K.: Cambridge Univ. Press, Jan. 2011.
[38] G. E. P. Box, "Evolutionary operation: A method for increasing industrial productivity," *Appl. Statist.*, vol. 6, no. 2, p. 81, Jun. 1957.
[39] R. M. Friedberg, "A learning machine: Part I," *IBM J. Res. Develop.*, vol. 2, no. 1, pp. 2–13, Jan. 1958.
[40] R. M. Friedberg, B. Dunham, and J. H. North, "A learning machine: Part II," *IBM J. Res. Develop.*, vol. 3, no. 3, pp. 282–287, Jul. 1959.
[41] T. Back, U. Hammel, and H.-P. Schwefel, "Evolutionary computation: Comments on the history and current state," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 3–17, Apr. 1997.
[42] P. A. Vikhar, "Evolutionary algorithms: A critical review and its future prospects," in *Proc. Int. Conf. Global Trends Signal Process., Inf. Comput. Commun. (ICGTSPICC)*, Dec. 2016, pp. 261–265.
[43] J.-Y. Li, Z.-H. Zhan, and J. Zhang, "Evolutionary computation for expensive optimization: A survey," *Mach. Intell. Res.*, vol. 19, no. 1, pp. 3–23, Feb. 2022.
[44] K. D. Jong, "Evolutionary computation," *Nat. Rev. Genet.*, vol. 2, pp. 428–436, Jul. 2007.
[45] T. Bäck and H. Schwefel, "Evolutionary computation: An overview," *Annu. Rev. Ecol. Evol. Syst.*, vol. 30, pp. 20–29, Dec. 2002.
[46] Z.-G. Chen, Z.-H. Zhan, S. Kwong, and J. Zhang, "Evolutionary computation for intelligent transportation in smart cities: A survey," *IEEE Comput. Intell. Mag.*, vol. 17, no. 2, pp. 83–102, May 2022.
[47] X.-S. Yang, *Recent Advances in Swarm Intelligence Evolutionary Computation*. Cham, Switzerland: Springer, Jan. 2015.
[48] E. H. Houssein, A. G. Gad, K. Hussain, and P. N. Suganthan, "Major advances in particle swarm optimization: Theory, analysis, and application," *Swarm Evol. Comput.*, vol. 63, Jun. 2021, Art. no. 100868.
[49] K. Arulkumaran, A. Cully, and J. Togelius, "AlphaStar: An evolutionary computation perspective," in *Proc. Genetic Evol. Comput. Conf. Companion*, Jul. 2019, pp. 314–315.
[50] K. Sörensen, "Metaheuristics—The metaphor exposed," *Int. Trans. Oper. Res.*, vol. 22, no. 1, pp. 3–18, Jan. 2015.
[51] J. Del Ser, E. Osaba, D. Molina, X.-S. Yang, S. Salcedo-Sanz, D. Camacho, S. Das, P. N. Suganthan, C. A. Coello Coello, and F. Herrera, "Bio-inspired computation: Where we stand and what's next," *Swarm Evol. Comput.*, vol. 48, pp. 220–250, Aug. 2019.
[52] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Comput. Surv.*, vol. 35, no. 3, pp. 268–308, Sep. 2003.
[53] M. Gendreau and J.-Y. Potvin, *Handbook of Metaheuristics*. New York, NY, USA: Springer, 2018.
[54] V. Tomar, M. Bansal, and P. Singh, "Metaheuristic algorithms for optimization: A brief review," *Eng. Proc.*, vol. 59, no. 1, p. 238, Mar. 2024.
[55] C.-W. Tsai and M.-C. Chiang, *Handbook of Metaheuristic Algorithms*. Amsterdam, The Netherlands: Elsevier, May 2023.
[56] G. Zhang, "Quantum-inspired evolutionary algorithms: A survey and empirical study," *J. Heuristics*, vol. 17, no. 3, pp. 303–351, Jun. 2011.
[57] S. Karmakar, A. Dey, and I. Saha, "Use of quantum-inspired metaheuristics during last two decades," in *Proc. 7th Int. Conf. Commun. Syst. Netw. Technol. (CSNT)*, Nov. 2017, pp. 272–278.
[58] K.-H. Han and J.-H. Kim, "Genetic quantum algorithm and its application to combinatorial optimization problem," in *Proc. Congr. Evol. Computation. CEC00*, vol. 2, 2000, pp. 1354–1360.
[59] Y. Wang, X.-Y. Feng, Y.-X. Huang, D.-B. Pu, W.-G. Zhou, Y.-C. Liang, and C.-G. Zhou, "A novel quantum swarm evolutionary algorithm and its applications," *Neurocomputing*, vol. 70, nos. 4–6, pp. 633–640, Jan. 2007.
[60] R. S. Pavithr and Gursaran, "Quantum inspired social evolution (QSE) algorithm for 0–1 knapsack problem," *Swarm Evol. Comput.*, vol. 29, pp. 33–46, Aug. 2016.

[61] O. Montiel, Y. Rubio, C. Olvera, and A. Rivera, "Quantum-inspired acromyrmex evolutionary algorithm," *Sci. Rep.*, vol. 9, no. 1, p. 12181, Aug. 2019.
[62] Z. Chen and P. Luo, "QISA: Incorporating quantum computation into simulated annealing for optimization problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2011, pp. 2480–2487.
[63] G. Acampora and A. Vitiello, "Implementing evolutionary optimization on actual quantum processors," *Inf. Sci.*, vol. 575, pp. 542–562, Oct. 2021.
[64] C. Durr and P. Hoyer, "A quantum algorithm for finding the minimum," 1996, *arXiv:9607014v2*.
[65] K. Kaparis, A. N. Letchford, and I. Mourtos, "Generalised 2-circulant inequalities for the max-cut problem," *Oper. Res. Lett.*, vol. 50, no. 2, pp. 122–128, Mar. 2022.
[66] R. Hassin and N. Leshenko, "Greedy differencing edge-contraction heuristic for the max-cut problem," *Oper. Res. Lett.*, vol. 49, no. 3, pp. 320–325, May 2021.
[67] C. Lu and Z. Deng, "A branch-and-bound algorithm for solving max-k-cut problem," *J. Global Optim.*, vol. 81, no. 2, pp. 367–389, Feb. 2021.
[68] S. S. Gill, A. Kumar, H. Singh, M. Singh, K. Kaur, M. Usman, and R. Buyya, "Quantum computing: A taxonomy, systematic review and future directions," *Software, Pract. Exper.*, vol. 52, no. 1, pp. 66–114, Oct. 2021.
[69] H. Al-Hraishawi, J. U. Rehman, M. Razavi, and S. Chatzinotas, "Characterizing and utilizing the interplay between quantum technologies and non-terrestrial networks," *IEEE Open J. Commun. Soc.*, vol. 5, pp. 1937–1957, 2024.
[70] J. Kelly, "State preservation by repetitive error detection in a superconducting quantum circuit," *Nature*, vol. 519, no. 7541, pp. 66–69, Mar. 2015.
[71] Y. Rubio, C. Olvera, and O. Montiel, "Quantum-inspired evolutionary algorithms on IBM quantum experience," *Eng. Lett.*, vol. 29, no. 4, pp. 1573–1584, Nov. 2021.
[72] E. T. Campbell, B. M. Terhal, and C. Vuillot, "Roads towards fault-tolerant universal quantum computation," *Nature*, vol. 549, no. 7671, pp. 172–179, Sep. 2017.
[73] A. Yu. Kitaev, "Quantum measurements and the Abelian stabilizer problem," 1995, *arXiv: quant-ph/9511026*.
[74] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proc. 28th Annu. ACM Symp. Theory Comput.*, 1996, pp. 212–219.
[75] F. M. Creevey, C. D. Hill, and L. C. L. Hollenberg, "GASP: A genetic algorithm for state preparation on quantum computers," *Sci. Rep.*, vol. 13, no. 1, p. 11956, Jul. 2023.
[76] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, and P. J. Coles, "Variational quantum algorithms," *Nat. Rev. Phys.*, vol. 3, no. 9, pp. 625–644, Aug. 2021.
[77] L. Arufe, M. A. González, A. Oddi, R. Rasconi, and R. Varela, "Quantum circuit compilation by genetic algorithm for quantum approximate optimization algorithm applied to MaxCut problem," *Swarm Evol. Comput.*, vol. 69, Mar. 2022, Art. no. 101030.
[78] T. Rindell, B. Yenilen, N. Halonen, A. Pönni, I. Tittonen, and M. Raasakka, "Exploring the optimality of approximate state preparation quantum circuits with a genetic algorithm," *Phys. Lett. A*, vol. 475, Jul. 2023, Art. no. 128860.
[79] A. G. Rattew, S. Hu, M. Pistoia, R. Chen, and S. Wood, "A domain-agnostic, noise-resistant, hardware-efficient evolutionary variational quantum eigensolver," 2019, *arXiv:1910.09694*.
[80] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, "A variational eigenvalue solver on a photonic quantum processor," *Nature Commun.*, vol. 5, no. 1, p. 4213, Jul. 2014.
[81] J. ur Rehman, H. Al-Hraishawi, and S. Chatzinotas, "Quantum approximate optimization algorithm for knapsack resource allocation problems in communication systems," in *Proc. IEEE Int. Conf. Commun.*, May 2023, pp. 2674–2679.
[82] H. Al-Hraishawi, J. U. Rehman, and S. Chatzinotas, "Quantum optimization algorithm for LEO satellite communications based on cell-free massive MIMO," in *Proc. IEEE Int. Conf. Commun. Workshops*, May 2023, pp. 1759–1764.
[83] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," 2014, *arXiv:1411.4028*.
[84] A. Pérez-Salinas, A. Cervera-Lierta, E. Gil-Fuster, and J. I. Latorre, "Data re-uploading for a universal quantum classifier," *Quantum*, vol. 4, p. 226, Feb. 2020.
[85] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195–202, Sep. 2017.
[86] S. Y.-C. Chen, C.-M. Huang, C.-W. Hsing, H.-S. Goan, and Y.-J. Kao, "Variational quantum reinforcement learning via evolutionary optimization," *Mach. Learn., Sci. Technol.*, vol. 3, no. 1, Feb. 2022, Art. no. 015025.
[87] D. Faílde, J. D. Viqueira, M. M. Juane, and A. Gómez, "Using differential evolution to avoid local minima in variational quantum algorithms," *Sci. Rep.*, vol. 13, no. 1, p. 16230, Sep. 2023.
[88] G. Acampora, A. Chiatto, and A. Vitiello, "Genetic algorithms as classical optimizer for the quantum approximate optimization algorithm," *Appl. Soft Comput.*, vol. 142, Jul. 2023, Art. no. 110296.
[89] S. Chand, H. K. Singh, T. Ray, and M. Ryan, "Rollout based heuristics for the quantum circuit compilation problem," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2019, pp. 974–981.
[90] A. Oddi and R. Rasconi, "Greedy randomized search for scalable compilation of quantum circuits," in *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*. Cham, Switzerland: Springer, 2018, pp. 446–461.
[91] S. Wang, E. Fontana, M. Cerezo, K. Sharma, A. Sone, L. Cincio, and P. J. Coles, "Noise-induced barren Plateaus in variational quantum algorithms," *Nature Commun.*, vol. 12, no. 1, p. 6961, Nov. 2021.
[92] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, "Cost function dependent barren Plateaus in shallow parametrized quantum circuits," *Nature Commun.*, vol. 12, no. 1, p. 1791, Mar. 2021.
[93] E. R. Anschuetz and B. T. Kiani, "Quantum variational algorithms are swamped with traps," *Nature Commun.*, vol. 13, no. 1, p. 7760, Dec. 2022.
[94] M. Ragone, B. N. Bakalov, F. Sauvage, A. F. Kemper, C. O. Marrero, M. Larocca, and M. Cerezo, "A lie algebraic theory of barren Plateaus for deep parameterized quantum circuits," 2023, *arXiv:2309.09342*.
[95] M. Cerezo, M. Larocca, and D. García-Martín, "Does provable absence of barren Plateaus imply classical simulability? Or, why we need to rethink variational quantum computing," *arXiv:2312.09121*, Dec. 2023.
[96] J. Nádori, G. Morse, Z. Majnay-Takács, Z. Zimborás, and P. Rakyta, "Line search strategy for navigating through barren Plateaus in quantum circuit training," 2024, *arXiv:2402.05227*.
[97] L. Franken, B. Georgiev, S. Mucke, M. Wolter, R. Heese, C. Bauckhage, and N. Piatkowski, "Quantum circuit evolution on NISQ devices," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2022, pp. 1–8.
[98] X. Yang, J. Li, and X. Peng, "An improved differential evolution algorithm for learning high-fidelity quantum controls," *Sci. Bull.*, vol. 64, no. 19, pp. 1402–1408, Oct. 2019.

JUNAID UR REHMAN received the B.S. degree in electrical engineering from the National University of Sciences and Technology (NUST), Islamabad, Pakistan, in 2013, and the Ph.D. degree in electronic engineering from Kyung Hee University (KHU), Yongin-si, South Korea, in 2019.

He was a Researcher (Postdoctoral Fellow/Research Associate/Research Scientist) with various research groups, including groups with KHU, Korea Institute of Science and Technology, and the Signal Processing and Communications (SIGCOM) Group, University of Luxembourg. Currently, he is an Assistant Professor with the Department of Electrical Engineering, King Fahd University of Petroleum and Minerals, Saudi Arabia. His research interests include quantum information sciences that include quantum communications, quantum computing, and quantum sensing.

Dr. Rehman has served as a technical program committee member for multiple IEEE conferences, including the IEEE International Conference on Communications (ICC) and the IEEE Global Communications Conference (Globecom). He has reviewed for numerous IEEE conferences and journals. He was an Exemplary Reviewer of IEEE Wireless Communications Letters, in 2022.

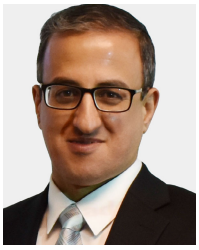
MUHAMMAD SHOHIBUL ULUM received the B.S. degree in electrical engineering from Bandung Institute of Technology, Bandung, Indonesia, in 2020. He is currently pursuing the Ph.D. degree with the Department of Electronics and Information Convergence Engineering, Kyung Hee University, South Korea. His research interests include quantum information science, quantum computing, and quantum metrology.

ABDURRAHMAN WACHID SHAFFAR received the B.S. degree from the Department of Physics, Universitas Gadjah Mada (UGM), Yogyakarta, Indonesia, in 2019. He is currently pursuing the Ph.D. degree with the Department of Electronics and Information Convergence Engineering, Kyung Hee University, South Korea. His research interests include quantum information science, quantum computing, and quantum communication.

AMIRUL ADLIL HAKIM received the B.S. degree from the Department of Physics, Universitas Gadjah Mada (UGM), Yogyakarta, Indonesia, in 2022. He is currently pursuing the Ph.D. degree with the Department of Electronics and Information Convergence Engineering, Kyung Hee University, South Korea. His research interests include quantum information science, quantum simulation, and quantum machine learning.

MUJIRIN received the B.S. degree in physics from Universitas Gadjah Mada, Indonesia, in 2015, and the M.Sc. degree in physics from Universitas Indonesia, Indonesia, in 2019. He is currently pursuing the Ph.D. degree in quantum information science with the Department of Electronics and Information Convergence Engineering, Kyung Hee University (KHU), South Korea. His research interests include quantum information science, quantum communication, and artificial intelligence.

ZAID ABDULLAH (Member, IEEE) received the Ph.D. degree in communications and signal processing from Newcastle University, Newcastle upon Tyne, U.K., in 2019. From 2019 to 2021, he was a Postdoctoral Researcher with the School of Engineering, University of Leicester, U.K. He is currently a Research Scientist with SnT, University of Luxembourg, Luxembourg. His research interests include beyond 5G networks, communications signal processing, metaheuristic optimization, algorithm design, and artificial intelligence.



HAYDER AL-HRAISHAWI (Senior Member, IEEE) received the Ph.D. degree from the Department of Electrical and Computer Engineering, Southern Illinois University Carbondale, USA, in 2017. Currently, he is an Assistant Professor with the Department of Electrical Engineering, University of South Florida, USA. Previously, he was a Research Scientist with the Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg. With over five years of industry experience in engineering and leadership roles, he has held positions at leading telecom companies, including Motorola Solutions, Huawei Technologies, and Nokia Networks. His research interests include terrestrial and non-terrestrial communication networks and signal processing, with a focus on spectrum-sharing, resource allocation for massive MIMO systems, reconfigurable intelligent surfaces, and satellite communications. He has also contributed as a Technical Program Committee Member for various IEEE conferences, including ICC, GLOBECOM, and WCNC. Since 2019, he has been an Associate Editor on the editorial board of IEEE Access and was recognized as an Exemplary Reviewer for IEEE Transactions on Communications, in 2022.

SYMEON CHATZINOTAS (Fellow, IEEE) received the M.Eng. degree in telecommunications from the Aristotle University of Thessaloniki, Greece, in 2003, and the M.Sc. and Ph.D. degrees in electronic engineering from the University of Surrey, U.K., in 2006 and 2009, respectively.

He is currently a Full Professor/Chief Scientist I and the Head of the Research Group SIGCOM, Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg. In parallel, he is an Adjunct Professor with the Department of Electronic Systems, Norwegian University of Science and Technology, and a Collaborating Scholar with the Institute of Informatics and Telecommunications, National Center for Scientific Research "Demokritos." In the past, he has lectured as a Visiting Professor with the University of Parma, Italy, and contributed in numerous research and development projects for the Institute of Telematics and Informatics, Center of Research and Technology Hellas, and the Mobile Communications Research Group, Center of Communication Systems Research, University of Surrey. He has authored more than 800 technical papers in refereed international journals, conferences, and scientific books.

Dr. Chatzinotas received numerous awards and recognitions, including the IEEE Fellowship and the IEEE Distinguished Contributions Award. He is on the editorial board of IEEE Transactions on Communications, IEEE Open Journal of Vehicular Technology, and *International Journal of Satellite Communications and Networking*.

HYUNDONG SHIN (Fellow, IEEE) received the B.S. degree in electronics engineering from Kyung Hee University (KHU), Yongin-si, South Korea, in 1999, and the M.S. and Ph.D. degrees in electrical engineering from Seoul National University, Seoul, South Korea, in 2001 and 2004, respectively. During his postdoctoral research with Massachusetts Institute of Technology (MIT), from 2004 to 2006, he was with the Laboratory for Information Decision Systems (LIDS). In 2006, he joined KHU, where he is currently a Professor with the Department of Electronic Engineering. His research interests include quantum information science, wireless communication, and machine intelligence. He received the IEEE Communications Society's Guglielmo Marconi Prize Paper Award and the William R. Bennett Prize Paper Award. He served as the Publicity Co-Chair for IEEE PIMRC and the Technical Program Co-Chair for IEEE WCNC and IEEE GLOBECOM. He was an Editor of IEEE Transactions on Wireless Communications and IEEE Communications Letters.