

Article

# Quantum Approximation for Wireless Scheduling

Jaeho Choi <sup>1</sup> , Seunghyeok Oh <sup>2</sup> and Joongheon Kim <sup>2,\*</sup> 

<sup>1</sup> School of Computer Science and Engineering, Chung-Ang University, Seoul 06974, Korea; bearsteak@cau.ac.kr

<sup>2</sup> School of Electrical Engineering, Korea University, Seoul 02841, Korea; seunghyeokoh@korea.ac.kr

\* Correspondence: joongheon@korea.ac.kr; Tel.: +82-2-3290-3223

Received: 2 September 2020; Accepted: 6 October 2020; Published: 13 October 2020



**Abstract:** This paper proposes an application algorithm based on a quantum approximate optimization algorithm (QAOA) for wireless scheduling problems. QAOA is one of the promising hybrid quantum-classical algorithms to solve combinatorial optimization problems and it provides great approximate solutions to non-deterministic polynomial-time (NP) hard problems. QAOA maps the given problem into Hilbert space, and then it generates the Hamiltonian for the given objective and constraint. Then, QAOA finds proper parameters from the classical optimization loop in order to optimize the expectation value of the generated Hamiltonian. Based on the parameters, the optimal solution to the given problem can be obtained from the optimum of the expectation value of the Hamiltonian. Inspired by QAOA, a quantum approximate optimization for scheduling (QAOS) algorithm is proposed. The proposed QAOS designs the Hamiltonian of the wireless scheduling problem which is formulated by the maximum weight independent set (MWIS). The designed Hamiltonian is converted into a unitary operator and implemented as a quantum gate operation. After that, the iterative QAOS sequence solves the wireless scheduling problem. The novelty of QAOS is verified with simulation results implemented via Cirq and TensorFlow-Quantum.

**Keywords:** quantum approximate optimization algorithm (QAOA); maximum weight independent set (MWIS); NP-hard; wireless scheduling; quantum application

## 1. Introduction

Nowadays, quantum computing and communications have received a lot of attention from academia and industry research communities. In particular, quantum computing-based non-deterministic polynomial-time (NP) hard problem solving is of great interest [1–4]. Among the available methods, the quantum approximate optimization algorithm (QAOA) is one of the well-known quantum computing-based optimization solvers, and it has been verified that the QAOA outperforms others in many combinatorial problems that are closely related to wireless scheduling problems [5–9]. Therefore, it is obvious that quantum computing can be used for various communications applications [10–13].

In this paper, a wireless scheduling problem is formulated with maximum weight independent set (MWIS) formulation, where the weight is defined as the queue backlog to be transmitted over wireless channels [14–17]. Due to the fact that the MWIS problem is an NP-hard problem, heuristic algorithms are desired; thus, a QAOA application algorithm, quantum approximate optimization for scheduling (QAOS), is designed to solve MWIS-based wireless scheduling problems.

The proposed QAOS works as follows. First of all, the objective function and constraint functions are formulated for MWIS. Next, the corresponding objective Hamiltonian and constraint Hamiltonian are designed, which map the objective function and the constraint function, respectively; then, the problem Hamiltonian, which should be optimized, is formulated as the form of linear

combinations of the objective Hamiltonian and constraint Hamiltonian. In addition, the mixing Hamiltonian is formulated using the Pauli-X operator, which provides the number of different cases via bit flip. Based on the definitions of the problem Hamiltonian and the mixing Hamiltonian, two corresponding unitary operators, i.e., the problem operator and mixing operator, can be defined, respectively; then, the parameterized state can be generated by alternately applying the two unitary operators. The sample solutions can be obtained by the measurement of the expectation value of the problem Hamiltonian on the parameterized state, and the parameters can be optimized in a classical optimization loop, using—for example—stochastic gradient methods. Here, the measurement is used to project a quantum state into one of the eigenstates (=eigenkets) of the problem operator, according to the Copenhagen interpretation [18]. Finally, the optimal solution of the MWIS problem can be obtained by the measurement of the expectation value of the problem Hamiltonian on the state generated by optimal parameters. As verified in performance evaluation, the QAOS outperforms the random search and greedy search.

QAOS is a novel attempt to carry out application research on wireless communication via QAOA. In the noisy intermediate-scale quantum (NISQ) era, research on hybrid quantum-classical algorithms such as QAOS is essential [6]. However, the research on QAOA-based applications such as QAOS is still in its infancy. Therefore, it is reasonable to increase the utilization of QAOA through convergence and transformation in various fields. From this point of view, QAOS, the novel attempt via QAOA, is very encouraging.

The rest of this paper is organized as follows. Section 2 presents the preliminary knowledge. Section 3 introduces MWIS-based wireless scheduling modeling. Section 4 presents the details of the proposed QAOS algorithm, and the performance is evaluated in Section 5. Finally, Section 6 concludes the paper.

## 2. Preliminaries

Prior to problem modeling, this section briefly explains bra–ket notation, basic quantum gates, and QAOA [5].

### 2.1. Bra–Ket Notation

In quantum computing, the bra–ket notation is generally used to represent qubit states (or quantum states). It is also called the Dirac notation, as well as the notation for observable vectors in Hilbert spaces. A ket and a bra can represent the column and row vectors, respectively. Thus, single qubit states, i.e.,  $|0\rangle$  and  $|1\rangle$ , are represented as follows:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \text{ and } |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad (1)$$

$$\text{and also } |0\rangle = \langle 0|^\dagger = \begin{bmatrix} 1 & 0 \end{bmatrix}^\dagger, \quad (2)$$

$$|1\rangle = \langle 1|^\dagger = \begin{bmatrix} 0 & 1 \end{bmatrix}^\dagger. \quad (3)$$

Note that  $\dagger$  means Hermitian transpose. Accordingly, the superposition state of a single qubit can be represented as follows:

$$c_1 |0\rangle + c_2 |1\rangle = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}, \quad (4)$$

where  $c_1$  and  $c_2$  are probability amplitudes that are complex numbers [19].

### 2.2. Basic Quantum Gates

This section introduces several commonly used basic quantum gates (or operators) that represent the single-qubit or 2-qubit operations [19]. The Hadamard gate  $H$ , Pauli-X gate  $X$ , Pauli-Y gate  $Y$ , and Pauli-Z gate  $Z$  are represented as follows:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \text{ and } Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{5}$$

The rotation-X gate  $RX(\theta)$ , rotation-Y gate  $RY(\theta)$ , and rotation-Z gate  $RZ(\theta)$  are represented as follows:

$$RX(\theta) = \begin{bmatrix} \cos \frac{\theta}{2} & -i \sin \frac{\theta}{2} \\ -i \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix}, RY(\theta) = \begin{bmatrix} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix}, \text{ and } RZ(\theta) = \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix}, \tag{6}$$

where  $\theta$  is the angle. The controlled-NOT gate  $CNOT$  and swap gate  $SWAP$  are represented as follows:

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \text{ and } SWAP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{7}$$

In this paper,  $H, X, Z, RX(\theta), RZ(\theta)$ , and  $CNOT$  are used.

### 2.3. Quantum Approximate Optimization Algorithm (QAOA)

QAOA is one of the well-known NISQ optimization algorithms to combat combinatorial problems [5–8]. QAOA formulates  $H_P$  (i.e., problem Hamiltonian) and  $H_M$  (i.e., mixing Hamiltonian) from the objective function  $f(y)$ ; and then generates the parameterized states  $|\gamma, \beta\rangle$  by alternately applying the  $H_P$  and  $H_M$  on initial state  $|s\rangle$ . Here,  $f(y), H_P|y\rangle, H_M$ , and  $|\gamma, \beta\rangle$  are defined as follows:

$$f(y) \triangleq f(y_1, y_2, \dots, y_n), \tag{8}$$

$$H_P|y\rangle \triangleq f(y)|y\rangle, \tag{9}$$

$$H_M \triangleq \sum_{k=1}^n X_k, \tag{10}$$

$$|\gamma, \beta\rangle \triangleq e^{-i\beta_p H_M} e^{-i\gamma_p H_P} \dots e^{-i\beta_2 H_M} e^{-i\gamma_2 H_P} e^{-i\beta_1 H_M} e^{-i\gamma_1 H_P} |s\rangle, \tag{11}$$

where  $n \in \mathbb{Z}^+, p \in \mathbb{Z}^+$ , and  $X_k$  is the Pauli-X operator applying on the  $k$ th qubit;  $\gamma$  and  $\beta$  are learnable parameters that tune the result. Note that  $H_P$  encodes  $f(y)$  in Equation (9), operating diagonally in the  $n$ -qubit quantum computational basis states [20].

In QAOA, through iterative measurement of  $|\gamma, \beta\rangle$ , the expectation value of  $H_P$  should be taken; then, eventually, the samples of  $f(y)$  should be computed as follows [5]:

$$\langle f(y) \rangle_{\gamma, \beta} = \langle \gamma, \beta | H_P | \gamma, \beta \rangle. \tag{12}$$

The optimal values of the parameters  $\gamma$  and  $\beta$  can be obtained via classical numerical optimization methods such as gradient descent [21,22]. Therefore, the solution can be computed from Equation (12) via the parameters obtained. Thus, QAOA is a hybrid quantum-classical optimization algorithm in which proper Hamiltonian design and the discovery of good parameters in a classical optimization loop are key [20,23,24].

### 3. Wireless Scheduling Modeling Using Maximum Weight Independent Set (MWIS)

Suppose a wireless network consists of a set of one-hop links under very high transmission power using millimeter wave [14]. Due to the high data transmission rate, the queue backlog should always be cleared; otherwise, the signal decoding always fails because of extremely high interference. For the scheduling, a conflict graph is organized where the set of vertices is (the links) and two vertices are connected by an edge if the corresponding links suffer from interference. The conflict graph can be formulated by its adjacency matrix, whose  $\mathcal{E}_{(i,j)}$  are defined as follows:

$$\mathcal{E}_{(i,j)} = \begin{cases} 1, & \text{if } l_i \text{ interferes with } l_j \text{ where} \\ & l_i \in \mathcal{L}, l_j \in \mathcal{L}, \text{ and } i \neq j, \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

For wireless network scheduling, the objective is for finding the set of links (i.e., nodes of the conflict graph) where two adjacent links connected via edges cannot be simultaneously selected because the two adjacent connected links are interfering with each other. This is equivalent to the case which maximizes the summation of weights of all possible independent sets in a given conflict graph. Thus, it is obvious that wireless network scheduling can be formulated with MWIS as follows:

$$\max : \quad \sum_{\forall l_k \in \mathcal{L}} w_k \mathcal{I}_k, \quad (14)$$

$$\text{s.t.} \quad \mathcal{I}_i + \mathcal{I}_j + \mathcal{E}_{(i,j)} \leq 2, \forall l_i \in \mathcal{L}, \forall l_j \in \mathcal{L}, \quad (15)$$

$$\mathcal{I}_i \in \{0, 1\}, \forall l_i \in \mathcal{L}, \quad (16)$$

$$\text{where} \quad \mathcal{I}_i = \begin{cases} 1, & \text{if } l_i \text{ is scheduled where } l_i \in \mathcal{L}, \\ 0, & \text{otherwise.} \end{cases} \quad (17)$$

Here,  $w_k$  is a positive integer weight at  $\forall l_k \in \mathcal{L}$ . The above formulation ensures that conflicting links are not scheduled simultaneously: if  $\mathcal{E}_{(i,j)} = 0$  (no edge between  $l_i$  and  $l_j$ ), then  $\mathcal{I}_i + \mathcal{I}_j \leq 2$ , i.e., both indicator functions can be 1. In contrast, if  $\mathcal{E}_{(i,j)} = 1$ ,  $\mathcal{I}_i + \mathcal{I}_j \leq 1$ , i.e., at most, one of the two indicators can be 1. In wireless communication research, the  $w_k$  where  $\forall l_k \in \mathcal{L}$  is usually considered as the transmission queue backlog which should be processed when the link is scheduled. More details are can be found in [14].

### 4. Quantum Approximate Optimization for Scheduling (QAOS)

In this section, Hamiltonians of QAOA are designed based on the scheduling model in Section 3; then, the quantum approximate optimization for scheduling (QAOS) algorithm is proposed by applying the designed Hamiltonian to QAOA.

#### 4.1. Design of the Problem Hamiltonian

The problem Hamiltonian  $H_P$  is designed by a linear combination of the objective Hamiltonian  $H_O$  and the constraint Hamiltonian  $H_C$ . The objectives and constraints of the problem are contained by  $H_O$  and  $H_C$ , respectively.

##### 4.1.1. Design of the Objective Hamiltonian

Suppose that a basic Boolean function  $B_1(x)$  exists as follows:

$$B_1(x) = x \text{ where } x \in \{0, 1\}. \quad (18)$$

Due to quantum Fourier expansion, Equation (18) can be mapped to Boolean Hamiltonian  $H_{B_1}$  where  $I$  and  $Z$  are an identity operator and the Pauli-Z operator, respectively [25]:

$$H_{B_1} = \frac{1}{2}(I - Z). \tag{19}$$

According to Equations (18) and (19), the objective function Equation (14) can be mapped to the following Hamiltonian:

$$H_{O'} = \sum_{\forall I_k \in \mathcal{L}} \frac{1}{2} w_k (I - Z_k), \tag{20}$$

where  $Z_k$  is the Pauli-Z operator applied to  $\mathcal{I}_k$ . Since  $H_{O'}$  is mapped from the objective function Equation (14), which should be maximized,  $H_{O'}$  should also be maximized. Therefore, the objective Hamiltonian  $H_O$  should be minimized is as follows:

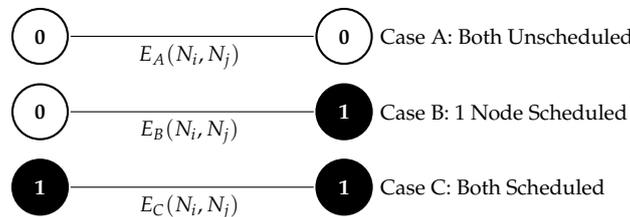
$$H_O = \sum_{\forall I_k \in \mathcal{L}} \frac{1}{2} w_k Z_k. \tag{21}$$

#### 4.1.2. Design of the Constraint Hamiltonian

In the MWIS-based wireless scheduling problem, a banned event is a case where both adjacent nodes of the conflict graph are scheduled, as shown in *Case C* of Figure 1. If the weights of  $N_i$  and  $N_j$  in *Case C* are defined as  $W_{N_i}$  and  $W_{N_j}$ , respectively; then the constraint function  $C'(i, j)$ , which counts the banned events, can be represented as follows:

$$C'(i, j) = \sum_{i=1}^n \sum_{j=1}^n (W_{N_i} + W_{N_j}) |E_C(N_i, N_j)| \text{ where } i > j. \tag{22}$$

Here,  $n$  is the number of nodes and  $|E_C(N_i, N_j)|$  is the number of  $E_C(N_i, N_j)$ ;  $i > j$  is a condition to avoid the duplication of the same edge.



**Figure 1.** The number of possible cases when a single edge exists between two nodes in the conflict graph. The scheduled and unscheduled nodes have states  $|1\rangle$  and  $|0\rangle$ , respectively.  $N_i$  and  $N_j$  represent arbitrary nodes, and  $E_A(N_i, N_j)$ ,  $E_B(N_i, N_j)$ , and  $E_C(N_i, N_j)$  represent edges in each case.

According to Equations (13)–(17),  $C'(i, j)$  can be redefined as  $C(i, j)$  with symbols in Section 3 as follows:

$$\begin{aligned} C(i, j) &= \sum_{\forall I_i \in \mathcal{L}} \sum_{\forall I_j \in \mathcal{L}} (w_i + w_j) \mathcal{E}_{(i,j)} \text{ where } i > j \\ &= \sum_{\forall I_i \in \mathcal{L}} \sum_{\forall I_j \in \mathcal{L}} (w_i + w_j) (\mathcal{I}_i \wedge \mathcal{I}_j) \text{ where } i > j. \end{aligned} \tag{23}$$

Here,  $\wedge$  is a Boolean AND operator and  $C(i, j)$ , which counts the banned events, must be 0 or the minimum value. Due to quantum Fourier expansion, the AND Boolean function  $B_2(x_1, x_2)$  can be mapped to the following Boolean Hamiltonian  $H_{B_2}$  [25]:

$$B_2(x_1, x_2) = x_1 \wedge x_2 \text{ where} \\ x_1 \in \{0, 1\} \text{ and } x_2 \in \{0, 1\}, \tag{24}$$

$$H_{B_2} = \frac{1}{4}(I - Z_1 - Z_2 + Z_1Z_2), \tag{25}$$

where  $Z_1$  and  $Z_2$  are the Pauli-Z operators applying on  $x_1$  and  $x_2$ , respectively.

According to Equations (24) and (25), the constraint function Equation (23) can be represented as following Hamiltonian:

$$H_{C'} = \sum_{\forall i \in \mathcal{L}} \sum_{\forall j \in \mathcal{L}} \frac{1}{4}(w_i + w_j)(I - Z_i - Z_j + Z_iZ_j) \text{ where } i > j. \tag{26}$$

Here,  $Z_i$  and  $Z_j$  are the Pauli-Z operators applied to  $\mathcal{I}_i$  and  $\mathcal{I}_j$ , respectively. Since  $C(i, j)$  must be 0 or the minimum value,  $H_{C'}$ , mapped from  $C(i, j)$ , should be minimized. Therefore, the constraint Hamiltonian  $H_C$ , which is a simplified form of  $H_{C'}$ , is as follows:

$$H_C = \sum_{\forall i \in \mathcal{L}} \sum_{\forall j \in \mathcal{L}} -\frac{1}{4}(w_i + w_j)(Z_i + Z_j - Z_iZ_j) \text{ where } i > j. \tag{27}$$

Based on the definitions of  $H_O$  and  $H_C$ , the problem Hamiltonian  $H_P$  can be defined as follows:

$$H_P = H_O + \rho H_C, \tag{28}$$

where  $\rho \geq 1$  is the penalty rate, which indicates the rate at which  $H_C$  affects  $H_P$  compared to  $H_O$  in the implementation.

#### 4.2. Design of the Mixing Hamiltonian

The mixing Hamiltonian, denoted by  $H_M$ , generates a variety of cases that can appear in the problem. MWIS can be formulated by a binary bit string that represents a set of nodes (e.g., |1010101>); thus, various cases can be created by flipping the state of each node, represented by |0> or |1>. The bit-flip can be handled by the Pauli-X operator, thus  $H_M$  is as follows:

$$H_M = \sum_{\forall l_k \in \mathcal{L}} X_k. \tag{29}$$

#### 4.3. Apply to QAOA Sequence

The application of the designed Hamiltonian to the QAOA sequence starts to occur when the design of Hamiltonians, i.e.,  $H_P$  and  $H_M$ , are completed. First, the parameterized state  $|\gamma, \beta\rangle$  can be generated by applying  $H_P$  and  $H_M$ , defined in Equations (21), (27), (28) and (29), to (11). Here, the initial state  $|s\rangle$  is set to the equivalent superposition state using the Hadamard gates. The expectation value of  $H_P$  can be measured on the generated parameterized state  $|\gamma, \beta\rangle$ . The parameters  $\gamma$  and  $\beta$  are iteratively updated in a classical optimization loop. When the QAOA sequence terminates, the optimal parameters  $\gamma_{opt}$  and  $\beta_{opt}$  are obtained. Thus, the scheduling solution can be obtained by the measurement of the expectation value of  $H_P$  on the optimal state  $|\gamma_{opt}, \beta_{opt}\rangle$  as follows:

$$\langle F \rangle = \langle \gamma_{opt}, \beta_{opt} | H_P | \gamma_{opt}, \beta_{opt} \rangle, \tag{30}$$

where  $\langle F \rangle$  is the expectation value of the objective function Equation (14) over the returned solution samples.

## 5. Performance Evaluation

The proposed QAOS algorithm is implemented using Cirq and TensorFlow-Quantum, which were developed for the NISQ algorithm and quantum machine learning computation [26].

### 5.1. Software Implementation

The application of the quantum gates, the basic units of the quantum circuit, is expressed by unitary operators. Based on the definitions of Hamiltonians in Section 4, the objective operator  $U_O(\gamma_\zeta)$ , constraint operator  $U_C(\gamma_\zeta)$ , problem operator  $U_P(\gamma_\zeta)$ , and mixing operator  $U_M(\beta_\zeta)$ , which are unitary operators, can be defined as follows:

$$U_O(\gamma_\zeta) = e^{-i\gamma_\zeta H_O}, \tag{31}$$

$$U_C(\gamma_\zeta) = e^{-i\gamma_\zeta \rho H_C}, \tag{32}$$

$$U_P(\gamma_\zeta) = U_O(\gamma_\zeta)U_C(\gamma_\zeta) = e^{-i\gamma_\zeta(H_O + \rho H_C)}, \tag{33}$$

$$U_M(\beta_\zeta) = e^{-i\beta_\zeta H_M}, \tag{34}$$

where  $\gamma_\zeta$  and  $\beta_\zeta$  are in  $\gamma \equiv \gamma_1 \cdots \gamma_p$  and  $\beta \equiv \beta_1 \cdots \beta_p$ , respectively:  $\zeta \in \mathbb{Z}^+$  and  $1 \leq \zeta \leq p$ . Note that the implementation of  $U_P(\gamma_\zeta)$  and  $U_M(\beta_\zeta)$  is the core of QAOS implementation.

In Figure 2, `cirq.rz()` and `cirq.CNOT()` are used for the implementation of  $U_P(\gamma_\zeta)$ . Note that, `cirq.rz()` and `cirq.CNOT()` represent the rotation-Z gate and the controlled-NOT gate, respectively. In addition,  $U_M(\beta_\zeta)$  is implemented using `cirq.rx()`, which represents the rotation-X gate.

The part that finds the optimal parameters using Keras (one of the well-known open-source deep learning computation libraries) is shown in Figure 2, from line 29 to line 36. Here, the parametrized quantum circuit (PQC) layer provides the auto-management of variables in the parameterized circuit. In this model, Adam is used as a gradient-based optimizer [27,28].

### 5.2. Experiments

This experiment demonstrates the possibility of a novel quantum approach via QAOS by focusing on simple and light methods to wireless scheduling problems. In the classical approaches, a message-passing algorithm consisting of linear programming relaxation, maximum product method, and maximum a posteriori estimation is mainly used to solve the MWIS-based wireless scheduling problem [14,16,17]. However, if the network requires a more simple and lighter algorithm, the random and greedy-based algorithms are also used for wireless scheduling [29,30]. Thus, the performance of the proposed QAOS, based on QAOA, one of the simple and light quantum algorithms that intuitively express the state with qubit rotation, is compared with the random search and greedy search, which are simple and light classical algorithms [31,32]. In addition, the QAOS algorithm is executed with different  $p$  value settings, where the  $p$  value means the number of alternations of  $U_P(\gamma_\zeta)$  and  $U_M(\beta_\zeta)$  in Equations (33) and (34), i.e.,  $\zeta \in \mathbb{Z}^+$  and  $1 \leq \zeta \leq p$ .

For the performance evaluation, random conflict graphs with 10 nodes are generated; then, random search, greedy search, and QAOS algorithms are performed for the given random conflict graphs. The measurement of each QAOS is performed 1000 times in each simulation (i.e., in each randomly generated conflict graph). The performance of each algorithm is quantitatively measured with  $\eta$  as follows:

$$\eta \triangleq \frac{a}{b}, \tag{35}$$

where  $a$  and  $b$  are the summations of weights of the scheduled nodes by the used algorithms and the summations of weights of the scheduled nodes by brute-force search (i.e., exhaustive search), respectively, for the given randomly generated graphs. As shown in Figure 3, the cumulative distribution functions (CDF) of  $\eta$  for each algorithm are computed.

As presented in Figure 3, QAOS algorithms with  $p \geq 8$  present a better performance than random search and greedy search, in any kind of randomly generated conflict graph. In these repeated simulations, the performances of QAOS algorithms are improved as the  $p$  value increases. In particular, the performance of the QAOS algorithm with  $p = 10$  is much better than the QAOS algorithms with  $p = 8$  and  $p = 9$ . As shown in Table 1, the QAOS algorithm with  $p = 10$  returns optimal solutions (i.e., equivalent to the solutions obtained by brute-force search) with a ratio of 69.50%. Through these results, we have verified that the proposed QAOS algorithm presents beautiful results in terms of the accuracy of the solutions.

```

1 # - QUBO Model for MWIS
2 def get_MWIS_QUBO(graph: nx.Graph, penalty_rate=1):
3     weight_set = np.array(
4         [graph.nodes(data=True)[i]['weight'] for i in range(graph.number_of_nodes())])
5     problem_QUBO={}
6     for i in range(graph.number_of_nodes()):
7         problem_QUBO[(i,)] = weight_set[i]
8         for _, j in graph.edges(i):
9             problem_QUBO[(i,)] -= (weight_set[i]+weight_set[j])*penalty_rate/2
10            if i < j:
11                problem_QUBO[(i, j)] = (weight_set[i]+weight_set[j])*penalty_rate/2
12    return problem_QUBO
13 ...
14 # - Problem Operator
15 def problem_operator(p_QUBO:dict, qubits, p, gamma):
16     key_iter=sorted(p_QUBO.keys(), key=lambda x: (len(x), x))
17     for nodes in key_iter:
18         for i in range(len(nodes)-1):
19             yield circq.CNOT(qubits[nodes[i]], qubits[nodes[i+1]])
20             yield circq.rz(gamma[p]*p_QUBO[nodes])(qubits[nodes[-1]])
21         for i in range(len(nodes)-1):
22             yield circq.CNOT(qubits[nodes[i]], qubits[nodes[i+1]])
23 ...
24 # - Mixing Operator
25 def mixing_operator(mwis_graph, qubits, p, beta):
26     for node in mwis_graph.nodes:
27         yield circq.rx(2*beta[p])(qubits[node])
28 ...
29 # - Optimal Parameter Search using Keras Model
30 model = tf.keras.Sequential()
31 model.add(tf.keras.layers.Input(shape=(), dtype=tf.dtypes.string))
32 model.add(tfq.layers.PQC(model_circuit, model_readout)) # Parameterized Quantum Circuit
33 model.add(tf.keras.layers.Lambda(correction))
34 model.compile(loss=tf.keras.losses.mean_absolute_error,
35               optimizer=tf.keras.optimizers.Adam(learning_rate=learning_rate))
36 history = model.fit(input_, optimum, epochs=epochs, verbose=0)

```

**Figure 2.** Parts of Python codes using Cirq and TensorFlow-Quantum for solving the maximum weight independent set (MWIS)-based scheduling problem.

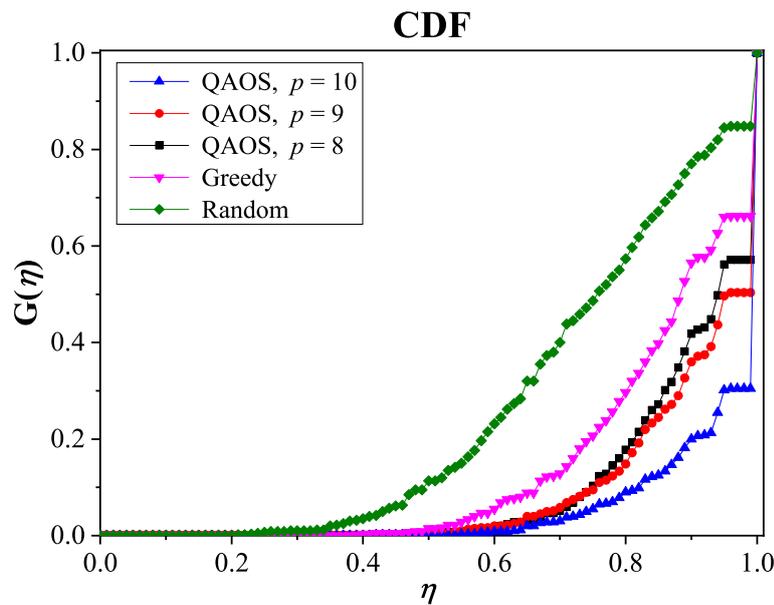


Figure 3. Performance evaluation results.  $G(\eta)$  is the cumulative distribution function (CDF) of  $\eta$ .

Table 1. Percentage of optimal solution computation.

QAOS, $p = 10$	QAOS, $p = 9$	QAOS, $p = 8$	Greedy	Random
69.50%	49.67%	42.83%	33.83%	15.17%

### 6. Concluding Remarks and Future Work

Wireless scheduling was modeled with the MWIS problem, which is one of the well-known NP-hard problems. In order to solve the MWIS problem, a QAOA-based scheduling algorithm, so-called quantum approximate optimization for scheduling (QAOS), was proposed. The proposed QAOS was implemented using Cirq and TensorFlow-Quantum. QAOS outperformed greedy search and random search in the performance evaluation on the random conflict graphs. Therefore, our novel quantum approach to the wireless scheduling problem via QAOS was meaningful.

Future research will focus on improving the performance of QAOS. In one method, introducing an error correction code to QAOS is considered. This method is expected to improve the sampling quality. Another method is to develop a new optimizer that can more accurately find the optimal parameters of QAOS. A novel optimizer is needed that is more suitable for quantum models than the mainly used optimizers such as Adam, Nelder–Mead (NM), and Broyden–Fletcher–Goldfarb–Shanno (BFGS). From the perspective of quantum machine learning, developing a novel optimizer for the parameterized quantum circuit like the QAOS circuit will be a meaningful challenge.

**Author Contributions:** J.C. was the main researcher who initiated and organized the research reported in the paper, and all authors including S.O. and J.K. were responsible for analyzing the simulation results and writing the paper. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the National Research Foundation of Korea (NRF-2019M3E4A1080391 and NRF-2019M3E3A1084054).

**Acknowledgments:** J.K. is the corresponding author of this paper.

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

## References

1. Arute, F.; Arya, K.; Babbush, R.; Bacon, D.; Bardin, J.C.; Barends, R.; Biswas, R.; Boixo, S.; Brandao, F.G.; Buell, D.A.; et al. Quantum Supremacy using a Programmable Superconducting Processor. *Nature* **2019**, *574*, 505–510. [[CrossRef](#)] [[PubMed](#)]
2. Farhi, E.; Goldstone, J.; Gutmann, S.; Lapan, J.; Lundgren, A.; Preda, D. A Quantum Adiabatic Evolution Algorithm Applied to Random Instances of an NP-Complete Problem. *Science* **2001**, *292*, 472–475. [[CrossRef](#)] [[PubMed](#)]
3. Kandala, A.; Mezzacapo, A.; Temme, K.; Takita, M.; Brink, M.; Chow, J.M.; Gambetta, J.M. Hardware-Efficient Variational Quantum Eigensolver for Small Molecules and Quantum Magnets. *Nature* **2017**, *549*, 242–246. [[CrossRef](#)] [[PubMed](#)]
4. Troyer, M.; Wiese, U.J. Computational Complexity and Fundamental Limitations to Fermionic Quantum Monte Carlo Simulations. *Phys. Rev. Lett.* **2005**, *94*, 170201. [[CrossRef](#)]
5. Farhi, E.; Goldstone, J.; Gutmann, S. A Quantum Approximate Optimization Algorithm. *arXiv* **2014**, arXiv:1411.4028.
6. Preskill, J. Quantum Computing in the NISQ Era and Beyond. *Quantum* **2018**, *2*, 79. [[CrossRef](#)]
7. Choi, J.; Oh, S.; Kim, J. The Useful Quantum Computing Techniques for Artificial Intelligence Engineers. In Proceedings of the 34th IEEE ICOIN, Barcelona, Spain, 7–10 January 2020; pp. 1–3.
8. Zhou, L.; Wang, S.T.; Choi, S.; Pichler, H.; Lukin, M.D. Quantum Approximate Optimization Algorithm: Performance, Mechanism, and Implementation on Near-Term Devices. *Phys. Rev. X* **2020**, *10*, 021067. [[CrossRef](#)]
9. Choi, J.; Kim, J. A Tutorial on Quantum Approximate Optimization Algorithm (QAOA): Fundamentals and Applications. In Proceedings of the 10th IEEE ICTC, Jeju Island, Korea, 16–18 October 2019; pp. 138–142.
10. Nawaz, S.J.; Sharma, S.K.; Wyne, S.; Patwary, M.N.; Asaduzzaman, M. Quantum Machine Learning for 6G Communication Networks: State-of-the-Art and Vision for the Future. *IEEE Access* **2019**, *7*, 46317–46350. [[CrossRef](#)]
11. Tariq, F.; Khandaker, M.R.; Wong, K.K.; Imran, M.A.; Bennis, M.; Debbah, M. A Speculative Study on 6G. *IEEE Wirel. Commun.* **2020**, *27*, 118–125. [[CrossRef](#)]
12. Viswanathan, H.; Mogensen, P.E. Communications in the 6G Era. *IEEE Access* **2020**, *8*, 57063–57074. [[CrossRef](#)]
13. Tang, F.; Kawamoto, Y.; Kato, N.; Liu, J. Future Intelligent and Secure Vehicular Network Toward 6G: Machine-Learning Approaches. *Proc. IEEE* **2020**, *108*, 292–307. [[CrossRef](#)]
14. Kim, J.; Caire, G.; Molisch, A.F. Quality-Aware Streaming and Scheduling for Device-to-Device Video Delivery. *IEEE/ACM Trans. Netw.* **2016**, *24*, 2319–2331. [[CrossRef](#)]
15. Basagni, S. Finding a Maximal Weighted Independent Set in Wireless Networks. *Telecommun. Syst.* **2001**, *18*, 155–168. [[CrossRef](#)]
16. Paschalidis, I.C.; Huang, F.; Lai, W. A Message-Passing Algorithm for Wireless Network Scheduling. *IEEE/ACM Trans. Netw.* **2015**, *23*, 1528–1541. [[CrossRef](#)] [[PubMed](#)]
17. Sanghavi, S.; Shah, D.; Willsky, A.S. Message Passing for Maximum Weight Independent Set. *IEEE Trans. Inf. Theory* **2009**, *55*, 4822–4834. [[CrossRef](#)]
18. Stapp, H.P. The Copenhagen Interpretation. *Am. J. Phys.* **1972**, *40*, 1098–1116. [[CrossRef](#)]
19. Duarte, F.J.; Taylor, T.S.; Slaten, J.C. On the Probability Amplitude of Quantum Entanglement and the Pauli Matrices. *Opt. Quantum Electron.* **2020**, *52*, 106. [[CrossRef](#)]
20. Hadfield, S.; Wang, Z.; O’Gorman, B.; Rieffel, E.G.; Venturelli, D.; Biswas, R. From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz. *Algorithms* **2019**, *12*, 34. [[CrossRef](#)]
21. Zinkevich, M.; Weimer, M.; Li, L.; Smola, A.J. Parallelized Stochastic Gradient Descent. In Proceedings of the 24th NIPS, Vancouver, BC, Canada, 6–9 December 2010; pp. 2595–2603.
22. Nawi, N.M.; Ransing, M.R.; Ransing, R.S. An Improved Learning Algorithm Based on the Broyden-Fletcher-Goldfarb-Shanno (BFGS) Method for Back Propagation Neural Networks. In Proceedings of the 6th IEEE ISDA, Jinan, China, 16–18 October 2006; pp. 152–157.
23. Streif, M.; Leib, M. Training the Quantum Approximate Optimization Algorithm without Access to a Quantum Processing Unit. *Quantum Sci. Technol.* **2020**, *5*, 034008. [[CrossRef](#)]

24. Wang, Z.; Hadfield, S.; Jiang, Z.; Rieffel, E.G. Quantum Approximate Optimization Algorithm for MaxCut: A Fermionic View. *Phys. Rev. A* **2018**, *97*, 022304. [[CrossRef](#)]
25. Hadfield, S. On the Representation of Boolean and Real Functions as Hamiltonians for Quantum Computing. *arXiv* **2018**, arXiv:1804.09130.
26. Broughton, M.; Verdon, G.; McCourt, T.; Martinez, A.; Yoo, J.; Isakov, S.V.; Massey, P.; Niu, M.Y.; Halavati, R.; Peters, E.; et al. TensorFlow Quantum: A Software Framework for Quantum Machine Learning. *arXiv* **2020**, arXiv:2003.02989.
27. Zhang, Z. Improved Adam Optimizer for Deep Neural Networks. In Proceedings of the 26th IEEE/ACM IWQoS, Banff, AB, Canada, 4–6 June 2018; pp. 1–2.
28. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
29. Feo, T.A.; Resende, M.G. Greedy Randomized Adaptive Search Procedures. *J. Glob. Optim.* **1995**, *6*, 109–133. [[CrossRef](#)]
30. Sakai, S.; Togasaki, M.; Yamazaki, K. A Note on Greedy Algorithms for the Maximum Weighted Independent Set Problem. *Discret. Appl. Math.* **2003**, *126*, 313–322. [[CrossRef](#)]
31. Farhi, E.; Harrow, A.W. Quantum Supremacy through the Quantum Approximate Optimization Algorithm. *arXiv* **2016**, arXiv:1602.07674.
32. Bravyi, S.; Gosset, D.; König, R. Quantum Advantage with Shallow Circuits. *Science* **2018**, *362*, 308–311. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).