Quantum Approximation for Multi-Scale Scheduling
Jaeho Choi, Seunghyeok Oh, and Joongheon Kim, Senior Member, IEEE

Abstract

This paper proposes a quantum approximate optimization algorithm (QAOA) method for multi-scale wireless scheduling problems. The QAOA is one of the promising hybrid quantum-classical algorithms for many applications and it provides highly accurate optimization solutions in NP-hard problems. QAOA maps the given problems into Hilbert spaces, and then it generates Hamiltonian for the given objectives and constraints. Then, QAOA finds proper parameters from classical optimization approaches in order to optimize the expectation value of generated Hamiltonian. Based on the parameters, the optimal solution to the given problem can be obtained from the optimum of the expectation value of Hamiltonian. Inspired by QAOA, a quantum approximate optimization for scheduling (QAOS) algorithm is proposed. First of all, this paper formulates a multi-scale scheduling problem using maximum weight independent set (MWIS) formulation. Then, for the given MWIS, the proposed QAOS designs the Hamiltonian of the problem. After that, the iterative QAOS sequence solves the scheduling problem. This paper verifies the novelty of the proposed QAOS via simulations implemented by Cirq and TensorFlow-Quantum.

Index Terms

Quantum Approximate Optimization Algorithm (QAOA), Maximum Weight Independent Set (MWIS), NP-Hard, Multi-Scale Wireless Networking

I. INTRODUCTION

Nowadays, quantum computing and communications have received a lot of attention by academia and industry research communities. In particular, quantum computing based NP-hard problem solving is of great interest [1]. Among them, quantum approximate optimization algorithm (QAOA) is one of the well-known quantum computing based optimization solvers [1], and it has been verified that the QAOA outperforms the others in many combinatorial problems. Based on this nature, it is obvious that quantum computing can be used for various multi-scale communications applications [2]–[4].

In this paper, a large-scale and multi-scale 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 [5]. According to the fact that the MWIS problem is

This research was supported by National Research Foundation of Korea (2019M3E4A1080391, Development of Quantum Deep Reinforcement Learning Algorithm using QAOA).

J. Choi is with the School of Computer Science and Engineering, Chung-Ang University, Seoul, Korea e-mail: jaehochoi2019@gmail.com.

S. Oh is with the Department of Physics, Chung-Ang University, Seoul, Korea e-mail: seunghyeokoh2019@gmail.com.

J. Kim is with the School of Electrical Engineering, Korea University, Seoul, Korea e-mail: joongheon@korea.ac.kr.

J. Kim is a corresponding author of this paper.
NP-hard, heuristic algorithms are desired, and in this paper, a novel QAOA-based algorithm is designed in order to solve MWIS-based wireless scheduling problems, so called quantum approximate optimization for scheduling (QAOS), in this paper.

The proposed QAOS works as follows. First of all, the objective function and constraint functions are formulated for MWIS. Next, corresponding objective Hamiltonian and constraint Hamiltonian are designed which map the objective function and the constraint function, respectively; and 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 a Pauli-$X$ operator. Based on the definitions of the problem Hamiltonian and the mixing Hamiltonian, two corresponding unitary operators, i.e., problem operator and mixing operator, can be defined, respectively; and then parameterized state can be generated by alternately applying the two unitary operators. Then, the sample solutions can be obtained by the measurement of the expectation value of problem Hamiltonian on the parameterized state, and the parameters can be optimized in a classical optimization loop. Finally, the optimal solution of the MWIS problem can be obtained by the measurement of the expectation value of problem Hamiltonian on the state generated by optimal parameters. As verified in performance evaluation, the QAOS outperforms the other algorithms, e.g., random search and greedy search.

II. PRELIMINARIES

A. Bra-ket Notation

In quantum computing, bra-ket notation is generally used to represent qubit states (or quantum states). It is also called Dirac notation as well as the notation for observable vectors in Hilbert spaces. Here, 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 presented as follows:

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

where $|0\rangle = \langle 0 |^\dagger = \begin{bmatrix} 1 & 0 \end{bmatrix}^\dagger$, $|1\rangle = \langle 1 |^\dagger = \begin{bmatrix} 0 & 1 \end{bmatrix}^\dagger$; and $\dagger$ means Hermitian transpose. Accordingly, the superposition state of a single qubit is as follows where $c_1$ and $c_2$ are probability amplitudes that are complex numbers:

$$c_1 |0\rangle + c_2 |1\rangle = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}.$$ (2)

B. Quantum Approximate Optimization Algorithm (QAOA)

QAOA is one of the well-known noisy intermediate-scale quantum (NISQ) optimization algorithms to combat combinatorial problems [6]. QAOA formulates $H_p$ (i.e., problem Hamiltonian) and $H_m$ (i.e., mixing Hamiltonian) from the optimization objective function $f(y)$; and then generates the parameterized states $|\gamma, \beta\rangle$ by alternately applying the $H_p$ and $H_m$ based on
initial state \(|s\rangle\). Here, \(f(y), H_P |y\rangle, H_M, \) and \(|\gamma, \beta\rangle\) are defined as follows.

\[
f(y) \triangleq f(y_1, y_2, \ldots, y_n),
\]

\[
H_P |y\rangle \triangleq f(y) |y\rangle,
\]

\[
H_M \triangleq \sum_{k=1}^{n} X_k,
\]

\[
|\gamma, \beta\rangle \triangleq e^{-i\beta_1 H_M} e^{-i\gamma_1 H_P} \ldots e^{-i\beta_2 H_M} e^{-i\gamma_2 H_P} \ldots e^{-i\beta_1 H_M} e^{-i\gamma_1 H_P} |s\rangle,
\]

where \(n \in \mathbb{Z}^+, \) \(p \in \mathbb{Z}^+\), and \(X_k\) is the Pauli-\(X\) operator applying on the \(k^{th}\) qubit.

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

\[
\langle f(y) \rangle_{\gamma, \beta} = \langle \gamma, \beta | H_P | \gamma, \beta \rangle.
\]

The optimal values of the parameters \(\gamma\) and \(\beta\) can be obtained by classical optimization methods, e.g., gradient descent. Therefore, the solution can be computed from (7) via the the parameters obtained. Eventually, it can be observed that QAOA is a hybrid quantum-classical optimizer which is needed the proper design of Hamiltonian; and the key is finding good parameters in the classical loop (7).

### III. Multi-Scale Scheduling Modeling using Maximum Weight Independent Set (MWIS)

Suppose a network consists of the set of one-hop links [5]. 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 \(E_{(i,j)}\) are defined as follows:

\[
E_{(i,j)} = \begin{cases} 
1, & \text{if } l_i \text{ interferes with } l_j \text{ where } l_i, l_j \in \mathcal{L}, \text{ and } i \neq j, \\
0, & \text{otherwise.}
\end{cases}
\]

For multi-scale scheduling, the objective is for finding the set of links (i.e., nodes of the conflict graph) where adjacent two connected links via edges cannot be simultaneously selected because the adjacent two connected links are interfering to 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 multi-scale scheduling can be formulated with MWIS as follows:

\[
\text{max : } \sum_{\forall l_k \in \mathcal{L}} w_k I_k,
\]

\[
\text{s.t. } I_i + I_j + E_{(i,j)} \leq 2, \forall l_i, l_j, \forall l_i, l_j \in \mathcal{L},
\]

\[
I_i \in \{0, 1\}, \forall l_i \in \mathcal{L},
\]

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

where \(w_k\) is a positive integer weight at \(\forall l_k \in \mathcal{L}\). The above formulation ensures that conflicting links are not scheduled simultaneously: If \(E_{(i,j)} = 0\) (no edge between \(l_i\) and \(l_j\)), then \(I_i + I_j \leq 2\), i.e., both indicator functions can be 1. In contrast, if \(E_{(i,j)} = 1\), \(I_i + I_j \leq 1\), i.e., at most one
Fig. 1: The number of possible cases when a single edge exists. The scheduled and unscheduled nodes have states $|1\rangle$ and $|0\rangle$. $N_i$ 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.

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 transmission queue-backlog at which should be processed when the link is scheduled. More details are in [5].

IV. QUANTUM APPROXIMATE OPTIMIZATION FOR SCHEDULING (QAOS)

In this section, Hamiltonian for QAOA is designed based on the scheduling model in Sec. III and then Quantum Approximate Optimization for Scheduling (QAOS) algorithm is proposed by applying the designed Hamiltonian to QAOA.

A. Design the problem Hamiltonian, $H_P$

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.

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

$$B_1(x) = x$$  \quad \text{where } x \in \{0, 1\}.  \quad (13)$$

Due to quantum Fourier expansion, (13) can be mapped to Boolean Hamiltonian $H_{B_1}$, where $I$ and $Z$ are an Identity operator and the Pauli-$Z$ operator, respectively [8]:

$$H_{B_1} = \frac{1}{2}(I - Z).  \quad (14)$$

According to (13)–(14), the objective function (2) can be represented as following Hamiltonian.

$$H_O = \sum_{\forall l_k \in \mathcal{L}} \frac{1}{2} w_k (I - Z_k),  \quad (15)$$

where $Z_k$ is the Pauli-$Z$ operator applying on $I_k$. The objective of the model is to maximize $H_O$, thus the objective Hamiltonian $H_O$ which should be minimized is as follows:

$$H_O = \sum_{\forall l_k \in \mathcal{L}} \frac{1}{2} w_k Z_k.  \quad (16)$$

DRAFT April 24, 2020
2) Constraint Hamiltonian: In MWIS problem, the banned condition is a case where both nodes directly connected to the edge are scheduled, as shown in Case C in Fig. 1. If the weights of the \( N_i \) 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 banned conditions can be represented as follows:

\[
C'(i,j) = \sum_{l=1}^{n} \sum_{j=1}^{n} (W_{N_i} + W_{N_j}) |E_C(N_i, N_j)|,
\]

where \( n \) is the number of nodes and \( |E_C(N_i, N_j)| \) is the number of \( E_C(N_i, N_j) \).

According to (8)–(12), \( C'(i,j) \) can be redefined to \( C(i,j) \) with symbols in Sec. III as follows:

\[
C(i,j) = \sum_{\forall i \in L} \sum_{\forall j \in L} \frac{1}{2} (w_i + w_j) \mathcal{E}(i,j) = \sum_{\forall i \in L} \sum_{\forall j \in L} \frac{1}{2} (w_i + w_j) (I_i \land I_j),
\]

(18)

where \( \land \) is a Boolean operator AND; and the reason why the coefficient is \( \frac{1}{2} \) in (18) because both \( \mathcal{E}(i,j) \) and \( \mathcal{E}(j,i) \) represent the same edge. The AND Boolean function \( B_2(x_1, x_2) \) can be mapped to Boolean Hamiltonian \( H_{B_2} \) as follows [8]:

\[
B_2(x_1, x_2) = x_1 \land x_2 \text{ where } x_1 \in \{0, 1\} \text{ and } x_2 \in \{0, 1\},
\]

(19)

\[
H_{B_2} = \frac{1}{4} (I - Z_1 - Z_2 + Z_1Z_2),
\]

(20)

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

According to (19)–(20), the objective function \( 18 \) can be represented as following Hamiltonian:

\[
H_{C'} = \sum_{\forall i \in L} \sum_{\forall j \in L} \frac{1}{8} (w_i + w_j) (I_i - Z_i - Z_j + Z_iZ_j),
\]

(21)

where \( Z_i \) and \( Z_j \) are the Pauli-Z operators applying on \( I_i \) and \( I_j \), respectively. The constraint of the model is to minimize \( H_{C'} \), and then the constraint Hamiltonian \( H_C \) is as follows:

\[
H_C = \sum_{\forall i \in L} \sum_{\forall j \in L} \frac{1}{8} (w_i + w_j) (Z_i + Z_j - Z_iZ_j).
\]

(22)

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

\[
H_P = H_O + \rho H_C,
\]

(23)

where \( \rho \in \mathbb{R}^+ \) is the penalty rate, which indicates the rate of which \( H_C \) affects \( H_P \) compared to \( H_O \). According to (16) and (22), both \( H_O \) and \( H_C \) should be minimized, thus \( H_P \) should be minimized as well.

B. Design the mixing Hamiltonian, \( H_M \)

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 flip the state of each node represented by \(|0\rangle \) or \(|1\rangle \). The bit-flip can be handled by the Pauli-X operator, thus \( H_M \) is as:

\[
H_M = \sum_{\forall k \in L} X_k,
\]

(24)
# (Step 1) Implementation of Operators with Cirq

- **Problem Operator**

```python
def problem_op(mwis_graph, weight, penalty_rate, qubits, p, gamma):
    for n in mwis_graph.nodes:  # n: node
        yield cirq.rz(-(1/2)*gamma[p]*weight[n])(qubits[n])
```

- **Objective Operator, Eqs. (16) & (26)**

```python
for e in mwis_graph.edges:  # e: edge
    weight_sum = weight[e[0]] + weight[e[1]]
    yield cirq.CZPowGate(exponent=(1/8)*gamma[p]*penalty_rate*weight_sum/np.pi,
                         global_shift=0)(qubits[e[0]], qubits[e[1]])
```

- **Constraint Operator, Eqs. (22) & (27)**

```python
for n in mwis_graph.nodes:  # n: node
    yield cirq.rx(beta[p][n])(qubits[n])
```

...  

# (Step 2) Training with TensorFlow-Quantum

```python
model.compile(loss=tf.keras.losses.mean_absolute_error,
              optimizer=tf.keras.optimizers.Adam(0.03))
model.fit(input_,optimum,epochs=1000,verbose=1)
```

Fig. 2: Parts of Python codes using Cirq and TensorFlow-Quantum for solving the MWIS-based scheduling problem.

where $X_k$ is the Pauli-X operator applying on $I_k$. In other words, $H_M$ is a transverse-field Hamiltonian [7].

C. Apply to QAOA sequence

The application of the designed Hamiltonian to QAOA sequence starts to conduct when the design of Hamiltonian, i.e., $H_P$ and $H_M$, is completed. First, the parameterized state $|\gamma, \beta\rangle$ can be generated by applying $H_P$ and $H_M$ defined in (16), (22), (23), and (24), to (6). 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 $|\gamma, \beta\rangle$ where the $\gamma$ 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 solution for link scheduling can be obtained by the measurement of the expectation value of $H_P$ on the optimal state $|\gamma_{opt}, \beta_{opt}\rangle$ as follows, where $\langle F \rangle$ is the expectation value of $|\gamma_{opt}, \beta_{opt}\rangle$ over the returned solution samples:

$$\langle F \rangle = \langle \gamma_{opt}, \beta_{opt}|H_P|\gamma_{opt}, \beta_{opt}\rangle.$$  \hspace{1cm} (25)

V. PERFORMANCE EVALUATION

The proposed QAOS algorithm is implemented using Cirq and TensorFlow-Quantum developed for NISQ algorithm and quantum machine learning computation [9], [10].
A. 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 Sec. IV, 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:

\begin{align}
U_O(\gamma_\zeta) &= e^{-i\gamma_\zeta H_O}, \\
U_C(\gamma_\zeta) &= e^{-i\gamma_\zeta \rho H_C}, \\
U_P(\gamma_\zeta) &= U_O(\gamma_\zeta)U_C(\gamma_\zeta) = e^{-i\gamma_\zeta (H_O+\rho H_C)}, \\
U_M(\beta_\zeta) &= e^{-i\beta_\zeta H_M},
\end{align}

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 implementing $U_P(\gamma_\zeta)$ and $U_M(\beta_\zeta)$ is the core of QAOS implementation.

In Fig. 2, `cirq.rz()` and `cirq.CZPowGate()` are used for the implementation of $U_O(\gamma_\zeta)$ and $U_C(\gamma_\zeta)$, respectively; and based on these, $U_P(\gamma_\zeta)$ is implemented as (28). Notice that `cirq.rz()` represents the rotation-Z gate, and `cirq.CZPowGate()` represents the quantum gate that applies a phase to the state $|11\rangle$. In addition, $U_M(\beta_\zeta)$ is implemented using `cirq.rx()` which means the rotation-X gate.

The part that finds the optimal parameters using Keras (one of well-known open-source deep learning computation libraries) is (Step 2), from line 16 to line 25, in Fig. 2. In this model, the parametrized quantum circuit (PQC) layer provides auto-management of variables in the parameterized circuit [10].

B. Results

The performance of our proposed QAOS algorithm is compared with random search and greedy search. In addition, the QAOS algorithm executes with different $p$ value settings where the $p$ value means the number of alternation of $U_P(\gamma_\zeta)$ and $U_M(\beta_\zeta)$ in (28) and (29), i.e., $\zeta \in \mathbb{Z}^+$ and $1 \leq \zeta \leq p$.

For the performance evaluation, we generate random graphs with 10 nodes, i.e., links in conflict graphs; and then random search, greedy search, and QAOS algorithms are performed for the given random graphs. The measurement of each QAOS is performed 1,000 times in each simulation (i.e., in each randomly generated conflict graph), and the solution that is returned with the maximum probability is selected as the solutions of each simulation. Then, the performance of each algorithm is quantitatively measured as $\eta \triangleq \frac{a}{b}$ where $a$ and $b$ are the summation of weights of the scheduled nodes by the used algorithms and the summations of weights of the scheduled nodes by brute-force full search (i.e., exhaustive search), respectively, for the given randomly generated graphs. Then, the cumulative distribution functions (CDF) of $\eta$ for each algorithm is computed and illustrated in Fig. 3.

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

TABLE I: 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%   |
VI. CONCLUDING REMARKS

In wireless network research, the large-scale and multi-scale scheduling can be modeled with the MWIS problem which is one of 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), is proposed. The performance of our proposed QAOS is evaluated via data-intensive simulations using Cirq and TensorFlow-Quantum. As a result, we confirm that our proposed QAOS outperforms the other methods for the MWIS-based multi-scale scheduling problem.

REFERENCES

[1] E. Farhi, J. Goldstone, and S. Gutmann, “A quantum approximate optimization algorithm,” arXiv:1411.4028, 2014.
[2] C. Rose and I. S. Mian, “Inscribed matter communication: Part I,” IEEE T-MBMC, vol. 2, no. 2, pp. 209–227, 2016.
[3] T. Nakano, “Molecular communication: A 10 year retrospective,” IEEE T-MBMC, vol. 3, no. 2, pp. 71–78, 2017.
[4] S. F. Bush, J. L. Paluh, G. Piro, V. S. Rao, R. V. Prasad, and A. W. Eckford, “Defining communication at the bottom,” IEEE T-MBMC, vol. 1, no. 1, pp. 90–96, 2015.
[5] J. Kim, G. Caire, and A. F. Molisch, “Quality-aware streaming and scheduling for device-to-device video delivery,” IEEE/ACM Transactions on Networking, vol. 24, no. 4, pp. 2319–2331, August 2016.
[6] J. Preskill, “Quantum computing in the NISQ era and beyond,” Quantum, vol. 2, p. 79, August 2018.
[7] S. Hadfield, Z. Wang, B. O’Gorman, E. G. Rieffel, D. Venturelli, and R. Biswas, “From the quantum approximate optimization algorithm to a quantum alternating operator ansatz,” Algorithms, vol. 12, no. 2, 2019.
[8] S. Hadfield, “On the representation of Boolean and real functions as Hamiltonians for quantum computing,” arXiv:1804.09130, 2018.
[9] M. Fingerhuth, T. Babar, and P. Wittek, “Open source software in quantum computing,” PLoS ONE, vol. 13, no. 12, December 2018.
[10] M. Broughton, G. Verdon, T. McCourt, A. J. Martinez, J. H. Yoo et al., “TensorFlow quantum: A software framework for quantum machine learning,” arXiv:2003.02989, 2020.