Quantum Search Algorithm for Weighted Solutions

YOUNGJIN SEO, YUJIN KANG, AND JUN HEO (Member, IEEE)
School of Electrical Engineering, Korea University, Seoul 02841, South Korea
Corresponding author: Jun Heo (junheo@korea.ac.kr)

This work was supported in part by the Samsung Research in Samsung Electronics, and in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) by the Ministry of Education under Grant 2019R1A2C2010061.

ABSTRACT This paper presents a quantum search algorithm (QSA) for weighted solutions. In a QSA, the final quantum state has an equal probability of solutions. However, solutions should have different probabilities in specific problems, such as dynamic spectrum management (DSM). We propose a method to make differences with probability in accordance with rewards for each solution, and present an application to DSM. The proposed algorithm has some error terms for the final quantum state, and we provide the error terms in accordance with the number of iterations. In addition, we present the simulation results of the proposed algorithm compared with a classical exact algorithm and QSA. The proposed algorithm has a lower complexity and relative difference than the classical exact algorithm. Furthermore, the proposed algorithm has a higher performance than the QSA in specific cases.

INDEX TERMS Dynamic spectrum management, graph coloring problem, quantum search algorithm.

I. INTRODUCTION
Quantum computation has the advantage of having a lower complexity than classical computation in many algorithms [1]–[6]. In particular, Grover’s quantum search algorithm (QSA) solves a data-searching problem with quadratic speed-up [5]. Given a search space with a size of \( N \), the QSA finds the desired data with complexity \( O(\sqrt{N}) \), which is much less than that of the classical random search \( O(N) \). After Grover had suggested the QSA, many improved quantum search algorithms have been proposed. Boyer-Brassard-Høyer-Tapp presented a tight bound of QSA and a method that solve the problem having unknown number of solutions [7]. Dürr and Høyer proposed a quantum minimum search algorithm [8]. Dürr-Høyer algorithm can find a minimum value through higher complexity rather than QSA. Hogg presented a heuristic algorithm for quantum method [9]. The quantum heuristic algorithm can solve a satisfiability problem with lower complexity than QSA. Fabrikant and Hogg presented the quantum heuristic algorithm for a graph coloring problem (GCP) [10]. In practice, QSA can be applied to a resource assignment problem such as the GCP [11].

A GCP is a coloring vertex that is connected by an edge with different colors in a graph where multiple vertices are drawn (called “interference condition”). We can describe the problem mathematically using graph theory, and find all possible color assignments that satisfy the given conditions using a search algorithm. Therefore, several studies have been conducted to address the resource assignment problem in communication systems where interference conditions exist by converting the assignment problem to GCP [12]–[14]. If two objects interfere with each other, these objects are denoted by two vertices connected by an edge. We can also find an assignment combination that maximizes the communication gain by adding rewards for each resource usage. Once we describe the system in graph theory, we can use a search algorithm in the same manner as GCP. This paper proposes a QSA that considers the sum of rewards of each solution, such that it can be used in GCPs with rewards.

The method of applying QSA to GCP and the implementation of corresponding gate-level circuits have been proposed in various ways. Since [11] reported that QSA is a faster solution for NP-complete problems such as GCP, [15] and [16] expressed vertices and available colors as qubits in the ternary domain, and they proposed a method of synthesizing the oracle, which marks states that satisfy interference conditions, as well as an amplification operator. In addition, [17]
addressed qubits in the binary domain and its oracle, while analyzing the cost of fault tolerance of the oracle.

In prior studies, if the number of solutions is greater than one, all solution states have the same probability. Thus, when we consider the sum of the rewards of each solution, the QSA does not reflect rewards. However, we introduce the conditioned oracle and phase inversion in Section III to increase the probability according to the sum of rewards. In addition, we propose procedures ranging from receiving the reward matrix (whose elements can be a real value) to entering values into the conditioned oracle. To maximize the probabilities of solutions, the upper and lower bounds on the number of iterations \( i \) of the process from the conditioned oracle to phase inversion are derived mathematically. Let \( N_v \) be the number of vertices and \( N_c \) be the number of colors; then, we reduce the complexity of the proposed algorithm from \( O(\sqrt{N}) \), where \( N = 2^{N_v N_c} \), to \( O(\sqrt{N}) \) where \( N = N_v N_c \), using the W-state in [18].

Therefore, we propose a QSA that solves GCP by considering rewards (called “Reward-GCP”). The simulation results also demonstrate the performance of the proposed algorithm. When comparing the proposed algorithm with the classical algorithm, we calculated the complexity to find the solution with the maximum reward from the result. It can be seen that the proposed algorithm has a lower complexity than the classical exact algorithm. In addition, when setting the classical exact algorithm and the proposed algorithm to the same complexity \( \sqrt{N} \), the expected reward of the proposed algorithm is much higher. Furthermore, we compared the relative difference of the proposed algorithm and QSA.

### II. PRELIMINARIES

In this section, we review dynamic spectrum management (DSM) using graph coloring problem (GCP) and quantum search algorithm (QSA) and present a method for the application of QSA to GCP.

#### A. DYNAMIC SPECTRUM MANAGEMENT (DSM)

In addition to 5G, there is the need for a technology that increases the data rate and manages a large number of users in a 6G network. Thus, a method for efficiently sharing idle resources of cognitive radio (CR) networks while preventing interference between adjacent users is being considered [19], [20]. In addition, the application of spectrum sharing to achieve the efficient utilization of limited spectrum resources has been consistently mentioned [21].

The CR network is an intelligent multi-user spectrum sharing technique that assigns an idle spectrum to a cognitive user (CU, or a secondary user). However, a primary user (PU), who is an original user who owns the spectrum in the primary network, is assigned to use a particular spectrum band by a communication service provider, while having priority over the CU. In this case, a spectrum hole refers to a spectrum band that is assigned to the PU, but which is not being utilized by its owner temporally or geographically. The CU shares the spectrum with the PU while detecting the spectrum hole in real time. Table 1 lists the components used in the CR network and their corresponding descriptions. The minimum unit for transmission and reception is called a CR unit. A pair of CR units where a transmitting CU and a receiving CU communicate with each other is called a CR link.

The DSM technique proposes a spectrum assignment scheme that considers the reward in a CR network. First, spectrum holes are activated and rewards that CR links can obtain from each spectrum are obtained. Then, the spectrum holes are dynamically assigned to CR links in an efficient manner to maximize the reward of the CR network. Therefore, the purpose of DSM is to find the optimal solution that satisfies interference conditions while considering the reward, and many previous studies have focused on developing a method to find the optimal solution quickly and accurately.

This paper describes the status of CR networks in graph theory and solving the resource assignment problem using GCP. DSM is converted to GCP, and a search algorithm is used to find solutions. GCP is a problem of painting vertices in some colors, but adjacent vertices connected by edges must be painted in different colors. In this problem, the number of available colors can be determined in advance and it can be extended to a problem in which rewards vary depending on the color assigned to each vertex. Each component of the DSM and its corresponding component of the GCP are shown in Table 2, and the solutions of the DSM will be color-assigned combinations of GCP [13]. Fig. 1 shows an example of converting DSM to GCP in an arbitrary CR network. Details of the GCP and search algorithms are described in the next section.

#### B. GRAPH-COLORING PROBLEM (GCP)

Graph \( \mathcal{G} \) consists of sets of \( N_v \) vertices, \( \mathcal{V} \), and \( N_e \) edges, \( \mathcal{E} \), which describe a graph using the notation \( \mathcal{G} = (\mathcal{V}, \mathcal{E}) \). In this paper, we define GCP as a problem coloring all vertices with \( N_c \) colors so that two vertices connected by one
edge (called the “adjacent vertices”) are assigned different colors (called “interference condition”). We can also define a solution for the graph. Let \( S_t(v_i) \) be a set of \( v_i \)'s colors in the \( t \)-th solution. Then, the graph is expressed as

\[
V = \{v_1, v_2, \ldots, v_N\},
\]

\[(v_i, v_j) \in E,\]

\[
C = \{c_1, c_2, \ldots, c_N\},
\]

\[
S_t(v_i) \subseteq C,
\]

\[
S_t(v_i) \cap S_t(v_j) = \emptyset, \quad \forall (v_i, v_j) \in E,
\]

where \( i, j = 1, \ldots, N_v \). In addition, \( t = 1, \ldots, m \) and \( m \) is the number of solutions. The \( t \)-th solution, \( S_t \), of graph \( G \) consists of \( N_v \) sets, that is, \( S_t \equiv \{S_t(v_1), S_t(v_2), \ldots, S_t(v_N)\} \).

The problem of assigning one color to one vertex is called a single coloring problem (SCP) where \( S_t \) satisfies the following equation.

\[
|S_t(v_i)| = 1. \tag{2}
\]

If any vertex should be assigned more than one color, we call this problem a multi-coloring problem (MCP).

When we define a weight \( \mathcal{W}(v_i) \) as the number of colors to assign a vertex \( v_i \in V \), SCP is the case where \( \mathcal{W}(v_i) = 1 \). However, the problem of assigning several colors to one vertex is called MCP where \( \mathcal{W}(v_i) \in \{1, 2, \ldots\} \) \( \forall v_i \in V \). In addition, there should be no colors in common at adjacent vertices. Thus, \( S_t \) in the MCP satisfies (1) and (3).

\[
|S_t(v_i)| = \mathcal{W}(v_i). \tag{3}
\]

Let a clique be a subset of \( G \) in which all vertices are mutually adjacent. Any MCP can be converted into SCP by replacing a vertex with \( \mathcal{W}(v_i) > 1 \) with a clique of size \( \mathcal{W}(v_i) \) [22].

Meanwhile, it is possible to establish lower bounds on \( N_c \) using a theorem such as Hoffman’s bound (1970). In this case, the minimum number of colors required to color graph \( G \) is denoted by \( N_{chr} \), and is called a chromatic number. In other cases, \( N_c \) is already determined to be a condition of the problem. A basic assumption in this study is that \( N_c \) is fixed as a condition of the GCP. Thus, we will discuss the GCP with the interference condition and a fixed value, \( N_c \).

Using graph theory described in this paper, the following example is available: Fig. 2 shows an example of a 3-coloring problem with three vertices that are fully connected and that use only three colors. Using (1), Fig. 2 can be expressed as

\[
V = \{v_1, v_2, v_3\},
\]

\[
E = \{(v_1, v_2), (v_1, v_3), (v_2, v_3)\},
\]

\[
C = \{c_1, c_2, c_3\},
\]

\[
S_t(v_i) \cap S_t(v_j) = \emptyset, \quad \forall (v_i, v_j) \in E. \tag{4}
\]

We can intuitively find a set of solution for this simple graph.

\[
S_t = \{S_t(v_1), S_t(v_2), S_t(v_3)\},
\]

\[
\text{Case 1 : } S_1 = \{\{c_1\}, \{c_2\}, \{c_3\}\},
\]

\[
\text{Case 2 : } S_2 = \{\{c_1\}, \{c_3\}, \{c_2\}\},
\]

\[
\text{Case 3 : } S_3 = \{\{c_2\}, \{c_1\}, \{c_3\}\},
\]

\[
\text{Case 4 : } S_4 = \{\{c_2\}, \{c_3\}, \{c_1\}\},
\]

\[
\text{Case 5 : } S_5 = \{\{c_3\}, \{c_1\}, \{c_2\}\},
\]

\[
\text{Case 6 : } S_6 = \{\{c_3\}, \{c_2\}, \{c_1\}\}. \tag{5}
\]
However, it is difficult to obtain a solution intuitively in more complex graphs with more vertices and edges. Thus, various search algorithms have been proposed to find solutions of the GCP that satisfy interference conditions with a fixed $N_c$. An exact algorithm always returns the best solution (or optimal solutions) in the problem of finding feasible combinations. The best solutions are all color-assigned combinations that satisfy all edge conditions and use the minimum number of colors, $N_{chr}$. Although $N_c$ is not equal to $N_{chr}$ and may be greater than $N_{chr}$, exact algorithms can always find solutions. Therefore, the running time for exact algorithms increases as the number of vertices or colors increases. Some exact algorithms for finding the best solutions are described as follows:

**Exhaustive search (brute-force search):** An exhaustive search is an enumeration method for GCP. It considers $N_c$ assignments (candidate solutions) on $N_v$ vertices and checks whether each assignment is legitimate. The algorithm then returns the best solutions using the fewest colors. A simple algorithm for Fig. 2 is shown in Fig. 3.

**Backtracking algorithm:** The backtracking algorithm gradually generates feasible solutions by checking the relevant edge conditions at each step. In the forward step, the algorithm assigns colors to vertices in a predefined order until a vertex cannot be colored with one of $N_c$ colors. In the backward step, the algorithm searches backward and identifies a colored vertex that can be colored with different colors. Then, the forward step resumes at this point, and the algorithm repeats these steps until we return to the root vertex $v_1$ in a backward step or reach a user-defined time limit. A simple algorithm for Fig. 2 is shown in Fig. 4.

So far, we have described GCP, where the number of available colors is predetermined, and the interference condition in (1) is satisfied. In this case, all solutions from the exact algorithm are equally significant. However, we can extend the GCP to a problem that considers a reward for each color of a vertex. The reward represents the gain obtained when a vertex is assigned a specific color, and has a non-negative value. Each solution has a different significance with respect to the reward.

Let $r$ be a reward matrix and $r_{ij}$ be an element of $r$ where $i = 1, \ldots, N_v$ and $j = 1, \ldots, N_c$. The element $r_{ij}$ represents the gain obtained when color $j$ is assigned to vertex $v_i$, that is, $S_i(v_i) = c_j$. $r = [r_{ij}]_{r_{ij} \geq 0}$, $S_i(v_i) = c_j \forall i, v_i$.

(6)

The reward of the entire system can be defined as follows:

$$U_{sum}(S_t) = \sum_{i=1}^{N_v} r_{ij}$$

(7)

where $S_i(v_i) = c_j$. We can find a specific solution based on the reward value among the solution set, $S_t$. Thus, when we consider colors as active resources and vertices as objects to be assigned, many resource assignment problems can be converted to GCP, and the reward is defined as the gain from each resource.

Therefore, it is possible to define Reward-GCP that finds a solution that maximizes the sum of rewards while satisfying the following conditions.

**Condition 1:** Interference condition $S_i(v_i) \cap S_j(v_j) = \emptyset$, associated with (1).

**Condition 2:** The number of available colors, $N_c$ is predetermined.

**Condition 3:** The non-negative reward, $r_{ij}$, is the gain obtained when the vertex $v_i$ is assigned to the color $c_j$, associated with (6).

The reward-GCP model is commonly used to solve the resource assignment problem of communication because clear factors are used as gains in the communication environment. Specific situations of the application of the reward-GCP to the communication system are described in Section IV.

Table 3 summarizes parameters that are used to describe GCP in this paper.

---

**C. QUANTUM SEARCH ALGORITHM (QSA)**

1) GROVER’S QUANTUM SEARCH ALGORITHM

Given a Boolean function $f : X \rightarrow \{0, 1\}$, we can find $x \in X$ such that $f(x) = 1$ using a QSA. Let $S_f$ be a conditional phase inversion operator defined by

$$S_f |x \rangle = (-1)^{f(x)} |x \rangle$$

(8)

Let $S_0$ be a phase inversion operator of $|0\rangle$, where $S_0 |x \rangle = - |x \rangle$ if and only if $x = 0$.

Assume that the space of $X$ is $N = 2^n$. There are $m$ solutions in $X$. The index $j \in X$ can be described in the range $0$ to $N - 1$.

QSA can be summarized in five steps: initialization, oracle, amplitude amplification, iteration, and measurement. During initialization, we can transform $|0\rangle$ to $|\psi\rangle$ using the Hadamard gate:

$$|\psi\rangle = H^{\otimes n} |0\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle = \cos \theta |\alpha\rangle + \sin \theta |\beta\rangle$$

(9)
TABLE 3. GCP parameters.

| Parameters | Description |
|-----------|-------------|
| $N_v$ | The number of vertices |
| $N_e$ | The number of edges |
| $N_c$ | The number of colors |
| $N_{ch}$ | The chromatic number of $G$ |
| $m$ | The number of solutions |
| $\mathcal{G}$ | Graph |
| $\mathcal{V}$ | Set of vertices |
| $\mathcal{E}$ | Set of edges |
| $\mathcal{C}$ | Set of available colors |
| $S_f(v_i)$ | Set of $v_i$’s colors in the $i$-th solution |
| $S_f$ | Each solution/set for $\mathcal{G}$ in multiple solution cases |
| $W(v_i)$ | The number of colors to assign to a vertex $v_i$ |
| $r$ | Reward matrix |
| $r_{ij}$ | A reward obtained when the vertex $v_i$ is colored in the color $c_j$ |
| $U_{sum}(S_f)$ | The sum of rewards for the entire system, $\sum_{i=1}^{N} r_{ij}, c_j = S_f(v_i)$ |

where $|\alpha\rangle$ and $|\beta\rangle$ are defined as a non-solution and solution quantum state, respectively, and $\sin \theta = \sqrt{m/N}$. We use the Hadamard gate for convenience, but any unitary transformation can be used [23].

In the oracle, $S_f$ is applied to $|\psi\rangle$. It transforms $|\psi\rangle$ into $|\psi'\rangle$ as follows:

$$|\psi'\rangle = S_f |\psi\rangle = \cos \theta |\alpha\rangle - \sin \theta |\beta\rangle .$$  \hspace{1cm} (10)

Subsequently, we use $-HS_0H^\dagger$ to amplify the probability of the solution quantum state in amplitude amplification. Operator $S_0$ is also expressed as $S_0 = I - 2|0\rangle\langle 0|$. Now we can define Grover’s operator as a unitary transformation

$$Q = -HS_0H^\dagger S_f .$$  \hspace{1cm} (11)

Hereafter, we focus on changes in $|\alpha\rangle$ and $|\beta\rangle$. First, we consider $Q |\alpha\rangle$.

$$Q |\alpha\rangle = \cos 2\theta |\alpha\rangle + \sin 2\theta |\beta\rangle .$$  \hspace{1cm} (12)

Second, we consider $Q |\beta\rangle$.

$$Q |\beta\rangle = -\sin 2\theta |\alpha\rangle + \cos 2\theta |\beta\rangle .$$  \hspace{1cm} (13)

This can be summarized by (14).

$$Q \begin{bmatrix} |\alpha\rangle \\ |\beta\rangle \end{bmatrix} = \begin{bmatrix} \cos 2\theta & \sin 2\theta \\ -\sin 2\theta & \cos 2\theta \end{bmatrix} \begin{bmatrix} |\alpha\rangle \\ |\beta\rangle \end{bmatrix} .$$  \hspace{1cm} (14)

Finally, we obtain the result of implementing Grover’s operator.

$$|\psi'\rangle = Q |\psi\rangle = \cos 3\theta |\alpha\rangle + \sin 3\theta |\beta\rangle .$$  \hspace{1cm} (15)

Consequently, the $Q$ operator rotates $|\alpha\rangle$ and $|\beta\rangle$ about $2\theta$ to positive side.

When we apply $Q$ to $|\psi\rangle$ $i$ times, we get

$$|\psi_i\rangle = Q^i |\psi\rangle = \cos (2i + 1) \theta |\alpha\rangle + \sin (2i + 1) \theta |\beta\rangle .$$  \hspace{1cm} (16)

The QSA process is described in Fig. 5. If we know the value of $m$, we can compute the appropriate number of $i$. The value of $i$ can be computed using the equation such that the probability of the solutions is close to 1.

$$\sin^2(2i + 1)\theta = \frac{1}{\pi}$$  \hspace{1cm} (17)

$$i = \frac{\pi}{2} - \frac{1}{2} \approx \frac{\pi}{4\sqrt{m/N}} .$$  \hspace{1cm} (18)

With the property of the sinusoid, $\theta$ approximates $\sin \theta = \sqrt{m/N}$ on a large scale of $N$. That is,

$$i = \frac{\pi}{4\sqrt{m/N}} .$$  \hspace{1cm} (19)

Finally, the state $|\psi_i\rangle$ is measured and yields a specific index $\chi$. If the index $\chi$ satisfies $f(\chi) = 1$, the algorithm succeeds. When the outcome of a measurement is not the solution, we restart QSA. In this case, the complexity of QSA is $O(\sqrt{N/m})$, as obtained by (19).

2) BOYER-BRASSARD-HAYER-TAPP QUANTUM SEARCH ALGORITHM

In QSA, to succeed with certainty, we should know the value of $m$ for the computation of the exact value of $i$. If the exact value of $i$ is unknown, QSA has a probability of failure.

When $m$ is unknown, we iterate the algorithm using a method presented by Boyer et al. [7]. First, initialize $j = 1$ and set $\lambda = 8/7$, which is any value strictly between 1 and 4/3. Then, an integer $i$ is chosen randomly such that $0 \leq i < j$, and apply $i$ iteration times are applied in QSA. Finally, if a solution is not obtained, we set $j$ to min($\lambda j$, $\sqrt{N}$) and return to the step choosing $i$.

3) Dürr-Hay-Hay-THAP QUANTUM SEARCH ALGORITHM

When we find a minimum value among $T[0, \cdots, N-1]$, we can use Dürr-Hay quantum search algorithm [8]. First, we choose a threshold index $y$ uniformly random such that $0 \leq y \leq N-1$. Then, set $x \in X$ such that

$$f(\arg_x, (T[x] < T[y]) = 1 .$$  \hspace{1cm} (20)

Afterward, we use the method in Section II-C2, and earn a measurement outcome $y'$. If $T[y'] < T[y]$, we set the threshold index $y$ to $y'$ and return to the step setting $x$. Finally, when the total running time is more than $22.5\sqrt{N}$, the algorithm ends.

D. QUANTUM METHOD FOR GCP

The quantum method employed to solve GCP is derived from an exact algorithm, for example, an exhaustive search. The exact algorithm can find an optimal solution with the maximum system reward among all possible combinations. The purpose of this study is to devise a QSA to solve the GCP satisfying Conditions ((1) and (6)). Based on prior studies, we describe how to solve the GCP satisfying Condition 1 and Condition 2 ((1)), and we analyze its limitation when Condition 3 ((6)) is added. In the five phases of QSA, the
initialization and the oracle present specific requirements for searching for solutions of GCP, and the rest remain the same.

When solving GCP using QSA, first, an input state of QSA uses qubits that can contain all the colors of all vertices. Thus, there are \( N_c \) pairs of \( N_v \) bits representing the number of available colors for each vertex, and the total \( N_v \times N_c \) qubits are allocated to the input state. \( q_{ij} \) is a qubit that indicates whether the \( j \)th color has been assigned to the \( i \)th node, and it has a value of 1 when it is assigned. Equation (21) shows an input state where \( N_v = 3 \) and \( N_c = 3 \) (\( G_1 \) in Fig. 2).

\[
\lvert q_{11}q_{12}q_{13}q_{21}q_{22}q_{23}q_{31}q_{32}q_{33} \rangle.
\] (21)

The input state of the QSA is a bare candidate that does not consider interference conditions (Condition 1). Candidates can represent all possible strings in a data space such as

\[
\lvert 000 000 000 \rangle, \lvert 000 000 001 \rangle, \ldots, \lvert 111 111 111 \rangle,
\] (22)

where the number of candidates is equal to the size of the data space, \( N = 2^{N_vN_c} \). Instead of using all possible strings, we use the \( W \) state to generate an input state, as described in [18]. This method is based on the fact that only one color can be assigned to a vertex, and only one qubit has a value of 1. For \( G_1 \), only one of \( q_{11}, q_{12} \) and \( q_{13} \) has a value of 1. Therefore, we can reduce the number of candidates to \( N_vN_c \), and an input state can be a superposition of these states:

\[
\lvert 001 001 001 \rangle, \lvert 001 001 010 \rangle, \ldots, \lvert 100 100 100 \rangle.
\] (23)

The initialization step generates a superposition of \( N = N_vN_c \) input states. Then, oracle \( S_f \), marks states that satisfy all interference conditions among the superpositioned input states. The oracle can be implemented using a controlled unitary gate or in various ways [15]–[17]. Afterward, the probability of the solution states marked by the oracle is maximally amplified through the amplitude amplification and iteration phases. The measurement result (index, \( N_{\text{CHR}} \)) is converted to classical information (bit), and can be used similar to the solution from the classical search algorithm. Using the procedure described so far, QSA can be applied to the GCP. GCP is interpreted as a problem of finding a solution that satisfies the given conditions in a space with a size of \( N = N_vN_c \). In addition, QSA can find a solution with a complexity of \( O(\sqrt{N_vN_c}) \), whereas classical counterparts can find one with \( O(N_vN_c) \).

Meanwhile, the solution state of the GCP can be a multiple rather than one. It is possible to amplify the probabilities of all solutions, and when performing measurements, all solutions are detected with equal probability. Because each solution has the same probability without considering rewards, QSA cannot be an optimal algorithm if Condition 3 is included.

However, the proposed algorithm can find an optimal solution of GCP satisfying Condition 3 while maintaining the complexity gain of QSA. In Section III, we present the mathematical development of the proposed algorithm and the manner in which the algorithm works. We also provide simulation results for the classical exact algorithm, QSA, and the proposed algorithm.

III. PROPOSED ALGORITHM

In this section, we present a QSA for weighted solutions with a reward value. First, we provide a simple case of the proposed algorithm with three parameters. We then consider the general case of the proposed algorithm. In addition, we state the algorithm in the case of the reward-GCP for DSM. Finally, we analyze the complexity of the classical and quantum methods, and we compare the performance of the proposed algorithm with that of QSA and the classical exact algorithm.

A. THE SIMPLE CASE OF THE PROPOSED ALGORITHM

Let us consider a problem where there are \( m \) different values of \( j \) such that \( f(j) = 1 \). We assume that \( \beta \) and \( \gamma \) are the high/low-weight solutions, respectively. In addition, \( \alpha \) represents a set of non-solutions. The space of the data in the problem is \( N \).

We can also prepare the quantum state for the initialization of QSA.

\[
\lvert \psi \rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \lvert j \rangle = \cos \theta \lvert \alpha \rangle + \sin \theta \sin \phi \lvert \beta \rangle + \sin \theta \cos \phi \lvert \gamma \rangle
\] (24)

where \( \sin \theta = \sqrt{m/N} \) and \( \sin^2 \phi \) is the ratio of \( \beta \) in the solutions. We also assume another conditional phase inversion operator, \( S_g \) and \( S_h \), which are different from \( S_f \).

\[
S_g \lvert x \rangle = (-1)^{y(x)} \lvert x \rangle, \quad x \in \beta
\]

\[
S_h \lvert x \rangle = (-1)^{y(x)} \lvert x \rangle, \quad x \in \gamma
\] (25)

First, we execute Grover’s operator with \( i \) iterations. Before the measurement, we have

\[
\lvert \psi_i \rangle = Q^i \lvert \psi \rangle = \cos (2i+1) \theta \lvert \alpha \rangle + \sin (2i+1) \theta \sin \phi \lvert \beta \rangle + \sin (2i+1) \theta \cos \phi \lvert \gamma \rangle.
\] (26)
Here, we explain the process of increasing the probability of $|\beta\rangle$ by reducing that of $|\gamma\rangle$. First, we define the iteration operator of the proposed algorithm.

$$K = -S_h H S_0 H^\dagger S_g. \quad (27)$$

When we apply $K$ to $|\psi_i\rangle$ $k$ times, we obtain

$$|\psi_{i,k}\rangle = K^k |\psi_i\rangle = \left[ \cos(2(i + k) + 1)\theta + \varepsilon_{a,k} \right] |\alpha\rangle + \left[ \sin(2(i + k) + 1)\theta \sin \phi + \varepsilon_{\beta,k}\right] |\beta\rangle + \left[ \sin(2(i - k) + 1)\theta \cos \phi + \varepsilon_{\gamma,k} \right] |\gamma\rangle \quad (28)$$

where $\varepsilon_{a,k}, \varepsilon_{\beta,k},$ and $\varepsilon_{\gamma,k}$ are the error terms of $|\alpha\rangle, |\beta\rangle,$ and $|\gamma\rangle$, respectively. The details of the error terms in simple case are mentioned in Appendix A.

To increase the probability of solutions, the probability of $|\alpha\rangle$ should decrease. Whereas $\varepsilon_{a,k}$ can be disregarded in small $k$, $\varepsilon_{a,k}$ increases the probability of $|\alpha\rangle$ in large $k$. Therefore, we should have a suitable value of $i$ and $k$ according to the probability of $|\alpha\rangle$. Because $\varepsilon_{\beta,k}$ and $\varepsilon_{\gamma,k}$ are much smaller than $\varepsilon_{a,k}$ depending on $k$, we can ignore $\varepsilon_{\beta,k}$ and $\varepsilon_{\gamma,k}$ in the low probability of $|\alpha\rangle$.

There were two conditions for $k$ iterations. First, the angle of $|\alpha\rangle$ is not allowed, which is larger than $\pi/2$. Second, the angle of $|\gamma\rangle$ is not allowed, which is lower than zero. Therefore, these two conditions can be combined. Then, we get

$$\begin{align*}
2(i + k) + 1)\theta &\leq \frac{\pi}{2}, \quad (2i - k) + 1)\theta \geq 0 \\
i + k &\leq \frac{\pi}{4\theta} - \frac{1}{2}, \quad i - k \geq -\frac{1}{2} \\
\vdots \quad k &\leq i \leq \frac{\pi}{4\theta} - k.
\end{align*} \quad (29)$$

When we know the number of solutions $n$, we can estimate the value of $\theta$. In other words, we can choose a suitable value of $i$ and $k$ with a high probability of the solutions. This means that the angle of $|\alpha\rangle$ is close to $\pi/2$. However, a high value of $k$ increases the probability of $|\alpha\rangle$ according to (64) in Appendix A. Consequently, we have to choose an appropriate value of $i$ and $k$ where the non-solutions have a low probability.

Now we can summarize the proposed algorithm of simple case to Algorithm 1. In the case of an unknown number of solutions, we can use the method of [7].

**B. CASE OF MULTIPLE WEIGHTED SOLUTIONS**

In this section, we present the proposed algorithm in $t$ different weight solution sets. In addition, we assume that there are $m$ different solutions in $N$ space of the problem. When all solutions have different weights, $t = m$. Now, we redefine the initialization state $|\psi\rangle$ given by

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle = \cos \theta |\alpha\rangle + \sin \theta \sum_{w=1}^{l} b_w |\beta_w\rangle. \quad (30)$$

where $b_w^2$ is the ratio of $\beta_w$ to the total solution such that $\sum_{w=1}^{l} b_w^2 = 1$ and $\sin \theta = \sqrt{m/N}$. Each solution state is denoted as $|\beta_w\rangle$, and $|\alpha\rangle$ represents the state of the incorrect answer. We also assume that the $l$-th conditional phase inversion operator $S_{w,l}$ is defined as

$$S_{w,l} |x\rangle = (-1)^{f(x)} |x\rangle, \quad x \in R_l \quad (31)$$

where $R_l$ is the $l$-th conditional weight solution set. That can be changed in each iteration. For example, if we want to increase the probability of $|\beta_1\rangle, |\beta_2\rangle, \ldots, |\beta_p\rangle$ in the $l$-th iteration, $R_l$ consists of $R_l = \{\beta_1, \beta_2, \ldots, \beta_p\}$. Specifically, when $R_l$ consists of all different weight solutions, $S_{w,l}$ equals $S_f$ in QSA. The function $f$ in (31) is identical to the $f$ function represented in (8).

The inversion operator of $S_{w,l}$ is defined by

$$S_{w,l}^\dagger |x\rangle = (-1)^{f(x)} |x\rangle, \quad x \notin R_l. \quad (32)$$

This operator is not described in QSA. If we set $R_l$ as a portion of the solution, we obtain the unintended results after amplitude amplification; the relative phase of others in $R_l$ will have a negative phase. To solve this problem, we must apply an additional phase inversion operator to the solution set not included in $R_l$. This operator can be considered as the generalized operator of $S_h$ in (25).

We also define the $l$-th iteration operator $K_l$ as follows:

$$K_l = -S_{w,l} H S_0 H^\dagger S_{w,l} \quad (33)$$

where $K_l$ is equal to $Q$ when $R_l = \{\beta_1, \beta_2, \ldots, \beta_l\}$. Similar to $K$ in (27), it rotates the states at approximately $2\theta$ with some errors.

In this algorithm, $Q$ should be applied $i$ times because the low-weight solutions may have a negative phase. The appropriate value of $i$ can be determined from the value of $k$, where $k$ represents the number of iterations operating $K_l$, not $Q$. After applying $Q'$ and $\prod_{l=1}^{t} K_l$ to $|\psi\rangle$, we obtain the final state $|\psi_{i,k}\rangle$,

$$|\psi_{i,k}\rangle = K_{k+1} \cdots K_1 Q' |\psi\rangle$$

$$= \left[ \cos(2(i + k) + 1)\theta + \varepsilon_{a,k} \right] |\alpha\rangle$$

$$\quad + \sum_{w=1}^{l} \left[ \sin(2(i + k\beta_w - k) + 1)\theta \cdot b_w + \varepsilon_{\beta_w,k} \right] |\beta_w\rangle \quad (34)$$

**Algorithm 1 Quantum Search Algorithm for Single-Weighted Solutions**

1. Initialize an integer $k$ from problem settings such as a reward matrix in GCP.
2. Set $j = k$ and $\lambda = 8/7$.
   ($\lambda$ can be any value between 1 and 4/3)
3. Choose an integer $i$ uniformly random such that $1 \leq i \leq j$.
4. Apply $K^k Q'$ to the initial state $|\psi\rangle$.
5. Measure the state $K^k Q' |\psi\rangle$ and let $\varphi$ be the outcome.
6. If $f(\varphi) = 1$, the algorithm is over.
7. Otherwise, reset $j$ to $\min(\lambda j, \frac{\pi}{4\theta} - k)$ and return to step 3.
where \( k_{\beta_w} \) denotes the number of \( \beta_w \) included in \( R_l \) for \( l = 1, 2, \cdots, k \). Note that \( k = \max_w k_{\beta_w} \). For example, let us include \( \beta_p \) in \( R_l \) where \( l = 1, 2, \cdots, p \). Then, we can compute the angle of \( |\beta_p| \) to be \( (2(i+p)+1)\theta \). In addition, \( \varepsilon_{\alpha,k} \) and \( \varepsilon_{\beta_w,k} \) are the \( k \)-th error terms of \( |\alpha| \) and \( |\beta_w| \), respectively. The details of coefficients are mentioned in Appendix B.

The entire process of the algorithm is illustrated in Fig. 6. In this paper, we denote the \( l \)-th conditioned oracle as \( S_{w,l} \) and the \( l \)-th phase inversion as \( S_{w,l}^† \). Because of the definitions of the conditioned oracle and the phase inversion, \( S_{w,l}^† S_{w,l} \) is equal to \( S_l \). The details of \( r \rightarrow r′ \) transformation are presented in Section III-C.

Similar to Section III-A, there are two conditions for \( k \) iterations. First, the angle of \( |\alpha| \) must be smaller than \( \pi/2 \). Second, the minimum angle of \( |\beta_w| \) must be greater than zero.

The first condition is represented by

\[
(2(i + k) + 1)\theta \leq \frac{\pi}{2}
\]

where the angle of \( |\alpha| \) is induced from \( (i + k) \)-th iteration states. Equation (35) can be represented by

\[
i \leq \frac{\pi}{4\theta} - k - 1 \approx \frac{\pi}{4} \sqrt{N/m} - k - 1/2
\]

where \( \theta \) approximates \( \sqrt{m/N} \) in \( N \gg m \). The second condition is represented by

\[
(2(i + 2 \cdot \min_w k_{\beta_w} - k) + 1)\theta \geq 0,
\]

where the minimum angle of \( |\beta_w| \) is also induced from the \( (i + k) \)-th iteration states. Equation (37) can be easily represented by

\[
i \geq k - 2 \cdot \min_w k_{\beta_w} - 1/2.
\]

We can obtain the bound of the \( i \) value from (36) and (38) as follows:

\[
k - 2 \cdot \min_w k_{\beta_w} \leq i \leq \frac{\pi}{4} \sqrt{N/m} - k.
\]

If the left side of (39) has a negative value, the minimum value of \( i \) is zero. The maximum value of \( i \) is related to the complexity of the algorithm. If \( i \) has a maximum value, the complexity of this algorithm is equal to that of QSA, that is, \( O(\sqrt{N/m}) \).

When we know the number of solutions \( m \), we can estimate the value of \( \theta \). If we have an unknown value of \( m \), we use the method in [7]. The entire algorithm is summarized in Algorithm 2.

**Algorithm 2 Quantum Search Algorithm for Multiple-Weighted Solutions**

1: Initialize an integer \( k \) and \( k_{\beta_w} \) from problem settings such as a reward matrix in GCP.
2: If \( k \geq 2 \cdot \min_w k_{\beta_w} \), set \( q = k - 2 \cdot \min_w k_{\beta_w} \), otherwise \( q = 0 \).
3: Set \( f = q \) and \( \lambda = 8/7 \).
   (\( \lambda \) can be any value between 1 and 4/3)
4: Choose an integer \( i \) uniformly random such that \( q \leq i \leq j \).
5: Apply \( \prod_{l=1}^{j} K_l \cdot Q^l \) to the initial state \( |\psi⟩ \).
6: Measure the state \( K_k \cdots K_1 Q^j |\psi⟩ \) and let \( \psi \) be the outcome.
7: If \( f(\psi) = 1 \), the algorithm is over.
8: Otherwise, reset \( f \) to \( \min \{\lambda j, \frac{\pi}{4} \sqrt{N/m} - k\} \) and return to step 4.

\( v_p \) is colored in color \( c_q \). That is, in the propose algorithm, we can substitute the element of the reward matrix as an element of \( R_l \). However, if \( r_{pq} \) is not included in the integer, we need a transformation to make integer \( r_{pq} \).

Assume that the quantization level of \( r_{pq} \) is decided for \( L \) in advance. In addition, a quantization reward value is decided for \( f_j(i) \), where \( j = 1, \cdots, L \). Now, we define an \( \hat{r} \) matrix using \( f \hat{r} \) as follows:

\[
\hat{r}_{pq} = f \hat{r} \left( \arg \min_j |f_j(i) - r_{pq}| \right).
\]

Then, we compose an \( r' \) matrix from an \( \hat{r} \) matrix. The element of \( r'_{pq} \) is decided as an index of \( \hat{r}_{pq} \). In other words, \( r'_{pq} = j - 1 \), where \( \hat{r}_{pq} = f_j(i) \). Finally, we operate the quantum state with \( R_l \) in accordance with the value of the row in \( r' \). Because the iteration is conducted serially through matrix \( r' \), the angle of the maximum reward state may be different from that of \( |\alpha| \).

For example, we consider matrix \( r \) such that

\[
r = \begin{bmatrix} 6 & 1 & 7 \\ 3 & 4 & 8 \\ 2 & 9 & 5 \end{bmatrix}.
\]

This example is for a fully connected graph of three vertices. In addition, we have a quantum state with no probability, such as

\[
|100 \ 100 \ 100⟩, |100 \ 100 \ 010⟩, \cdots, |001 \ 001 \ 001⟩.
\]

We can transform matrix \( r \) to matrix \( \hat{r} \), when \( L = 2 \). \( f_2(1) = 4 \), \( f_2(2) = 7 \).

\[
r = \begin{bmatrix} 6 & 1 & 7 \\ 3 & 4 & 8 \\ 2 & 9 & 5 \end{bmatrix} \rightarrow \hat{r} = \begin{bmatrix} 7 & 4 & 7 \\ 4 & 4 & 7 \\ 4 & 7 & 4 \end{bmatrix}.
\]

Then, we can also transform matrix \( \hat{r} \) to matrix \( r' \) as follows:

\[
\hat{r} = \begin{bmatrix} 7 & 4 & 7 \\ 4 & 4 & 7 \\ 4 & 7 & 4 \end{bmatrix} \rightarrow r' = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.
\]
Subsequently, we first operate (42) with $R_l$ in accordance with $r_{0}^{'1}$, where
\[
R_1 = \{|100\cdots, 001\cdots\}. \tag{45}
\]
Because the quantum states containing the first vertex ($v_1$) colored with the first color ($c_1$) should be included in $R_1$ once, when $r_{11}^{'1} = 1$, (45) is decided. Using the same method, we obtain $R_2$ and $R_3$.
\[
R_2 = \{|\cdots 001\cdots\}, \quad R_3 = \{|\cdots 010\cdots\}. \tag{46}
\]
From (45) and (46), $|100\ 001\ 010\rangle$ has a maximum $k_{\omega}$ value, that is, $k_{100\ 001\ 010} = 3$. Therefore, we determined that $k = 3$. The number of iterations for reward-GCP is summarized in Table 4.

**D. THE COMPLEXITY**

Prior to quantum methods including QSA and the proposed algorithm, we can consider the complexity of classical methods. The classical methods such as the exhaustive search (brute-force search) or the backtracking algorithm have the complexity of $O(N_c^N_v)$ when we solve the GCP. On the other hand, in quantum sides, the complexity of QSA [5] is $O(\sqrt{N_c^N_v})$. Boyer-Brassard-Høyer-Tapp (BBHT) algorithm [7] and Dürr-Høyer (DH) algorithm [8] have the same complexity of QSA. However, the quantum heuristic algorithm [9] has lower complexity than QSA. The proposed algorithm with oracles which can be implemented at gate-level circuits has the complexity of $O(\sqrt{N_c^N_v})$. The comparison of the complexity according to each method is presented in Table 5.

**IV. EXPERIMENTAL RESULTS**

In this section, we compare the performance of the proposed algorithm with that of QSA and the classical exact algorithm in terms of finding the optimal solution quickly and accurately.

The simulation was conducted in MATLAB, and we used benchmark instances of GCP on the computational symposium website [24]. Because it is necessary to obtain all possible color combinations using the classical exact algorithm, we used GEOM20b and queen5_5 models, which have 20 and 25 vertices ($3^{20}$ and $5^{25}$ candidates), respectively, as well as two additional graph models with the density of edges changed. Table 6 describes the properties of the models, and the minimum number of colors in [25] was used.

**A. COMPARISON WITH CLASSICAL METHODS AND QUANTUM ALGORITHMS**

First, we measured the complexity of the actual GCP using simulations. In the simulation settings, we generated a random matrix having values with 0 to 20 for GEOM20b and GEOM20b_d5. A random matrix having values with less than 50 was used for queen5_5_d3 and queen5_5. A random matrix was used as the reward matrix $r'$ in Section III-C. Based on the solution with the maximum reward, which is the most interesting solution, we calculated the number of iterations required in the quantum algorithms as
\[
\sqrt{N}/p_m, \tag{47}
\]
where $p_m$ is the probability of the solution with the maximum reward. The maximum reward solution can be multiple. In this case, we calculated $p_m$ as the product of the number of solutions and $p_m$. We compared it with the complexity of finding the solution using the classical algorithm. The results are presented in Table 7 and Fig. 7. As a result of conducting simulations 10 times while changing the combination of the reward matrix, the complexity of QSA and the proposed algorithm were found to be much lower as the number of vertices increased or the density increased. Therefore, the complexity of QSA and the proposed algorithm is lower than...
that of the classical exact algorithm. The complexity of the QSA is similar to that of the proposed algorithm because the values of the elements on the reward matrix are small.

Meanwhile, we can calculate the expected reward when the classical exact algorithm, QSA, and the proposed algorithm are applied with the same complexity as $\sqrt{N}$. As we assumed that selecting the input data in the classical exact algorithm follows a uniform distribution, the probability of finding each solution is $m/N$. Let $p_u$ be the probability of the solution in the QSA. In addition, let $p_t$ be the probability of the $t$-th solution in the proposed algorithm. The expected reward of the classical exact algorithm is

$$
\sum_{t=1}^{m} U_{sum}(S_t) \times (m/N)
$$

and that of QSA is

$$
\sum_{t=1}^{m} U_{sum}(S_t) \times p_u.
$$

In addition, the expected reward of the proposed algorithm is

$$
\sum_{t=1}^{m} U_{sum}(S_t) \times p_t.
$$

Table 8 presents the expected rewards of the algorithms. As the number of vertices or density increases, the expected reward increases. In all cases, quantum methods have a higher expected reward.

Now, we present the relative difference of the algorithms in Table 9 and Fig. 8. If the average expected reward is $T$ and the maximum reward value is $T_{opt}$, then the relative difference is $1 - T/T_{opt}$ [14]. The relative difference is another performance parameter in the graph coloring approach. This means the difference of system utility given by the algorithms and the global optimum. Overall, the performance of the quantum algorithms is higher than that of the classical algorithm for the same complexity.

### B. COMPARISON WITH QUANTUM ALGORITHMS (RANDOM CASE)

Previously, we compared quantum algorithms and the classical algorithm in terms of their complexity and relative difference. However, for a small reward value under 50, the proposed algorithm is similar to QSA because the problem space is much larger than the reward value. A brief example for queen5_5 is denoted by Fig. 9. To determine the effect of the reward value, we applied the proposed algorithm and QSA to the same graph model (queen5_5) with multiple reward matrices from $1 \times 1$ to $800 \times 800$. The comparisons of complexity and relative difference are presented in Tables 10 and 11, respectively. In random reward matrix cases, the proposed algorithm has almost the same performance as QSA.
### TABLE 7. Average value of the complexity of applying the classical exact algorithm, QSA, and the proposed algorithm for graph models with 20 or 25 vertices (10 times).

| Model (\(N_v, \text{density}\)) | Complexity (Classical) | Complexity (QSA) | Complexity (Proposed) | Ratio (Proposed/Classical) |
|----------------------------------|------------------------|------------------|-----------------------|---------------------------|
| GEOM20b (20.0)                   | 1.162e+09              | 1.512e+07        | 1.494e+07             | 1.285e-02                 |
| GEOM20b_d5 (20.0.5)              | 3.076e+13              | 2.268e+09        | 2.268e+09             | 7.373e-05                 |
| queen5_5_d3 (25.0.3)             | 9.382e+15              | 8.724e+09        | 8.724e+09             | 9.298e-06                 |
| queen5_5 (25.0.5)                | 2.709e+17              | 1.191e+11        | 1.191e+11             | 4.396e-07                 |

### TABLE 8. Average expected reward when applying the classical exact algorithm, QSA, and the proposed algorithm for graph models with 20 or 25 vertices (10 times).

| Model (\(N_v, \text{density}\)) | Average expected reward (Classical) | Average expected reward (QSA) | Average expected reward (Proposed) | Ratio (Proposed/Classical) |
|----------------------------------|-------------------------------------|-------------------------------|-----------------------------------|---------------------------|
| GEOM20b (20.0.2)                 | 4.271e-02                           | 2.525e+02                     | 2.426e+02                         | 7.388e+01                 |
| GEOM20b_d5 (20.0.5)              | 1.376e-06                           | 2.531e+02                     | 2.531e+02                         | 1.356e+04                 |
| queen5_5_d3 (25.0.3)             | 5.940e-08                           | 6.870e+02                     | 6.870e+02                         | 1.075e+05                 |
| queen5_5 (25.0.5)                | 1.340e-10                           | 6.933e+02                     | 6.933e+02                         | 2.275e+06                 |

**FIGURE 7.** Plot showing the complexity for each model in Table 7.

**FIGURE 8.** Plot showing the relative difference for each model in Table 9.

### C. COMPARISON WITH QUANTUM ALGORITHMS (SPECIFIC CASE)

To increase the gain of the proposed algorithm, we simulated queen5_5 with a reward matrix having a specific value in the case of Section III-A. First, we divided the possible cases of the specific reward matrix into 4 instances. In these cases, the first row of the specific reward matrix has \([r'_b 0 0 0 0]\), \([r'_b r'_b 0 0 0]\), \([r'_b r'_b r'_b 0 0]\), or \([r'_b r'_b r'_b r'_b 0]\).

Specifically, we can explain the specific case of \([r'_b r'_b 0 0 0]\) using Fig. 10. In a practical system applied to graph queen5_5, we can consider a specific user with a high reward for spectrum 1 and 2. We can also consider that only user 1 has a gain for a specific resource, and that the remaining users have an equal gain regardless of the resources that they use. For example, user 1 must use a specific frequency because of the poor status of the channel (e.g., shadow area, inside buildings), but other users having a good status can use any frequency. Otherwise, user 1 is given priority over the desired frequency because user 1 has paid a lot of money. In these cases, we can quantize the reward of a specific user to \(r'_b\) and other rewards to 0. That is, the value of \(r'_{11}\) and \(r'_{12}\) is \(r'_b\), and others in the reward matrix are 0.

Now, we introduce a specific scenario employing the reward matrices of Fig. 10. This scenario is considered in the mobile communication engineering scenario [26], [27]. To explain the scenario, we suppose a few assumptions. First,
Y. Seo et al.: Quantum Search Algorithm for Weighted Solutions

FIGURE 9. A practical network example for queen5_5. Reward matrix values were changed for 10 times.

TABLE 9. Relative difference when applying the classical exact algorithm, QSA, and the proposed algorithm for graph models with 20 or 25 vertices (10 times).

| Model (N_v, density) | Relative difference (Classical) | Relative difference (QSA) | Relative difference (Proposed) |
|----------------------|---------------------------------|---------------------------|-------------------------------|
| GEOM20b (20,0.2)    | 1                               | 0.1828                    | 0.2149                        |
| GEOM20b, d5 (20,0.5) | 1                               | 0.2012                    | 0.2012                        |
| queen5_5, d3 (25,0.3) | 1                              | 0.1717                    | 0.1717                        |
| queen5_5 (25,0.5)   | 1                               | 0.1909                    | 0.1909                        |

TABLE 10. Average value of the complexity when applying QSA and the proposed algorithm for queen5_5 with multiple reward matrices (10 times).

| Multiple value of reward matrices | Complexity (QSA) | Complexity (Proposed) |
|-----------------------------------|------------------|-----------------------|
| ×1                                | 1.191087e+11     | 1.191087e+11          |
| ×100                              | 1.191087e+11     | 1.190541e+11          |
| ×200                              | 1.191087e+11     | 1.190210e+11          |
| ×800                              | 1.191087e+11     | 1.191157e+11          |

TABLE 11. Relative difference when applying QSA and the proposed algorithm for queen5_5 with multiple reward matrices (10 times).

| Multiple value of reward matrices | Relative difference (QSA) | Relative difference (Proposed) |
|-----------------------------------|----------------------------|-------------------------------|
| ×1                                | 0.199044                  | 0.199044                      |
| ×100                              | 0.199044                  | 0.199779                      |
| ×200                              | 0.199044                  | 0.190717                      |
| ×800                              | 0.199044                  | 0.192014                      |

TABLE 12. Complexity of applying QSA and the proposed algorithm for graph models with specific reward matrix.

| Model (N_v, density) | Complexity (QSA) | Complexity (Proposed) |
|----------------------|------------------|-----------------------|
| queen5_5, (25,0.5), type1 | 9.099e+08       | 8.565e+08             |
| queen5_5, (25,0.5), type2 | 2.730e+09       | 2.423e+09             |

TABLE 13. Relative difference when applying QSA, and the proposed algorithm for graph models with specific reward matrix.

| Model (N_v, density) | Relative difference (QSA) | Relative difference (Proposed) |
|----------------------|---------------------------|-------------------------------|
| queen5_5, (25,0.5), type1 | 0.3998                  | 0.3625                        |
| queen5_5, (25,0.5), type2 | 0.7998                  | 0.7745                        |

a base station (BS) intends to allocate a spectrum to multiple users with no interference. Second, each user sends a “channel report information” to BS periodically. The “channel report information” means channel qualities of a communication link. In Fig. 10, the “channel report information” is employed to reward matrix values. Third, only one user can report the “channel report information” at a time, and other users send nothing. In Fig. 10, since only user 1 sends the “channel report information”, user 1 has reward value r'_b and other users have a zero value. The third assumption can be considered in the case that the user 1 has a priority of spectrum assignment. Finally, BS assigns the spectrum for each user in consideration of the interference and “channel report information”. In Fig. 10, we set the reward matrix value of the user u_1 to [r'_b, 0, 0, 0]. The other users have the reward matrix value of [0, 0, 0, 0]. Consequently, the user u_1 has a high weight about the spectrum s_1 and s_2.

In Fig. 11, we compare the complexity of the QSA and the proposed algorithm. We calculated the complexity using (47). The main difference with the previous simulation (in Section IV-B) is that p_m is related to the number of solutions with the maximum reward value. In most of the previous simulations, the number of maximum reward solutions was mostly 1. Therefore, the number of solutions having the maximum reward value has an effect on the complexity, as shown in Fig. 11. The proposed algorithm decreases the complexity up to 43.6% compared to QSA in the 1:4 case. In addition, in all the cases, the proposed algorithm reduces the complexity by at least 14.7%. Consequently, we can see that the proposed algorithm has a higher gain in complexity than QSA in accordance with a higher reward value.

In Fig. 12, we compare QSA and the proposed algorithm with the relative difference. As opposed to Fig. 11, the proposed algorithm has a better performance in 4:1 rather than 1:4. The proposed algorithm has a lower relative difference up to 69% in comparison with QSA. As a result, we observed a low relative difference of the proposed algorithm compared to QSA in accordance with a higher reward value.
FIGURE 10. Practical system is applied to the graph queen5_5. Only user 1 has a reward value for spectrum 1 and 2. The reward value is $r_{b}'$. Considering the communication system with a high-priority(paying more) user or a user with better channel conditions, we can compose a graph $G$.

FIGURE 11. The graph shows the complexity for queen5_5 in 4 cases. The reward value in the graph is used as the value of $r_{b}'$. In addition, we progressively simulated the ratio of $r_{b}'$ to 0 in the first row of the reward matrix, i.e., 1:4(Proposed) indicate the number of $r_{b}'$:0 in the reward matrix by applying the proposed algorithm.

FIGURE 12. The graph shows the relative difference for queen5_5 in 4 cases. The simulation settings of the horizontal axis and legend are identical to those in Fig. 11.

We also simulated queen5_5 with a reward matrix that has a specific value in the case of Section III-B. Each element of the reward matrices has one of $r_{b}', r_{c}'$, 0, where $r_{b}' = 5 \times 10^6$ and $r_{c}' = 5 \times 10^3$. At the same time, the values of the reward matrix are arranged such that the set of solutions is divided

FIGURE 13. A communication system that is applied to the type 1 of queen5_5.

FIGURE 14. The graph shows the complexity for each model in Table 12.
In type 2, the first row of the reward matrix has second and third groups, the sums of rewards are 

In this paper, we presented a QSA that can make the weighted solution states. The proposed algorithm has the same complexity as the QSA, BBHT-QSA, and DH-QSA. In addition, we proposed a method that solves the reward-OCP using the proposed algorithm. Then, we compared the classical exact algorithm and quantum algorithms in terms of the complexity and relative difference. Consequently, quantum algorithms have lower complexity and relative difference than the classical exact algorithm. In addition, we compared the QSA and the proposed algorithm in terms of the complexity and relative difference. In Tables 10-13 and Figs. 11, 12, 14, 15, we also compared QSA and the proposed algorithm with diverse cases. Based on the results, we demonstrated that the proposed algorithm has a higher expected reward than QSA in some cases.

APPENDIX A
THE ERROR TERMS IN SIMPLE CASE OF THE PROPOSED ALGORITHM
In this section, we focus on the error terms of \( |\alpha\rangle \), \( |\beta\rangle \), and \( |\gamma\rangle \) in simple case of the proposed algorithm. First, we considered \( K |\alpha\rangle \),

\[
K |\alpha\rangle = -S_h H S_0 H^\dagger S_g |\alpha\rangle \\
= \cos 2\theta |\alpha\rangle + \sin 2\theta \sin \phi |\beta\rangle - \sin 2\theta \cos \phi |\gamma\rangle .
\]

(51)

In the second, we consider \( K |\beta\rangle \),

\[
K |\beta\rangle = -S_h H S_0 H^\dagger S_g |\beta\rangle \\
= - \sin 2\theta \sin \phi |\alpha\rangle + (1 - 2 \sin^2 \theta \sin^2 \phi) |\beta\rangle + 2 \sin^2 \theta \sin \phi \cos \phi |\gamma\rangle .
\]

(52)

And then, we consider \( K |\gamma\rangle \),

\[
K |\gamma\rangle = -S_h H S_0 H^\dagger S_g |\gamma\rangle \\
= \sin 2\theta \cos \phi |\alpha\rangle + 2 \sin^2 \theta \sin \phi \cos \phi |\beta\rangle + (1 - 2 \sin^2 \theta \cos^2 \phi) |\gamma\rangle .
\]

(53)

From the results of implementing operator \( K \) to the basis states, we obtain \(|\psi_{i,1}\rangle\) by applying \( K \) to \(|\psi_i\rangle\) once.

\[
|\psi_{i,1}\rangle = \left\{ \cos (2i + 3)\theta + \epsilon_{a,1} \right\} |\alpha\rangle \\
+ \left\{ \sin (2i + 3)\theta \sin \phi + \epsilon_{\beta,1} \right\} |\beta\rangle \\
+ \left\{ \sin (2i - 1)\theta \cos \phi + \epsilon_{\gamma,1} \right\} |\gamma\rangle
\]

(54)

where \( \epsilon_{a,1}, \epsilon_{\beta,1}, \) and \( \epsilon_{\gamma,1} \) are the error terms of \(|\alpha\rangle, |\beta\rangle, \) and \(|\gamma\rangle\), respectively. The error terms in (54) are represented as (55), (56), and (57).

\[
\epsilon_{a,1} = 2 \cos^2 \phi \sin 2\theta \sin (2i + 1)\theta .
\]

(55)

\[
\epsilon_{\beta,1} = 4 \cos^2 \phi \sin^2 \theta \sin (2i + 1)\theta \cdot \sin \phi .
\]

(56)

\[
\epsilon_{\gamma,1} = 4 \sin^2 \phi \cos^2 \theta \sin (2i + 1)\theta \cdot \cos \phi .
\]

(57)

Now, we consider the effect of the error terms. When we apply \( K \) to \(|\psi_{i,1}\rangle\), we also represent \(|\psi_{i,2}\rangle\) using \(|\alpha\rangle, |\beta\rangle, \) and \(|\gamma\rangle\).

\[
|\psi_{i,2}\rangle = \left\{ \cos (2i + 5)\theta + \epsilon_{a,2} \right\} |\alpha\rangle \\
+ \left\{ \sin (2i + 5)\theta \sin \phi + \epsilon_{\beta,2} \right\} |\beta\rangle \\
+ \left\{ \sin (2i - 3)\theta \cos \phi + \epsilon_{\gamma,2} \right\} |\gamma\rangle
\]

(58)

where \( \epsilon_{a,2}, \epsilon_{\beta,2}, \) and \( \epsilon_{\gamma,2} \) are the error terms of \(|\alpha\rangle, |\beta\rangle, \) and \(|\gamma\rangle\) from \(|\psi_{i,2}\rangle\), respectively. The details of \( \epsilon_{a,2} \) are represented in (59).

\[
\epsilon_{a,2} = \cos 2\theta \cdot \epsilon_{a,1} + \sin 2\theta \epsilon_{\gamma,1} \cos \phi - \epsilon_{\beta,1} \sin \phi \\
+ \cos^2 \phi \sin 2\theta \sin (2i + 3)\theta + \sin (2i - 1)\theta \cdot \cos \phi .
\]

(59)

Because \( \epsilon_{\beta,1} \) and \( \epsilon_{\gamma,1} \) are significantly smaller than the other terms in (59), we can ignore the second term on the right side in (59). If a space of the problem is significantly more than the number of solutions \( (N \gg m) \), then \( \cos 2\theta \) approximates to “one.” In this case, we can summarize (59) as follows:

\[
\epsilon_{a,2} \simeq \epsilon_{a,1} + \cos^2 \phi \sin 2\theta \sin (2i + 3)\theta + \sin (2i - 1)\theta .
\]

(60)

By employing the same approximation method, we can also represent \( \epsilon_{\beta,2} \) and \( \epsilon_{\gamma,2} \) as follows.

\[
\epsilon_{\beta,2} \simeq \epsilon_{\beta,1} + \sin 2\theta \sin \phi \cdot \epsilon_{a,1} \\
+ 2 \sin^2 \theta \cos^2 \phi \sin \phi \sin (2i - 1)\theta - \sin (2i + 3)\theta .
\]

(61)
\( \varepsilon_{y,2} \simeq \varepsilon_{y,1} - \sin 2\theta \cos \phi \cdot \varepsilon_{a,1} \\
+ \cos \phi \sin 2\theta \{ \cos (2i - 1)\theta - \cos (2i + 3)\theta \} \\
+ 2\sin^2 \theta \sin^2 \phi \cos \phi \{ \sin (2i + 3)\theta - \sin (2i - 1)\theta \}. \) (62)

When we apply \( K \) to \( |\psi_i\rangle \) \( k \) times, we can obtain
\[
|\psi_{i,k}\rangle = K^k |\psi_i\rangle = \left[ \cos (2(i + k) + 1)\theta + \varepsilon_{a,k} \right] |\alpha\rangle \\
+ \left[ \sin (2(i + k) + 1)\theta \sin \phi + \varepsilon_{\phi,k} \right] |\beta\rangle \\
+ \sin (2(i - k) + 1)\theta \cos \phi + \varepsilon_{y,k} |\gamma\rangle \] (63)

where \( \varepsilon_{a,k}, \varepsilon_{\phi,k}, \) and \( \varepsilon_{y,k} \) are denoted by (64), (65), and (66), respectively.

\[
\varepsilon_{a,k} \simeq \cos^2 \phi \sin 2\theta \sum_{j=0}^{k-1} \left[ \sin (2(i + j) + 1)\theta \\
+ \sin (2(i - j) + 1)\theta \right]. \) (64)
\[
\varepsilon_{\phi,k} \simeq \sum_{j=1}^{k-1} \sin 2\theta \sin \phi \cdot \varepsilon_{a,j} \\
+ 2\sin^2 \theta \cos^2 \phi \sin \phi \sum_{j=0}^{k-1} \left[ \sin (2(i + j) + 1)\theta \\
+ \sin (2(i - j) + 1)\theta \right]. \) (65)
\[
\varepsilon_{y,k} \simeq \cos \phi \sin 2\theta \sum_{j=1}^{k-1} \left[ \cos (2(i + j) + 1)\theta \\
- \cos (2(i - j) + 1)\theta \right] - \sum_{j=1}^{k-1} \sin 2\theta \cos \phi \cdot \varepsilon_{a,j} \\
+ 2\sin^2 \theta \sin^2 \phi \cos \phi \sum_{j=0}^{k-1} \left[ \sin (2(i + j) + 1)\theta \\
+ \sin (2(i - j) + 1)\theta \right]. \) (66)

Equations (65) and (66) are defined in \( k \geq 2. \)

**APPENDIX B**

**THE ERROR TERMS IN MULTIPLE CASE OF THE PROPOSED ALGORITHM**

In this section, we focus on the error terms of \( |\alpha\rangle \) and \( |\beta_w\rangle \) in multiple case of the proposed algorithm. We first apply \( Q \) to \( |\psi\rangle \) \( t \) times, and then we have
\[
Q^t |\psi\rangle = \cos (2i + 1)\theta |\alpha\rangle + \sin (2i + 1)\theta \sum_{w=1}^{t} b_w |\beta_w\rangle. \) (67)

To avoid confusion of the notation, we consider \( Q^t \) as \( K_1 \cdots K_2 K_1, \) where \( R_i = \{ \beta_1, \beta_2, \ldots, \beta_t \} \) for \( i = 1, \ldots, i. \) In this case, there are no error terms of \( Q^t |\psi\rangle, \) i.e., \( \varepsilon_{a,l} = \varepsilon_{\phi,l} = \varepsilon_{y,l} = 0 \) for \( l = 1, \ldots, i \) and \( w = 1, \ldots, t. \)

Now, assume that \( R_{i+1} = \{ \beta_1, \beta_2, \ldots, \beta_p \}, \) where \( p < t. \) After applying \( Q^t, \) we apply \( K_{i+1} \) to \( Q^t |\psi\rangle. \) \( K_{i+1}Q^t |\psi\rangle \) is considered in three parts: \( |\alpha\rangle, \beta_w \in R_{i+1}, \beta_w \notin R_{i+1}. \)

First, we consider \( K_{i+1} |\alpha\rangle \) as follows:
\[
K_{i+1} |\alpha\rangle = -S_{w,1}^{t}H_{s}H_{1}^{t}S_{w,i+1} |\alpha\rangle \\
= \cos 2\theta |\alpha\rangle + \sin 2\theta \sum_{w=1}^{p} b_w |\beta_w\rangle \\
- \sin 2\theta \sum_{w=p+1}^{t} b_w |\beta_w\rangle. \] (68)

Solution states should be considered in two ways. We assume that \( \beta_m \) is an element of \( R_{i+1} = \{ \beta_1, \ldots, \beta_m, \ldots, \beta_p \}. \)
\[
K_{i+1} |\beta_m\rangle = -S_{w,1}^{t}H_{s}H_{1}^{t}S_{w,i+1} |\beta_m\rangle \\
= -\sin 2\theta \cdot b_m |\alpha\rangle \\
- 2\sin^2 \theta \cdot b_m \sum_{w=1, w \neq m}^{p} b_w |\beta_w\rangle \\
+ (1 - 2\sin^2 \theta \cdot b_m^2) |\beta_m\rangle \\
+ 2\sin^2 \theta \cdot b_m \sum_{w=p+1}^{t} b_w |\beta_w\rangle. \] (69)

Then, we consider \( |\beta_n\rangle, \) which is not the element of \( R_{i+1}. \)
\[
K_{i+1} |\beta_n\rangle = -S_{w,1}^{t}H_{s}H_{1}^{t}S_{w,i+1} |\beta_n\rangle \\
= \sin 2\theta \cdot b_n |\alpha\rangle \\
+ 2\sin^2 \theta \cdot b_n \sum_{w=1}^{p} b_w |\beta_w\rangle \\
+ (1 - 2\sin^2 \theta \cdot b_n^2) |\beta_n\rangle \\
- 2\sin^2 \theta \cdot b_n \sum_{w=p+1, w \neq n}^{t} b_w |\beta_w\rangle \] (70)

where \( p + 1 \leq n \leq t. \) These computations are used when calculating the coefficient of the states.

When we apply \( K_{i+1} \) to \( Q^t |\psi\rangle, \) the coefficients of \( |\alpha\rangle \) and \( |\beta_n\rangle \) are denoted as follows:
\[
K_{i+1}Q^t |\psi\rangle \\
= \{ \cos (2i + 3)\theta + \varepsilon_{a,i+1} \} |\alpha\rangle \\
+ \sum_{n=1}^{p} \sin (2i + 3)\theta + \varepsilon_{\phi,n,i+1} \} \cdot b_m |\beta_m\rangle \\
+ \sum_{n=p+1}^{t} \{ \sin (2i - 1)\theta + \varepsilon_{y,n,i+1} \} \cdot b_n |\beta_n\rangle \] (71)

where \( \beta_m \in R_{i+1} \) and \( \beta_n \notin R_{i+1}. \) Here, we assume the unintended terms of coefficients as error terms such as \( \varepsilon_{a,i+1}, \varepsilon_{\phi,w,i+1}, \) and \( \varepsilon_{y,w,i+1}. \) The error terms are represented as follows:
\[
\varepsilon_{a,i+1} = 2 \sin 2\theta \sin (2i + 1)\theta \sum_{w=p+1}^{t} b^2_w \] (72)
\[
\varepsilon_{\phi,w,i+1} = 4 \sin^2 \theta \sin (2i + 1)\theta \sum_{w=p+1}^{t} b^2_w \] (73)
\[
e_{\alpha_{i+1}} = 4\sin^2 \theta \sin(2i + 1)\theta \sum_{w=1}^{p} b_w^2
\]

where \(m = 1, \cdots, p\) and \(n = p + 1, \cdots, l\).

Using the same method as in Appendix A, \(e_{\alpha,k}\) in (34) can be denoted by

\[
e_{\alpha,k} \approx \sin 2\theta \left[ \sum_{i=1}^{k} \left( \sin \{2(i + (l - 1)) + 1\right)\theta + \sum_{\beta_i \in R_l} \sin \beta_{i,0} \cdot b_{i,0} - \sum_{\beta_i \in R_l} \sin \beta_{i,1} \cdot b_{i,1} \right]
\]

where \(\beta_{i,j}\) is the \(j\)-th angle, \(\beta_{i,0}\). The error term \(\epsilon_{\beta_{i,k}}\) is also ignored with a low probability of \(|\alpha|\) because \(\epsilon_{\beta_{i,k}}\) is much smaller than \(e_{\alpha,k}\).

REFERENCES

[1] D. Deutsch, “Quantum theory, the church-turing principle and the universal quantum computer,” Proc. Roy. Soc. London A, Math. Phys. Sci., vol. 400, no. 1818, pp. 97–117, Jul. 1985.

[2] 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.

[3] D. R. Simon, “On the power of quantum computation,” in Proc. 35th Annu. Symp. Found. Comput. Sci., 1994, pp. 124–134, doi: 10.1109/SFCS.1994.365700.

[4] P. W. Shor, “Algorithms for quantum computation: Discrete logarithms and factoring,” in Proc. 35th Annu. Symp. Found. Comput. Sci., 1994, pp. 124–134, doi: 10.1109/SFCS.1994.365700.

[5] L. K. Grover, “A fast quantum mechanical algorithm for database search,” in Proc. 28th Annu. ACM Symp. Theory Comput. (STOC), 1996, pp. 212–219.

[6] A. W. Harrow, A. Hassidim, and S. Lloyd, “Quantum algorithm for linear systems of equations,” Phys. Rev. Lett., vol. 103, no. 15, Oct. 2009, Art. no. 150502.

[7] M. Boyer, G. Brassard, P. Høyer, and A. Tapp, “Tight bounds on quantum search,” Fortschrritte Physik, vol. 46, nos. 4–5, pp. 493–505, Jun. 1998.

[8] C. Dürr and P. Høyer, “A quantum algorithm for finding the minimum,” Jul. 1996, arXiv:quant-ph/9607014.

[9] T. Hogg, “Quantum search heuristics,” Phys. Rev. A, Gen. Phys., vol. 61, no. 5, Apr. 2000, Art. no. 052311.

[10] A. Fabrikant and T. Hogg, “Graph coloring with quantum heuristics,” in Proc. 18th Nat. Conf. Artif. Intell., 2002, pp. 22–27, [Online]. Available: https://www.aaai.org/.library/AAAI2002/aaai02-004.pdf.

[11] M. Furer, “Solving NP-complete problems with quantum search,” in LATIN 2008: Theoretical Informatics, vol. 4957, Berlin, Germany: Springer, 2008.

[12] Y. Seo, Y. Kang, J. Park, and J. Heo, “A complexity reduction method of quantum search algorithm applied to singlegraph coloring problem,” presented at the KICS Summer Conf., Aug. 2020.

[13] L. Zhang, M. Xiao, G. Wu, M. Alam, Y.-C. Liang, and S. Li, “A survey of advanced techniques for spectrum sharing in 5G networks,” IEEE Wireless Commun., vol. 24, no. 5, pp. 44–51, Oct. 2017, doi: 10.1109/MWC.2017.700069.

[14] M. Z. Chowdhury, M. Shahjalal, S. Ahmed, and Y. M. Jang, “6G wireless communication systems: Applications, requirements, technologies, challenges, and research directions,” IEEE Open J. Commun. Soc., vol. 1, pp. 957–975, 2020, doi: 10.1109/OJCOMS.2020.3010270.

[15] M. Matinmikko-Blue, S. Yrgjola, and P. Aholangas, “Spectrum management in the 6G era: The role of regulation and spectrum sharing,” in Proc. 2nd 6G Wireless Summit (6G SUMMIT), 2020, pp. 1–5, doi: 10.1109/6GSUMMIT49458.2020.9083851.

[16] R. M. R. Lewis, A Guide to Graph Colouring, Berlin, Germany: Springer, 2015.

[17] L. K. Grover, “Quantum computers can search rapidly by using almost any transformation,” Phys. Rev. Lett., vol. 80, no. 19, pp. 4329–4332, May 1998, doi: 10.1103/PhysRevLett.80.4329.

[18] M. A. Trick. (2012). Computational Symposium: Graph Coloring and Its Generalizations. [Online]. Available: http://mat.gsia.cmu.edu/COLOR03

[19] A. Lim, X. Zhang, and Y. Zha, “A hybrid methods for the graph coloring and related problems,” presented at the 5th Metaheuristics Int. Conf. (MIC), 2003.

[20] Y. Seo et al.: Quantum Search Algorithm for Weighted Solutions

[21] YOUNGJIN SEO received the B.S. degree in electrical engineering from Korea University, Seoul, South Korea, in 2018, and he is currently pursuing the Ph.D. degree with the School of Electrical Engineering.

His research interests include quantum algorithm and quantum random number generation.

[22] YUJIN KANG received the B.S. degree in electrical engineering from Konkuk University, Seoul, South Korea, in 2018. She is currently pursuing the Ph.D. degree with the School of Electrical Engineering, Korea University, Seoul.

From 2017 to 2019, she was a Research Engineer in digital design verification part at Silicon Works Company, Seoul. Her research interest includes quantum algorithm.

[23] JUN HEO (Member, IEEE) received the B.S. and M.S. degrees in electronics engineering from Seoul National University, Seoul, South Korea, in 1989 and 1991, respectively, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 2002.

From 1991 to 1997, he was a Senior Research Engineer with LG Electronics Company Inc. He is currently a Professor with the School of Electrical Engineering, Korea University, Seoul. His research interests include channel coding theory and quantum communication.