

Received October 12, 2021; revised February 8, 2022; accepted February 8, 2022; date of publication February 14, 2022; date of current version March 11, 2022.

Digital Object Identifier 10.1109/TQE.2022.3151137

A Grover Search-Based Algorithm for the List Coloring Problem

SAYAN MUKHERJEE 

Department of Mathematics, Statistics and Computer Science, University of Illinois Chicago, Chicago IL 60608 USA
blueqat Co. Ltd., Tokyo 150-0002, Japan
(e-mail: sayan@blueqat.com)

ABSTRACT Graph coloring is a computationally difficult problem, and currently the best known classical algorithm for k -coloring of graphs on n vertices has runtimes $\Omega(2^n)$ for $k \geq 5$. The list coloring problem asks the following more general question: given a *list* of available colors for each vertex in a graph, does it admit a proper coloring? We propose a hybrid classical-quantum algorithm based on Grover search 12 to quadratically speed up exhaustive search. Our algorithm loses in complexity to classical ones in specific restricted cases, but improves exhaustive search for cases, where the lists and graphs considered are arbitrary in nature.

INDEX TERMS Graph coloring, Grover search, hybrid algorithm.

I. INTRODUCTION

Graph coloring problems provide for a rich family of NP-complete problems in theoretical computer science. While exhaustive search is believed to be the fastest classical approach for several NP-complete problems including satisfiability and hitting set [8], there are much better classical algorithms using dynamic programming, inclusion–exclusion, and other structural approaches for problems such as graph coloring [4], [10], [18], the traveling salesman problem [13], [24], set cover [15], etc. Several authors have obtained quantum speedup on these classical algorithms [2], [30], [33]; however, all of these algorithms have the limitation that they cannot be easily generalized to the list coloring problem.

Given a finite graph $G = (V, E)$, a proper coloring of G is a function $\chi : V \rightarrow \mathbb{N}$ such that for every edge $uv \in E$, $\chi(u) \neq \chi(v)$. The list coloring problem tries to determine a proper coloring χ of a graph $G = (V, E)$, given a list L_v of available colors for each vertex v . In other words, it is forced that $\chi(v) \in L_v$. When $L_v = \{1, 2, \dots, k\}$ for every vertex v this reduces to the well-studied k -coloring problem. We propose a simple Grover search-based approach to obtain a quadratic speedup on exhaustive search for the list coloring problem.

Grover's algorithm [12] is known to speed up unstructured search quadratically using the technique of *amplitude amplification*. In its simplest form, to find some marked elements from a list of $N = 2^n$ entries, the algorithm starts with a uniform quantum superposition of all 2^n basis states of an n -qubit register. It then amplifies the amplitudes of the

searched state and reduces those of the other states, such that a measurement of the n qubits leads to one of the searched states with high probability.

Grover's algorithm has been used to obtain quantum speedups for various problems in combinatorial optimization and computer science (see, for example, [16], [17], [19], [22], [34]). Needless to say, graph coloring problems are also not an exception in the literature, and have been attacked using quantum annealing [21], [36], hybrid approaches [6], [35], as well as using Grover search [32], [33], [39].

In [39], a qutrit-based approach has been used to demonstrate the cost efficiency of ternary quantum logic; however, their main algorithm is not realizable right now on NISQ devices. The algorithm of [33] has the same issue as it requires quantum RAM, which has not been realized at this moment. On the other hand, the authors of [32] and [31] demonstrate a quantum algorithm solving the k -coloring problem on NISQ devices, comparing the efficiency of their algorithm against the reduction of 3-SAT to 3-coloring approach of Hu *et al.* [14].

All of these algorithms use an oracle design, which uses binary comparators, and provide solutions, where almost all binary strings have positive probabilities of being selected, including those that do not represent valid colorings. Our approach circumvents this problem via a modified initialization and diffusion operator that restricts the evolution of the quantum algorithm to the only $\prod_{v \in L_v} |L_v|$ plausible states. Note that, this is the total number of valid colorings when the underlying graph is empty. We achieve this via the restricted version of Grover search [11], [12].

Proposition 1 (Restricted Grover Search): Let $S \subseteq \{1, 2, \dots, 2^n - 1\}$, and suppose $S' \subsetneq S$ is a set of marked states. Let O be an oracle that marks these states and requires a ancillas. Then, there is a quantum circuit on $n + a + 1$ qubits, which makes $O(\sqrt{2^n/|S'|})$ queries, which when measured, gives one of the marked states with high probability. Further, states outside S are never measured.

Additionally, we use an oracle design different from those in [32] and [39], and give a classical algorithm in Section III that can reduce the complexity of this oracle in several special cases (such as for the 3-coloring or 4-coloring problems). As a corollary of Proposition 1, our main theorem provides an algorithm for the list coloring problem.

Theorem 2 (Quantum List Coloring Algorithm): Given a graph $G = (V, E)$ on n vertices and m edges and lists of available colors $\{L_v : v \in V\}$, there exists a $(\sum_{v \in V} \lceil \log_2 |L_v| \rceil + m + 1)$ -qubit quantum algorithm with query complexity $O(\prod_{v \in V} |L_v|^{1/2})$ that returns a valid list coloring of G with high probability.

This article is organized as follows. In Section II, we describe Grover’s algorithm and a gate-level implementation. Section III is devoted to tackling the list coloring problem, and proves Theorem 2. In Section IV, we run experiments on classical simulators as well as real quantum machines, and compare the outcomes. Finally, Section V concludes this article.

II. GROVER’S ALGORITHM

In this section, we provide a concise exposition on Grover search. The main idea behind Grover search is to amplify the amplitudes of some number of marked states (states that are being searched for), and consequentially decrease that of unmarked states. Grover’s algorithm requires three different operators: Initialization, oracle, and diffusion. Below we present two formulations of the algorithm.

A. UNRESTRICTED SEARCH SPACE

When searching for a marked state among the full search space $S = \{0, 1\}^n$, the initialization step of the algorithm creates a uniform superposition of all the possible states of an n -qubit system. This is achieved via appending Hadamard gates on each qubit

$$H^{\otimes n}|0\rangle_n = |+\rangle^n = \frac{1}{\sqrt{2^n}} \sum_{i=1}^n |i\rangle_n.$$

Here we abuse notation and write $|i\rangle_n$ to denote the state corresponding to an n -digit binary representation of i .

Next, Grover’s algorithm requires an oracle O that, given a uniform superposition of all 2^n possible states, can change the sign of the marked states. Let $S' \subseteq \{0, 1, \dots, 2^n - 1\}$ be a set of marked states. The oracle O then switches the signs of the states in S' , i.e.,

$$O|i\rangle_n = \begin{cases} |i\rangle_n, & i \notin S' \\ -|i\rangle_n, & i \in S'. \end{cases}$$

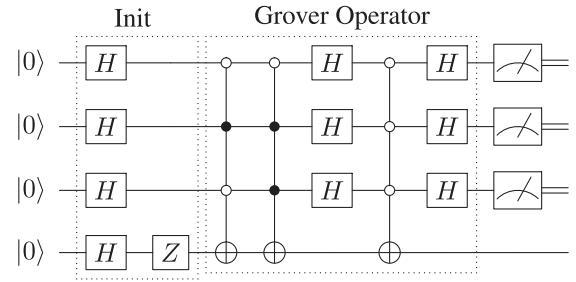


FIGURE 1. Example of Grover search implementation. In this standard circuit implementation of Grover search, the initialization is achieved by the Hadamard operator $H^{\otimes 3}$. Phase kickback from the fourth qubit initialized to the state $|-\rangle$ is used to negate the amplitudes of $|010\rangle$ and $|011\rangle$. Finally, diffusion is achieved via another phase kickback from the same qubit.

The circuit implementation of the oracle O usually is the most difficult (and computationally expensive) part of the algorithm, and one of the most basic implementations requires the usage of phase kickback [7].

The final component of Grover’s algorithm is the diffusion operator D , which can be thought of as a reflection around the vector $|0\rangle^n$. As an operator, we have

$$D = 2|0\rangle_n\langle 0|_n - I.$$

D is usually implemented using phase kickback in the same fashion as the oracle O .

Grover’s algorithm requires repeated usage of the operator $G = H^{\otimes n}DH^{\otimes n}O$, which has the net effect of reflecting around $|+\rangle^n$, amplifying the amplitudes of marked states and decreases those of other states. Measuring the state $G^r H^{\otimes n}|0\rangle_n$ ($r \geq 1$) gives one of the marked states with high probability, and this probability is maximum when $r = \lfloor \frac{\pi}{4} \sqrt{2^n/|S'|} \rfloor$. Since $|S'|$ is not known in general, the r is either randomly selected [5], [9], or is estimated using quantum counting algorithms [1], [26].

See Fig. 1 for an example of a circuit implementing unrestricted Grover search with $n = 3$, $S' = \{|010\rangle_n, |011\rangle_n\}$, $S = \{0, 1, \dots, 7\}$.

B. RESTRICTED SEARCH SPACE

Let us now consider a search space $S \subsetneq \{0, 1\}^n$. In this case, the algorithm is designed to only evolve over the states of S , and this is achieved via an initialization operator A , such that

$$A|0\rangle_n = \frac{1}{\sqrt{|S|}} \sum_{i \in S} |i\rangle_n.$$

And the Grover operator is changed to a reflection around $A|0\rangle_n$ instead of $|+\rangle^n$

$$G = ADA^\dagger O.$$

The usage of ADA^\dagger instead of $H^{\otimes n}DH^{\otimes n}$ makes sure that the evolution of the quantum states remains in the subspace spanned by S instead of the entire space $\{0, 1\}^n$, and this leads to probability distribution of the measured outcomes being supported on the state S .

The only detail missing in this formulation is the construction of the initialization operator A . As we shall see in Section III, for graph coloring problems (and most applications in general), A can be represented as a block matrix and can be implemented in time linear in the number of qubits.

We make a remark here that the graph coloring algorithm of [32] uses the unrestricted formulation of Grover's algorithm, but modifies the oracle to discard states that represent invalid colorings. On the other hand, they do not modify their diffusion operator, leading to states outside the search space having positive probabilities of being measured.

III. QUANTUM LIST COLORING ALGORITHM

Our goal in this section is to prove Theorem 2. For the remainder of this section, assume that $G = (V, E)$ is an arbitrary graph with $|V| = n$, $|E| = m$. Further, for every vertex v , let L_v denote the list of admissible colors for vertex v . Then the i th admissible color of the vertex v is denoted as $L_v[i]$. As our algorithm is based on Grover search, we shall discuss the four basic steps of the algorithm: Circuit setup, initialization, oracle, and diffusion, each in their respective subsections.

A. SETUP AND QUBIT LABELS

Our algorithm design requires three different qubit registers as follows.

- 1) A *vertex register* to keep track of vertex colors. For each vertex v , we require the usage of $\lceil \log_2 |L_v| \rceil$ qubits to represent each color in L_v . Let us denote $j_v := \lceil \log_2 |L_v| \rceil$, and let the qubits corresponding to vertex $v \in V$ be labeled by $q_v^1, \dots, q_v^{j_v}$.
- 2) An *edge register* consisting of m qubits, one corresponding to each edge. Let q_{uv}^i denote the qubit corresponding to an edge $uv \in E$.
- 3) A single qubit *ancilla register*, used for phase kickback in Grover's algorithm. Let q^* denote this ancilla.

The total number of qubits required is $\sum_{v \in V} j_v + m + 1$. Now we take a closer look at our list coloring algorithm.

B. INITIALIZATION

For each $v \in V$, we initialize qubits $q_v^1, \dots, q_v^{j_v}$ to a uniform superposition $\chi_v = \frac{1}{\sqrt{|L_v|}} \sum_{i=0}^{|L_v|-1} |i\rangle$. This can be achieved via a unitary operator U_v , such that

$$U_v|0\rangle = \frac{1}{\sqrt{|L_v|}} \sum_{i=0}^{|L_v|-1} |i\rangle.$$

We show one way of constructing the operator U_v . First, consider the standard basis $B = \{|i\rangle : i \in \{0, 1, \dots, 2^{j_v} - 1\}\}$. We shall replace any one entry $|i\rangle$ with χ_v , where $i \in L_v$: Let $B' = \{\chi_v\} \cup (B \setminus \{|i\rangle\})$. It can be seen that $\text{Span}(B) = \text{Span}(B')$. We can now consider B' as an ordered basis with its first entry as χ_v , and apply the Gram–Schmidt process to turn B' into an orthonormal basis B'' [29]. Note that as $\|\chi_v\| = 1$, it remains unchanged in B'' . The transpose of the coefficients of the vectors in B'' constitutes a change of basis operator that

maps $|0\rangle$ to χ_v , and this is how one can construct the matrix for U_v . It can then be implemented with quantum gates using the results of [23], [27], and [37], for example.

For the sake of clarity of the above procedure, let us consider an example with $j_v = 2$ and $L_v = \{0, 1, 2\}$. Note that $\chi_v = \frac{1}{\sqrt{3}}(|0\rangle + |1\rangle + |2\rangle)$. Then, $B = \{|0\rangle, |1\rangle, |2\rangle, |3\rangle\}$, and we can take $B' = \{\chi_v, |1\rangle, |2\rangle, |3\rangle\}$. After the Gram–Schmidt process, we obtain

$$B'' = \left\{ \frac{1}{\sqrt{3}}|0\rangle + \frac{1}{\sqrt{3}}|1\rangle + \frac{1}{\sqrt{3}}|2\rangle - \frac{1}{\sqrt{6}}|0\rangle + \frac{\sqrt{2}}{\sqrt{3}}|1\rangle - \frac{1}{\sqrt{6}}|2\rangle - \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|2\rangle, |3\rangle \right\}.$$

Hence, in this case, we get

$$U_v = \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & 0 \\ -\frac{1}{\sqrt{6}} & \frac{\sqrt{2}}{\sqrt{3}} & -\frac{1}{\sqrt{6}} & 0 \\ -\frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^T.$$

Observe that by construction, $U_v|0\rangle = U_v \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T = \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & 0 \end{bmatrix}^T$, as desired.

Finally, let us denote $A = \bigotimes_{v \in V} U_v$. Then

$$A \otimes \bigotimes_{e \in E} I \otimes ZH \quad (\text{III.1})$$

is the full initialization operator applied to the circuit starting from $|0\rangle_{\sum_{v \in V} j_v} \otimes |0\rangle_m \otimes |0\rangle$. This creates the quantum state $\bigotimes_{v \in V} \chi_v \otimes |0\rangle_m \otimes |-\rangle$.

C. ORACLE

Traditionally for the graph coloring problem, each vertex color is represented using the same number of qubits, and binary comparator circuits [39] are used to make sure that the two colors corresponding to two adjacent vertices are different.

While this approach is very efficient for the k -coloring problem, where every vertex has the same set of admissible colors, the list coloring problem may sometimes require a large number of qubits. In fact, the total number of qubits required for implementing a comparator-based oracle would be $n \cdot \max_{v \in V} \log_2 [\max L_v] + m + 1$, which can be much higher than our proposed oracle when the j_v 's are not all equal.

In short, for every edge $uv \in E$, we shall encode all possible colorings in $L_u \times L_v$ via flipping the amplitudes of the states corresponding to valid colorings

$$S = \left\{ |i_1\rangle_{j_u} |i_2\rangle_{j_v} : \begin{array}{l} 0 \leq i_1 < |L_u|, 0 \leq i_2 < |L_v|, \\ L_u[i_1] \neq L_v[i_2] \end{array} \right\}.$$

Algorithm 1: Oracle $O_{u,v}$ for marking valid colorings.

Input : Sets L_u, L_v (denote $j_u = \lceil \log_2 |L_u| \rceil$, $j_v = \lceil \log_2 |L_v| \rceil$).
Output: A $j_u + j_v + 1$ -qubit quantum circuit $O_{u,v}$ satisfying (III.2).

- 1 Let $W' = \text{oracleReduction}(L_u, L_v)$;
- 2 Create a circuit C with quantum wires $q_1, \dots, q_{j_u+j_v+1}$;
- 3 **for every pair** (J, s) **in** W' **do**
- 4 Add a multicontrolled NOT gate to C with controls on wires $\{q_j : j \in J, s_j = 1\}$, anticontrols on wires $\{q_j : j \in J, s_j = 0\}$ and target $q_{j_u+j_v+1}$.
- 5 **return** C

We propose a classical $O(|L_u|^2|L_v|^2)$ -time algorithm to construct an efficient oracle $O_{u,v}$ for flipping these amplitudes. In short, $O_{u,v}$ should have the following net effect:

$$O_{u,v} \left(\frac{1}{\sqrt{|L_u||L_v|}} \sum_{\substack{0 \leq i_1 < |L_u| \\ 0 \leq i_2 < |L_v|}} |i_1\rangle|i_2\rangle|-\rangle \right) = \frac{1}{\sqrt{|L_u||L_v|}} \left(\sum_{|i_1\rangle|i_2\rangle \in S} -|i_1\rangle|i_2\rangle + \sum_{|i_1\rangle|i_2\rangle \notin S} |i_1\rangle|i_2\rangle \right) |-\rangle. \quad (\text{III.2})$$

Given a string s of length ℓ and a subset $J = \{j_1, j_2, \dots, j_r\} \subseteq \{1, 2, \dots, \ell\}$ we use s_J to denote the substring $s_{j_1}s_{j_2}\dots s_{j_r}$. \mathbb{F}_2 denotes the finite field of two elements. Our algorithm for implementing $O_{u,v}$ (Algorithm 1) makes use of a subroutine called Algorithm 2 that can significantly simplify the complexity and the number of controlled not operations required in many cases.

We shall now demonstrate the correctness of Algorithm 1.

Theorem 3: Algorithm 1 gives a circuit C satisfying (III.2). Further, assuming that the cost of implementing a k -controlled NOT operation is k (refer to Remark 4), the cost of C is the smallest among all circuits that can be made using only controlled NOT operations onto the phase flip qubit.

Proof: It is sufficient to verify the action of C on the states $|i_1\rangle|i_2\rangle|-\rangle$, where $i_1 < |L_u|$ and $i_2 < |L_v|$. Notice that a single controlled NOT gate corresponding to a pair (J, s) in C effectively flips the amplitudes of all basis states represented by $\{0, 1\}$ -strings x of length $j_u + j_v$ for which $x_i = s_i, i \in J$. In other words, all states of the following form are flipped (here $*$ denotes a wildcard, and $J = \{j_1, \dots, j_k\}, s = s_1 \dots s_k$):

$$| * \dots * \underset{j_1^{\text{th}}}{s_1} * \dots * \underset{j_2^{\text{th}}}{s_2} \dots \dots \dots \underset{j_k^{\text{th}}}{s_k} * \dots * \rangle.$$

Function $\text{oracleReduction}(L_u, L_v)$

Input : Sets L_u, L_v .
Output: A set of pairs W' .

- 1 Denote $j_u = \lceil \log_2 |L_u| \rceil, j_v = \lceil \log_2 |L_v| \rceil$;
- 2 Let $A = \{0, \dots, |L_u| - 1\}$ and $B = \{0, \dots, |L_v| - 1\}$;
- 3 Convert each element of A and B into $\{0, 1\}$ -strings of lengths j_u and j_v , respectively;
- 4 Let $X = \{ab : a \in A, b \in B, L_u[a] \neq L_v[b]\}$, where “ ab ” denotes concatenation;
- 5 Let $Y = \{ab : a \in A, b \in B\}$, then $|Y| = |A||B| = |L_u||L_v|$;
- 6 Set $W = \emptyset$;
- 7 **for** $k = 1$ **to** $j_u + j_v$ **do**
- 8 **for every** k -**element subset** J of $\{1, 2, \dots, j_u + j_v\}$ **do**
- 9 **for every** $\{0, 1\}$ -**string** s of length k **do**
- 10 Add a $\{0, 1\}$ -variable x_J^s to W ;
- 11 Create an empty linear system of equations \mathcal{L} over \mathbb{F}_2 with variables W ;
- 12 **for every** $\{0, 1\}$ -**string** $t \in Y$ **do**
- 13 Calculate the expression $f(W, t) = \sum \{x_J^s \in W : t_J = s\}$;
- 14 **if** $t \in X$ **then**
- 15 Add $f(W, t) = 1$ to \mathcal{L} ;
- 16 **else**
- 17 Add $f(W, t) = 0$ to \mathcal{L} ;
- 18 Solve the $|L_u||L_v| \times \sum_{j=1}^k \binom{j_u+j_v}{j} \cdot 2^j$ system \mathcal{L} using Gaussian Elimination over \mathbb{F}_2 ;
- 19 **if the system** \mathcal{L} **is solvable then**
- 20 Solve the linear program minimizing $\sum_{x_J^s \in W} |J| \cdot x_J^s$ subject to \mathcal{L} ;
- 21 Let $W' = \{(J, s) : x_J^s \in W, x_J^s = 1\}$;
- 22 **return** W'

Thus, after application of all the controlled NOT gates, only states which appeared in an odd number of (J, s) pairs in W' will survive, and those appearing an even number of times will not have their amplitudes flipped.

Let us now fix a string $t = t_1 t_2 \dots t_{j_u+j_v}$, where $t_1 \dots t_{j_u} \in L_u$ and $t_{j_u+1} \dots t_{j_u+j_v} \in L_v$, and analyze the function 2 closely. Recall that $A = \{0, \dots, |L_u| - 1\}, B = \{0, \dots, |L_v| - 1\}, Y = \{ab : a \in A, b \in B\}$ and $X = \{ab : a \in A, b \in B, L_u[a] \neq L_v[b]\}$. We now make a crucial observation: The number of times the amplitude of $|t\rangle$ gets flipped by C is

$$\begin{aligned} |\{(J, s) \in W' : t_J = s\}| &= |\{x_J^s \in W : x_J^s = 1, t_J = s\}| \\ &= f(W, t) \end{aligned}$$

as x_J^s are all $\{0, 1\}$ valued. Since the linear system \mathcal{L} over \mathbb{F}_2 exactly contains the equations $f(W, t) = 1$, if $t \in X$ and

Algorithm 2: Full oracle O for marking valid list colorings.

Input : Graph $G = (V, E)$, lists $\{L_v : v \in V\}$, and a $\sum_{v \in V} j_v + m + 1$ -qubit quantum circuit C .
Output: C appended with a graph coloring oracle O .

- 1 **for** every edge $uv \in E$ **do**
- 2 Apply Oracle $O_{u,v}$ on qubits $q_u^1, \dots, q_u^{j_u}$, $q_v^1, \dots, q_v^{j_v}$, q'_{uv} of C ;
- 3 Apply a controlled NOT operation with controls q'_* and target q^* ;
- 4 **for** every edge $uv \in E$ **do**
- 5 Apply Oracle $O_{u,v}$ on qubits $q_u^1, \dots, q_u^{j_u}$, $q_v^1, \dots, q_v^{j_v}$, q'_{uv} to de-entangle the vertex register of C ;
- 6 **return** C

$f(W, t) = 0$ if $t \in Y \setminus X$, any solution to the system \mathcal{L} will give a correct circuit satisfying (III.2). ■

Now we come to the second assertion of the theorem. Note that, for each variable $x_j^s \in W$ with $x_j^s = 1$, we require a $|J|$ -controlled NOT operation onto the phase flip ancilla. This means that the cost of C is exactly $\sum_{x_j^s \in W} |J| \cdot x_j^s$. As we have minimized this cost via a linear program in Step 20 of 2, this proves the second claim. □

Remark 4: Suppose that the actual cost of implementing k -controlled operations is $f(k)$. In the construction of 2, we could use any $f(k)$ by modifying the objective function in the linear program of Step 20 to $\sum_{x_j^s \in W} f(|J|) \cdot x_j^s$. For an increasing function $f(k)$, the optimization problem discourages usage of gates with high number of controls. It can be seen in Barenco *et al.* [3, Corollary 7.4], that $f(k) \leq 48k - 204$ for $k \geq 7$. This implies $f(k) = O(k)$, hence our choice of $f(k) = k$ since the presence of constants does not change the optimization problem in Step 20. Finding the best $f(k)$ is still an active area of research (see, e.g., [20], [28]).

We now have all the ingredients required to implement our full oracle, which is presented as Algorithm 3 in the following.

D. DIFFUSION

Our diffusion operator is very straightforward, and is a direct application of the restricted search space diffusion mentioned in Section II-B. Let A be the initialization operator we implemented in (III.1), then a diffusion is achieved by $A D A^\dagger$, where $D = 2|0\rangle\langle 0| - I$ can be implemented by a controlled NOT with anticontrols on each of the vertex qubits and target the phase flip ancilla.

Refer to Fig. 2 for an illustration of the full list coloring algorithm.

IV. RESULTS

We implement our list coloring algorithm in python 3.8 using blueqat-sdk. In order to gauge the efficiency of our

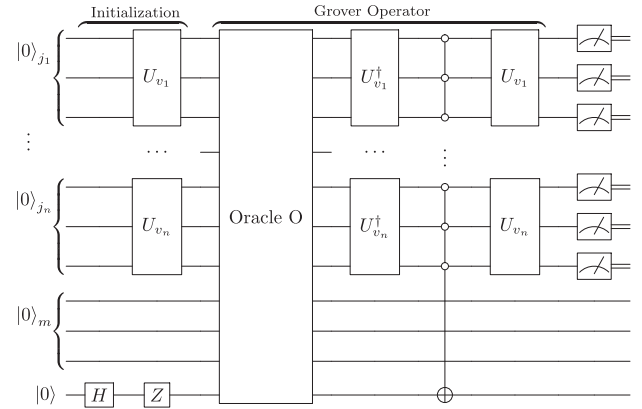
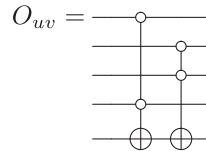


FIGURE 2. Outline of our list coloring circuit for a single Grover iteration.

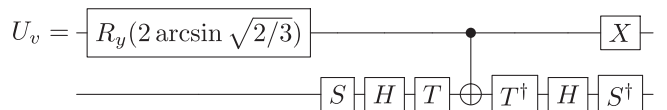
result, we run experiments for the 3- and 4-coloring problems used in [32] on the the Amazon Statevector Simulator.

A. 3-COLORING K_3

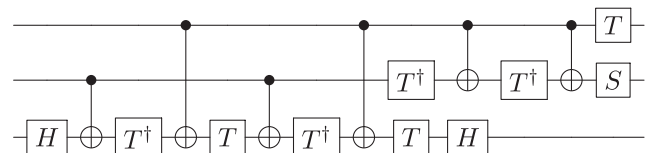
3-coloring the triangle graph $G = (\{1, 2, 3\}, \{12, 23, 13\})$ is equivalent to the list-coloring problem on G with $L_1 = L_2 = L_3 = \{|00\rangle, |01\rangle, |10\rangle\}$. In this case, our oracle component O_{uv} obtained from Algorithm 1 is



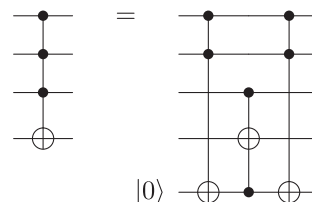
The initialization operator U_v can be written as the following circuit:



Each Toffoli gate can be decomposed into two-qubit gates using the standard decomposition



To implement a $cccX$ gate, we use a clean ancilla qubit to reduce circuit depth as demonstrated in the following:



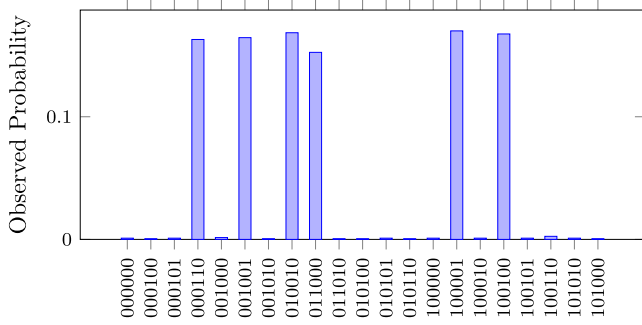


FIGURE 3. Observed frequencies for 2000 shots for 3-coloring K_3 .

TABLE 1 Theoretical Probability Density for 3-Coloring K_3

Set of states	Representation	PDF value
Marked states	A	0.165066
Unmarked but valid combinations	$B \setminus A$	0.000457
Invalid combinations	$\{ 0\rangle, 1\rangle\}^{\otimes 6} \setminus B$	0

TABLE 2 Theoretical Probability Density for 4-Coloring K_4

Set of states	Representation	PDF value
Marked states	A'	0.041657
Unmarked states	$B' \setminus A'$	10^{-6}

Finally, we are able to run our circuit on the Amazon Statevector Simulator after decomposing into these elementary single and two-qubit operations. Our resulting histogram for one Grover iteration is shown in Fig. 3.

It is seen that states in the set $A = \{|000110\rangle, |001001\rangle, |011000\rangle, |011000\rangle, |100001\rangle, |100100\rangle\}$ are all massively amplified. Each state of A represents a valid 3-coloring of K_3 . Let $B = \{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}^{\otimes 3}$. The statevector at the end of our algorithm can be calculated theoretically, and we present the probability density function over the 6-qubit states in Table 1. In particular, each state in A has a probability p_A of 0.165066 to be measured, and states containing $|11\rangle$ as a color have probability 0 of being observed. Our algorithm has a single-shot accuracy of $p_A \cdot |A| \approx 0.9904$.

B. 4-COLORING K_4

Let $G = (\{1, 2, 3, 4\}, \{12, 13, 14, 23, 24, 34, 14\})$ be the complete graph on four vertices, and suppose $L_1 = L_2 = L_3 = L_4 = \{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$. In this case, the simple Hadamard operator $H^{\otimes 2}$ initializes each vertex register to a uniform superposition of its valid colors, and we can then run Algorithm 1 to figure out the component O_{uv} . It turns out that one of the valid solutions minimizing the cost of gates used is the following circuit:

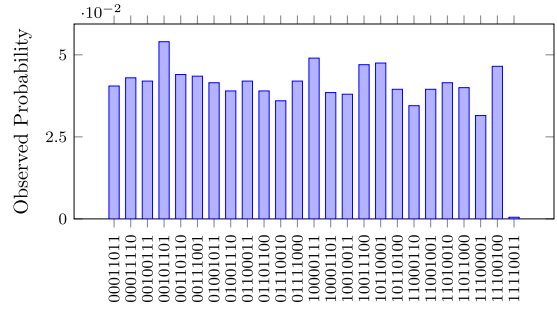
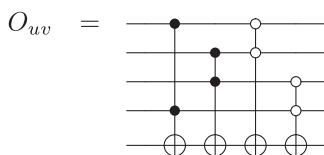


FIGURE 4. Observed frequencies for 2000 shots for 4-coloring K_4 .

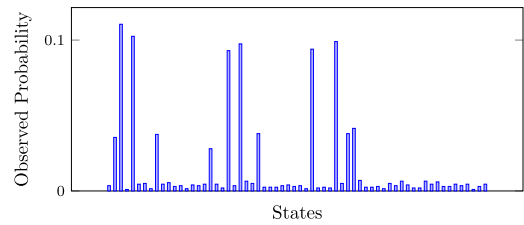


FIGURE 5. Observed probabilities for 2000 shots for 3-coloring K_3 using the algorithm of [32].

Fig. 4 shows the results of running our circuit with one Grover iteration on the statevector simulator.

In this case, the result of a statevector computation after one Grover iteration is shown in Table 2. Here A' denotes the set of all $4! = 24$ marked states corresponding to valid colorings of K_4 , and $B' = \{|0\rangle, |1\rangle\}^{\otimes 8}$ is the full state space.

The result of our experiment can also be seen to follow this distribution.

C. COMPARISON WITH PREVIOUS WORK: 3-COLORING K_3

We take [32] as the state-of-the-art result for the k -coloring problem. We compare the performance of our algorithm with theirs for $k = 3$. Fig. 5 depicts the empirical probability distribution obtained via implementing their algorithm for one Grover rotation and running on the Amazon Statevector Simulator. We observe that our algorithm has much higher probability of selecting a correct state (around 0.9904), whereas their method has an empirical success rate of 0.5965 for 2000 shots.

Finally, we compare the gate count of their circuit versus ours for the 3-coloring K_3 problem for one Grover iteration in Table 3.

The code we used for this comparison is available at the following GitHub repository: <https://github.com/Potla1995/Grover-ListColoring>.

D. EXPERIMENTS ON NISQ DEVICES AND LIMITATIONS

We also ran our circuits on the IonQ physical device. However, reasonable results were not obtained. There might be several possible reasons behind this as follows.

TABLE 3 Comparison of Gate Counts for One Grover Iteration

Gates	Saha et. al. [32]	Our Circuit
X, H	58	66
R_y, R_z	0	45
CX	20	15
CCX	24	12
$CCCX$	3	2
$CCCCX$	1	0
Total elementary gates	686	429

- CX is not a gate natively implemented on the IonQ processor. Although its fully connected topology is ideal for the circuits we construct, controlled operations need to be expressed in terms of Mølmer-Sørensen gates [25], which makes them expensive.
- Our circuit has a depth of 30 when expressed in terms of the elementary gates (including CX). This makes the circuit much more susceptible to errors from noise and decoherence. Although we can implement and run our algorithm on quantum devices currently available, the results suggest that devices with lesser intrinsic noise are needed for practical use of our algorithm.

These limitations suggest that while our algorithm is designed for running on NISQ devices, the circuits generated by it are too complex, giving unusable results due to the intrinsic noise in the currently available quantum devices. In any case, classical computers can brute force the list coloring problem for the small-sized graphs that can be currently encoded on an NISQ device using our encoding scheme.

V. CONCLUSION

The list coloring problem is ubiquitous in real life, as it not only generalizes an already well-appearing problem of graph coloring (scheduling, satisfiability, etc.) but is also applicable to several other scenarios, as follows.

- Wireless Network Allocation* [38]: In a wireless network, each radio is allocated special frequencies which it can connect to. Suppose that radios in close proximity cannot operate on the same frequency due to interference. The problem of which radio is connected to which network frequency can be modeled as a list coloring problem in the following graph: Let the radios be represented by vertices, and add an edge between two radios if they are in close proximity of each other. The lists for each vertex will be the set of available frequencies for its corresponding radio.
- Register Allocation*: In compiler optimization, register allocation is the process of assigning a large number of target program variables (n) onto a small number of CPU registers (k), which reduces to a k -coloring problem on an n -vertex graph.
- Sudoku*: We can represent every cell in a sudoku problem with a vertex, and join two vertices with an edge if they are in same row or same column or same block.

Given x already filled cells, we can formulate the sudoku problem as a list-coloring problem on $81 - x$ vertices and at most nine colors.

We proposed a Grover search-based quantum algorithm that achieves quadratic speedup in query complexity compared to a classical brute-force search, and also proposed a classical algorithm that can simplify the oracle design for several special instances of the list coloring problem. We demonstrate the efficiency of our method in comparison with previous work by running our algorithm on the Amazon Statevector Simulator for the 3- and 4-coloring problems.

Unfortunately, the list coloring problem is difficult to solve both classically and using quantum algorithms; as for generic lists with no known structure, brute force seems to be the only way to attack the problem. As our algorithm is basically a brute-force quantum search with some optimizations in the oracle, it is expected to perform better in general cases, where the structure of the lists are unknown. However, the existence of clever hybrid algorithms exploiting specific structures for known lists cannot be ignored, and is a very promising future direction.

Finally, we note that, one can obtain an improvement on our algorithm by just changing a given list coloring problem to a reduced problem. For example, if $G = (V, E)$ has a vertex v with $|L_v| = 1$, we can color v first and remove its color from each L_u , such that $uv \in E$, and iterate until all lists have at least two colors.

ACKNOWLEDGMENT

A part of this work was completed during a summer internship at Elyah Co. in 2020. The author would like to thank S. Andrews, R. Nagai, and G. Rajendran for several helpful discussions and comments; Y. Minato and S. Al Jimeely for encouragement and support; and the anonymous reviewers whose insightful comments and suggestions helped drastically improve the quality of the manuscript.

REFERENCES

- S. Aaronson and P. Rall, "Quantum approximate counting, simplified," in *Proc. Symp. Simplicity Algorithms*, 2020, pp. 24–32, doi: [10.1137/1.9781611976014.5](https://doi.org/10.1137/1.9781611976014.5).
- A. Ambainis, K. Balodis, J. Iraids, M. Kokainis, K. Prūsis, and J. Vihrovs, "Quantum speedups for exponential-time dynamic programming algorithms," in *Proc. 30th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2019, pp. 1783–1793, doi: [10.1137/1.9781611975482.107](https://doi.org/10.1137/1.9781611975482.107).
- A. Barenco et al., "Elementary gates for quantum computation," *Phys. Rev. A*, vol. 52, no. 5, 1995, Art. no. 3457, doi: [10.1103/PhysRevA.52.3457](https://doi.org/10.1103/PhysRevA.52.3457).
- L. H. Bodlaender and D. Kratsch, "An exact algorithm for graph coloring with polynomial memory," Dept. Inf. Comput. Sci., Utrecht Univ., Utrecht, Netherlands, Tech. Rep. UU-CS-2006-015, 2006.
- M. Boyer, G. Brassard, P. Høyer, and A. Tapp, "Tight bounds on quantum searching," *Fortschritte der Physik: Prog. Phys.*, vol. 46, no. 4/5, pp. 493–505, 1998, doi: [10.1002/\(SICI\)1521-3978\(199806\)46:4/5<493::AID-FRPA4649330001>3.0.CO;2-3](https://doi.org/10.1002/(SICI)1521-3978(199806)46:4/5<493::AID-FRPA4649330001>3.0.CO;2-3).
- S. Bravyi, A. Kliesch, R. Koenig, and E. Tang, "Hybrid quantum-classical algorithms for approximate graph coloring," 2020, *arXiv:2011.13420*.
- R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca, "Quantum algorithms revisited," *Proc. Roy. Soc. London. Ser. A: Math., Phys. Eng. Sci.*, vol. 454, no. 1969, pp. 339–354, 1998, doi: [10.1098/rspa.1998.0164](https://doi.org/10.1098/rspa.1998.0164)

- [8] M. Cygan *et al.*, “On problems as hard as CNF-SAT,” *ACM Trans. Alg.*, vol. 12, no. 3, pp. 1–24, 2016, doi: [10.1145/2925416.10.1109/CCC.2012.36](https://doi.org/10.1145/2925416.10.1109/CCC.2012.36).
- [9] C. Durr and P. Hoyer, “A quantum algorithm for finding the minimum,” 1996, *arXiv: quant-ph/9607014*.
- [10] D. Eppstein, “Small maximal independent sets and faster exact graph coloring,” in *Proc. Workshop Algorithms Data Structures*, 2001, pp. 462–470, doi: [10.1007/3-540-44634-6_42](https://doi.org/10.1007/3-540-44634-6_42).
- [11] A. Gilliam, S. Woerner, and C. Goniculea, “Grover adaptive search for constrained polynomial binary optimization,” *Quantum*, vol. 5, 2021, Art. no. 428, doi: [10.22331/q-2021-04-08-428](https://doi.org/10.22331/q-2021-04-08-428).
- [12] K. L. Grover, “A fast quantum mechanical algorithm for database search,” in *Proc. 28th Annu. ACM Symp. Theory Comput.*, 1996, pp. 212–219, doi: [10.1145/237814.237866](https://doi.org/10.1145/237814.237866).
- [13] K. L. Hoffman *et al.*, “Traveling salesman problem,” *Encyclopedia Operations Res. Manage. Sci.*, vol. 1, pp. 1573–1578, 2013, doi: [10.1007/1-4020-0611-X_1068](https://doi.org/10.1007/1-4020-0611-X_1068).
- [14] S. Hu, P. Liu, C. R. Chen, M. Pistoia, and J. Gambetta, “Reduction-based problem mapping for quantum computing,” *Computer*, vol. 52, no. 6, pp. 47–57, 2019, doi: [10.1109/MC.2019.2909709](https://doi.org/10.1109/MC.2019.2909709).
- [15] Q.-S. Hua, Y. Wang, D. Yu, and F. CM Lau, “Dynamic programming based algorithms for set multicover and multiset multicover problems,” *Theor. Comput. Sci.*, vol. 411, no. 26–28, pp. 2467–2474, 2010, doi: [10.1016/j.tcs.2010.02.016](https://doi.org/10.1016/j.tcs.2010.02.016).
- [16] S. Jeffery, R. Kothari, F. Le Gall, and F. Magniez, “Improving quantum query complexity of Boolean matrix multiplication using graph collision,” *Algorithmica*, vol. 76, no. 1, pp. 1–16, 2016, doi: [10.1007/s00453-015-9985-x](https://doi.org/10.1007/s00453-015-9985-x).
- [17] K. Khadiev and A. Ilikaev, “Quantum algorithms for the most frequently string search, intersection of two string sequences and sorting of strings problems,” in *Proc. Int. Conf. Theory Pract. Natural Comput.*, 2019, pp. 234–245, doi: [10.1007/978-3-030-34500-6_17](https://doi.org/10.1007/978-3-030-34500-6_17).
- [18] M. Koivisto, “An $o^*(2^n)$ algorithm for graph coloring and other partitioning problems via inclusion-exclusion,” in *Proc. IEEE 47th Annu. IEEE Symp. Found. Comput. Sci.*, 2006, pp. 583–590, doi: [10.1109/FOCS.2006.11](https://doi.org/10.1109/FOCS.2006.11).
- [19] D. Kravchenko, K. Khadiev, and D. Serov, “On the quantum and classical complexity of solving subtraction games,” in *Proc. Int. Comput. Sci.-Theory Appl.*, 2019, pp. 228–236, doi: [10.1007/978-3-030-61739-4_6](https://doi.org/10.1007/978-3-030-61739-4_6).
- [20] A. M. Krol, A. Sarkar, I. Ashraf, Z. Al-Ars, and K. Bertels, “Efficient decomposition of unitary matrices in quantum circuit compilers,” *Appl. Sci.*, vol. 12, no. 2, 2022, Art. no. 759, doi: [10.3390/app12020759](https://doi.org/10.3390/app12020759).
- [21] K. Kudo, “Constrained quantum annealing of graph coloring,” *Phys. Rev. A*, vol. 98 no. 2, 2018, Art. no. 022301, doi: [10.1103/PhysRevA.98.022301](https://doi.org/10.1103/PhysRevA.98.022301).
- [22] T. Lee, F. Magniez, and M. Santha, “Improved quantum query algorithms for triangle detection and associativity testing,” *Algorithmica*, vol. 77, no. 2, pp. 459–486, 2017, doi: [10.1007/s00453-015-0084-9](https://doi.org/10.1007/s00453-015-0084-9).
- [23] C. Li, R. Roberts, and X. Yin, “Decomposition of unitary matrices and quantum gates,” *Int. J. Quantum Inf.*, vol. 11, no. 1, 2013, Art. no. 1350015, doi: [10.1142/S0219749913500159](https://doi.org/10.1142/S0219749913500159).
- [24] J. DC Little, K. G. Murty, D. W. Sweeney, and C. Karel, “An algorithm for the traveling salesman problem,” *Oper. Res.*, vol. 11, no. 6, pp. 972–989, 1963, doi: [10.1287/opre.11.6.972](https://doi.org/10.1287/opre.11.6.972).
- [25] K. Mølmer and A. Sørensen, “Multiparticle entanglement of hot trapped ions,” *Phys. Rev. Lett.*, vol. 82, pp. 1835–1838, Mar. 1999, doi: [10.1103/PhysRevLett.82.1835](https://doi.org/10.1103/PhysRevLett.82.1835).
- [26] M. Mosca, “Counting by quantum eigenvalue estimation,” *Theor. Comput. Sci.*, vol. 264, no. 1, pp. 139–153, Aug. 2001, doi: [10.1016/S0304-3975\(00\)00217-6](https://doi.org/10.1016/S0304-3975(00)00217-6).
- [27] M. Möttönen, J. J. Vartiainen, V. Bergholm, and M. M. Salomaa, “Quantum circuits for general multiqubit gates,” *Phys. Rev. Lett.*, vol. 93, no. 13, 2004, Art. no. 130502, doi: [10.1103/PhysRevLett.93.130502](https://doi.org/10.1103/PhysRevLett.93.130502).
- [28] K. M. Nakanishi, T. Satoh, and S. Todo, “Quantum-gate decomposer,” 2021, *arXiv:2109.13223*.
- [29] L. Pursell and S. Y. Trimble, “Gram–Schmidt orthogonalization by gauss elimination,” *Amer. Math. Mon.*, vol. 98, no. 6, pp. 544–549, 1991, doi: [10.2307/2324877](https://doi.org/10.2307/2324877).
- [30] P. Ronagh, “Quantum algorithms for solving dynamic programming problems,” 2019, *arXiv:1906.02229*.
- [31] A. Saha, A. Chongder, S.B. Mandal, and A. Chakrabarti, “Synthesis of vertex coloring problem using Grover’s algorithm,” in *Proc. IEEE Int. Symp. Nanoelectron. Inf. Syst.*, 2015, pp. 101–106, doi: [10.1109/iNIS.2015.55](https://doi.org/10.1109/iNIS.2015.55).
- [32] A. Saha, D. Saha, and A. Chakrabarti, “Circuit design for k-coloring problem and its implementation on near-term quantum devices,” in *Proc. IEEE Int. Symp. Smart Electron. Syst. (Formerly iNIS)*, 2020, pp. 17–22, doi: [10.1109/iSES50453.2020.00015](https://doi.org/10.1109/iSES50453.2020.00015).
- [33] K. Shimizu and R. Mori, “Exponential-time quantum algorithms for graph coloring problems,” in *Proc. Latin Amer. Symp. Theor. Informat.*, 2021, pp. 387–398, doi: [10.1007/978-3-030-61792-9_31](https://doi.org/10.1007/978-3-030-61792-9_31).
- [34] A. Shukla and P. Vedula, “Trajectory optimization using quantum computing,” *J. Glob. Optim.*, vol. 75, no. 1, pp. 199–225, 2019, doi: [10.1007/s10898-019-00754-5](https://doi.org/10.1007/s10898-019-00754-5).
- [35] O. Titiloye and A. Crispin, “Graph coloring with a distributed hybrid quantum annealing algorithm,” in *Agent and Multi-Agent Systems: Technologies and Applications*. ser. Lecture Notes in Computer Science, vol. 6682, Berlin, Germany: Springer, 2011, pp. 553–562, doi: [10.1007/978-3-642-22000-5_57](https://doi.org/10.1007/978-3-642-22000-5_57).
- [36] O. Titiloye and A. Crispin, “Quantum annealing of the graph coloring problem,” *Discrete Optim.*, vol. 8, no. 2, pp. 376–384, 2011, doi: [10.1016/j.disopt.2010.12.001](https://doi.org/10.1016/j.disopt.2010.12.001).
- [37] J. J. Vartiainen, M. Möttönen, and M. M. Salomaa, “Efficient decomposition of quantum gates,” *Phys. Rev. Lett.*, vol. 92, no. 17, 2004, Art. no. 177902, doi: [10.1103/PhysRevLett.92.177902](https://doi.org/10.1103/PhysRevLett.92.177902).
- [38] W. Wang and X. Liu, “List-coloring based channel allocation for open-spectrum wireless networks,” in *Proc. IEEE 62nd Veh. Technol. Conf.*, vol. 1, pp. 690–694, 2005, doi: [10.1109/VETEFC.2005.1558001](https://doi.org/10.1109/VETEFC.2005.1558001).
- [39] Y. Wang and M. Perkowski, “Improved complexity of quantum oracles for ternary grover algorithm for graph coloring,” in *Proc. 41st IEEE Int. Symp. Mult.-Valued Log.*, 2011, pp. 294–301, doi: [10.1109/ISMVL.2011.42](https://doi.org/10.1109/ISMVL.2011.42).