# Concrete Quantum Circuits to Prepare Generalized Dicke States on a Quantum Machine

Shintaro Narisada[a], Shohei Beppu[b], Kazuhide Fukushima[c] and Shinsaku Kiyomoto[d]
*KDDI Research, Inc., Fujimino, Japan*

Keywords: Dicke State, Hamming Weight, Noisy Computation, Quantum Circuit, Quantum Computing, Generalization.

Abstract: A Dicke state is a superposition of $n$-qubit with Hamming weight $k$, denoted by $|D^n_k\rangle$. Dicke states are frequently employed to prepare input superpositions for quantum search algorithms (e.g., Grover search and quantum walks) that solve combinatorial problems with a certain number $\binom{n}{k}$ of candidate solutions. Bärtschi and Eidenbenz propose a concrete quantum circuit to construct the Dicke state $|D^n_k\rangle$ with polynomial quantum gates, and they generalize the circuit in terms of Hamming weight $k$ to prepare a superposition of Dicke states. Subsequently, Esser et al. present another quantum circuit to generate the Dicke state $|D^n_k\rangle$ with polynomial gates and a few auxiliary quantum bits. In this paper, we generalize Esser's state preparation circuit to construct a superposition of Dicke states. We conduct a concrete comparison with two generalized Dicke state preparation circuits. We perform noisy simulations and experiments using real quantum machines from the IBM quantum experience service (IBMQ). Both circuits successfully construct the generalized Dicke state superposition using a noisy intermediate-scale quantum (NISQ) device, albeit somewhat affected by noise.

## 1 INTRODUCTION

The exponential hardness of combinatorial problems is one of the foundations of the security of modern society. With the advent of quantum computers, quantum search algorithms to solve combinatorial problems such as the Grover algorithm (Grover, 1996), quantum amplitude amplification (QAA) (Brassard et al., 2002) and quantum walks (Ambainis, 2004) are being realized (Mandviwalla et al., 2018; Acasiete et al., 2020). The input to these algorithms is normally a quantum state superposition associated with the candidate solutions to a specific problem to be solved. While uniform superposition of all $2^n$ $n$-qubit states can be prepared with $n$ Hadamard gates, it is known that constructing an arbitrary superposition requires $\Theta(2^n)$ quantum gates (Shende et al., 2006). There are several studies of quantum circuits that efficiently prepare superpositions of a certain class of states, including circuits to generate linear-size superpositions called the $|GHZ_n\rangle$ and $|W_n\rangle$ states (Cruz et al., 2019), and circuits to generate exponential-size superpositions called the Dicke state $|D^n_k\rangle$ probabilistically (Childs et al., 2000; Chakraborty et al., 2014) or deterministically (Kaye and Mosca, 2001; Bärtschi and Eidenbenz, 2019; Mukherjee et al., 2020; Esser et al., 2021; Bärtschi and Eidenbenz, 2022) in polynomial time.

$|D^n_k\rangle$ is the superposition of all $n$-qubit states with Hamming weight $k$ of size $\binom{n}{k}$, which has a close relationship to combinatorial problems such as the $k$-vertex cover problem (Cook et al., 2020), extractive summarization (Niroula et al., 2022) and syndrome decoding (SD) problem (Esser et al., 2022; Perriello et al., 2021; Chailloux et al., 2021), which is the security basis of code-based cryptosystems. A generalized Dicke state preparation circuit to construct a superposition of Dicke states $|D^n_k\rangle$ for any $k \leq n$ with polynomial gates is also proposed in (Kaye and Mosca, 2001; Bärtschi and Eidenbenz, 2019). For certain combinatorial problems, the number of combinations for candidate solutions may be a sum of $\binom{n}{i}$ for any $0 \leq i \leq n$ rather than $\binom{n}{k}$. For example, the SD problem can be easier to solve if it is extended to choose binary vectors with weight $\leq k$ than vectors with weight $k$ from $n$ vectors.

In this paper, we generalize the Dicke state preparation circuit shown in (Esser et al., 2021) and propose another circuit to construct an equal superposi-

[a] https://orcid.org/0000-0002-9399-9778
[b] https://orcid.org/0000-0002-8220-9515
[c] https://orcid.org/0000-0003-2571-0116
[d] https://orcid.org/0000-0003-0268-0532

Narisada, S., Beppu, S., Fukushima, K. and Kiyomoto, S.
Concrete Quantum Circuits to Prepare Generalized Dicke States on a Quantum Machine.
DOI: 10.5220/0011618000003405
In *Proceedings of the 9th International Conference on Information Systems Security and Privacy (ICISSP 2023)*, pages 329-338
ISBN: 978-989-758-624-8; ISSN: 2184-4356

tion of $|D^n_k\rangle$ for any $0 \leq i \leq k$ with polynomial size quantum gates. We implement the generalized circuit and Bärtschi's circuit using the Qiskit library and give a comparison between two generalized Dicke state preparation circuits. Quantum simulations of our implementations in both noise-free models and noisy circuit models are also conducted. We run our implementations on the quantum device from IBMQ and achieved superposition for some generalized Dicke states.

## 2 PRELIMINARIES

We write a qubit as $|x\rangle, x \in \{0,1\}$. A tensor product of two qubits $|x\rangle$ and $|y\rangle$ is $|x\rangle \otimes |y\rangle$ or simply $|x\rangle|y\rangle$. A tensor product of two identical qubits $|x\rangle$ is shortened by $|x\rangle^{\otimes 2} = |x\rangle \otimes |x\rangle$. Let $|x_1x_2...x_n\rangle, x_i \in \{0,1\}, 1 \leq i \leq n$ be an $n$-qubit quantum state. $|x_ix_{i+1}...\rangle_i$ denotes the sequence of qubits starting at $i$. We may treat a quantum state $|x_1x_2...x_n\rangle$ as a binary string $b_n = x_1x_2...x_n$ for simplicity of explanation. The Hamming weight of a binary string $b_n$ is $\mathrm{wt}(b_n) = |\{i \mid x_i = 1\}|$. A binary string of length $n$ and Hamming weight $k$ is written by $b_{n,k}$. The empty string is denoted by $\varepsilon$. $|\psi_n\rangle = c_{00...0}|00...0\rangle + c_{00...1}|00...1\rangle + ... + c_{11...1}|11...1\rangle$ denotes the $n$-qubit superposition, where $\sum_{i_1,...,i_n} |c_{i_1...i_n}|^2 = 1$. Each $|c_{i_1...i_n}|^2$ can be considered as the existence probability $p_{i_1...i_n}$ of the corresponding quantum state $|i_1...i_n\rangle$. Measuring the $n$-qubit superposition yields one $n$-qubit state $|i_1...i_n\rangle$ with probability $p_{i_1...i_n}$. A Dicke state is defined as follows:

**Definition 1.** *A Dicke state $|D^n_k\rangle$ is the equal superposition of all n-qubit states $|x\rangle$ with Hamming weight* $\mathrm{wt}(x) = k$,

$$|D^n_k\rangle = \sqrt{\frac{1}{\binom{n}{k}}} \sum_{x \in \{0,1\}^n, \mathrm{wt}(x)=k} |x\rangle. \qquad (1)$$

For example, $|D^3_1\rangle = \frac{1}{\sqrt{3}}(|100\rangle + |010\rangle + |001\rangle)$. A superposition of Dicke states is written by

$$\sum_{0 \leq i \leq n} \alpha_i |D^n_i\rangle, \qquad (2)$$

where $\alpha_i \in \mathbb{C}$, $\alpha_0^2 + \ldots + \alpha_n^2 = 1$ (Kaye and Mosca, 2001; Bastin et al., 2009). Hereafter, we consider an equal superposition of Dicke states for arbitrary weights. For an integer set $K \subseteq [0,n]$, we will refer to equal superpositions of Dicke states with Hamming weights $i \in K$ as a *generalized Dicke state*:

**Definition 2.** *Generalized Dicke state $|D^n_K\rangle$ for an integer set $K \subseteq [0,n]$ is the equal superposition of all n-qubit states $|x\rangle$ with Hamming weight* $\mathrm{wt}(x) \in K$,

$$|D^n_K\rangle = \sqrt{\frac{1}{|C^n_K|}} \sum_{x \in \{0,1\}^n, \mathrm{wt}(x) \in K} |x\rangle, \qquad (3)$$

where $|C^n_K| = \sum^n_{i \in K} \binom{n}{i}$. For example, $|D^3_K\rangle = \frac{1}{\sqrt{5}}(|000\rangle + |100\rangle + |010\rangle + |001\rangle + |111\rangle)$ for $K = \{0,1,3\}$. Using Equation 2, the generalized Dicke state is a case when $\alpha_i = \sqrt{\binom{n}{i}/|C^n_K|}$ for all $i \in K$ and $\alpha_i = 0$ otherwise.

Any unitary operation on the quantum state $|x\rangle$ is denoted by $U|x\rangle$, where $U$ is a unitary matrix. We will introduce certain basic unitary operations on quantum states.

- $I$: identity operator $I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $I|0\rangle = |0\rangle$ and $I|1\rangle = |1\rangle$.
- $X$: single qubit gate defined as $X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$. $X|0\rangle = |1\rangle$ and $X|1\rangle = |0\rangle$ as with the classical NOT gate.
- CNOT (controlled-X): 2-qubits gate that applies $X$ to the target qubit when the control qubit is $|1\rangle$ and does nothing when the control qubit is $|0\rangle$.
- $C^nX$ (multi-controlled-$X$): $(n+1)$-qubits gate that applies $X$ to the target qubit when the control $n$-qubit satisfy some quantum state $|b_n\rangle$ and does nothing when the control is not $|b_n\rangle$. When $n = 2$ and $b_2 = 11$, $C^nX$ is called a Toffoli gate.
- $R_y(\theta)$: single qubit gate defined by $R_y(\theta) = \begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{bmatrix}$. Note that since $R_y(2\cos^{-1}(\sqrt{p}))|0\rangle = \sqrt{p}|0\rangle + \sqrt{1-p}|1\rangle$, it is helpful when we want to partition the probability of $|0\rangle$ into $p$ and $1-p$. We also use $CR_y(\theta)$ (controlled-$R_y(\theta)$) gates and $C^nR_y(\theta)$ (multi-controlled-$R_y(\theta)$) gates.

Thereafter, these basic gates are combined to achieve our desired unitary operation. A tensor product of two unitary operators $U_1$ and $U_2$ is $U_1 \otimes U_2$. A tensor product of two identical unitary operators $U$ is shortened by $U^{\otimes 2} = U \otimes U$. A quantum circuit is a sequence of unitary operations starting from $|00...0\rangle$. The purpose of a quantum circuit is to obtain the desired quantum state $|\psi\rangle$ (e.g., corresponding to the solution of a combinatorial problem) with high probability when measuring the qubits at the endpoint of the circuit.

The computational complexity of a quantum circuit (circuit complexity) is evaluated by its *width* and *depth*. The width of a quantum circuit is the total number of qubits required for the circuit. Depth is the longest path in the circuit. Auxiliary qubits (ancilla) are extra qubits that are often employed to simplify gating operations. For example, ancilla qubits are employed to store the carry of the quantum full adder circuit.

# 3 DETERMINISTIC DICKE STATE PREPARATION

We will explain previous deterministic Dicke state preparation circuits proposed by Bärtschi (Bärtschi and Eidenbenz, 2019) and Esser (Esser et al., 2021) et al. The former uses no ancilla qubits, but its circuit is somewhat complicated. The latter is intuitive construction by using some ancilla qubits as a counter.

## 3.1 Dicke State Preparation Without Ancilla Qubits

The central part of their circuit is the construction of a unitary operator $U_{n,k}$ such that $|\underbrace{00..00}_{n-\ell}\underbrace{1..1}_{\ell}\rangle$ is an input and $|D^n_\ell\rangle$ is an output for *any* $0 \le \ell \le k$.

**Definition 3** ($U_{n,k}$)**.** *n-qubit unitary gate satisfying* $U_{n,k}|0\rangle^{\otimes n-\ell}|1\rangle^{\otimes \ell} = |D^n_\ell\rangle$ *for all* $0 \le \ell \le k$.

By setting $\ell \le k$ instead of $\ell = k$, we can inductively construct $U_{n,k}$ by using the following property for the Dicke state.

**Lemma 1.** *(Lamata et al., 2013; Moreno and Parisio, 2018)* $|D^n_\ell\rangle$ *has the following inductive sum form:*

$$|D^n_\ell\rangle = \sqrt{\frac{\ell}{n}}|D^{n-1}_{\ell-1}\rangle \otimes |1\rangle + \sqrt{\frac{n-\ell}{n}}|D^{n-1}_{\ell}\rangle \otimes |0\rangle. \quad (4)$$

Intuitively, $|D^{n-1}_{\ell-1}\rangle$ contains $\binom{n-1}{\ell-1}$ strings, and $|D^{n-1}_{\ell}\rangle$ contains $\binom{n-1}{\ell}$ strings with equal existence probabilities $\binom{n-1}{\ell-1}^{-1}$ and $\binom{n-1}{\ell}^{-1}$, respectively. The sum of $|D^{n-1}_{\ell-1}\rangle \otimes |1\rangle$ and $|D^{n-1}_{\ell}\rangle \otimes |0\rangle$ then contains $\binom{n-1}{\ell-1} + \binom{n-1}{\ell} = \binom{n}{\ell}$ distinct strings. The coefficients of each term are intended to equalize the probability.

Both Dicke states $|D^{n-1}_{\ell-1}\rangle$ and $|D^{n-1}_{\ell}\rangle$ can be generated by the same unitary $U_{n-1,k}$ given the inputs $|0\rangle^{\otimes n-\ell}|1\rangle^{\otimes \ell-1}$ and $|0\rangle^{\otimes n-1-\ell}|1\rangle^{\otimes \ell}$, respectively. To inductively construct $U_{n,k}$, we need a unitary gate that corresponds to the following operation that changes (actually *left shift*) the last qubit $|1\rangle$ to $|0\rangle$ with probability $\frac{n-\ell}{n}$ for inputs $|0\rangle^{\otimes n-\ell}|1\rangle^{\otimes \ell}$ for all $\ell \le k$

$$\begin{aligned}|0\rangle^{\otimes n-\ell}|1\rangle^{\otimes \ell} \mapsto \sqrt{\frac{\ell}{n}}|0\rangle^{\otimes n-\ell}|1\rangle^{\otimes \ell-1}|1\rangle +& \\ \sqrt{\frac{n-\ell}{n}}|0\rangle^{\otimes n-\ell-1}|1\rangle^{\otimes \ell-1}|0\rangle.& \quad (5)\end{aligned}$$

Note that we can disregard the first $n-k-1$ qubits since they are always 0. Next, we can consider a unitary $S_{n,k}$ (referred to as split and cyclic shift in the original paper) such that for the last $k+1$ qubits it performs a left shift operation with probability $\frac{n-\ell}{n}$ for all $\ell \le k$.

**Definition 4** ($S_{n,k}$)**.** $(k+1)$*-qubits unitary gate for all* $1 \le \ell \le k$ *satisfying*

$$\begin{aligned}S_{n,k}|0\rangle^{\otimes k+1} &= |0\rangle^{\otimes k+1} \\ S_{n,k}|0\rangle^{\otimes k+1-\ell}|1\rangle^{\otimes \ell} &= \sqrt{\frac{\ell}{n}}|0\rangle^{\otimes k+1-\ell}|1\rangle^{\otimes \ell-1}|1\rangle + \\ &\quad \sqrt{\frac{n-\ell}{n}}|0\rangle^{\otimes k-\ell}|1\rangle^{\otimes \ell-1}|0\rangle \\ S_{n,k}|1\rangle^{\otimes k+1} &= |1\rangle^{\otimes k+1}.\end{aligned}$$

One can construct the $S_{n,k}$ gate concretely by combining the CNOT, $CR_y(\theta)$ gate and $C^nR_y(\theta)$ gate:

**Definition 5** (Building blocks of $S_{n,k}$)**.** *$S_{n,k}$ is the connection of k basic gates $B_{n,1}$ to $B_{n,k}$*

$$\begin{aligned}B_{n,1}: &|00\rangle_{n-1} &&\to |00\rangle_{n-1} \\ &|11\rangle_{n-1} &&\to |11\rangle_{n-1} \\ &|01\rangle_{n-1} &&\to \sqrt{\frac{1}{n}}|01\rangle_{n-1} + \sqrt{\frac{n-1}{n}}|10\rangle_{n-1} \\ B_{n,\ell}: &|00\rangle_{n-\ell}|0\rangle_n &&\to |00\rangle_{n-\ell}|0\rangle_n \\ &|01\rangle_{n-\ell}|0\rangle_n &&\to |01\rangle_{n-\ell}|0\rangle_n \\ &|00\rangle_{n-\ell}|1\rangle_n &&\to |00\rangle_{n-\ell}|1\rangle_n \\ &|11\rangle_{n-\ell}|0\rangle_n &&\to |11\rangle_{n-\ell}|0\rangle_n \\ &|01\rangle_{n-\ell}|1\rangle_n &&\to \sqrt{\frac{\ell}{n}}|01\rangle_{n-\ell}|1\rangle_n + \\ & && \quad \sqrt{\frac{n-\ell}{n}}|11\rangle_{n-\ell}|0\rangle_n\end{aligned}$$

*where* $2 \le \ell \le k$*. Namely,* $S_{n,k} = \underbrace{B_{n,k}B_{n,k-1}...B_{n,2}B_{n,1}}_{k \text{ basic gates}}$*.*

*Figure 1 shows the basic gate components of $B_{n,1}$ and $B_{n,\ell}$.*

Using the inductive property $U_{i,i} = (U_{i-1,i-1} \otimes I^{\otimes k-i}) \cdot S_{i,i-1}$ for $2 \le i \le k$ and $U_{i,k} = (U_{i-1,k} \otimes I^{\otimes n-i}) \cdot (I^{\otimes i-k-1} \otimes S_{i,k})$ for $k+1 \le i \le n$, $U_{n,k}$ is decomposed by $S_{n,k}$ gates as follows:

**Lemma 2.** *(Bärtschi and Eidenbenz, 2019) The following inductive construction of $U_{n,k}$ is consistent with Definition 3.*

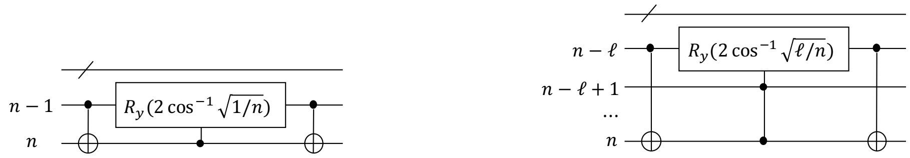

Figure 1: $B_{n,1}$ gate (left) and $B_{n,\ell}$ gate for $2 \leq \ell \leq k$ (right).

$$U_{n,k} := \prod_{i=2}^{k}(S_{i,i-1} \otimes I^{\otimes n-i}) \cdot \prod_{i=k+1}^{n}(I^{\otimes i-k-1} \otimes S_{i,k} \otimes I^{\otimes n-i}).$$

For instance, $U_{5,3}|00111\rangle = (S_{2,1} \otimes I^{\otimes 3}) \cdot (S_{3,2} \otimes I^{\otimes 2}) \cdot (S_{4,3} \otimes I) \cdot (I \otimes S_{5,3})|00111\rangle = |D^5_3\rangle$. Since $|0\rangle^{\otimes n-k}|1\rangle^k$ can be constructed from $|00...0\rangle$ by applying $X$ gates, Dicke state $|D^n_k\rangle$ can be prepared from the initial state $|00...0\rangle$. The circuit complexity is shown as follows:

**Theorem 1.** *(Bärtschi and Eidenbenz, 2019) Dicke states $|D^n_k\rangle$ can be prepared with a circuit of width $n$ and depth $O(n)$ using $k$ $X$ gates and one $U_{n,k}$ gate.*

We can further reduce the number of gates (Mukherjee et al., 2020) or the circuit depth to $O(k \log \frac{n}{k})$ for constant $k$ (Bärtschi and Eidenbenz, 2022). However, for the sake of circuit simplicity, only the circuit from (Bärtschi and Eidenbenz, 2019) is considered in this paper.

## 3.2 Dicke State Preparation with Ancilla Qubits

Esser's circuit is also based on the inductive property of the Dicke state (Lemma 1). However, unlike Bärtschi's circuit, which determines the qubit sequence from right to left, Esser's circuit determines the quantum bit string in left-to-right order by considering Lemma 1 as follows:

$$|D^n_k\rangle = \sqrt{\frac{n-k}{n}}|0\rangle|D^{n-1}_k\rangle + \sqrt{\frac{k}{n}}|1\rangle|D^{n-1}_{k-1}\rangle,$$
$$|D^n_0\rangle = |0\rangle^{\otimes n}, |D^n_n\rangle = |1\rangle^{\otimes n}.$$

The lemma can be written like a Markov chain. For example, $|D^4_2\rangle = \sqrt{\frac{1}{2}}|0\rangle|D^3_2\rangle + \sqrt{\frac{1}{2}}|1\rangle|D^3_1\rangle = \sqrt{\frac{1}{6}}|00\rangle|D^2_2\rangle + \sqrt{\frac{1}{3}}|01\rangle|D^2_1\rangle + \sqrt{\frac{1}{3}}|10\rangle|D^2_1\rangle + \sqrt{\frac{1}{6}}|11\rangle|D^2_0\rangle = \sqrt{\frac{1}{6}}(|0011\rangle + |0101\rangle + |0110\rangle + |1001\rangle + |1010\rangle + |1100\rangle)$. Recall that the $R_y(\theta)$ gate can divide the existence probability of qubits to $\alpha^2$ and $1-\alpha^2$ by $R_y(2\cos^{-1}(\alpha))|0\rangle = \alpha|0\rangle + \sqrt{1-\alpha^2}|1\rangle$. In particular, when the $i$-th qubit is rotated, $|b_{i-1,j}\rangle|D^{n-(i-1)}_{k-j}\rangle = \alpha_{i,j}|b_{i-1,j}\rangle|0\rangle|D^{n-i}_{k-j}\rangle + \sqrt{1-\alpha^2_{i,j}}|b_{i-1,j}\rangle|1\rangle|D^{n-i}_{k-j-1}\rangle$ is satisfied for probability $\alpha_{i,j} = \sqrt{\binom{n-i}{k-j}/\binom{n-i+1}{k-j}} = \sqrt{(n-i+1-k+j)/(n-i+1)}$ and any binary prefix $b_{i-1,j}$ of length $i-1$ with weight $j$ for $1 \leq i \leq n$ and $0 \leq j \leq k-1$. The probability distribution is independent of the *pattern* of bit subsequence $b_{i-1,j}$ but depends on the *length* and *weight* of $b_{i-1,j}$, and thus, the number of required $R_y(\theta)$ gates to divide the probability is estimated to be $O(nk)$.

To achieve such a Markov chain for $|D^n_k\rangle$ on a quantum circuit, we just need to keep track of the Hamming weight $j$ of the $b_{i-1,j}$. We prepare ancilla $\lceil\log(k+1)\rceil$-qubits $c$ to store the weight as binary digits. When $i = 1$, $c$ is initialized to 0 and $b_{0,0} = \varepsilon$. Applying the $R_y(2\cos^{-1}(\alpha_{i,j}))$ gate yields $|b_{i-1,j}\rangle R_y(2\cos^{-1}(\alpha_{i,j}))|0\rangle_i \rightarrow \sqrt{\alpha_{i,j}}|b_{i-1,j}\rangle|0\rangle_i + \sqrt{1-\alpha^2_{i,j}}|b_{i-1,j}\rangle|1\rangle_i$. Next, $c$ is incremented by 1 for the quantum state $|b_{i-1,j}\rangle|1\rangle_i \rightarrow |b_{i,j+1}\rangle$. We show a pseudocode for constructing $|D^n_k\rangle$ with auxiliary variable $c$ in Algorithm 1.

Algorithm 1: Preparation of $|D^n_k\rangle$ with auxiliary variable $c$.

```
Input: integer $k \leq n$, $n$-qubit
       $|x_1x_2...x_n\rangle = |0\rangle^{\otimes n}$ and $\lceil\log(k+1)\rceil$
       ancilla qubits to store $c$
Output: $|D^n_k\rangle$
$c \leftarrow 0$
for $i \leftarrow 1, \dots, n$ do
    for $j \leftarrow 0, \dots, k-1$ do
        if $c = j$ then
            $|0\rangle_i \rightarrow \alpha_{i,j}|0\rangle_i + \sqrt{1-\alpha^2_{i,j}}|1\rangle_i$
    if $x_i = 1$ then
        $c \leftarrow c+1$
return $|x_1x_2...x_n\rangle$
```

### Quantum Gates for Algorithm 1

Similar to $U_{n,k}$, we construct a unitary operator $A_{n,k}$ such that $|0\rangle^{\otimes n+\lceil\log(k+1)\rceil}$ is an input and $|D^n_k\rangle|0\rangle^{\otimes\lceil\log(k+1)\rceil}$ is an output.

**Definition 6** ($A_{n,k}$)**.** *$n$-qubit unitary gate satisfying $A_{n,k}|0\rangle^{\otimes n+\lceil\log(k+1)\rceil} = |D^n_k\rangle|0\rangle^{\otimes\lceil\log(k+1)\rceil}$.*

The building blocks of $A_{n,k}$ are $C^nX$ and $C^nR_y(\theta)$

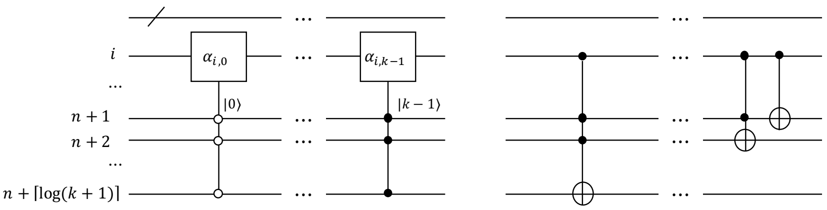

Figure 2: $R_i$ gate (left) and $N_i$ gate (right) for $1 \le i \le n$. The $R_y(2\cos^{-1}(\alpha_{i,j}))$ gate is abbreviated by $\alpha_{i,j}$.

gates. We prepare two quantum registers $|x_1...x_n\rangle$ and $|c_1...c_{\lceil\log(k+1)\rceil}\rangle$ to store the Dicke state and weight for bit strings. For these two registers, the rotation unitary $R_i$ and increment unitary $N_i$ are applied in alternating order from 1 to $n$.

**Definition 7** (Building blocks of $R_i$). *$R_i$ rotates the i-th qubit $|0\rangle_i$ according to the value of the ancilla qubit for $1 \le i \le n$ and $0 \le j \le k-1$*

$$R_i : |0\rangle_i |j\rangle_{n+1} \quad \to \left(\alpha_{i,j}|0\rangle_i + \sqrt{1-\alpha_{i,j}^2}|1\rangle_i\right)|j\rangle_{n+1},$$

*where $\alpha_{i,j} = \sqrt{\frac{n-i+1-k+j}{n-i+1}}$. $|j\rangle_{n+1}$ is the ancilla qubits $|c_1...c_{\lceil\log(k+1)\rceil}\rangle$ corresponding to the binary representation of an integer j for $0 \le j \le k-1$. $R_i$ is constructed by k $C^nR_y(\theta)$ gates shown in Figure 2 (left).*

**Definition 8** (Building blocks $N_i$). *$N_i$ increments the ancilla qubit if $|x\rangle_i = |1\rangle_i$ for $1 \le i \le n$*

$$\begin{aligned} N_i : |0\rangle_i|j\rangle_{n+1} &\to |0\rangle_i|j\rangle_{n+1} \\ |1\rangle_i|j\rangle_{n+1} &\to |1\rangle_i|j+1\rangle_{n+1} \end{aligned}$$

*$N_i$ can be constructed by k $C^nX$ gates in Figure 2 (right).*

$|D_k^n\rangle$ can be generated using a circuit with ancilla qubits, whose circuit complexity is computed as in the next theorem.

**Theorem 2.** *(Esser et al., 2021; da Silva and Park, 2022) Dicke states $|D_k^n\rangle$ can be prepared with a circuit of width $n + \lceil\log(k+1)\rceil$ and depth $O(nk\log k)$ using one $A_{n,k}$ gate.*

Note that the original circuit depth in (Esser et al., 2021) is $O(nk\log k\log\log k)$ using $2\lceil\log(k+1)\rceil$ ancilla qubits since the $C^nR_y(\theta)$ gate with $\log(k+1)$ control can be decomposed into a depth $O(\log k\log\log k)$ elementary circuit with additional $\lceil\log(k+1)\rceil$ ancilla. However, due to (da Silva and Park, 2022), one can implement any $n$-controlled single-qubit unitary gate $C^nU$ with $O(n)$ depth and $O(n^2)$ elementary gates without ancilla qubits. Therefore, we removed the $\log\log k$ term and $\lceil\log(k+1)\rceil$ additional ancilla from the original theorem.

The circuit can also be a concrete instantiation of the circuit in (Kaye and Mosca, 2001) with a smaller number of ancilla qubits of size $\lceil\log(k+1)\rceil$ than $\log(n/\varepsilon)$ of the original circuit, where $\varepsilon \ll 1$ is a precision parameter.

# 4 GENERALIZED DICKE STATE PREPARATION

In this section, we review the generalized Dicke state preparation circuit proposed in (Bärtschi and Eidenbenz, 2019). Based on the circuit, we generalize another Dicke state preparation circuit proposed in (Esser et al., 2021) with the same circuit complexity.

## 4.1 Dicke State Generalization Without Ancilla Qubits

Algorithm 2: Prepare-$|D_K^n\rangle$ without ancilla.

```
Input: $n, k$, empty quantum circuit qc
Output: qc constructs $|D_K^n\rangle$
qc.ry($2\cos^{-1}(\beta_0, n)$)
for $i \leftarrow 1, \ldots, k-1$ do
    qc.cry($2\cos^{-1}(\beta_i), n-i+1, n-i$)
qc.$U_{n,k}$ // concat $U_{n,k}$ gate
return qc
```

First, we review the concrete construction of $|D_K^n\rangle$ for any integer set $i \in K$ (Bärtschi and Eidenbenz, 2019). To construct $|D_K^n\rangle$ in polynomial gates, the authors use the property of the $U_{n,k}$ gate such that $U_{n,k}|0\rangle^{\otimes n-\ell}|1\rangle^{\otimes\ell} = |D_\ell^n\rangle$ for all $0 \le \ell \le k$. Recall that the input of $U_{n,k}$ for the original algorithm is $|0\rangle^{\otimes n-k}|1\rangle^{\otimes k}$. However, because $U_{n,k}$ accepts weights $\ell \le k$, $U_{n,k}$ also returns $|D_\ell^n\rangle$ for the input $|0\rangle^{\otimes n-\ell}|1\rangle^{\otimes\ell}$ for all $\ell < k$. Thus, to achieve a superposition of Dicke states, one can set $k \leftarrow \max K$ and input the superposition of $|00..00\rangle$, $|00..01\rangle$, …, $|00..\underbrace{11..11}_{k}\rangle$ to the $U_{n,k}$ gate. Since we want the equal superposition of Dicke states, the input superposition for $U_{n,k}$ is given by $\sum_{0\le i\le n}\alpha_i|D_i^n\rangle$, where $\alpha_i = \sqrt{\binom{n}{i}/|C_K^n|}$

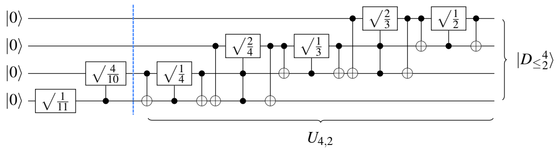

Figure 3: Quantum circuit to prepare $|D_{\leq 2}^{4}\rangle = \sqrt{\frac{1}{11}} \sum_{x\in\{0,1\}^4,\mathrm{wt}(x)\leq 2} |x\rangle$ generated from Algorithm 2.

for all $i \in K$ and $\alpha_i = 0$ otherwise. Such a superposition can be constructed with one $R_y(\theta)$ gate and $k-1$ $CR_y(\theta)$ gates. Algorithm 2 is a Qiskit-like pseudocode to generate a quantum circuit to prepare $|D_K^n\rangle$, where $\beta_i = (\sqrt{\alpha_i^2/(1-\sum_{j=0}^{i-1}\alpha_j^2}).$ $\mathtt{qc.a}$ represents the append of gate $\mathtt{a}$ to circuit $\mathtt{qc}$. The $R_y(\theta)$ gate is denoted by $\mathtt{ry}(\theta, i)$, where $i$ is the target qubit to which the gate is applied. The $CR_y(\theta)$ gate is denoted by $\mathtt{cry}(\theta, i, j)$, where $i$ is the control and $j$ is the target qubit. We show an example of the circuit for $|D_K^4\rangle$ with $K = \{0,1,2\}$ in Figure 3.

**Theorem 3.** *(Bärtschi and Eidenbenz, 2019) Generalized Dicke states $|D_K^n\rangle$ can be prepared with a circuit of width $n$ and depth $O(n)$ using one $R_y(\theta)$ gate, $k-1$ $CR_y(\theta)$ gates and one $U_{n,k}$ gate with $k \leftarrow \max K$.*

## 4.2 Dicke State Generalization with Ancilla Qubits

We show that Esser's quantum circuit can also be generalized by considering the Markov chain for the sum of combinations $\sum_{i\in K}\binom{n}{i}$ by using the same building blocks $R_i$ and $N_i$. To do so, we set $k \leftarrow \max K$ to store the weight $w$ in the ancilla qubits for all $0 \leq w \leq i$, $i \in K$. The angle value $\alpha_{i,j}$ in block $R_i$ is replaced by

$$\alpha_{i,j} = \beta_{i,j}/\beta_{i-1,j}, \tag{6}$$

where $\beta_{i,j} = \sqrt{\sum_{\ell\in K,\ell\geq j}\binom{n-i}{\ell-j}}$. The entire algorithm to construct $|D_K^n\rangle$ is described in Algorithm 3. The $C^nR_y(\theta)$ gate is denoted by $\mathtt{mcry}(\theta, j, i)$, where $j$ is the control state and $i$ is the target. Increment$(i, \mathtt{qc})$ in Line 7 appends $N_i$ unitary to $\mathtt{qc}$. The correctness of the Algorithm 3 is shown as follows.

**Theorem 4.** *Algorithm 3 outputs a circuit to prepare $|D_K^n\rangle$ from $|0\rangle^{\otimes n+\lceil\log(k+1)\rceil}$ with a circuit of width $n+\lceil\log(k+1)\rceil$ and depth $O(nk\log k)$ , where $k \leftarrow \max K$.*

*Proof.* In Line 3, Algorithm 3 rotates the 1st qubits by $R_y(2\cos^{-1}(\alpha_{1,0}))$:

$$|0\rangle^{\otimes n} \xrightarrow{\alpha_{1,0}} \alpha_{1,0}|0\rangle^{\otimes n} + \sqrt{1-\alpha_{1,0}^2}|1\rangle|0\rangle^{\otimes n-1}$$
$$= \frac{\beta_{1,0}}{\beta_{0,0}}|0\rangle^{\otimes n} + \frac{\beta_{1,1}}{\beta_{0,0}}|1\rangle|0\rangle^{\otimes n-1}$$

Algorithm 3: Prepare-$|D_K^n\rangle$ with ancilla.

```
  Input: $n, k \leftarrow \max K$, qc with registers
         $|\mathtt{b}\rangle \leftarrow |0\rangle^{\otimes n}, |\mathtt{c}\rangle \leftarrow |0\rangle^{\otimes\lceil\log(k+1)\rceil}$
  Output: qc constructs $|D_K^n\rangle$
1 for $i \leftarrow 1,\ldots,n$ do
2     if $i = 1$ then
3         qc.ry($2\cos^{-1}(\alpha_{1,0}), 1$)
4     else
5         for $j \leftarrow 0,\ldots,k-1$ do
6             qc.mcry($2\cos^{-1}(\alpha_{i,j}), |\mathtt{c}\rangle = j, i$)
7     qc $\leftarrow$ Increment($i$, qc)
8 return qc
```

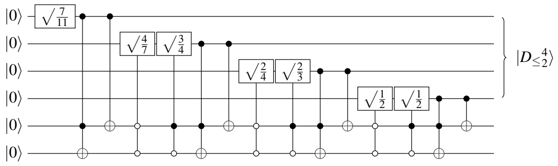

Figure 4: Quantum circuit to prepare $|D_{\leq 2}^{4}\rangle$ generated from Algorithm 3.

since $\sqrt{1-\alpha_{i,j}^2} = \beta_{i,j+1}/\beta_{i-1,j}$ from $\binom{i}{j} = \binom{i-1}{j} + \binom{i-1}{j-1}$. After $n$ rotations, we obtain

$$\sum_{b_{n,w_n}} \left(\prod_{i=1}^{n} \frac{\beta_{i,w_i}}{\beta_{i-1,w_{i-1}}}\right) |b_{n,w_n}\rangle = \sum_{b_{n,w_n}} \frac{\beta_{n,w_n}}{\beta_{0,w_0}} |b_{n,w_n}\rangle, \tag{7}$$

where $w_0 = 0$ and $w_{i-1} \leq w_i \leq w_{i-1}+1$. If $w_n \in K$, $\beta_{n,w_n} = \sqrt{\binom{0}{0}} = 1$ and 0 otherwise. Since $\beta_{0,0} = \sqrt{|C_K^n|}$, we obtain the equal superposition for all states $|b_{n,w_n}\rangle$, $w_n \in K$ with probability $\frac{1}{|C_K^n|}$. The circuit complexity is the same as in (Esser et al., 2021) since we only change the angle of the $R_y(\theta)$ gate and $C^nX$ gates. □

The whole quantum circuit for $|D_K^4\rangle$ with $K = \{0,1,2\}$ is presented in Figure 4.

# 5 EXPERIMENTS

To verify the performance of the generalized Dicke state preparation circuits, we conduct several experiments on both the quantum simulator and the actual quantum machine from IBM Quantum. All algorithms are implemented in Qiskit.

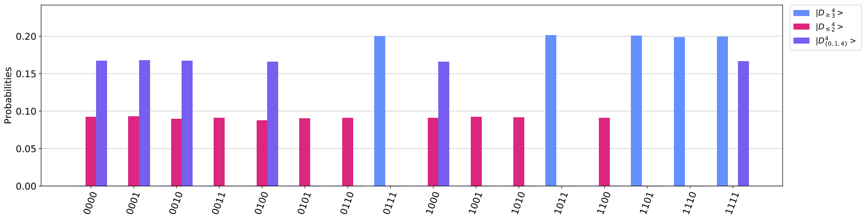

Figure 5: Probability distribution obtained from the noise-free simulated quantum circuit for $10^5$ shots corresponding to $|D^4_{\geq 3}\rangle$, $|D^4_{\leq 2}\rangle$, and $|D^4_{\{0,1,4\}}\rangle$.

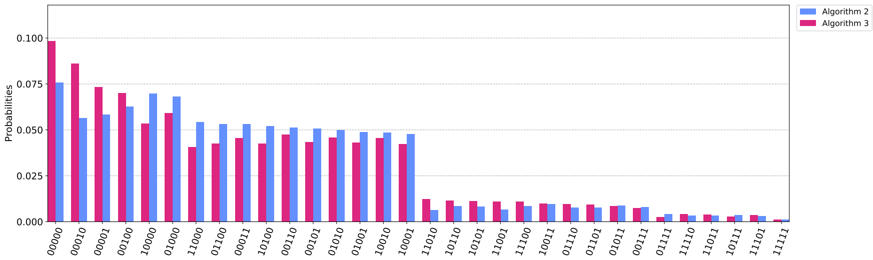

Figure 6: Comparison of the noisy probability distribution obtained from Algorithm 2 and Algorithm 3 corresponding to $|D^5_{\leq 2}\rangle$.

## 5.1 Noise Free Simulation

First, we verify that Algorithm 2 and Algorithm 3 indeed construct generalized Dicke states using the QASM quantum simulator under the condition that the circuit is noise-free. Circuit runs and measurements were repeated numerous times specified by *shots*. We observe whether the desired probability distribution (superposition) is constructed by aggregating the quantum state measured at each run. In our experiments, we built circuits that prepare $|D^4_{\geq 3}\rangle = \sqrt{\frac{1}{5}}\sum_{x\in\{0,1\}^4,\mathrm{wt}(x)\geq 3}|x\rangle$, $|D^4_{\leq 2}\rangle = \sqrt{\frac{1}{11}}\sum_{x\in\{0,1\}^4,\mathrm{wt}(x)\leq 2}|x\rangle$ and $|D^4_{\{0,1,4\}}\rangle = \sqrt{\frac{1}{6}}\sum_{x\in\{0,1\}^4,\mathrm{wt}(x)\in\{0,1,4\}}|x\rangle$. The results of Algorithm 3 for $10^5$ shots are shown in Figure 5. The figure shows that our algorithms indeed generate uniform superpositions $|D^4_{\geq 3}\rangle$, $|D^4_{\leq 2}\rangle$, and $|D^4_{\{0,1,4\}}\rangle$ by observing the distribution aggregated from the measured quantum state for each shot. For Algorithm 2, we confirm that the uniform distributions are correctly constructed. Note that since Algorithm 2 and Algorithm 3 are deterministic, no quantum state with undesired Hamming weights can be obtained as long as the quantum circuit is noise-free.

## 5.2 Noisy Simulation

Next, we compared the probability distribution gathered from Algorithm 2 and Algorithm 3 in the noisy simulation model. In the experiment, the error rate of a 1-qubit gate such as $X$ is set to $10^{-3}$ and that of CNOT is set to $10^{-2}$. These values are plausible and are derived from the error rates of IBM's quantum computers, such as Falcon (IBM, 2022). Figure 6 shows the results of running the circuit generated from each algorithm that obtains $|D^5_{\leq 2}\rangle = \sqrt{\frac{1}{16}}\sum_{x\in\{0,1\}^5,\mathrm{wt}(x)\leq 2}|x\rangle$ for $10^5$ shots. Unlike the noiseless case, quantum states with incorrect weights $\geq k$ were observed with a small probability. For the desired states, Algorithm 3 generated $|00000\rangle$ and states near $|00000\rangle$ with a higher probability than other states. Therefore, the result is somewhat different from the uniform superposition. Algorithm 2 seems to generate correct weight states more uniformly than Algorithm 3. We transpile two circuits generated from Algorithm 2 and Algorithm 3. The backend of the transpiler is `ibm_nairobi` from IBMQ, which is a 7-qubits quantum machine we will use in our experiments in the next subsection. Basic gates implemented in `ibm_nairobi` are `['id', 'rz', 'sx', 'x', 'cx', 'reset']`. The depth of the transpiled circuit of Algorithm 3 is 382, which is twice as large as 189 of Algorithm 2. The result is reasonable given that the depth complexity of Algorithm 3 is $O(k\log k)$ larger than that of Algorithm 2.

## 5.3 Real Quantum Machine

We deployed Algorithm 2 and Algorithm 3 on a real quantum machine produced by IBM Quantum for toy examples. The 7-qubits quantum

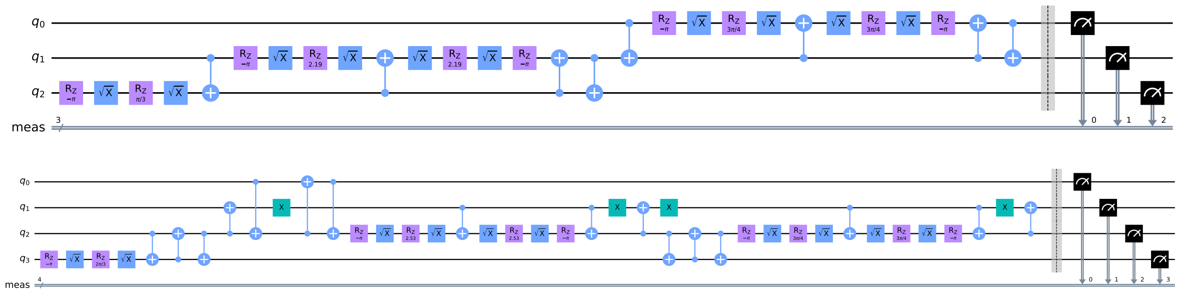

Figure 7: Transpiled quantum circuits for $|D^3_{\leq 1}\rangle$ generated from Algorithm 2 (top) and Algorithm 3 (bottom).

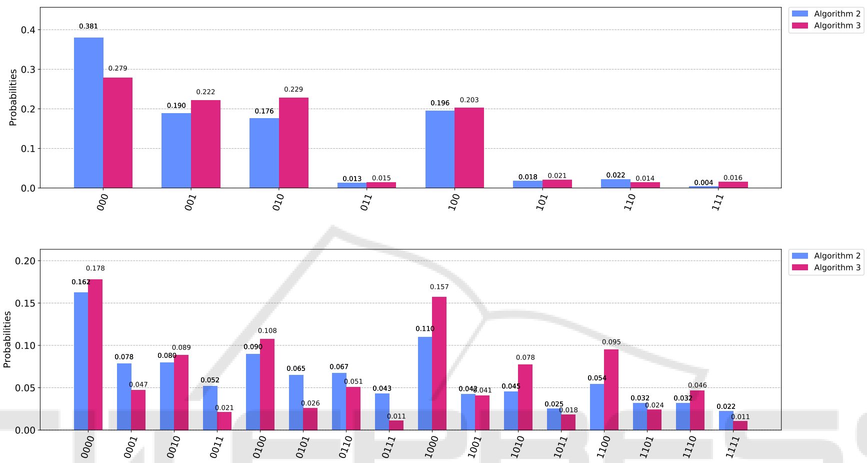

Figure 8: Probability distribution obtained from Algorithm 2 and Algorithm 3 on `ibm_nairobi` quantum machine for $|D^3_{\leq 1}\rangle$ (top) and $|D^4_{\leq 2}\rangle$ (bottom).

Table 1: Gate summary of quantum circuits for Algorithm 2 and Algorithm 3 transpiled in `ibm_nairobi` quantum machine.

| Target State | $\|D^3_{\leq 1}\rangle$ | | $\|D^4_{\leq 2}\rangle$ | |
|---|---|---|---|---|
| Algorithm | Alg. 2 | Alg. 3 | Alg. 2 | Alg. 3 |
| $X$ gates | 0 | 4 | 0 | 6 |
| $\sqrt{X}$ gates | 10 | 10 | 28 | 37 |
| $Rz$ gates | 10 | 10 | 52 | 141 |
| CNOT gates | 8 | 16 | 79 | 234 |
| Circuit Width | 3 | 4 | 4 | 6 |
| Circuit Depth | 29 | 39 | 133 | 298 |

machine `ibm_nairobi` was utilized in the experiment. Our target generalized Dicke states are $|D^3_{\leq 1}\rangle = \frac{1}{2}(|000\rangle + |001\rangle + |010\rangle + |100\rangle)$ and $|D^4_{\leq 2}\rangle = \frac{1}{\sqrt{11}}(|0000\rangle + |0001\rangle + |0010\rangle + |0100\rangle + |1000\rangle + |0011\rangle + |0110\rangle + |1100\rangle + |0101\rangle + |1010\rangle + |1001\rangle)$. Again, we compared the probability distribution achieved by Algorithm 2 and Algorithm 3. Table 1 summarizes the number of gates for each circuit transpiled in the `ibm_nairobi` quantum machine. Figure 7 displays the transpiled circuits to construct $|D^3_{\leq 1}\rangle$ for both algorithms. You can also see the circuits for $|D^4_{\leq 2}\rangle$ in Appendix 6. The average CNOT error during the experiment was 0.01225. The number of shots is 20000. To eliminate coincidence, 100 batch executions for the same circuit were performed and we calculated the average value. The result is displayed in Figure 8. For $|D^3_{\leq 1}\rangle$, both algorithms can generate correct weight states with high probability (94.2% for Algorithm 2 and 93.4% for Algorithm 3). For the standard deviation in the correct states, Algorithm 3 achieves 0.118, which is smaller than 0.145 for Algorithm 2. Thus, Algorithm 3 obtained better uniformity than Algorithm 2 for $|D^3_{\leq 1}\rangle$. The success probability for $|D^4_{\leq 2}\rangle$ is 84.6% for Algorithm 2 and

88.9% for Algorithm 3. The standard deviation in the correct states is 0.0516 for Algorithm 2 and 0.0631 for Algorithm 3. In particular, it is surprising that the success probability of Algorithm 3, which uses more than 200 CNOT gates, is higher than that of Algorithm 2 even though the CNOT error rate exceeds 0.01. The presumed reason for this is that Algorithm 3 has a circuit structure with no dependencies among the qubits except for the ancilla. Since errors are assumed to accumulate in the ancilla, quantum error correction should be performed preferentially on the ancilla qubits. To construct generalized Dicke states for larger $n, k$ precisely, a quantum computer with error correction capability is essentially needed.

## 6 CONCLUSION

We generalized a quantum circuit to prepare a superposition of Dicke states proposed by (Esser et al., 2021) with the same circuit complexity. We implemented and compared two generalized Dicke state preparation circuits using the Qiskit library. In our experiments, we validated that both circuits can construct the generalized Dicke state in a noisy quantum simulator and a real quantum device. Future work includes developing end-to-end quantum circuits to solve a combinatorial problem employing the superposition constructed from a generalized Dicke state preparation circuit.

## REFERENCES

Acasiete, F., Agostini, F. P., Moqadam, J. K., and Portugal, R. (2020). Implementation of quantum walks on IBM quantum computers. *Quantum Information Processing*, 19(12):1–20.

Ambainis, A. (2004). Quantum walk algorithm for element distinctness. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pages 22–31.

Bärtschi, A. and Eidenbenz, S. (2019). Deterministic preparation of Dicke states. In *International Symposium on Fundamentals of Computation Theory*, pages 126–139. Springer.

Bärtschi, A. and Eidenbenz, S. (2022). Short-depth circuits for Dicke state preparation. In *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE. To appear.

Bastin, T., Thiel, C., von Zanthier, J., Lamata, L., Solano, E., and Agarwal, G. S. (2009). Operational determination of multiqubit entanglement classes via tuning of local operations. *Phys. Rev. Lett.*, 102:053601.

Brassard, G., Hoyer, P., Mosca, M., and Tapp, A. (2002). Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 305:53–74.

Chailloux, A., Debris-Alazard, T., and Etinski, S. (2021). Classical and quantum algorithms for generic syndrome decoding problems and applications to the lee metric. In *International Conference on Post-Quantum Cryptography*, pages 44–62. Springer.

Chakraborty, K., Choi, B.-S., Maitra, A., and Maitra, S. (2014). Efficient quantum algorithms to construct arbitrary Dicke states. *Quantum information processing*, 13(9):2049–2069.

Childs, A. M., Farhi, E., Goldstone, J., and Gutmann, S. (2000). Finding cliques by quantum adiabatic evolution. *arXiv preprint quant-ph/0012104*.

Cook, J., Eidenbenz, S., and Bärtschi, A. (2020). The quantum alternating operator ansatz on maximum $k$-vertex cover. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 83–92. IEEE.

Cruz, D., Fournier, R., Gremion, F., Jeannerot, A., Komagata, K., Tosic, T., Thiesbrummel, J., Chan, C. L., Macris, N., Dupertuis, M.-A., et al. (2019). Efficient quantum algorithms for *ghz* and *w* states, and implementation on the IBM quantum computer. *Advanced Quantum Technologies*, 2(5-6):1900015.

da Silva, A. J. and Park, D. K. (2022). Linear-depth quantum circuits for multiqubit controlled gates. *Phys. Rev. A*, 106:042602.

Esser, A., Ramos-Calderer, S., Bellini, E., Latorre, J. I., and Manzano, M. (2021). An optimized quantum implementation of ISD on scalable quantum resources. *arXiv preprint arXiv:2112.06157*.

Esser, A., Ramos-Calderer, S., Bellini, E., Latorre, J. I., and Manzano, M. (2022). Hybrid decoding – classical-quantum trade-offs for information set decoding. In *International Conference on Post-Quantum Cryptography*. Springer. To appear.

Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219.

IBM (2022). IBM quantum. https://quantum-computing.ibm.com/.

Kaye, P. and Mosca, M. (2001). Quantum networks for generating arbitrary quantum states. In *Optical Fiber Communication Conference and International Conference on Quantum Information*, page PB28. Optica Publishing Group.

Lamata, L., Lopez, C. E., Lanyon, B., Bastin, T., Retamal, J. C., and Solano, E. (2013). Deterministic generation of arbitrary symmetric states and entanglement classes. *Physical Review A*, 87(3):032325.

Mandviwalla, A., Ohshiro, K., and Ji, B. (2018). Implementing grover's algorithm on the IBM quantum computers. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 2531–2537. IEEE.

Moreno, M. and Parisio, F. (2018). All bipartitions of arbitrary Dicke states. *arXiv preprint arXiv:1801.00762*.

Mukherjee, C. S., Maitra, S., Gaurav, V., and Roy, D. (2020). Preparing Dicke states on a quantum computer. *IEEE Transactions on Quantum Engineering*, 1:1–17.

Niroula, P., Shaydulin, R., Yalovetzky, R., Minssen, P., Herman, D., Hu, S., and Pistoia, M. (2022). Constrained quantum optimization for extractive summarization on a trapped-ion quantum computer.

Perriello, S., Barenghi, A., and Pelosi, G. (2021). A complete quantum circuit to solve the information set decoding problem. In *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 366–377. IEEE.

Shende, V. V., Bullock, S. S., and Markov, I. L. (2006). Synthesis of quantum-logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(6):1000–1010.

# APPENDIX

We show the quantum circuits for $|D_{\leq 2}^{4}\rangle$ with elementary gates transpiled by the `ibm_nairobi` backend in the IBM quantum experience service in Figure 9 and Figure 10.

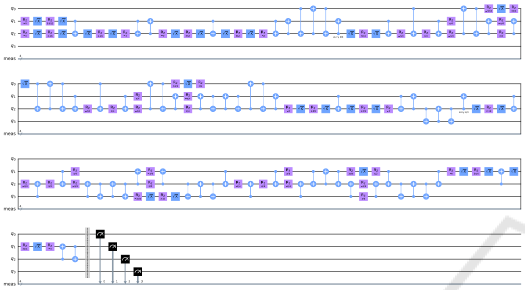

Figure 9: Transpiled quantum circuit for $|D_{\leq 2}^{4}\rangle$ generated from Algorithm 2.

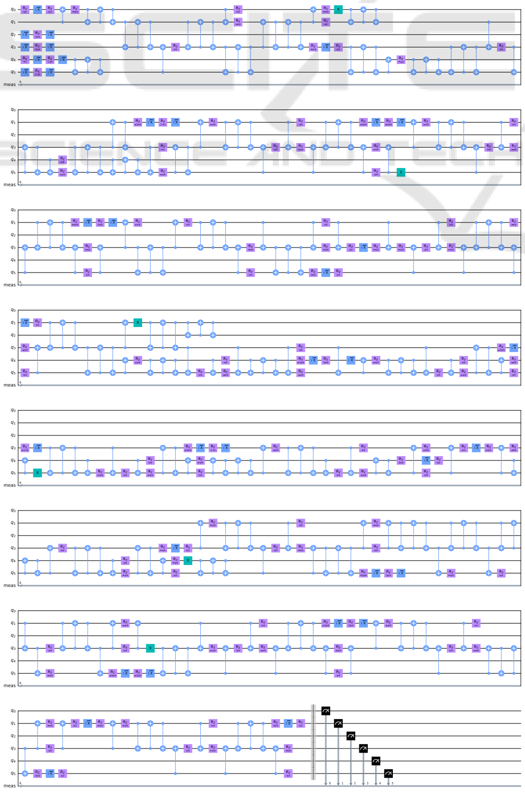

Figure 10: Transpiled quantum circuit for $|D_{\leq 2}^{4}\rangle$ generated from Algorithm 3.