




OPEN

Multi-qubit quantum computing using discrete-time quantum walks on closed graphs

Prateek Chawla^{1,2}, Shivani Singh^{1,5}, Aman Agarwal^{1,3}, Sarvesh Srinivasan^{1,4} & C. M. Chandrashekar^{1,2,6}

Universal quantum computation can be realised using both continuous-time and discrete-time quantum walks. We present a version based on single particle discrete-time quantum walk to realize multi-qubit computation tasks. The scalability of the scheme is demonstrated by using a set of walk operations on a closed lattice form to implement the universal set of quantum gates on multi-qubit system. We also present a set of experimentally realizable walk operations that can implement Grover's algorithm, quantum Fourier transformation and quantum phase estimation algorithms. An elementary implementation of error detection and correction is also presented. Analysis of space and time complexity of the scheme highlights the advantages of quantum walk based model for quantum computation on systems where implementation of quantum walk evolution operations is an inherent feature of the system.

Quantum computing is poised to provide supremacy over classical computing using quantum mechanical phenomena such as superposition, interference and entanglement. Physical systems like, superconducting circuits^{1–4}, nuclear magnetic resonance (NMR) systems^{5–9}, ion traps^{10,11}, ultra-cold atoms in optical lattice^{12,13} and photonics^{14–18} have been successfully engineered to demonstrate small scale quantum processors and implement quantum simulations and computational tasks. The noisy-intermediate scale quantum processors we have today are still far from the one that can be used for performing useful tasks that are inaccessible by the existing powerful classical computers. Different models for quantum computation and the engineering of different physical systems and architecture to build scalable processors has been explored for a long time now. For example, measurement based quantum computing model^{19–22}, adiabatic quantum computing model^{23–27}, and KLM-linear optical quantum computing¹⁵ are some of the examples in addition to standard circuit based quantum computation model. The use of quantum walks^{28–31}, which are part of several quantum algorithms^{32–34} developed to outperform classical algorithms at computational tasks has also been proposed to develop a scheme for universal quantum computation model.

A quantum walk based quantum computing model was first introduced on unweighted graphs using the continuous-time quantum walk³⁵ and a corresponding scheme using discrete-time quantum walk was proposed later³⁶. Recently, we proposed a new scheme using a single qubit discrete-time quantum walk on a closed lattice setting³⁷. Compared to the earlier discrete-time quantum walk scheme which requires a large number of real qubits and higher dimensional coin operations, our scheme defines computation purely as a sequence of position dependent coin and shift operations on a system with a single real qubit and position space as an additional computational basis. With the advent of photonics-based quantum computing systems^{17,38} and the efficient realization of quantum walks³⁹, the potential realizability of our proposed scheme gains more relevance. Decreased resource requirements for implementation of this scheme is one of the major factors that make it experimentally more feasible to implement in any hardware platform where controllable discrete-time quantum walks have been demonstrated.

Here we present a detailed extension of the simple, implementable quantum computing scheme using a single particle discrete-time quantum walk which can be scaled to higher dimensions³⁷. Along with the position Hilbert space on which the quantum walks are defined, the discrete-time quantum walk provides additional degree of

¹The Institute of Mathematical Sciences, C. I. T. Campus, Taramani, Chennai 600113, India. ²Homi Bhabha National Institute, Training School Complex, Anushakti Nagar, Mumbai 400094, India. ³BITS-Pilani, K. K. Birla Goa Campus, NH17B, Bypass Road, Zuarinagar, Goa 403726, India. ⁴Birla Institute of Technology and Science, Pilani Campus, Pilani 333031, India. ⁵FNSPE, Czech Technical University in Prague, Brehova 7, 119 15, Praha 1, Prague, Czech Republic. ⁶Quantum Optics and Quantum Information, Department of Instrumentation and Applied Physics, Indian Institute of Science, Bengaluru, India. ✉email: prateekc@imsc.res.in

freedom in the form of coin Hilbert space that can be exploited to achieve control over the states to perform computing operations. This model can be implemented on a photonic or lattice based quantum systems where one photon or free particle can act as coin that can be used to perform computation when entangled with the position space. We propose the use of multiple sets of closed graph with four sites and four edges to act as a system with $2^{(N-1)}$ -dimensional position space. Each graph is equivalent to a two-qubit state and n -sets of closed graph provides $2n$ -qubit equivalent states. With the help of the coin and shift operations, the coin and position state of the particle can be evolved into the desired output state⁴⁰. We also demonstrate the effectiveness of our scheme by presenting a combination of quantum walk operations to implement the quantum algorithms like Grover's search algorithm, quantum Fourier transformation and phase estimation algorithms. Further, an elementary implementation of single qubit error detection (3-qubit code) for both bit-flip and phase-flip errors, and error correction using a 5-qubit code is presented. We also discuss the space and time complexity of the scheme in a generic sense to highlight the possible advantages of the quantum walk based scheme.

In "Quantum walk on higher-qubit equivalent systems", we present a brief discussion on the discrete-time quantum walk and show the scalability of the single qubit quantum computational scheme to N -qubit equivalent system by expanding the position space. "Implementing hadamard, phase, and controlled-NOT gates on N -qubit equivalent system" shows the implementation of universal gates on this N -qubit equivalent system, and in "Grover's search algorithm on three qubit equivalent quantum walk scheme", "Quantum Fourier transformation on three-qubit equivalent quantum walk scheme" and "Phase estimation algorithm on three qubit equivalent quantum walk scheme" we present schemes for realization of Grover's search algorithm, quantum Fourier transformation and phase estimation algorithm on the DTQW-based system, respectively. We discuss the space and time complexity of quantum walk based scheme in "Quantum space and time complexity", and explore a basic implementation of quantum error detection and correction codes in "Single-qubit error detection". We present our conclusions and future outlook for this work in "Conclusion".

Quantum walk on higher-qubit equivalent systems

The dynamics of the discrete-time quantum walk on a closed graph is defined on a Hilbert space $\mathcal{H} = \mathcal{H}_c \otimes \mathcal{H}_p$ where, \mathcal{H}_c is the coin Hilbert space with internal degrees of freedom and \mathcal{H}_p is the position Hilbert space defined by closed set of points in the position space⁴¹. For the computation model proposed in this work, we choose the position Hilbert space to be defined by the multiple sets of closed graphs of 4-states spanned by $|x\rangle = \{|0\rangle, |1\rangle, |2\rangle, |3\rangle\}$. The evolution operation on this setup of discrete-time quantum walk is described by the action of the unitary quantum coin operation \hat{C} on the coin state followed by the conditional position shift operation on the desired set of closed graph of the position space.

The general form of position shift operator for discrete-time quantum walk on a closed graph, that translates to the left or right conditioned on the coin states with μ internal degrees of freedom is given as,

$$\hat{S}_{\pm}^{\alpha} = \sum_{l \in \mathbb{Z}} \left[|\alpha\rangle\langle\alpha| \otimes |l \pm 1 \pmod{4}\rangle\langle l| + \sum_{\beta \neq \alpha}^{\mu} (|\beta\rangle\langle\beta| \otimes |l\rangle\langle l|) \right]. \quad (1)$$

Here, $\{|\alpha\rangle, |\beta\rangle\} \in \mathcal{H}_c$ are the basis states of coin Hilbert space \mathcal{H}_c and $|l\rangle$ are the basis states of position Hilbert space \mathcal{H}_p . The general form of the quantum coin operator with two internal degree of freedom $\mathcal{H}_c = \text{span}\{|0\rangle, |1\rangle\}$ is given by SU(2) operator of the form,

$$\hat{C}(\xi, \zeta, \theta) = \begin{bmatrix} e^{i\xi} \cos(\theta) & e^{i\zeta} \sin(\theta) \\ -e^{-i\xi} \sin(\theta) & e^{-i\zeta} \cos(\theta) \end{bmatrix}. \quad (2)$$

This set of operators along with the identity operator \mathbb{I} can be considered a generic set of operators that describes the scalable quantum computation scheme using discrete-time quantum walk, hereafter called the quantum walk in this text.

Quantum computation using quantum walk. The scheme presented for universal quantum computation on quantum walk for three qubit equivalent system³⁷ can be scaled to a larger qubit system by using the same coin in conjunction with different sets of closed graph of the position space. This method will expand the shift operator with the increase of the number of closed graphs of four-sites, but can be scaled as far as the scheme goes.

The form of shift operators which is used throughout for scaling of the universal computation model for input state $|k\rangle \otimes_{i=1}^n |m_i\rangle$ will be given as,

$$S_{j,\pm}^k = \sum_l [|k\rangle\langle k| \otimes \mathbb{I}^{\otimes j-1} \otimes |l \pm 1 \pmod{4}\rangle\langle l| \otimes \mathbb{I}^{\otimes n-j} + |p \neq k\rangle\langle p| \otimes \mathbb{I}^{\otimes n}], \quad (3)$$

where n is the total number of closed graphs and j indicates the closed graph on which the shift operation is performed. $\{|k\rangle, |p\rangle\} \in \mathcal{H}_c$ are states in the coin Hilbert space with two internal degree of freedom and $|l\rangle$ represents the four states on the four-site closed graph, and number of closed graph is n . The number of states for this case will be equivalent to the number of states in the combined state of the Hilbert-space $\mathcal{H}_c \otimes \mathcal{H}_p$, where \mathcal{H}_p has dimension $2^{(N-1)}$, and N is the total number of qubits in the system. The evolution operation on this system can be interpreted as the shift operation on the j th closed graph representing the 'selected' position space and identity operation on the rest of the closed sets of the position space, as shown in Fig. 1.

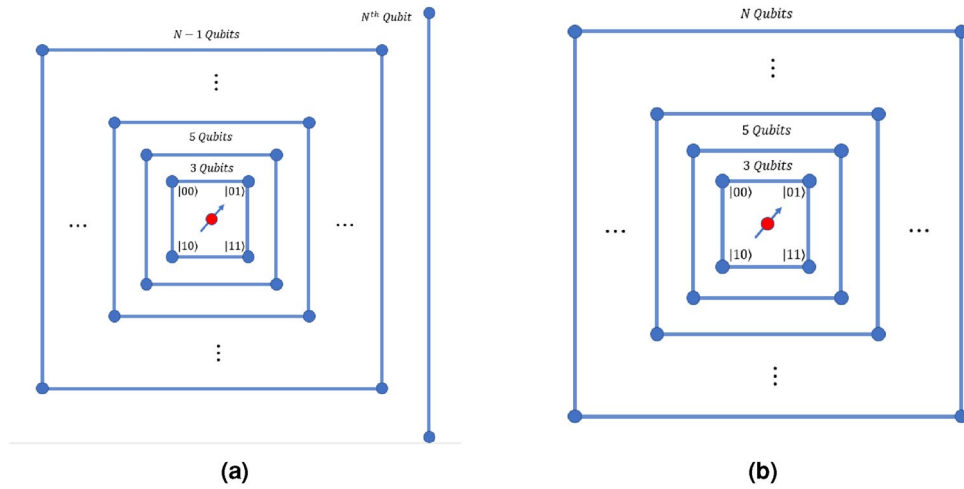


Figure 1. Scaling of the quantum walk scheme to N -qubit systems. **(a)** Case where N is even, and **(b)** the case where N is odd. When N is even (case (a)), then $(N/2 - 1)$ graphs with four sites and one graph with two sites are required. In case (b), $(N - 1)/2$ graphs with four sites are required to implement universal quantum computation.

This can be then used to derive the \hat{W} operator, in order to implement the Hadamard gate on N -qubit system and a specific case of this \hat{W} -operator is used in Ref.³⁷. An N -qubit system will require $n = \lfloor \frac{N-2}{2} \rfloor$ sets of four-site closed graph and one set of two-site graph with one edge if N is even, and $n = \lfloor \frac{N-1}{2} \rfloor$ sets of four-site closed graphs if N is odd. In order to simplify notation, we choose $|m_j\rangle$ to represent the position state of the j th set of closed graphs. The complete state of the position space is given by $|m\rangle$, which is defined as,

$$|m\rangle \equiv \bigotimes_{i=1}^n |m_i\rangle. \tag{4}$$

Then, the \hat{W} operators on state $|m_j\rangle$ with $1 < j < n$ is defined as,

$$\hat{W}_{j,\pm}^0 |k\rangle |m\rangle = \left[(\sigma_x \otimes |m\rangle \langle m| + \mathbb{I} \otimes \sum_{l \neq m} |l\rangle \langle l|) S_{j,\pm}^0 (\sigma_x \otimes \mathbb{I}^{\otimes n}) \right] |k\rangle |m\rangle, \tag{5}$$

$$\hat{W}_{j,\pm}^1 |k\rangle |m\rangle = \left[(\sigma_x \otimes |m\rangle \langle m| + \mathbb{I} \otimes \sum_{l \neq m} |l\rangle \langle l|) S_{j,\pm}^1 (\sigma_z \otimes \mathbb{I}^{\otimes n}) \right] |k\rangle |m\rangle. \tag{6}$$

A note on notation Here, uppercase letters are used to represent a particular qubit and lowercase letters refer to the order of the closed graph. It may also be observed from Fig. 1 that the l th qubit belongs to the closed graph of order $i = \frac{l}{2}$ if l is even and $i = \frac{l-1}{2}$ if l is odd.

In an abbreviated notation, the shift operator is written as,

$$S_{j,\pm}^k = \sum_{l,m} \left[|k\rangle \langle k| \otimes |l \pm 1 \pmod 4\rangle \langle l| + |m \neq k\rangle \langle m| \otimes \mathbb{I}_p \right]. \tag{7}$$

Implementing Hadamard, Phase, and controlled-NOT gates on an N -qubit equivalent system

Hadamard gate. In this scheme, the Hadamard gate can be implemented on any qubit of N -qubit equivalent system by redefining the Hadamard gates \hat{H}_2 and \hat{H}_3 in Ref.³⁷. Hadamard operation on the j th level of the closed graph, when the coin state is $|k\rangle$ and position state is $|m\rangle$ as given by Eq. (4), can be implemented on the quantum walk scheme by evolving the initial state by using Eq. (8) when the Hadamard gate is applied on the $(2j)$ th-qubit and by using Eq. (9) when the Hadamard gate is applied on the $(2j + 1)$ th-qubit.

$$\hat{H}_{2,j}^k |k\rangle |m\rangle = \left[\hat{W}_{j,-}^{k \pmod 2} |0_j\rangle \langle 0_j| + \hat{W}_{j,+}^{k \pmod 2} |1_j\rangle \langle 1_j| + \hat{W}_{j,-}^{(k+1) \pmod 2} |3_j\rangle \langle 3_j| + \hat{W}_{j,+}^{(k+1) \pmod 2} |2_j\rangle \langle 2_j| \right] \left(\hat{H}_1 \otimes \mathbb{I}_2^{\otimes N} \right), \tag{8}$$

$$\hat{H}_{3,j}^k |k\rangle |m\rangle = \left[\hat{W}_{j,+}^{k \bmod 2} |0_j\rangle \langle 0_j| + \hat{W}_{j,-}^{(k+1) \bmod 2} |1_j\rangle \langle 1_j| + \hat{W}_{j,+}^{(k+1) \bmod 2} |3_j\rangle \langle 3_j| + \hat{W}_{j,-}^{k \bmod 2} |2_j\rangle \langle 2_j| \right] \left(\hat{H}_1 \otimes \mathbb{I}_2^{\otimes N} \right). \quad (9)$$

Thus, the \hat{H} corresponds to a position-dependent evolution operator in quantum walk scheme which applies to the appropriate vertices of the desired closed graph in the scaling diagram as shown in Fig. 1. Here the eigenstates of the 2-qubit equivalent j th closed system are $|m_j\rangle$, $m = \{0, 1, 2, 3\}$. The Hadamard on any qubit $Q > 1$ can be expressed on discrete-time quantum walk scheme in the form of evolution operator $\hat{H}_{i,j}^k$, where $i \in \{2, 3\}$ and j is the level of the closed graph such that the relation between j and Q is $j = \lfloor \frac{Q}{2} \rfloor$, i.e.,

$$\hat{H}_Q^k = \begin{cases} \hat{H}_{2,j}^k & \text{for even } Q \\ \hat{H}_{3,j}^k & \text{for odd } Q. \end{cases} \quad (10)$$

A special case arises when the last qubit $Q = N$ is even and scaling is illustrated by Fig. 1(a). In this case,

$$\hat{H}_Q^k = \hat{H}_{3,n}^k. \quad (11)$$

In case $Q = 1$, the Hadamard gate can be reduced to a coin operation $\hat{H}_1 = \hat{C}(0, 0, \frac{\pi}{4}) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ with an identity shift operator.

nnPhase gate. The phase gate can be implemented on an N -qubit equivalent system in a manner similar to the Hadamard gate. Therefore, phase applied to the Q th qubit ($Q \in \{2, 3, \dots, N\}$) can be expressed in terms of the level j of the closed graph as,

$$\hat{P}_Q = \begin{cases} \hat{P}_{2,j} & \text{for even } Q \\ \hat{P}_{3,j} & \text{for odd } Q, \end{cases} \quad (12)$$

where $\hat{P}_{2,j}$ and $\hat{P}_{3,j}$ are given as,

$$P_{2,j} = \mathbb{I} \otimes (|0_j\rangle \langle 0_j| + |1_j\rangle \langle 1_j|) + e^{i\phi} \mathbb{I} \otimes (|3_j\rangle \langle 3_j| + |2_j\rangle \langle 2_j|) \quad (13)$$

$$P_{3,j} = \mathbb{I} \otimes (|0_j\rangle \langle 0_j| + |2_j\rangle \langle 2_j|) + e^{i\phi} \mathbb{I} \otimes (|3_j\rangle \langle 3_j| + |1_j\rangle \langle 1_j|) \quad (14)$$

For the special case when $Q = N$ is even, analogous to the Hadamard gate, phase gate can be given as,

$$\hat{P}_N = \hat{P}_{3,n}. \quad (15)$$

When $Q = 1$, the phase operation on the first qubit is simply a coin operation, $C = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{bmatrix}$ with an identity operation on the position space.

Controlled-NOT gate. Since, controlled-NOT gate (CNOT) is a two qubit gate (unlike Hadamard and phase gate), the gate implementation scheme changes form based on which two qubits are being addressed in the N -qubit equivalent system. The different cases which will cover all the possibilities of controlled-NOT gate between control qubit Q_c (here assumed to be on the i th level) and target qubit Q_t (assumed to be on the j th level) on N -qubit equivalent system (with n levels) are:

Case 1: $Q_c = 1$ or $Q_t = 1$. Case 1a: $Q_c = 1$, Q_t is even, and $j = n$,

$$CNOT_{1,N} = \left[S_{j,+}^1 (|0_j\rangle \langle 0_j|) + S_{j,-}^1 (|1_j\rangle \langle 1_j|) \right]. \quad (16)$$

Case 1b: $Q_t = 1$, Q_c is even, and $i = n$,

$$CNOT_{N,1} = \left[\mathbb{I} \otimes \mathbb{I}(|0_i\rangle \langle 0_i|) + \sigma_x \otimes \mathbb{I}(|1_i\rangle \langle 1_i|) \right]. \quad (17)$$

Case 1c: $Q_c = 1$, Q_t is even, and is on j th level, with $j \neq n$,

$$CNOT_{1,Q_t} = \left[S_{j,+}^1 (|1_j\rangle \langle 1_j| + |2_j\rangle \langle 2_j|) + S_{j,-}^1 (|0_j\rangle \langle 0_j| + |3_j\rangle \langle 3_j|) \right]. \quad (18)$$

Case 1d: $Q_c = 1$, Q_t is odd, and on the j th-level for $j \neq n$,

$$CNOT_{1,Q_t} = \left[S_{j,+}^1 (|0_j\rangle \langle 0_j| + |3_j\rangle \langle 3_j|) + S_{j,-}^1 (|1_j\rangle \langle 1_j| + |2_j\rangle \langle 2_j|) \right]. \quad (19)$$

Case 1e: $Q_t = 1$, for even Q_c such that Q_c is on the i th-level, and $i \neq n$,

$$\text{CNOT}_{Q_c,1} = \left[\mathbb{I} \otimes \mathbb{I}(|0_i\rangle\langle 0_i| + |1_i\rangle\langle 1_i|) + \sigma_x \otimes \mathbb{I}(|2_i\rangle\langle 2_i| + |3_i\rangle\langle 3_i|) \right]. \quad (20)$$

Case 1f: $Q_t = 1$, for odd Q_c such that Q_c is on the i th-level and $i \neq n$,

$$\text{CNOT}_{Q_c,1} = \left[\mathbb{I} \otimes \mathbb{I}(|0_i\rangle\langle 0_i| + |2_i\rangle\langle 2_i|) + \sigma_x \otimes \mathbb{I}(|1_i\rangle\langle 1_i| + |3_i\rangle\langle 3_i|) \right]. \quad (21)$$

Case 2: Q_c and Q_t are on the same level i.e., $i = j$. Case 2a: Q_c is odd and Q_t is even,

$$\text{CNOT}_{Q_c,Q_t} = \left[\mathbb{I} \otimes \mathbb{I}(|0_j\rangle\langle 0_j| + |1_j\rangle\langle 1_j|) + S_{j,+}^1 S_{j,+}^0 (|2_j\rangle\langle 2_j|) + S_{j,-}^1 S_{j,-}^0 (|3_j\rangle\langle 3_j|) \right]. \quad (22)$$

Case 2b: Q_c is even and Q_t is odd,

$$\text{CNOT}_{Q_c,Q_t} = \left[\mathbb{I} \otimes \mathbb{I}(|0_j\rangle\langle 0_j| + |2_j\rangle\langle 2_j|) + S_{j,+}^1 S_{j,+}^0 (|1_j\rangle\langle 1_j|) + S_{j,-}^1 S_{j,-}^0 (|3_j\rangle\langle 3_j|) \right]. \quad (23)$$

Case 3: $i \neq j$, where Q_c and Q_t are on i th and j th levels, respectively, and $Q_t \neq N$ if N is even. Case 3a: Q_c is odd and Q_t is odd,

$$\begin{aligned} \text{CNOT}_{Q_c,Q_t} = & \left[\mathbb{I} \otimes \mathbb{I}(|0_i\rangle\langle 0_i| + |2_i\rangle\langle 2_i|) + S_{j,+}^1 S_{j,+}^0 (|1_i\rangle\langle 1_i| + |3_i\rangle\langle 3_i|) (|0_j\rangle\langle 0_j| + |3_j\rangle\langle 3_j|) \right. \\ & \left. + S_{j,-}^1 S_{j,-}^0 (|1_i\rangle\langle 1_i| + |3_i\rangle\langle 3_i|) (|1_j\rangle\langle 1_j| + |2_j\rangle\langle 2_j|) \right]. \end{aligned} \quad (24)$$

Case 3b: Q_c is odd and Q_t is even,

$$\begin{aligned} \text{CNOT}_{Q_c,Q_t} = & \left[\mathbb{I} \otimes \mathbb{I}(|0_i\rangle\langle 0_i| + |1_i\rangle\langle 1_i|) + S_{j,+}^1 S_{j,+}^0 (|2_i\rangle\langle 2_i| + |3_i\rangle\langle 3_i|) (|0_j\rangle\langle 0_j| + |3_j\rangle\langle 3_j|) \right. \\ & \left. + S_{j,-}^1 S_{j,-}^0 (|2_i\rangle\langle 2_i| + |3_i\rangle\langle 3_i|) (|1_j\rangle\langle 1_j| + |2_j\rangle\langle 2_j|) \right]. \end{aligned} \quad (25)$$

Case 3c: Q_c and Q_t are both even,

$$\begin{aligned} \text{CNOT}_{Q_c,Q_t} = & \left[\mathbb{I} \otimes \mathbb{I}(|0_i\rangle\langle 0_i| + |1_i\rangle\langle 1_i|) + S_{j,+}^1 S_{j,+}^0 (|2_i\rangle\langle 2_i| + |3_i\rangle\langle 3_i|) (|1_j\rangle\langle 1_j| + |2_j\rangle\langle 2_j|) \right. \\ & \left. + S_{j,-}^1 S_{j,-}^0 (|2_i\rangle\langle 2_i| + |3_i\rangle\langle 3_i|) (|0_j\rangle\langle 0_j| + |3_j\rangle\langle 3_j|) \right]. \end{aligned} \quad (26)$$

Case 3d: Q_c is even and Q_t is odd,

$$\begin{aligned} \text{CNOT}_{Q_c,Q_t} = & \left[\mathbb{I} \otimes \mathbb{I}(|0_i\rangle\langle 0_i| + |2_i\rangle\langle 2_i|) + S_{j,+}^1 S_{j,+}^0 (|1_i\rangle\langle 1_i| + |3_i\rangle\langle 3_i|) (|1_j\rangle\langle 1_j| + |2_j\rangle\langle 2_j|) \right. \\ & \left. + S_{j,-}^1 S_{j,-}^0 (|1_i\rangle\langle 1_i| + |3_i\rangle\langle 3_i|) (|0_j\rangle\langle 0_j| + |3_j\rangle\langle 3_j|) \right]. \end{aligned} \quad (27)$$

Case 4: $i \neq j$, where Q_c and Q_t are on i th and j th level, respectively, and $Q_t = N$, for even N . Case 4a: Q_c is even,

$$\begin{aligned} \text{CNOT}_{Q_c,Q_t} = & \left[\mathbb{I} \otimes \mathbb{I}(|0_i\rangle\langle 0_i| + |1_i\rangle\langle 1_i|) + S_{j,+}^1 S_{j,+}^0 (|2_i\rangle\langle 2_i| + |3_i\rangle\langle 3_i|) (|0_j\rangle\langle 0_j|) \right. \\ & \left. + S_{j,-}^1 S_{j,-}^0 (|2_i\rangle\langle 2_i| + |3_i\rangle\langle 3_i|) (|1_j\rangle\langle 1_j|) \right]. \end{aligned} \quad (28)$$

Case 4b: Q_c is odd,

$$\begin{aligned} \text{CNOT}_{Q_c,Q_t} = & \left[\mathbb{I} \otimes \mathbb{I}(|0_i\rangle\langle 0_i| + |2_i\rangle\langle 2_i|) + S_{j,+}^1 S_{j,+}^0 (|1_i\rangle\langle 1_i| + |3_i\rangle\langle 3_i|) (|0_j\rangle\langle 0_j|) \right. \\ & \left. + S_{j,-}^1 S_{j,-}^0 (|1_i\rangle\langle 1_i| + |3_i\rangle\langle 3_i|) (|1_j\rangle\langle 1_j|) \right]. \end{aligned} \quad (29)$$

based on the two qubit on which CNOT gate is applied, different cases from above can be selected. A different scheme of implementing the universal set of quantum gates on the same quantum walk scaling model is shown in Supplementary information Sec. S1. This shows that on this model of quantum walk, we can have different forms of the evolution operators to achieve desired operation based on the suitability of the available quantum processors. This above scheme can be very easily implemented on photonic system with different sets of four-sided closed graph.

Grover's search algorithm on three qubit equivalent quantum walk scheme

For searching a target state $|x\rangle$, Grover's search algorithm uses an oracle \hat{C} on state $|\Psi\rangle = \sum_x \psi_x |x\rangle$ of the form,

$$\widehat{O}|\Psi\rangle \rightarrow \begin{cases} -|x\rangle, & \text{when } x \text{ is the target element} \\ |x\rangle, & \text{else} \end{cases} \tag{30}$$

Grover’s algorithm requires an oracle for the task of marking the targeted state by applying a negative sign to the desired search result state.

The possible states of three-qubit system are $|000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle, |111\rangle$. On three qubit equivalent quantum walk scheme, we need one real qubit on square lattice (closed graph of four sites). Oracle can be implemented by applying a position dependent evolution operator. The operator involves the coin operation,

$$\begin{aligned} \hat{N}_1 &= \sigma_z \otimes \mathbb{I} \\ \hat{N}_0 &= \hat{C}(0, 0, \pi) \otimes \mathbb{I} \\ \hat{C}(\xi, \zeta, \theta) &= \begin{bmatrix} e^{i\xi} \cos(\theta) & e^{i\zeta} \sin(\theta) \\ e^{-i\zeta} \sin(\theta) & -e^{-i\xi} \cos(\theta) \end{bmatrix}. \end{aligned} \tag{31}$$

and the form of oracle on quantum walk scheme is shown in Fig. 2.

Quantum walk scheme for three qubit Grover’s search algorithm, when the coin and position state is initialized to $|0\rangle_c \otimes |x = 0\rangle$, involves following steps,

1. A quantum walker starts with an equal superposition of all the states of the form $|\psi_c\rangle \otimes |x\rangle$ in both coin and position space. It can be achieved by applying operation $\hat{H}_2\hat{H}_3$ on position state according to the quantum walk scheme as given in Ref.³⁷ and then Hadamard operation on coin state.
2. The oracle is applied on the walker according to Fig. 2 to search for the desired marked state.
3. Hadamard operation is again applied on the coin state followed by the operation $\hat{H}_3\hat{H}_2$ on position state according to quantum walk scheme.
4. The iteration method can be applied on the walker using position dependent \hat{N} operators as defined in Eq. (31) and illustrated in Fig. 3 which will perform a conditional phase shift on every state except $|000\rangle$.

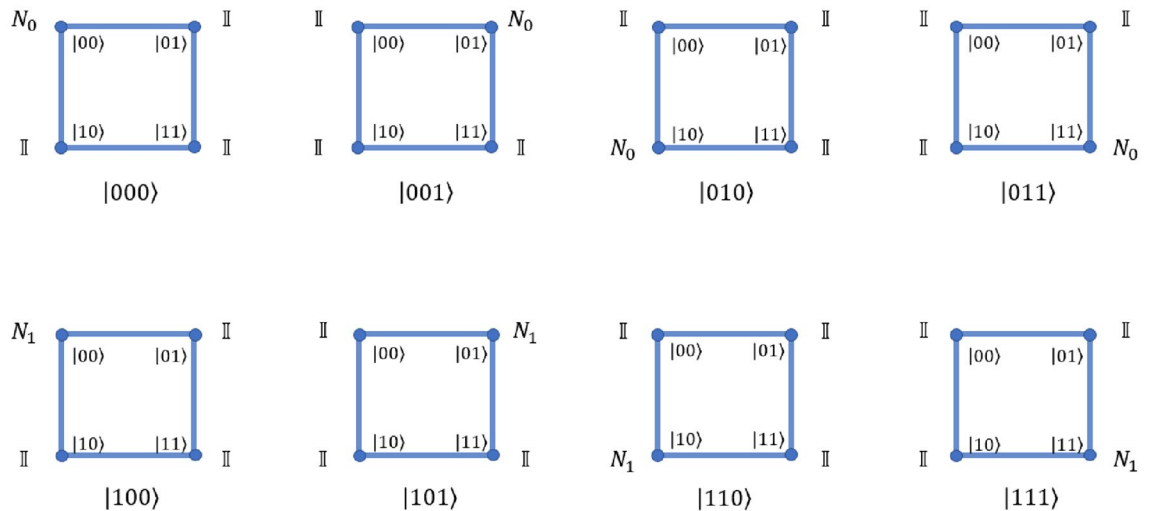


Figure 2. A schematic illustration of the oracle operation on the position state of the three-qubit equivalent quantum walk system using position dependent operators. The states below each square correspond to the target states of Grover’s search. The definition of the various N operators have been defined in Eq. (31).

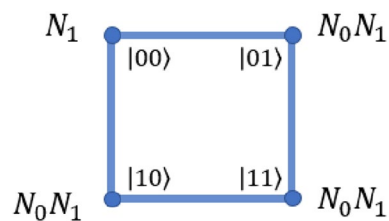


Figure 3. A schematic illustration of the iteration operation on the position basis of the three qubit system using position dependent operators. All the states except $|000\rangle$ will get a negative sign in this one step operation. The definition of the various N operators have been defined in Eq. (31).

- Again apply Hadamard operation on the coin state followed by the operation $\hat{H}_3\hat{H}_2$ on position state according to quantum walk scheme.
- Repeating steps 2 and 5 (also called the Grover iteration) for less or equal to $\left\lceil \frac{\pi}{4} \sqrt{\frac{U}{V}} \right\rceil$ times where, V = number of target entries in the search space and $U = 2^N$. For $N = 3$ and $V = 1$, $\left\lceil \frac{\pi}{4} \sqrt{\frac{U}{V}} \right\rceil \leq 3 = 2$.
- Measurement in coin and position basis will give us our target state.

Section S1 of the supplementary file verifies the quantum walk based search algorithm by taking an example on search space of three qubit.

Quantum Fourier transformation on three-qubit equivalent quantum walk scheme

The quantum Fourier transform is defined on orthonormal basis $|0\rangle, |1\rangle \dots |X - 1\rangle$ as a linear operator of the form,

$$|\alpha\rangle = \frac{1}{\sqrt{X}} \sum_{l=0}^{X-1} e^{2\pi i \alpha l / X} |l\rangle \tag{32}$$

It can be transformed into a more easily implementable format as,

$$\begin{aligned} |\alpha\rangle &\rightarrow \frac{1}{\sqrt{X}} \sum_{l=0}^{X-1} e^{2\pi i \alpha l / X} |l\rangle \\ &\rightarrow \frac{1}{\sqrt{X}} (|10\rangle + e^{2\pi i 0, \alpha N} |1\rangle)(|10\rangle + e^{2\pi i 0, \alpha N - 1} |1\rangle) \dots (|10\rangle + e^{2\pi i 0, \alpha_1 \dots \alpha_{N-1} \alpha_N} |1\rangle) \\ & [e^{2\pi i \alpha_1 \dots \alpha_{N-1} \alpha_N} = e^{2\pi i 0, \alpha_N}] \end{aligned} \tag{33}$$

where $X = 2^N$ and N is the number of qubits in the system. Quantum Fourier transformation on three-qubit quantum walk scheme requires a controlled-SWAP operation which, on quantum walk scheme can be obtained by applying the following operations,

$$\begin{aligned} \hat{A}_+^0 |k, m\rangle &= \hat{\sigma}_x^{m+1} \hat{S}_{1,+}^0 |k, m\rangle \\ \hat{A}_+^1 |k, m\rangle &= \hat{\sigma}_x^{m+1} \hat{S}_{1,+}^1 |k, m\rangle \\ \hat{A}_-^0 |k, m\rangle &= \hat{\sigma}_x^{m-1} \hat{S}_{1,-}^0 |k, m\rangle \\ \hat{A}_-^1 |k, m\rangle &= \hat{\sigma}_x^{m-1} \hat{S}_{1,-}^1 |k, m\rangle \\ \hat{T}_+ |k, m\rangle &= \hat{S}_{1,+}^1 \hat{S}_{1,+}^0 \hat{S}_{1,+}^1 |k, m\rangle; \end{aligned} \tag{34}$$

where $\hat{S}_{1,\pm}^k$ are conditional shift operators in the position space of the walker and are given by Eq. (1) on the position state $|m\rangle$ conditioned on the state of coin $|k\rangle$ and $\hat{\sigma}_x^m$ is given by,

$$\hat{\sigma}_x^m = \hat{\sigma}_x \otimes (|m\rangle\langle m|)_p + \mathbb{I} \otimes \sum_{j \neq m} (|j\rangle\langle j|) \tag{35}$$

Equations (35) and (34) and Fig. 4 outlines the operations which swaps two qubits.

Thus, quantum Fourier transformation on quantum walk scheme can be given by the operation as shown in Fig. 5, after producing the initial state, where,

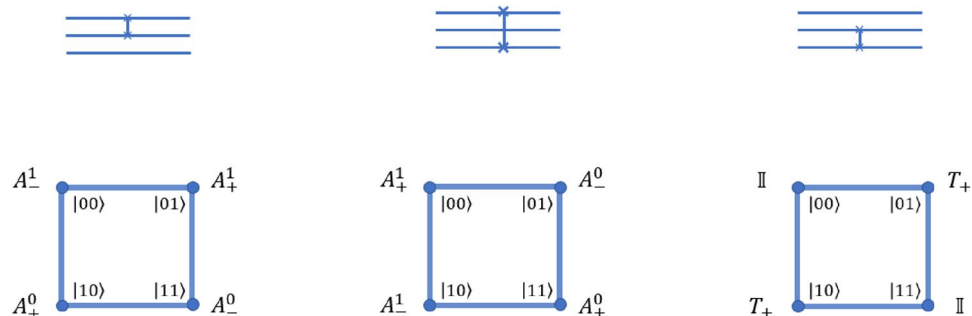


Figure 4. A schematic illustration of the controlled swap gate operation on the position basis of the three qubit equivalent quantum walk system using position dependent operators. The definition of the various A and T operators have been defined in Eq. (34).

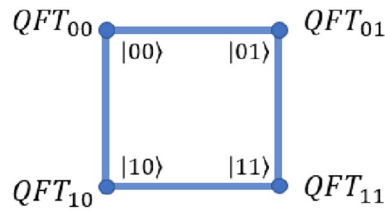


Figure 5. A schematic illustration of quantum Fourier transformation on three-qubit equivalent quantum walk scheme using position dependent operators.

$$\begin{aligned}
 QFT_{00} &= \hat{A}_+^1 \hat{H}_3 \hat{H}_2 \hat{H}_1 \\
 QFT_{01} &= \hat{A}_+^0 \hat{H}_3 \hat{H}_2 \hat{P}(\pi/4) \hat{H}_1 \\
 QFT_{11} &= \hat{A}_+^0 \hat{H}_3 \hat{\Phi}(\pi/2) \hat{H}_2 \hat{P}(\pi/4) \hat{P}(\pi/2) \hat{H}_1 \\
 QFT_{10} &= \hat{A}_+^1 \hat{H}_3 \hat{H}_2 \hat{\Phi}(\pi/2) \hat{H}_1
 \end{aligned}
 \tag{36}$$

and operator \hat{H}_2, \hat{H}_3 and $\hat{P}(\phi), \hat{\Phi}(\phi)$ on the quantum walk scheme is given in the Ref.³⁷. $\hat{A}_{+,-}^0, \hat{A}_{+,-}^1$ are given by Eq. (34) and \hat{H}_1 is Hadamard operation on coin operation.

Phase estimation algorithm on three qubit equivalent quantum walk scheme

To estimate the phase φ induced by an operator \hat{U} on one of its eigenvectors $|\psi\rangle$ using single qubit on three-qubit equivalent quantum walk system, we consider the eigenvector $|\psi\rangle$ as the coin state and the position Hilbert space represents the state of the control qubits. The quantum circuit for phase estimation on three-qubit system is given in Fig. 6.

Algorithm for phase estimation on quantum walk scheme according to quantum circuit as given in Fig. 6, when coin and position state is initialised to state $|0\rangle_c \otimes |x = 0\rangle$ is,

1. Bringing the position states in equal superposition by implementing Hadamard operation H_2 and H_3 on second and third qubit, respectively. The state after this operation will have form,

$$\begin{aligned}
 |\phi_1\rangle &= |0\rangle_c \otimes \frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{2} \\
 &= |0\rangle_c \otimes \frac{|x = 0\rangle + |x = 1\rangle + |x = 3\rangle + |x = 2\rangle}{2}
 \end{aligned}
 \tag{37}$$

2. Bringing the coin state to $|\psi\rangle_c$ using unitary operation G such that $|\psi\rangle_c = G|0\rangle_c$. Here, $|\psi\rangle_c$ is an eigenvector of the unitary operator U with eigenvalue $e^{2\pi i\varphi}$, where the value of φ is unknown. The state after this operation will have form,

$$|\phi_2\rangle = |\psi\rangle_c \otimes \frac{|0\rangle + |1\rangle + |2\rangle + |3\rangle}{2}
 \tag{38}$$

3. The effect of the controlled \hat{U} -operations can be thought of as different powers of \hat{U} being operated on each of the position states as position-dependent coin operation given by,

$$\begin{aligned}
 \hat{C}'_U &= \mathbb{I}_C \otimes |0\rangle\langle 0| + \hat{U} \otimes |1\rangle\langle 1| \\
 &\quad + \hat{U}^2 \otimes |3\rangle\langle 3| + U^3 \otimes |2\rangle\langle 2|.
 \end{aligned}
 \tag{39}$$

The form of the state after this operation is

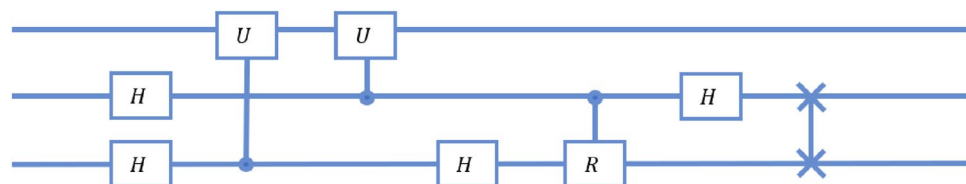


Figure 6. Schematic of quantum circuit for phase estimation procedure on three qubit system. The state of the first qubit of the system is equivalent to the coin state and last two qubit shows the equivalence to the position states of the quantum walk scheme.

$$\begin{aligned}
|\phi_3\rangle &= \hat{C}'_U |\phi_2\rangle \\
&= \frac{|\psi\rangle|0\rangle + \hat{U}|\psi\rangle|1\rangle + \hat{U}^2|\psi\rangle|3\rangle + \hat{U}^3|\psi\rangle|2\rangle}{2} \\
&= |\psi\rangle \otimes \frac{|0\rangle + e^{i\varphi}|1\rangle + e^{2i\varphi}|3\rangle + e^{3i\varphi}|2\rangle}{2}
\end{aligned} \tag{40}$$

4. Then applying inverse quantum Fourier transformation in the standard basis such that final state is,

$$\begin{aligned}
|\phi_f\rangle &= QFT^{-1}|\phi_3\rangle \\
&= |\psi\rangle|\tilde{\varphi}\rangle
\end{aligned} \tag{41}$$

The position dependent evolution operator for inverse Fourier transformation on state $|\phi_3\rangle$ in quantum walk scheme is given as,

$$QFT^{-1} = (G \otimes \mathbb{I}) V_2^x V_1^x (G^\dagger \otimes \mathbb{I}), \tag{42}$$

where G is the operator given in step-2 of the algorithm and the form of V_1^x and V_2^x position dependent operator is given as,

$$\begin{aligned}
V_1^{x=0} &= \hat{S}_1^+(\hat{H} \otimes \mathbb{I}) \\
V_1^{x=1} &= \hat{S}_1^-(\hat{H} \otimes \mathbb{I}) \\
V_1^{x=3} &= \hat{S}_1^-(\hat{H} \otimes \mathbb{I}) \\
V_1^{x=2} &= \hat{S}_1^+(\hat{H} \otimes \mathbb{I})
\end{aligned} \tag{43}$$

and

$$\begin{aligned}
V_2^{x=0} &= \hat{S}_1^-(\hat{H} \otimes \mathbb{I}) \\
V_2^{x=1} &= \hat{S}_1^+(\hat{H} \otimes \mathbb{I})(\hat{\Phi}_{-\frac{\pi}{2}} \hat{\sigma}_x \otimes \mathbb{I}) \\
V_2^{x=3} &= (\hat{\sigma}_x \otimes \mathbb{I}) \hat{S}_0^+(\hat{H} \otimes \mathbb{I}) \\
V_2^{x=2} &= (\hat{\sigma}_x \otimes \mathbb{I}) \hat{S}_0^-(\hat{H} \otimes \mathbb{I})(\hat{\Phi}_{\frac{\pi}{2}} \hat{\sigma}_x \hat{\sigma}_z \otimes \mathbb{I}).
\end{aligned} \tag{44}$$

Using this scheme on quantum walk, phase φ induced by an operator \hat{U} on one of its eigenvectors $|\psi\rangle_c$ can be estimated up to a certain accuracy. The accuracy in the estimation can be increased by using large position Hilbert space.

Quantum space and time complexity

An analysis of complexity is mainly concerned with the inherent cost of solving a problem, where the cost is measured in terms of some well-defined resources. In this section, we shall be considering two ways of expressing complexity, namely *quantum space complexity* and *quantum time complexity*. We define these terms as follows.

1. **Quantum space complexity** is defined as the number of real qubits required to implement the circuit. This is analogous to the classical space complexity.
2. **Quantum time complexity** is defined as the smallest number of time steps required to perform a computation on the circuit. In other words, it describes the least number of simultaneous elementary operations required to perform a single computation on the circuit. This is also in direct analogy to classical time complexity.

Note that these definitions are expressed keeping in mind the specific discrete-time quantum walk-based scheme presented in this manuscript. In the circuit model of quantum computation, they are equivalent to the notions of circuit width and depth, respectively. Complex multi-qubit gates can be composed of elementary gates from the universal gate set, and we define the complexity of implementing a (complex) multi-qubit gate as the quantum time-complexity of the equivalent circuit constructed with elementary operations.

In case of a standard circuit model, an elementary operation can be a single-qubit Hadamard gate, a single-qubit phase gate, or a two-qubit CNOT operation. Every other gate may be composed of these gates as they form a universal set³⁵.

In case of our model based on the quantum walk, an elementary operation is defined as a walk operation, i.e. a coin operation, followed by a shift operation. In case multiple quantum walk operations can be done with a common step, then the time complexity reduces.

As an example, consider the sequence of steps $\hat{\Phi}(\frac{\pi}{2})\hat{P}(\frac{\pi}{4})\hat{P}(\frac{\pi}{2})$, as used in the definition of QFT_{11} . With the way that Φ and P gates are described in Ref.³⁷, all the gates can effectively be implemented by a coin operation, and can thus be combined into a single P operation with a global phase. Thus, the time complexity of this 3 gate sequence is actually 1 time step.

Compared to the earlier universal quantum computation scheme with quantum walks³⁶, our scheme defines computation purely as a sequence of walks that achieve the same effect as certain gates, instead of actually simulating gates from quantum walk steps, and then creating mirroring the circuit model. The existing models thus impose significant resource requirements to achieve the implementations of algorithms, thereby becoming prohibitively resource-intensive.

We now detail an analysis of circuits for implementation of quantum algorithms considered in this paper, both in terms of the standard circuit model and our proposed quantum walk model of computation.

Grover’s search. In this work, have considered Grover’s search algorithm for 3 qubits, and have searched for the state $|011\rangle$ as an example.

Quantum space complexity The proposed quantum walk model of computation requires 3 qubits for implementation of the walk, however, only one qubit is a real (particle) qubit. The other two qubits are implemented with the position space. Thus, the quantum space complexity is 1. In case of the standard circuit model (Fig. 7), the implementation requires 3 qubits for the algorithm, and 1 ancilla qubit, thus making the total quantum space complexity to be 4.

Quantum time complexity In our quantum walk model, the generation of the initial superposition (done by the operator H_2H_3) takes 6 time steps. The oracle operation requires 1 time step, each ensuing Hadamard operation requires 3 time steps, and the final iteration operator needs another 2 time steps. Since 2 Grover iterations are required for a 3-qubit implementation, the total quantum time complexity becomes 39.

In the standard circuit, the superposition requires 4 parallel single-qubit gates on all 4 qubits and can be achieved in 1 time step. The various gates required to implement the algorithm on a 3-qubit system are the 4-qubit *CCCNOT*, which requires the Toffoli (*CCNOT*) gate implementation, the single qubit *X* gate, and the *CCZ* gate. The various gates and their quantum time complexities are shown in Figs. 8, 9, 10 and 11. Accounting for everything, the complete implementation has a quantum time complexity of 72.

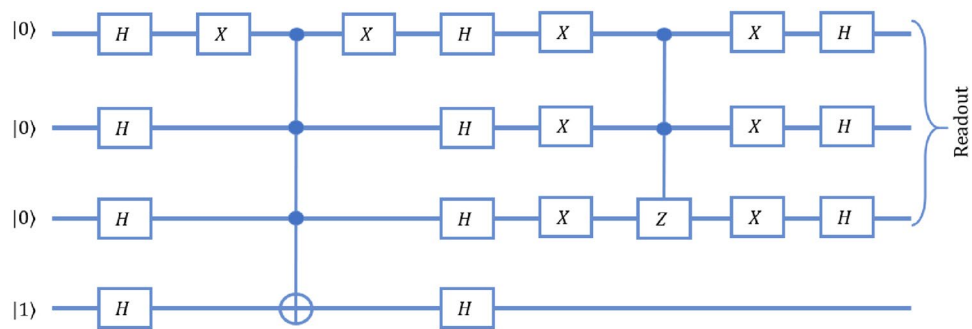


Figure 7. Schematic of quantum circuit for implementation of Grover’s search algorithm on a three qubit system. The oracle is designed here to search for the state $|011\rangle$.

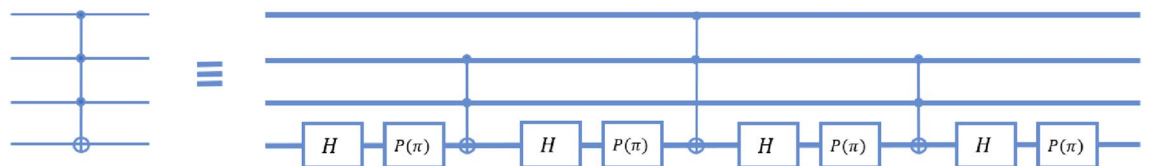


Figure 8. Schematic of quantum circuit for implementation of the *CCCNOT* gate on a 4-qubit system. This gate has a quantum time complexity of 45.

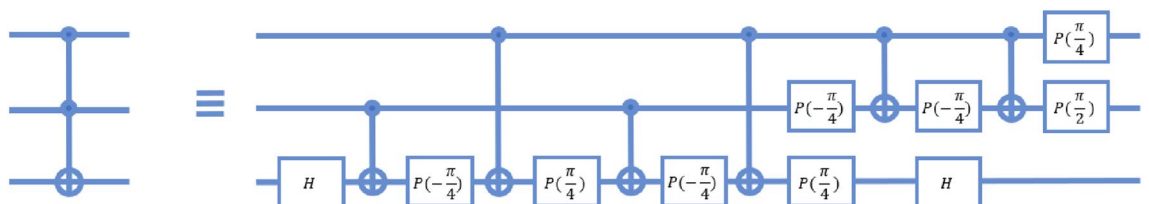


Figure 9. Schematic of quantum circuit for implementation of the Toffoli gate on a three qubit system. The quantum time complexity of this implementation is 13.



Figure 10. Schematic of quantum circuit for implementation of the Pauli X gate on a single qubit. This gate has a quantum time complexity of 3.

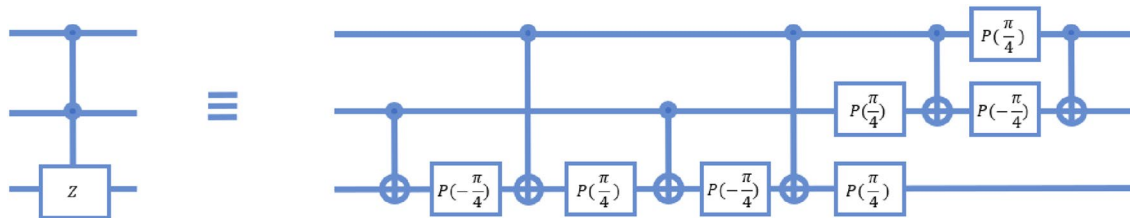


Figure 11. Schematic of quantum circuit for implementation of the CCZ gate on a three qubit system. This implementation is similar to the 3-qubit Toffoli gate, except it has a few less operations. The quantum time complexity of this gate is thus 11.

Quantum Fourier transform. We have considered the problem of computing the quantum Fourier transform for a 3-qubit system.

Quantum space complexity In our circuit, we require 1 real qubit to achieve a 3-qubit quantum Fourier transform. The standard circuit model requires 3 real qubits.

Quantum time complexity In our quantum walk-based model, the operations A_{\pm}^i are essentially a single step of the walk, and can be implemented in one time step. The A_{\pm}^i operation is then followed by the sequence $H_1H_3H_2$, which requires 7 time steps to implement. The maximum time is required by QFT_{01} and QFT_{11} operators each of which require 9 time steps. This is due to the fact that the position-dependent Phase operations may be applied simultaneously, as they are all simply coin operations. Thus, the quantum-walk based model can implement this algorithm in 9 time steps.

In the standard circuit, the QFT is implemented as shown in Fig. 12. The circuit begins with a Hadamard gate, followed by two controlled phase gates on the first qubit. The implementation of a controlled phase gate is shown in Fig. 13, it may be seen that a single controlled phase gate requires 5 time steps to implement. The final gate we require to implement is a two-qubit swap gate, which can be implemented efficiently as a series of 3 two-qubit CNOT gates, which requires 3 time steps to implement. The circuit is shown in Fig. 14. As a result, the standard circuit will require a total of 21 steps to implement.

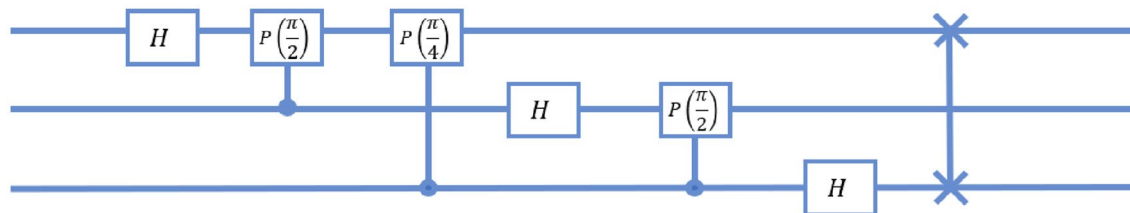


Figure 12. Schematic for the quantum circuit model implementation of the quantum Fourier transform on a three qubit system. The quantum time complexity of this implementation is 21.

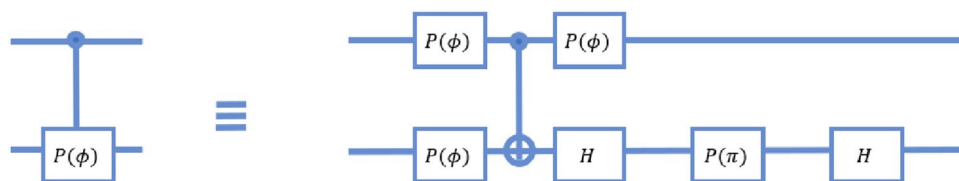


Figure 13. Schematic for the quantum circuit model implementation of the controlled phase gate on two qubits. The quantum time complexity of this gate is 5.

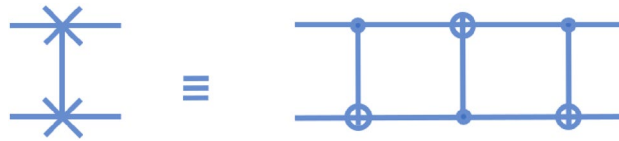


Figure 14. Schematic for the quantum circuit model implementation of the swap gate on two qubits. The quantum time complexity of this gate is 3.

Phase estimation algorithm. We apply the phase estimation algorithm to an unknown unitary operation U .

Quantum space complexity In our circuit, we require only 1 real qubit in order to implement phase estimation. In a standard circuit, we need 3 real qubits to implement this algorithm.

Quantum time complexity In our circuit, the initial superposition required can be made in 6 time steps by the application of the operator H_2H_3 . 1 time step is then required to implement the operator G , required to bring the coin into the correct state. It is sure that this will require only 1 time step as the coin qubit can be affected by a coin operator and an identity shift operator on the system. The controlled- U operations are then realised as position-dependent operations, which require 3 time steps to implement (assuming U will require 1 step to implement). The inverse Fourier transform on a 2-qubit system requires a worst case time of 7 steps. The complete quantum time complexity of this circuit thus becomes 17.

In a standard circuit as shown in Fig. 6, the two initial Hadamard gates require one time step to implement, as they can be implemented in parallel. Going by the reduction for a controlled- U gate, as shown in Fig. 15, the controlled- U and controlled- U^2 gates would require 5 time steps each. The remaining circuit for an inverse QFT on two qubits requires 1 time step each for the Hadamard gates, 5 time steps for the controlled Phase, and 3 time steps for the swap gate. In total, the circuit requires 21 time steps to be implemented.

By this analysis, proposed quantum walk scheme uses a lesser number of real qubits to implement algorithmic operations than the circuit model. It also requires a lesser number of time steps than the circuit model in order to implement the algorithms shown here.

Single-qubit error detection

The proposed model of quantum computation also lends itself to an elementary representation of a quantum encoding. In this section, we present two examples of [3, 1] codes, and an example of a [5, 1] code. The [3, 1] code is able to detect either one of single-qubit errors, namely, bit-flip and phase-flip errors, and the [5, 1] code saturates the quantum Hamming bound, and is thus able to protect against arbitrary single-qubit errors.

Bit-flip code. The [3, 1] bit-flip encoding and decoding in the circuit model of computation is realised as shown in Fig. 16. The encoding uses 2 auxiliary qubits to generate error syndromes which can be corrected by the decoding circuit, which is shown post the introduction of error. The decoding of the syndrome and correc-

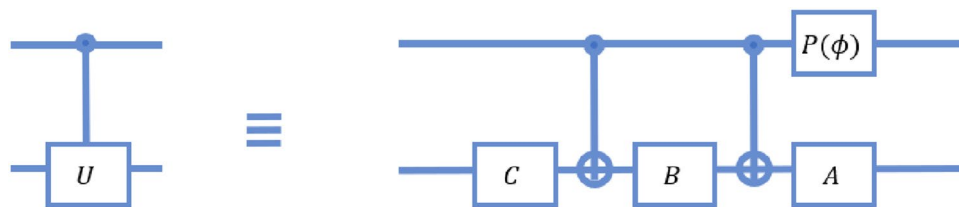


Figure 15. The circuit model of a controlled- U gate, where U is an unknown unitary, as given in Ref.⁴². Here $P(\phi)$ represents the phase gate as described in³⁷, and A, B, C, ϕ satisfy $e^{i\phi}AXBXC = U$, and $ABC = \mathbb{1}$. X is the Pauli- X operation.

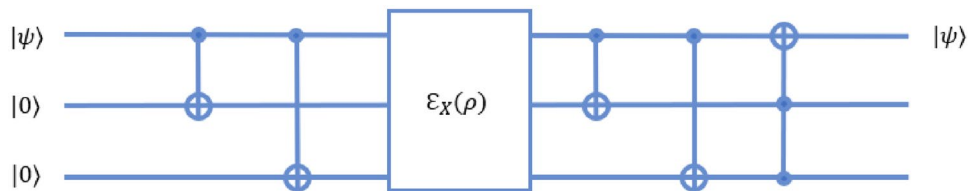


Figure 16. A circuit-model representation of the bit-flip code, implemented on a 3-qubit system. The figure is based on from the code as described in⁴².

tion of error in a single qubit case requires the implementation of a Toffoli gate as shown. The Toffoli gate may be implemented with the gates belonging to the universal set as shown in Fig. 9.

The equivalent operation on a 3-qubit quantum walk system as detailed in³⁷ may be performed by the operations $CNOT_{1,2}$ and $CNOT_{1,3}$ applied to the system. The final correction step is implemented with the Toffoli gate as demonstrated in Fig. 17.

Phase-flip code. The phase flip encoding is also a [3, 1] code, and is able to detect and correct single-qubit phase flip errors. The circuit representation for encoding and decoding in the phase flip code is shown in Fig. 18. The circuit for the phase flip code is similar to that used for the bit flip encoding, except that it requires an extra Hadamard operation on each qubit after the bit-flip encoding. On a 3-qubit equivalent quantum walk system, this corresponds to applying the operations H_1 , H_2 , and H_3 on the system after applying the bit-flip encoding.

Error correcting code. Figure 19 shows a circuit model implementation of a [5, 1] encoding, a more elaborate description of which was given by Laflamme et al.⁴³. The code enables error correction, and is able to correct against general single-qubit errors. This encoding may be implemented on a 2-level (5-qubit equivalent) graph in a quantum walk system, with one level consisting of a two-site closed graph and the second level being a four-site closed graph. This setup would require 2 real qubits to implement this code, however, in order to reduce the space complexity, it is possible to use a pair of 4-site closed graphs with a single particle executing a discrete time walk on them.

The quantum walk implementation would also require the implementation of the twin $CNOT$ gate, the controlled-controlled-Z (CCZ) gate, and the $CCCZ$ gate with two of the inputs inverted. The circuit model implementation of these gates is shown in Figs. 11 and 20. The sequence of steps required to achieve the CCZ gate on a quantum walk system is the same as illustrated in³⁷. The $CCCZ$ implementation will vary depending on the system topology chosen, however, it is illustrated here considering a 2-level implementation, where the

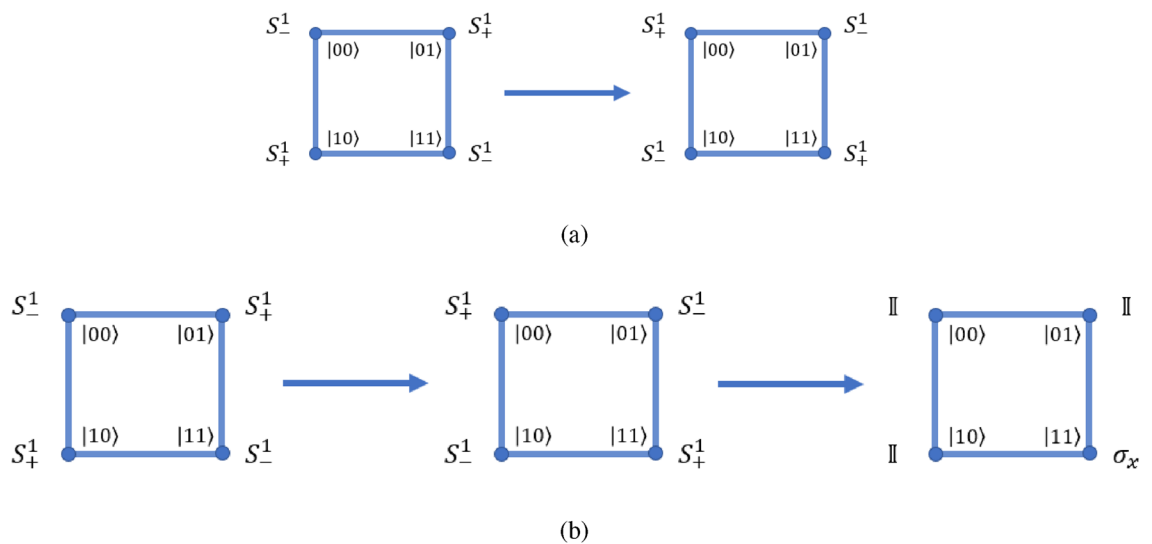


Figure 17. A possible realization of the bit-flip encoding in the quantum walk paradigm. The figure (a) describes the steps in encoding, and (b) describes the decoding steps. The quantum time complexity of the complete encoding and decoding scheme is 5 (2 for encoding and 3 for decoding). In the circuit formalism, the quantum time complexity becomes 15 (1 for encoding, 14 for decoding). The reason for this disparity is that the quantum walk formalism allows for a simple realization of the Toffoli gate.

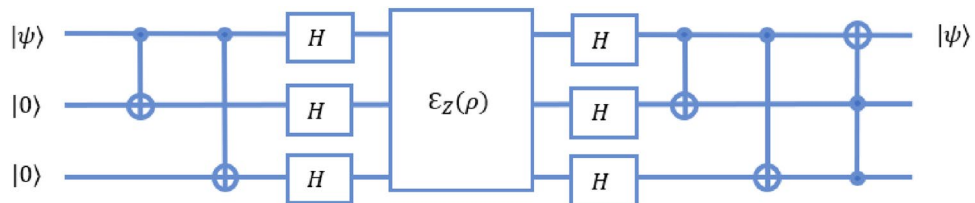


Figure 18. A circuit-model representation of the phase-flip code, implemented on a 3-qubit system. The figure is based on from the code as described in⁴². This circuit is very similar to the bit-flip code, except that it requires an extra Hadamard operation on each qubit during both the detection and correction steps.

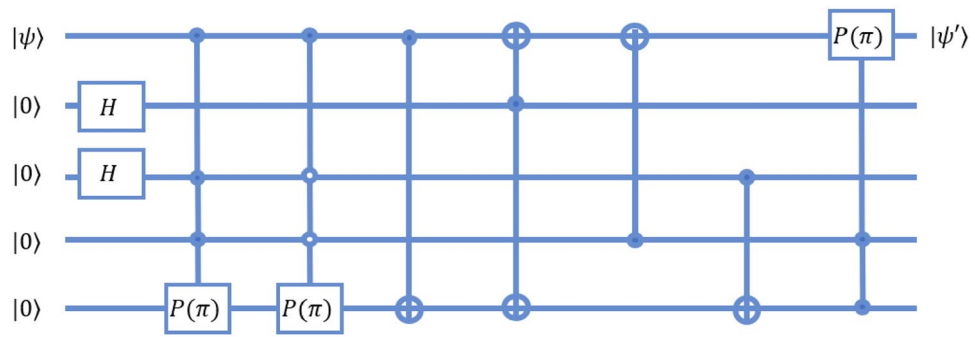


Figure 19. The quantum circuit for computing syndromes according to the [5, 1] code described in⁴³. The circuit for recovery of the original qubit $|\phi\rangle$ is exactly the reverse of this circuit. The empty circle for the control qubit implies that the gate is activated if the qubit is in the $|0\rangle$ state. Gates are applied from the left column to the right column. Gates in a single column may be applied simultaneously.

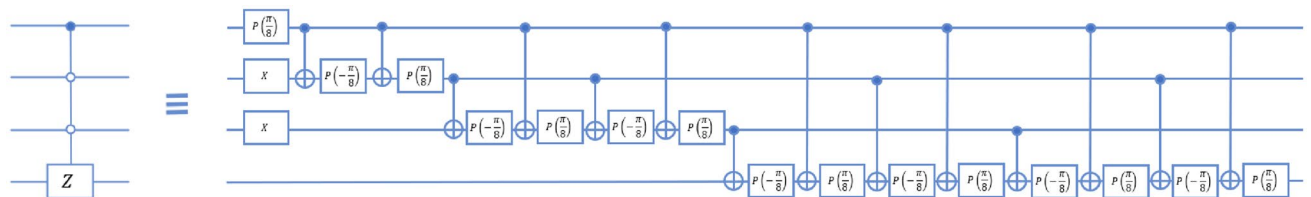


Figure 20. A circuit-model realization of the modified CCCZ-gate as required for implementing the [5, 1] quantum error-correcting code. In order to realize the CCCZ-gate which activates as usual, i.e. when all inputs are $|1\rangle$, one may substitute the single qubit pauli X rotations executed at the first time step with identity operations in this figure. The method to create this realization has been described in⁴⁴.

both levels are 4-site graphs, traversed by a single particle. The forms of the operators required are illustrated in detail in “[Implementing hadamard, phase, and controlled-NOT gates on N-qubit equivalent system](#)”. The twin CNOT gate may be designed in both the models in the same way, namely, by applying two CNOT operations simultaneously.

The CCCZ-gate requires a modified form of the controlled phase operation, which is given by the operator $\tilde{P}_{3,j}$, which will cause a conditional phase to be applied in case the control qubit is in the state $|0\rangle$. This is described in Fig. 21. The form of the complete operation is given by Eq. (45).

$$CCCZ_{Q,\bar{b},\bar{c},d} = \mathbb{1}_Q \otimes (|00\rangle_{j=1} + |10\rangle_{j=1}) \otimes \tilde{P}_{3,j=2} + \mathbb{1}_Q \otimes (|01\rangle_{j=1} + |11\rangle_{j=1}) \otimes \mathbb{1}_{j=2}, \quad (45)$$

where j denotes the level at which the operation is applied, Q is the first qubit in the system (assuming the qubit to be encoded is mapped to the real qubit), and the operation $\tilde{P}_{3,j=2}$ is as illustrated in Fig. 21.

The error correcting codes considered here are based on models of decoherence, as the error on a physical qubit can be modeled as a decoherence-inducing process. Since the proposed scheme uses a single physical system in configuration of position space, the whole computation is a bit less susceptible to errors due to reduced interactions between multiple components of physical systems.

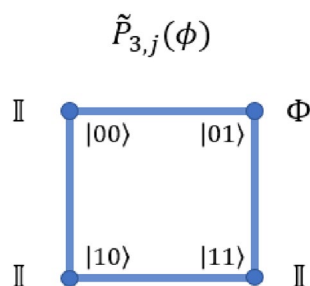


Figure 21. The modified form of the controlled-phase operation in the quantum walk regime. This operator applies the phase when the control qubit is $|0\rangle$. The operation Φ adds a global phase, and is defined as described in³⁷.

Conclusion

In this paper, we have presented a more generalized form of the quantum computation using single particle quantum walk³⁷, and have shown the scaling of the scheme. Our proposed model can be scaled to system of a higher number of qubits by considering different sets of three-qubit equivalent closed graph as position space. To implement quantum universal gates on larger qubit equivalent system, the coin operation will control the evolution of the walker's position space by changing the probability amplitude of the targeted closed set. Using appropriate conditional position dependent evolution operators, the quantum walk based quantum computing scheme can be easily implemented. We have also shown that on this scheme on an N -qubit system, universal gate implementation technique is not unique but can be changed according to the available resources. Since quantum walks on closed graph have been experimentally implemented on photonic system before^{45,46}, with the help of available photonic quantum processors, universal gates model based on single particle quantum walk can be implemented.

We have also presented the scheme for implementing quantum algorithms such as Grover's search, quantum Fourier transform and quantum phase estimation on this scheme for three-qubit equivalent system. A comparison of circuit complexity and circuit depth shows that the proposed quantum walk scheme reduces the complexity when compared to circuit model in all of the cases. However, with a careful designing of position dependent evolution operators one can engineer the implementation of various quantum computational tasks.

Data availability

All data generated or analyzed during this study are included in this article itself.

Received: 19 January 2023; Accepted: 19 July 2023

Published online: 26 July 2023

References

- Kaminsky, W. M., Lloyd, S. & Orlando, T. P. Scalable superconducting architecture for adiabatic quantum computation (2004). [arXiv:quant-ph/0403090](https://arxiv.org/abs/quant-ph/0403090).
- Clarke, J. & Wilhelm, F. K. Superconducting quantum bits. *Nature* **453**, 1031–1042. <https://doi.org/10.1038/nature07128> (2008).
- Gong, M. *et al.* Quantum walks on a programmable two-dimensional 62-qubit superconducting processor. *Science* **372**, 948–952. <https://doi.org/10.1126/science.abg7812> (2021).
- Singh, S., Adhikari, B., Dutta, S. & Zueco, D. Perfect state transfer on hypercubes and its implementation using superconducting qubits. *Phys. Rev. A* **102**, 062609. <https://doi.org/10.1103/PhysRevA.102.062609> (2020).
- Jones, J. NMR quantum computation: A critical evaluation. *Fortschr. Phys.* **48**, 909–924. [https://doi.org/10.1002/1521-3978\(200009\)48:9/11<909::AID-PROP909>3.0.CO;2-2](https://doi.org/10.1002/1521-3978(200009)48:9/11<909::AID-PROP909>3.0.CO;2-2) (2000).
- Prawer, S. & Greentree, A. D. Diamond for quantum computing. *Science* **320**, 1601–1602. <https://doi.org/10.1126/science.1158340> (2008).
- Ryan, C. A., Laforest, M. & Laflamme, R. Randomized benchmarking of single- and multi-qubit control in liquid-state NMR quantum information processing. *New J. Phys.* **11**, 013034. <https://doi.org/10.1088/1367-2630/11/1/013034> (2009).
- Childress, L. & Hanson, R. Diamond NV centers for quantum computing and quantum networks. *MRS Bull.* **38**, 134–138. <https://doi.org/10.1557/mrs.2013.20> (2013).
- Singh, G., Dorai, K. & Arvind. Experimental quantum state transfer of an arbitrary single-qubit state on a cycle with four vertices using a coined quantum random walk (2023). [arXiv:2305.02106](https://arxiv.org/abs/2305.02106).
- Paul, W. Electromagnetic traps for charged and neutral particles. *Rev. Mod. Phys.* **62**, 531–540. <https://doi.org/10.1103/RevModPhys.62.531> (1990).
- Blinov, B. B., Leibfried, D., Monroe, C. & Wineland, D. J. Quantum computing with trapped ion hyperfine qubits. *Quantum Inf. Process.* **3**, 45–59. <https://doi.org/10.1007/s11128-004-9417-3> (2004).
- Jaksch, D. Optical lattices, ultracold atoms and quantum information processing. *Contemp. Phys.* **45**, 367–381. <https://doi.org/10.1080/00107510410001705486> (2004).
- Gross, C. & Bloch, I. Quantum simulations with ultracold atoms in optical lattices. *Science* **357**, 995–1001. <https://doi.org/10.1126/science.aal3837> (2017).
- Lloyd, S. & Braunstein, S. L. Quantum computation over continuous variables. *Phys. Rev. Lett.* **82**, 1784–1787. <https://doi.org/10.1103/PhysRevLett.82.1784> (1999).
- Knill, E., Laflamme, R. & Milburn, G. J. A scheme for efficient quantum computation with linear optics. *Nature* **409**, 46–52. <https://doi.org/10.1038/35051009> (2001).
- Kok, P. *et al.* Linear optical quantum computing with photonic qubits. *Rev. Mod. Phys.* **79**, 135–174. <https://doi.org/10.1103/RevModPhys.79.135> (2007).
- Tang, H. *et al.* Generating Haar-uniform randomness using stochastic quantum walks on a photonic chip. *Phys. Rev. Lett.* **128**, 050503. <https://doi.org/10.1103/PhysRevLett.128.050503> (2022).
- Di Colandrea, F. *et al.* Ultra-long quantum walks via spin-orbit photonics. *Optica* **10**, 324. <https://doi.org/10.1364/OPTICA.474542> (2023).
- Raussendorf, R., Browne, D. E. & Briegel, H. J. Measurement-based quantum computation on cluster states. *Phys. Rev. A* **68**, 022312. <https://doi.org/10.1103/PhysRevA.68.022312> (2003).
- Walther, P. *et al.* Experimental one-way quantum computing. *Nature* **434**, 169–176. <https://doi.org/10.1038/nature03347> (2005).
- Wei, T.-C., Haghnegahdar, P. & Raussendorf, R. Hybrid valence-bond states for universal quantum computation. *Phys. Rev. A* **90**, 042333. <https://doi.org/10.1103/PhysRevA.90.042333> (2014).
- Larsen, M. V., Chamberland, C., Noh, K., Neergaard-Nielsen, J. S. & Andersen, U. L. Fault-tolerant continuous-variable measurement-based quantum computation architecture. *PRX Quantum* **2**, 030325. <https://doi.org/10.1103/PRXQuantum.2.030325> (2021).
- Farhi, E., Goldstone, J., Gutmann, S. & Sipser, M. Quantum computation by adiabatic evolution (2000). [arXiv:quant-ph/0001106](https://arxiv.org/abs/quant-ph/0001106).
- Santoro, G. E. & Tosatti, E. Optimization using quantum mechanics: Quantum annealing through adiabatic evolution. *J. Phys. A Math. Gen.* **39**, R393–R431. <https://doi.org/10.1088/0305-4470/39/36/R01> (2006).
- Biamonte, J. D. & Love, P. J. Realizable Hamiltonians for universal adiabatic quantum computers. *Phys. Rev. A* **78**, 012352. <https://doi.org/10.1103/PhysRevA.78.012352> (2008).
- Arthur, D. & Date, P. Balanced k-means clustering on an adiabatic quantum computer. *Quantum Inf. Process.* **20**, 294. <https://doi.org/10.1007/s11128-021-03240-8> (2021).
- Hegade, N. N. *et al.* Shortcuts to adiabaticity in digitized adiabatic quantum computing. *Phys. Rev. Appl.* **15**, 024038. <https://doi.org/10.1103/PhysRevApplied.15.024038> (2021).

28. Meyer, D. A. From quantum cellular automata to quantum lattice gases. *J. Stat. Phys.* **85**, 551–574. <https://doi.org/10.1007/BF02199356> (1996).
29. Chandrashekar, C. M., Srikanth, R. & Laflamme, R. Optimizing the discrete time quantum walk using a SU(2) coin. *Phys. Rev. A* **77**, 032326. <https://doi.org/10.1103/PhysRevA.77.032326> (2008).
30. Venegas-Andraca, S. E. Quantum walks: A comprehensive review. *Quantum Inf. Process.* **11**, 1015–1106. <https://doi.org/10.1007/s11128-012-0432-5> (2012).
31. Campos, E., Venegas-Andraca, S. E. & Lanzagorta, M. Quantum tunneling and quantum walks as algorithmic resources to solve hard K-SAT instances. *Sci. Rep.* **11**, 16845. <https://doi.org/10.1038/s41598-021-95801-1> (2021).
32. Ambainis, A. Quantum walks and their algorithmic applications. *Int. J. Quantum Inf.* **01**, 507–518. <https://doi.org/10.1142/S0219749003000383> (2003).
33. Chawla, P., Mangal, R. & Chandrashekar, C. M. Discrete-time quantum walk algorithm for ranking nodes on a network. *Quantum Inf. Process.* **19**, 158. <https://doi.org/10.1007/s11128-020-02650-4> (2020).
34. Carrette, T., Laurière, M. & Magniez, F. Extended learning graphs for triangle finding. *Algorithmica* **82**, 980–1005. <https://doi.org/10.1007/s00453-019-00627-z> (2020).
35. Childs, A. M. Universal computation by quantum walk. *Phys. Rev. Lett.* **102**, 180501. <https://doi.org/10.1103/PhysRevLett.102.180501> (2009).
36. Lovett, N. B., Cooper, S., Everitt, M., Trevers, M. & Kendon, V. Universal quantum computation using the discrete-time quantum walk. *Phys. Rev. A* **81**, 042330. <https://doi.org/10.1103/PhysRevA.81.042330> (2010).
37. Singh, S., Chawla, P., Sarkar, A. & Chandrashekar, C. M. Universal quantum computing using single-particle discrete-time quantum walk. *Sci. Rep.* **11**, 11551. <https://doi.org/10.1038/s41598-021-91033-5> (2021).
38. Zhong, H.-S. *et al.* Quantum computational advantage using photons. *Science* **370**, 1460–1463. <https://doi.org/10.1126/science.abe8770> (2020).
39. Zhang, P. *et al.* Implementation of one-dimensional quantum walks on spin-orbital angular momentum space of photons. *Phys. Rev. A* **81**, 052322. <https://doi.org/10.1103/PhysRevA.81.052322> (2010).
40. Giordani, T. *et al.* Experimental engineering of arbitrary qudit states with discrete-time quantum walks. *Phys. Rev. Lett.* **122**, 020503. <https://doi.org/10.1103/PhysRevLett.122.020503> (2019).
41. Banerjee, S., Srikanth, R., Chandrashekar, C. M. & Rungta, P. Symmetry-noise interplay in a quantum walk on an n-cycle. *Phys. Rev. A* **78**, 052316. <https://doi.org/10.1103/PhysRevA.78.052316> (2008).
42. Nielsen, M. A. & Chuang, I. L. *Quantum Computation and Quantum Information* 10th anniversary. (Cambridge University Press, 2010).
43. Laflamme, R., Miquel, C., Paz, J. P. & Zurek, W. H. Perfect quantum error correcting code. *Phys. Rev. Lett.* **77**, 198–201. <https://doi.org/10.1103/PhysRevLett.77.198> (1996).
44. Schuch, N. & Siewert, J. Programmable networks for quantum algorithms. *Phys. Rev. Lett.* **91**, 027902. <https://doi.org/10.1103/PhysRevLett.91.027902> (2003).
45. Qiang, X. *et al.* Efficient quantum walk on a quantum processor. *Nat. Commun.* **7**, 11511. <https://doi.org/10.1038/ncomms11511> (2016).
46. Nejadstarrari, F. *et al.* Experimental realization of wave-packet dynamics in cyclic quantum walks. *Optica* **6**, 174. <https://doi.org/10.1364/OPTICA.6.000174> (2019).

Acknowledgements

CMC would like to thank Department of Science and Technology, Government of India for the Ramanujan Fellowship Grant No.: SB/S2/RJN-192/2014. We also acknowledge the support from Interdisciplinary Cyber Physical Systems (ICPS) programme of the Department of Science and Technology, India, Grant No.: DST/ICPS/QuST/Theme-1/2019/1.

Author contributions

C.M.C. designed the study, and P.C., S.S., A.A. and S.Sr. carried out the further work under the guidance of C.M.C. and prepared the figures. P.C., S.S. and C.M.C. wrote the manuscript, and all authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1038/s41598-023-39061-1>.

Correspondence and requests for materials should be addressed to P.C.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2023