



Morley-Short, S., Bartolucci, S., Gimeno-Segovia, M., Shadbolt, P., Cable, H., & Rudolph, T. (2018). Physical-depth architectural requirements for generating universal photonic cluster states. *Quantum Science and Technology*, 3(1), [015005]. <https://doi.org/10.1088/2058-9565/aa913b>

Publisher's PDF, also known as Version of record

License (if available):
CC BY

Link to published version (if available):
[10.1088/2058-9565/aa913b](https://doi.org/10.1088/2058-9565/aa913b)

[Link to publication record in Explore Bristol Research](#)
PDF-document

This is the final published version of the article (version of record). It first appeared online via IOP at <http://iopscience.iop.org/article/10.1088/2058-9565/aa913b/meta> . Please refer to any applicable terms of use of the publisher.

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:
<http://www.bristol.ac.uk/pure/about/ebr-terms>

PAPER • OPEN ACCESS

Physical-depth architectural requirements for generating universal photonic cluster states

To cite this article: Sam Morley-Short *et al* 2018 *Quantum Sci. Technol.* **3** 015005

View the [article online](#) for updates and enhancements.

Related content

- [Relative multiplexing for minimising switching in linear-optical quantum computing](#)
Mercedes Gimeno-Segovia, Hugo Cable, Gabriel J Mendoza et al.
- [Loss tolerant linear optical quantum memory by measurement-based quantum computing](#)
Michael Varnava, Daniel E Browne and Terry Rudolph
- [Phase transition of computational power in the resource states for one-way quantum computation](#)
Daniel E Browne, Matthew B Elliott, Steven T Flammia et al.

Quantum Science and Technology



PAPER

Physical-depth architectural requirements for generating universal photonic cluster states

OPEN ACCESS

RECEIVED

26 June 2017

REVISED

20 September 2017

ACCEPTED FOR PUBLICATION

5 October 2017

PUBLISHED

15 November 2017

Original content from this work may be used under the terms of the [Creative Commons Attribution 3.0 licence](#).

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.



Sam Morley-Short^{1,2,5} , Sara Bartolucci³, Mercedes Gimeno-Segovia^{1,3,4}, Pete Shadbolt³, Hugo Cable¹ and Terry Rudolph³

¹ Quantum Engineering Technology Labs, H. H. Wills Physics Laboratory and Department of Electrical and Electronic Engineering, University of Bristol, BS8 1FD, United Kingdom

² Quantum Engineering Centre for Doctoral Training, H. H. Wills Physics Laboratory and Department of Electrical and Electronic Engineering, University of Bristol, Tyndall Avenue, BS8 1FD, United Kingdom

³ Department of Physics, Imperial College London, London SW7 2AZ, United Kingdom

⁴ Institute for Quantum Science and Technology, University of Calgary, Alberta T2N 1N4, Canada

⁵ Author to whom any correspondence should be addressed.

E-mail: sam.morley-short@bristol.ac.uk and hugo.cable@bristol.ac.uk

Keywords: quantum computing, linear optical quantum computing, applied percolation theory, quantum computer architectures

Supplementary material for this article is available [online](#)

Abstract

Most leading proposals for linear-optical quantum computing (LOQC) use cluster states, which act as a universal resource for measurement-based (one-way) quantum computation. In *ballistic* approaches to LOQC, cluster states are generated passively from small entangled resource states using so-called fusion operations. Results from percolation theory have previously been used to argue that universal cluster states can be generated in the ballistic approach using schemes which exceed the critical threshold for percolation, but these results consider cluster states with unbounded size. Here we consider how successful percolation can be maintained using a physical architecture with fixed physical depth, assuming that the cluster state is continuously generated and measured, and therefore that only a finite portion of it is visible at any one point in time. We show that universal LOQC can be implemented using a constant-size device with modest physical depth, and that percolation can be exploited using simple pathfinding strategies without the need for high-complexity algorithms.

1. Introduction

Within the last decade, great progress has been made in the theoretical field of quantum computer architectures. Modern fault-tolerant schemes rely on the use of many error-prone *physical* qubits to create individual *logical* qubits with fewer errors. Whilst we understand these methods of abstraction theoretically, implementing them in reality is not a trivial task when experimental constraints are applied. The study of quantum computation architectures must therefore incorporate both an understanding of high-level theoretical models and experimental limitations.

While there are many attractive aspects of photonic qubits, utilizing them for linear-optical quantum computation (LOQC) presents some unique architectural challenges [1]. Most significantly, LOQC suffers from a lack of deterministic entangling gates, with initial proposals requiring large resource overheads to compensate [2, 3]. However, the main challenge for modern LOQC architectures [4–14] remains the generation and utilization of highly-entangled resource states. This is now generally addressed within the paradigm of cluster states [15, 16] applied to LOQC [4] and the use of entangling fusion gates [5, 17, 18].

One particularly appealing approach to LOQC uses ideas from percolation theory as first proposed in [8]. The main idea is to passively entangle small resource states (also called microclusters), using fusion gates, to generate a large cluster state which can enable universal quantum computing. The cluster state which is generated corresponds to a random graph on a geometric lattice with missing sites and bonds. By using schemes which exceed the critical threshold for percolation on the lattice [8–10, 19], a cluster state which supports

universal quantum computation can be guaranteed. A lattice of logical qubits can then be identified using methods such as renormalization as given in [8], or the lattice concentration algorithm of [20]. The main virtue of using the percolation approach to LOQC is that it enables *ballistic* architectures that sidestep requirements for extensive adaptive switching networks, which are technologically very challenging [21].

In this work, we address a vital question that must be addressed for any high-level LOQC architecture based on percolation: can successful percolation be sustained using a physical device of fixed finite size, and what size (cross-section and depth) of percolating cluster state must be kept *online* at any point in time to do so? The methods we use to answer this question differ from conventional treatments of percolation, and are based on pathfinding algorithms which must exploit information in real-time about the outcomes of recent fusion operations. We assume that photons making up the percolating cluster state can only be kept online for modest periods using optical delays, which provide limited *lookahead* capability before measurements must be performed on the photons. Our analysis can have implications for all aspects of LOQC architecture by impacting hardware specifications at the component level. Specifically, this work presents three key results: (i) spanning paths can exist on extremely elongated blocks of edge-percolated cluster state lattice, but only when the cross-sectional side length exceeds some minimum length set by the lattice edge probability; (ii) an LOQC device with a physical-depth of only 10–20 layers is sufficient to produce measurement-based quantum computation (MBQC) qubit channels (within a loss- and error-less LOQC architecture model); (iii) long-range limited-lookahead pathfinding can be achieved with algorithms with minimal complexity, thereby reducing associated classical co-processing requirements for LOQC.

The structure of this work is as follows: in section 2 we briefly review recent work on percolation-based architectures for LOQC. In section 3 we consider the minimum resource requirements of percolated cluster state lattices for producing long-range single-qubit channels. In section 4 we present the main results of our work, where we define the Random-node pathfinding process, conjecture a condition of pathfinding success and present results from numerical pathfinding simulations. Section 5 considers implications of the results presented for LOQC architectures, identifying key architectural trade-offs and specifications. Finally, a selection of open questions for future work are presented in section 6.

2. Percolation-based architectures for LOQC

The fundamental challenge of LOQC is the construction of large graph states. Graph states are a subset of stabilizer states [22] that can be uniquely described by simple graphs (for a review of graph states see [23]). In this formalism, a graph $G(V, E)$ containing *vertices* (or *nodes*) V and *edges* (or *bonds*) E , uniquely represents the state

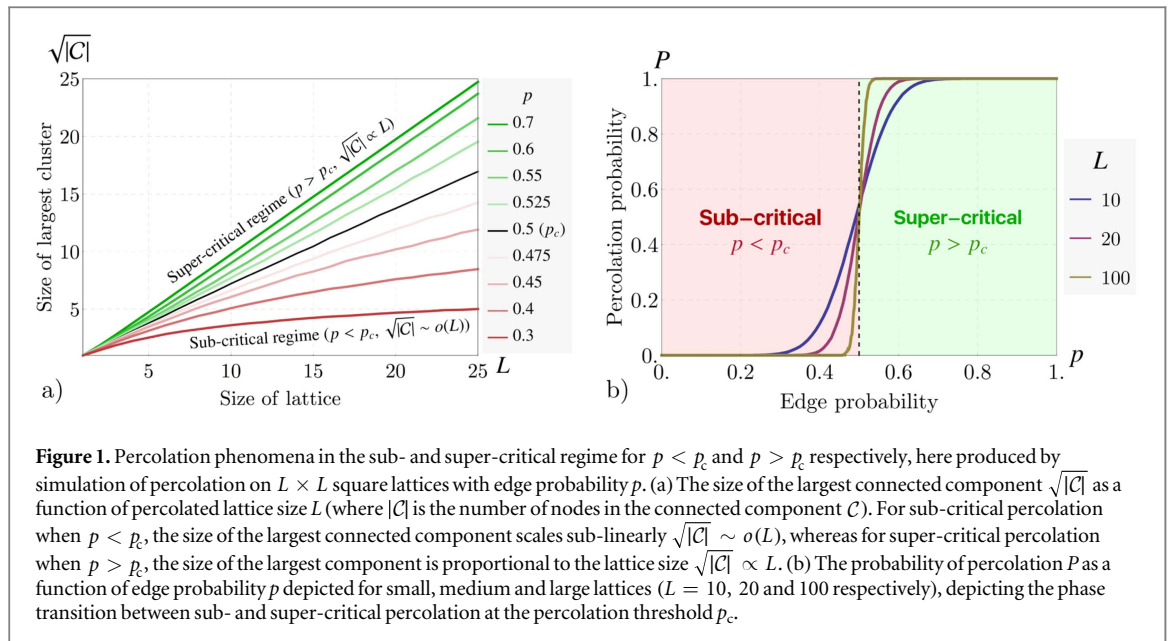
$$|\Psi_G\rangle = \prod_{(i,j) \in E} CZ_{i,j} \bigotimes_{v \in V} |+\rangle_v, \quad (1)$$

where $CZ_{i,j} = |00\rangle\langle 00|_{i,j} + |01\rangle\langle 01|_{i,j} + |10\rangle\langle 10|_{i,j} - |11\rangle\langle 11|_{i,j}$ and $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. We specifically refer to graph states represented by regular lattices as *cluster states*. In LOQC, cluster states can be probabilistically built using two types of fusion gate [5]. Known as type-I and -II fusion gates, these gates destructively consume 1 and 2 photonic qubits respectively and on success produce entanglement between the remaining qubits in the clusters (and on failure the input qubits are subjected to single-qubit measurements). Whilst type-I fusion consumes fewer qubits, it cannot herald photon loss, whereas type-II can herald such loss, but at the cost of consuming an extra qubit. In standard operation, both gates operate with a 50% success rate. However, Type-II fusion can be *boosted* to increase the success rate above 50% through the consumption of additional auxiliary resources [17, 18]. For example, a success rate of 75% can be achieved through either the consumption of a Bell pair or 4 single photons.

To overcome nondeterministic entangling gates, renormalization is used to produce an idealized lattice \mathcal{L}^* from a coarse graining of some percolated lattice \mathcal{L} . For example, in one common strategy, microcluster states are placed on the sites of a lattice and fusion gates of success probability p_f are applied to produce entanglement between the centre qubits of adjacent microclusters. Once \mathcal{L} is constructed, a single *central* qubit is identified on each renormalization block that is path-connected to central qubits of adjacent blocks by sets of *path* qubits⁶. As in MBQC protocols [15, 16], all other qubits in the lattice are then removed by adaptive single-qubit measurements, thereby producing \mathcal{L}^* . An example of this is depicted in figure 3, where a single-qubit MBQC channel is produced from the renormalization of a 2D lattice.

The size of blocks on \mathcal{L} required for renormalization to a fixed \mathcal{L}^* depends only on the percolation threshold p_c of \mathcal{L} , as produced by the lattice's structure. Reducing the overall resource requirements for a LOQC device therefore relies on producing a lattice with low p_c without the need for high-degree and therefore costly

⁶ For the renormalization of 2D lattices, a different method based on the identification of topological minors is also known [20], however this has yet to be extended to higher dimensional lattices.



microcluster resource states. Initial work on renormalization identified cubic, diamond and pyrochlore lattices as potential candidates, requiring 7-, 5- and 4-qubit microcluster resources respectively [8]. By extending a percolation approach to the generation of resource states, it was shown that both microcluster creation and fusion could be achieved from boosted fusion [17, 18] of 3-photon GHZ states to produce a ‘brickwork’ diamond lattice with $p_c < p_f$ [10] and pyrochlore [9]. Recently, this scheme was further generalized for higher-dimensional lattices and n -qubit microclusters [19]. After \mathcal{L} has been constructed, renormalization can be abstracted to the graph-theoretical problem of finding crossing clusters on percolated lattices, which can be solved efficiently [24].

Commonly, schemes for generating \mathcal{L} correspond to a bond-percolation, where successful bonds correspond to *open* edges [25, 26]. On percolated lattices with bond probability p , the existence of an infinite open cluster exhibits threshold behaviour. In the limit of an infinite lattice \mathcal{L}_∞ , the probability $P_\infty(p, \mathcal{L}_\infty)$ that an infinite open cluster \mathcal{C}_∞ undergoes a phase transition (from 0 to 1) at $p = p_c$. This threshold represents the division between two distinct percolation regimes for $p < p_c$ and $p > p_c$, known respectively as the sub- and super-critical regime. The degree of connectivity within the lattice is fundamentally different between these regimes; for example, the scaling in size of the largest connected component transitions from sub-linear to linear across the threshold, as depicted in figure 1(a). For finite lattices \mathcal{L} , the finite-sized analogue to P_∞ is probability $P_i(p, \mathcal{L})$ that a spanning cluster \mathcal{C} exists along the i direction, thereby containing a path connecting opposite faces of the lattice block along axis i . Thresholds for $P_i(p, \mathcal{L})$ correspond to continuous functions, becoming sharper for larger lattices and converge to $P_\infty(p, \mathcal{L}_\infty)$, as depicted by figure 1(b). In practise, percolation thresholds can be found by identifying the crossing point of functions $P_i(p, \mathcal{L})$ for various sizes of \mathcal{L} [26], or numerically using the Newman–Ziff algorithm [27].

In order to exploit percolation phenomena within a scheme for quantum computation, [10] also considered percolation on a subregion of the lattice with a small cross section which is to be used as a single-qubit channel for MBQC. By simulating $P_i(p_f = 0.75, \mathcal{L})$ for $L_t \times L \times L$ brickwork diamond lattices over a range of L (for $L_t \gg L$), it was shown that long-range percolation, and hence a single-qubit channel, was produced above some minimum L . This result can also be applied to finding long-range renormalization.

3. Long-range percolation for single-qubit channels

Our first set of new results extends the study of lattice percolation for single-qubit channels presented in [10], which was limited to the generation of the partially amorphous⁷ and anisotropic brickwork diamond lattice, built specifically with $p_f = 0.75$ fusion gates. To do so, we present a generalized model of percolation on elongated bond-percolated cubic lattices and establish a relationship between the minimum side-length L_{\min} required for consistent long-range percolation and edge probability p .

⁷ Here *partially amorphous* describes a lattice that may contain bonds other than those defined by the lattice structure, such as diagonal edges or edges between non-adjacent nodes. When constructing a brickwork diamond lattice by the scheme presented in [10], this occurs for certain choices of fusion gate bases.

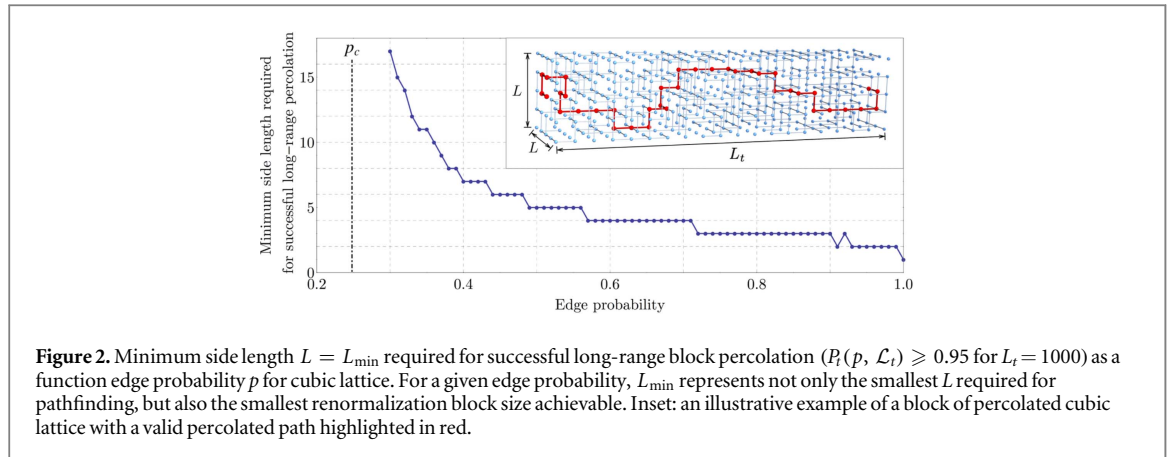


Figure 2. Minimum side length $L = L_{\min}$ required for successful long-range block percolation ($P_t(p, \mathcal{L}_t) \geq 0.95$ for $L_t = 1000$) as a function edge probability p for cubic lattice. For a given edge probability, L_{\min} represents not only the smallest L required for pathfinding, but also the smallest renormalization block size achievable. Inset: an illustrative example of a block of percolated cubic lattice with a valid percolated path highlighted in red.

The model we use is as follows: consider a block of percolated $L_t \times L \times L$ cubic lattice \mathcal{L}_t with edge probability p , where $L_t \gg L$, depicted in the inset of figure 2. On \mathcal{L}_t , we examine the existence of an end-to-end spanning cluster, occurring with probability $P_t(p, \mathcal{L}_t)$. To produce a reliable single-qubit channel, we specifically consider probabilities of percolation near unity, $P_t(p, \mathcal{L}_t) \approx 1$. We therefore generally consider *successful* outcomes (for percolation and, in later sections, pathfinding) as having probability of at least 0.95, and *long-range* as referring to $L_t \geq 1000$. These definitions are chosen such that if the above conditions are satisfied, a renormalized qubit loss rate below 10^{-3} can be achieved (given reasonable assumptions of renormalization blocks with side-length $\mathcal{O}(10)$ in the scheme of Kieling *et al* [8])⁸. Given the known trade-off between correctability of qubit error and qubit loss for topological codes [28], minimizing loss rates is essential for maximizing tolerance for unavoidable computational errors. Such a low rate is also expected to be a negligible contribution to renormalized qubit loss in the face of other potential sources of error within the architecture (such as photonic qubit loss, detector inefficiencies, distinguishability, etc).

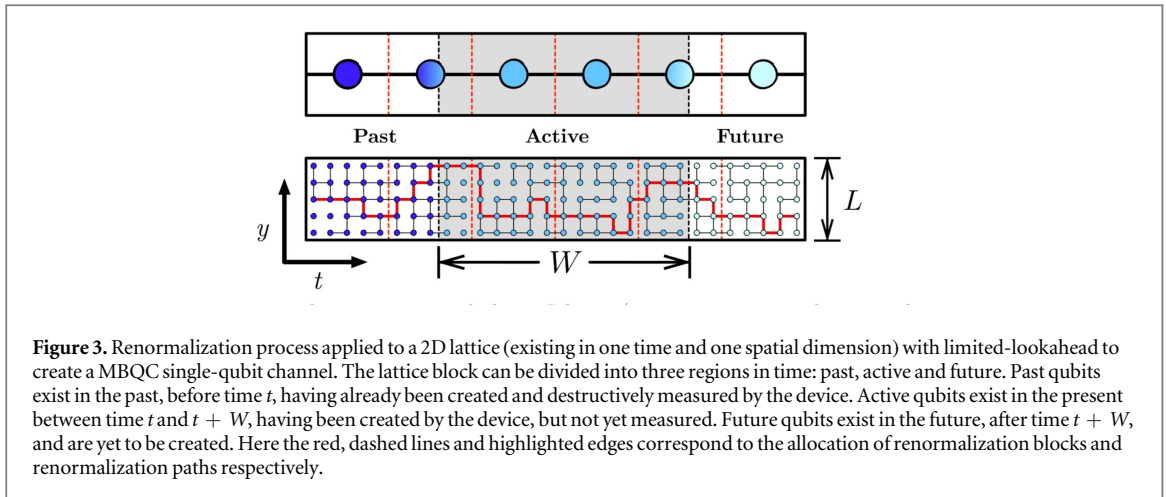
However, within this model, percolation phenomena are less-well studied than in the standard regime. When considering finite-sized, elongated lattices such as \mathcal{L}_t , it is challenging to make analytic statements about the existence of spanning clusters, as can often be done for the limit of infinite lattices. For example, while for a lattice \mathcal{L}_t , one can find some $p < 1$ such that $P_t(p, \mathcal{L}_t) \approx 1$, it is necessarily true⁹ that as $L_t \rightarrow \infty$, $P_t(p, \mathcal{L}_t) \rightarrow 0$. As such, we highlight that all results presented in this work are expected to have some minor functional dependence on our specific definition of *successful* and *long-range* given above. Therefore, we apply a more phenomenological and empirical approach to the relevant percolation effects, and within the context of LOQC such results provide important information for designing an architecture.

We now consider the following question: what is the minimum side length L_{\min} required to successfully produce a long-range spanning cluster \mathcal{C} on \mathcal{L}_t as a function of edge probability p ? To answer this question numerically, we have generated instances of $1000 \times L \times L$ sized \mathcal{L}_t for a given p , and identified the minimum value $L = L_{\min}$ for which $P_t(p, \mathcal{L}_t) \geq 0.95$. In figure 2 we show values of L_{\min} over a range of $p > p_c$. We observe that for edge probabilities well above $p_c = 0.248$ (the percolation threshold for a simple cubic lattice [29]), small L_{\min} can be achieved (such as $L_{\min} = 5$ for $p = 0.5$), with small increases in L_{\min} providing large reductions in p . However, as p approaches p_c , the scaling in L_{\min} is less favourable, requiring progressively greater increases in L_{\min} for incremental reductions in p . This scaling region suffers from particularly punitive resource costs if used for MBQC, as the number of qubits in $\mathcal{L}_t = 1000L^2$ scales quadratically in L . We also note that such a relationship for $L_{\min}(p)$ can be inverted to define $p_{\min}(L)$, such that for a given L , long-range percolation can only be achieved for some $p \geq p_{\min}$.

Furthermore, we can consider the implications of these results for a renormalization-based LOQC scheme. In this context, L_{\min} provides a lower bound on the side length for renormalization blocks. Whether or not this bound can be reached depends on finding intersections between spanning clusters connecting pairs of opposing faces within a single block as well as between adjacent blocks. This is especially problematic for p close to p_c as inter- and intra-block connectivity is sparse; however for p well above p_c , the increased connectivity also increases the likelihood such intersections occur.

⁸ This can be seen by noting that if the probability of creating 100 renormalized qubits is greater than 0.95, then the probability of creating a single renormalized qubit is (to a reasonable approximation) greater than $0.95^{1/100} \approx 0.9995$, and thus the loss rate for said qubit is less than 10^{-3} .

⁹ This can be seen by considering that the probability of no open edges occurring between two layers spanning the cross-section of the block is $(1-p)^{L^2}$, and hence the probability that this never occurs over L_t layers is $\Gamma = (1 - (1-p)^{L^2})^{L_t} \leq 1$. Since a spanning cluster is contingent on this never occurring then $P_t(\mathcal{L}) < \Gamma$, but for $p < 1$, $L_t \rightarrow \infty \Rightarrow \Gamma \rightarrow 0$, and therefore in the limit of infinite length, percolation never occurs.



4. Limited-lookahead pathfinding

In a physical LOQC device, \mathcal{L} exists in one time and two spatial dimensions with \mathbb{Z}^3 node coordinates (t, y, z) and size $L_t \times L_y \times L_z$. To construct \mathcal{L} , at each time t from $t = 0$ to $t = L_t$, a $L_y \times L_z$ layer of \mathcal{L} is created and entangled to the previous layer at $t - 1$, where L_y and L_z are fixed by the renormalization protocol. However, all-optical storage of L_t lattice layers in time would require lengthy delay lines, producing a physical qubit loss rate that scales with computation length (for some applications L_t is effectively unbounded); under such conditions, it is unlikely such a scheme could succeed.

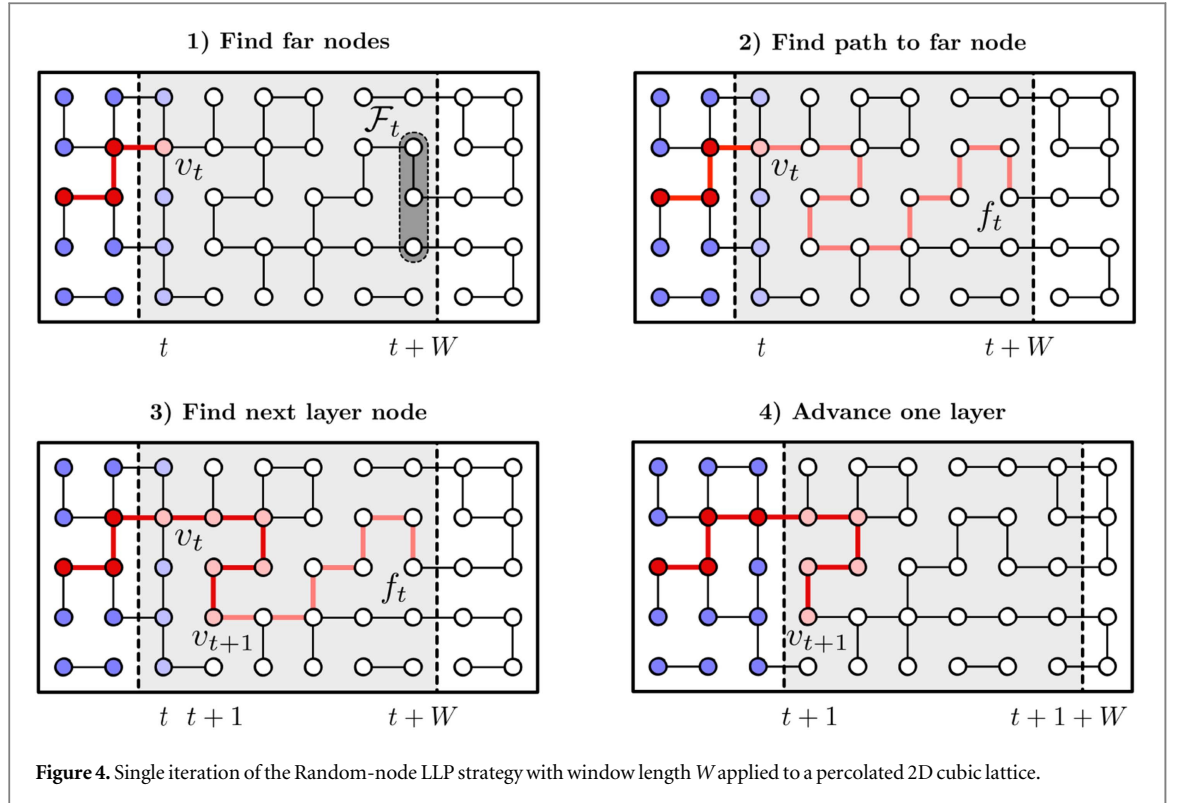
It is therefore expected that an LOQC device will have a finite fixed depth, storing only a finite-depth *window* W of the lattice at any time t . In this model, depicted in figure 3, any classical co-processing algorithms applied to \mathcal{L} suffer from a *limited-lookahead*, preventing analysis of a complete \mathcal{L} (as previously assumed by algorithms for MBQC and renormalization). Under this limitation, previously-considered algorithms no longer apply, or their optimality proofs and scaling efficiencies are no longer guaranteed. To address this, new non-trivial dynamic algorithms must be designed.

However, finding optimality proofs for graph algorithms that only ever have partial knowledge of a problem is highly non-trivial, and different input scenarios may require different algorithm strategies for optimal performance. To study the limitations of the necessary dynamic algorithms, we consider the aforementioned task of identifying single-qubit channels on percolated lattices. Specifically, we extend the task of finding a spanning cluster presented in section 3 to the identification of a single end-to-end path, given a limited-lookahead. To do so, we next construct a basic *limited lookahead pathfinding* (LLP) algorithm.

4.1. Random-node pathfinding

We now introduce some notation needed for describing the LLP algorithm. Consider again the lattice \mathcal{L}_t as defined in section 3, with nodes labelled by their coordinates (t, y, z) . We define a *layer* l_t as the subgraph of \mathcal{L}_t induced by the 2D $L \times L$ layer of nodes at time t , that is $l_t = \mathcal{L}[\{v = (t, y, z), \forall y, z = 1, \dots, L\}]$, where $G' = G[V]$ denotes the *induced subgraph* G' of G by the node set V . We define a *block* $\mathcal{B}_{a,b}$ as the subgraph of \mathcal{L}_t induced by the 3D block of nodes within layers a to b (inclusive), that is $\mathcal{B}_{a,b} = \mathcal{L}[\{v = (t, y, z), \forall t = a, \dots, b, y, z = 1, \dots, L\}]$. Note that under this definition $\mathcal{L}_t = \mathcal{B}_{0,L_t}$. The nodes within $\mathcal{B}_{a,b}$ that are also part of spanning cluster \mathcal{C} of \mathcal{L}_t are denoted $\mathcal{C}_{a,b} = \mathcal{C} \cap \mathcal{B}_{a,b}$ and represent nodes that are potentially usable for pathfinding. Similarly, $\mathcal{C}_t = \mathcal{C} \cap l_t$. In some $\mathcal{B}_{a,b}$, $\mathcal{C}_{a,b}$ may contain more than one connected component. Therefore, we also define $\mathcal{C}_{a,b}(v) = \mathcal{L}[\{v' \in \mathcal{C}_{a,b} : \langle v \leftrightarrow v' \rangle\}]$ as the connected component of $\mathcal{C}_{a,b}$ containing node v , where $\langle v \leftrightarrow v' \rangle$ indicates that there exists an open path connecting v and v' . Hence, if two nodes u and v are not path-wise connected within $\mathcal{C}_{a,b}$, then they must exist in disjoint connected components and $\mathcal{C}_{a,b}(u) \cap \mathcal{C}_{a,b}(v) = \emptyset$. Lastly, the superscript $E-E$ denotes components that extend *end-to-end* across the layers indicated, e.g. $\mathcal{C}_{a,b}^{E-E}$ are the components in $\mathcal{C}_{a,b}$ that have nodes in both l_a and l_b with the number of separate end-to-end components given by $n(\mathcal{C}_{a,b}^{E-E})$.

To represent a limited lookahead, we consider the restriction that at a given time t , we can only have knowledge of the lattice structure within the finite block $\mathcal{B}_{t,t+W}$ of fixed *window-length* W . This 'visible' block of lattice is known as the *active block*. At the end of every time-step, the next far layer of lattice l_{t+1+W} is revealed and nearest layer l_t is removed, the active block now becoming $\mathcal{B}_{t+1,t+W+1}$ for time $t + 1$.



This limitation requires us to consider an iterative approach to finding spanning paths, which we shall call *limited-lookahead pathfinding*, where each time-step the algorithm must choose a path inside the lattice based on only partial information of the lattice. Specifically, we shall consider a low-complexity instance of pathfinding, which we call *Random-node pathfinding*. We consider a naive algorithm such as this to both identify a lower bound on the success rates of general pathfinding strategies as well as their computational complexities. To find a path \mathcal{P} the following pathfinding algorithm is applied (depicted visually in figure 4), starting at $t = 0$, (with $\mathcal{P} = v_0$ for some $v_0 \in \mathcal{C}_0$) and is repeated until success or failure occurs:

Random-node pathfinding:

- 1) **Find far nodes.** From the current path node v_t in the nearest layer l_t , find the set of all nodes $\mathcal{F}_t = \{v \in l_{t+W} : v_t \leftrightarrow v\}$ in the farthest active block layer l_{t+W} to which a path exists (only considering nodes and edges within the active block). If $\mathcal{F} = \emptyset$, pathfinding fails.
- 2) **Find path to far node.** Randomly pick a far node f_t from \mathcal{F}_t , and find the shortest path $\mathcal{P}_t = (v_t, \dots, f_t)$ within the window to it.
- 3) **Find next layer node.** Find the node in layer l_{t+1} that occurs furthest along \mathcal{P}_t and assign it to the next time-step path node v_{t+1} . Append the (v_t, \dots, v_{t+1}) section of \mathcal{P}_t , to \mathcal{P} . If the final node f_t in \mathcal{P} is a member of l_{L_c} , pathfinding succeeds.
- 4) **Advance one layer.** Remove layer l_t and reveal layer l_{t+1+W} .

The first thing to note is that this algorithm is far from optimal, and in fact is almost the worst strategy one could apply (other than making deliberately bad path choices). The only non-trivial analysis of structure occurs at step 3, where the action of finding the furthest l_t layer node allows the inclusion of paths that double-back on themselves, advancing forwards and then back to layer l_t before eventually reaching the final layer, an example of which is shown in step 3 of figure 4. The most computationally expensive operation in the algorithm occurs in step 1, when finding \mathcal{F}_t . This operation consists of running Dijkstra's algorithm (for finding shortest paths on arbitrary graphs) from v_t , thus providing Random-node pathfinding with an overall worst-case performance of $\mathcal{O}(|E| + |V| \log|V|)$ [30].

Finding optimal pathfinding strategies which demand only minimal values for W is very challenging in general and the Random-node strategy can be used to explore the worst-case scenario, from which improvement may be made. Inevitably, more complex strategies that require detailed analysis of the active

block's configuration are computationally expensive, which is a major concern for real-time implementation in hardware devices. A secondary aim of our work is therefore also to minimize the computational overhead required for pathfinding, and the Random-node strategy also adheres to this goal.

4.2. Successful long-range pathfinding

We now consider the conditions required for successful long-range LLP and show that these can be framed in terms of standard block percolation. This aims to reduce the complexity of analysing a dynamic pathfinding algorithm to the simpler problem of calculating percolation statistics on small lattices.

First and foremost, pathfinding fails if no spanning cluster exists. To ensure that a path does exist (with probability $P_t \geq 0.95$ for a given pathfinding distance L_t), we immediately require two conditions: $p > p_c$ and $L \geq L_{\min}(p)$. Having satisfied these, we then seek to identify the conditions such that pathfinding almost certainly succeeds. In this section, we prove that pathfinding always succeeds if the number of end-to-end components in each active block never exceeds one, and subsequently conjecture that successful pathfinding is only achieved if the probability of this number exceeding one is less than some small ϵ .

Before outlining our argument we assert two key assumptions made. Firstly, we assume a unique spanning cluster always exists across \mathcal{L} (where *unique* specifies that only one ever exists), and hence exclude any cases where long-range block percolation does not exist (e.g. by assuming $L > L_{\min}(p)$). The validity of this assumption is provided by recalling that for $p > p_c$ the mean size of a finite cluster decreases exponentially in p [25], thereby preventing more than one cluster from spanning the lattice. Given this assumption, failure therefore only occurs from incorrect choices made during pathfinding. Secondly, we assume that at any given time, the pathfinding algorithm may only have access to information of the lattice's structure within the active block, i.e. it cannot store in memory any information about past lattice structure, nor gain preemptive knowledge of any future lattice structure. This allows us to consider each individual active block as a single instance of block percolation on a small lattice, and hence percolation statistics are constant across all active blocks.

Under these assumptions, the probability $P_{pf}(\mathcal{L}_t, W)$ of pathfinding across \mathcal{L}_t with window length W , is given by the product of the probabilities $P_{pf}^t(\mathcal{B}_{t,t+W})$ that, at each time-step t , a path node v_{t+1} in $\mathcal{B}_{t,t+W}$ is chosen that still allows for successful pathfinding to distance L_t , that is

$$P_{pf}(\mathcal{L}_t, W) = \prod_{t=0}^{L_t-W} P_{pf}^t(\mathcal{B}_{t,t+W}, v_t), \quad (2)$$

such that for $W = L_t$, $P_{pf}(p, W) = P_{pf}^0(\mathcal{B}_{0,L_t}) = 1$ (from our first assumption). However, for $W < L_t$, the values of P_{pf}^t are less easily computed.

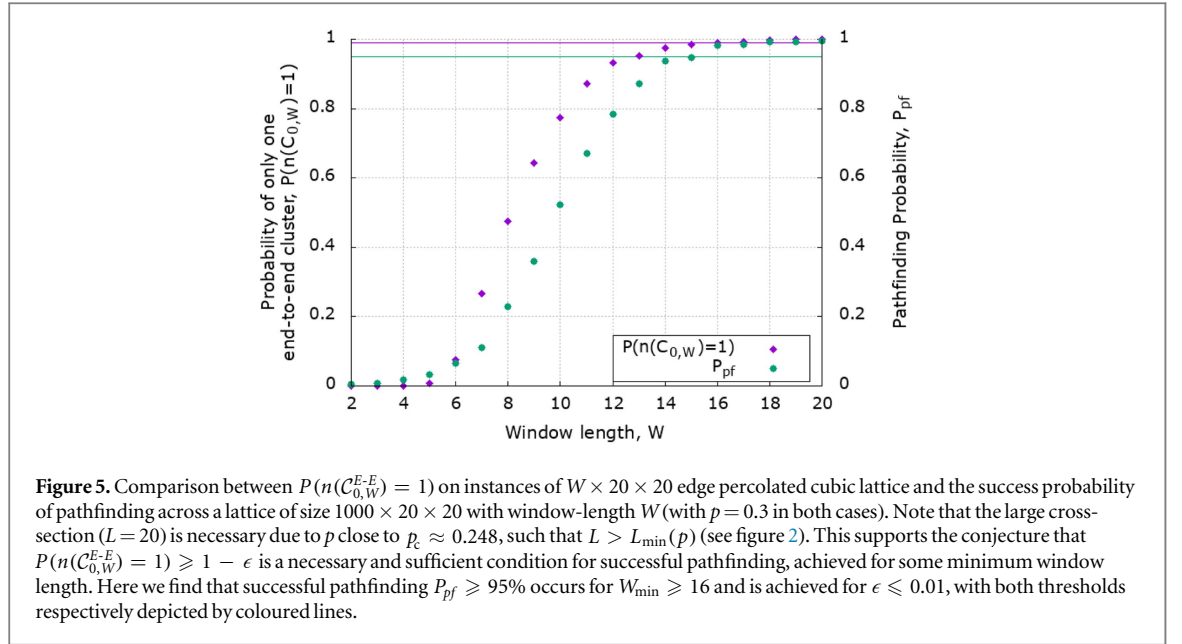
We can easily see that the probability of successful pathfinding given next node choice v_{t+1} depends on the probability that v_{t+1} exists in a component extending to the farthest layer, that is $P_{pf}^t(\mathcal{B}_{t,t+W}|v_{t+1}) = P(v_{t+1} \in \mathcal{C}_{t+1,L_t}^{E-E})$ (where we recall that $\mathcal{C}_{t+1,L_t}^{E-E}$ are the *end-to-end* connected components contained within \mathcal{B}_{t+1,L_t} that have one or more nodes in both l_{t+1} and l_{L_t}). However, at any given time step, we cannot know whether $v_{t+1} \in \mathcal{C}_{t+1,L_t}^{E-E}$ or not when $t \leq L_t - W$ (by our second assumption). Instead, we desire some active block proxy condition for $P(v_{t+1} \in \mathcal{C}_{t+1,L_t}^{E-E})$ based only on block percolation statistics.

Specifically, we are interested here in the case of $P_{pf}(\mathcal{L}_t, W) \approx 1$ and hence $P_{pf}^t(\mathcal{B}_{t,t+W}) \geq 1 - \epsilon$ (where $\epsilon \ll 1$) for all t . Ideally, we therefore desire some *feature function* of active blocks $F: \mathcal{B}_{t,t+W} \mapsto \{0, 1\}$, such that if $F(\mathcal{B}_{t,t+W}) = 1$, then $P_{pf}^t(\mathcal{B}_{t,t+W}) = 1$ surely, but if $F(\mathcal{B}_{t,t+W}) = 0$ then $P_{pf}^t(\mathcal{B}_{t,t+W}) < 1$. From this, we then conjecture that if lattice parameters can be found such that $P(F(\mathcal{B}_{t,t+W}) = 1) \geq 1 - \epsilon \forall t$, successful long-range pathfinding will be achieved. We now prove one such feature function to be the number of end-to-end connected components within an active block, and thereby define a condition for W such that $P(F(\mathcal{B}_{t,t+W}) = 1) \geq 1 - \epsilon$.

We find that one such feature function can be defined from the uniqueness of end-to-end connected components, such that

$$F(\mathcal{B}_{t,t+W}) = \begin{cases} 1 & \text{if } n(\mathcal{C}_{t,t+W}^{E-E}) = 1 \\ 0 & \text{if } n(\mathcal{C}_{t,t+W}^{E-E}) > 1 \end{cases} \quad (3)$$

To see that this satisfies our feature function requirements, consider two possible structures of $\mathcal{B}_{t,t+W}$, either $n(\mathcal{C}_{t,t+W}^{E-E}) = 1$, or $n(\mathcal{C}_{t,t+W}^{E-E}) > 1$. In the case of $n(\mathcal{C}_{t,t+W}^{E-E}) = 1$, the previous choice of path node was essentially irrelevant, since all connected nodes in the far layer \mathcal{C}_{t+W} can be reached from any $v_t \in \mathcal{C}_t$. If this condition is satisfied for *every* active block, then at each time-step all choices of path node (using our pathfinding process) are practically equivalent, and thus $P_{pf}(\mathcal{L}_t, W) = 1$. Alternatively, one can understand this by saying that if $n(\mathcal{C}_{t,t+W}^{E-E}) = 1$, the current connected component $\mathcal{C}_{t,t+W}^{E-E}(v_t)$ *must* be part of the spanning component \mathcal{C}_{t,L_t} extending to the final layer, or



$$\mathcal{C}_{t,t+W}^{E-E}(v_t) \cap \mathcal{C}_{t,L_t}^{E-E} \neq \emptyset, \quad (4)$$

assuming that $n(\mathcal{C}_{t+W,L_t}^{E-E}) = 1$ (which holds for $t + W \ll L_t$, from the uniqueness of \mathcal{C}_{0,L_t}).

Conversely, if $n(\mathcal{C}_{t,t+W}^{E-E}) > 1$, no choice of $v_t \in \mathcal{C}_t$ can possibly allow for all nodes in \mathcal{C}_{t+W} to be reached, and hence presents a possibility that v_t is not in a component that extends forward to the final layer, $v_t \notin \mathcal{C}_{t,L_t}^{E-E}$. In such a scenario, two possibilities exist: either, equation (4) is ultimately satisfied, indicating that a path passing through $\mathcal{C}_{t,t+W}(v_t)$ can reach layer L_D , and therefore allows successful pathfinding, or

$$\mathcal{C}_{t,t+W}^{E-E}(v_t) \cap \mathcal{C}_{t,L_t}^{E-E} = \emptyset, \quad (5)$$

indicating that structure within $\mathcal{C}_{t,t+W}(v_t)$ cannot contribute to pathfinding, and therefore represents a dead-end, which causes pathfinding to fail. Note, due to effect of finite block side lengths L , there is always some non-zero probability that equation (5) is satisfied (such as no open edges existing between nodes in $\mathcal{C}_{t+W}(v_t)$ and \mathcal{C}_{t+W+1}) and thus $P_{pf}(\mathcal{L}_t, W) < 1$. This proves that the condition $n(\mathcal{C}_{t,t+W}^{E-E}) = 1$ satisfies our desired feature function requirements.

We now consider the lattice requirements such that $P(n(\mathcal{C}_{t,t+W}^{E-E}) = 1) \geq 1 - \epsilon \forall t$. To satisfy this requirement, we define (for a given p and L) the *minimum window length* $W_{\min}(L, p)$ as the smallest W such that $P(n(\mathcal{C}_{t,t+W}^{E-E}) = 1) \geq 1 - \epsilon \forall t$. Note that for $L \geq L_{\min}$ such a minimum window length must exist; for any \mathcal{L}_t , clearly $P(n(\mathcal{C}_{0,W}^{E-E}) = 1) = 1$ when $W = L_t$ (from the uniqueness of $\mathcal{C}_{0,W}^{E-E}$), but as W is decreased from L_t , either W_{\min} is found when $P(n(\mathcal{C}_{t,t+W}^{E-E}) = 1) < 1 - \epsilon$ occurs, or else no lookahead is required (and $W_{\min} = 2$). We further define $W_{\max}(L, p_{\min})$ as the maximum window length required by LLP occurring for a given L at p_{\min} , above which any further increase in W provides no advantage.

We have shown that for a given \mathcal{L}_t with lattice parameters p and L , $n(\mathcal{C}_{t,t+W}^{E-E}) = 1$ is a sufficient feature function. We hence conjecture that $P(n(\mathcal{C}_{0,W}^{E-E}) = 1) \geq 1 - \epsilon$ is a necessary and sufficient condition for successful long-range LLP. That is, if this condition is not satisfied then no strategy (regardless of complexity) can ever produce successful long-range LLP, and that this condition is always satisfied for $W \geq W_{\min}$ as $\epsilon \rightarrow 0$.

4.3. Numerical simulation

We now consider numerical simulation of LLP applying a Random-node strategy.

Firstly, we address the conjecture that $P(n(\mathcal{C}_{0,W}^{E-E}) = 1) \geq 1 - \epsilon$ is a necessary and sufficient condition for successful pathfinding. Figure 5 depicts simulation of both LLP and $\mathcal{B}_{0,W}$ block percolation over a range of W for a cubic lattice. We observe that successful pathfinding occurs for minimum window length $W_{\min}(20, 0.3) = 16$, where $P_{pf}(p, W) = 0.983$ and $P(n(\mathcal{C}_{0,W}^{E-E}) = 1) = 0.991$, such that $\epsilon = 10^{-2}$. We also note that $P_{pf}(p, W)$ drops significantly as $P(n(\mathcal{C}_{0,W}^{E-E}) = 1)$ decreases below $1 - \epsilon$, further validating our choice of feature function. In conjunction with the proofs of our feature function presented in section 4.2, these results support our conjecture that $P(n(\mathcal{C}_{0,W}^{E-E}) = 1) \geq 1 - \epsilon$ is a necessary and sufficient condition for successful pathfinding.

We now consider the interdependence of pathfinding parameter W_{\min} and lattice parameters L and p . To do so, we consider the probability of successful pathfinding $P_{pf}(p, W)$ on instances of cubic lattice \mathcal{L}_t with dimensions $1000 \times L \times L$ over a range of p and W . Figure 6 depicts such a simulation for $L = 7$.

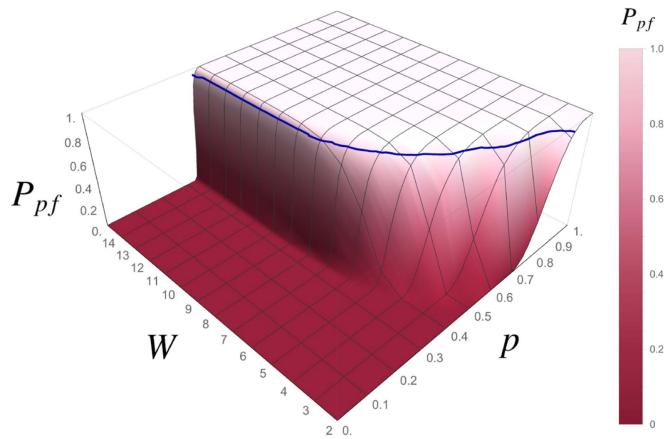


Figure 6. Limited-lookahead pathfinding success probability $P_{pf}(p, W)$ for $1000 \times 7 \times 7$ cubic lattices found over a range of window sizes W and edge probabilities p . These results clearly depict the combination of both long-range block percolation phenomena and the effect of a limited lookahead on pathfinding. Firstly, a clear percolation threshold is observed at $p_{\min} \approx 0.4$, as predicted (by numeric simulations for L_{\min} in figure 2). Secondly, the detrimental effect of a limited lookahead on pathfinding for window sizes $W < 10$ is also observed. This shows that for $p = p_{\min}$, a maximum window length $W_{\max}(L, p_{\min})$ exists, below which pathfinding can only be achieved by a complementary increase in p . The region of successful long-range pathfinding ($P_{pf}(p, W) \geq 0.95$) is found above the highlighted blue contour.

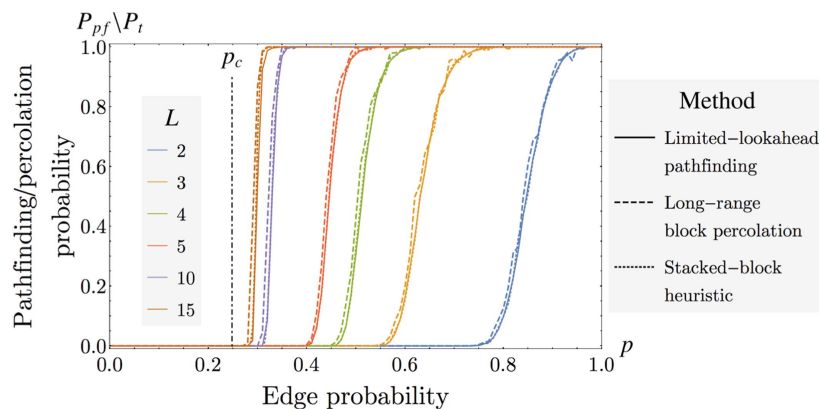
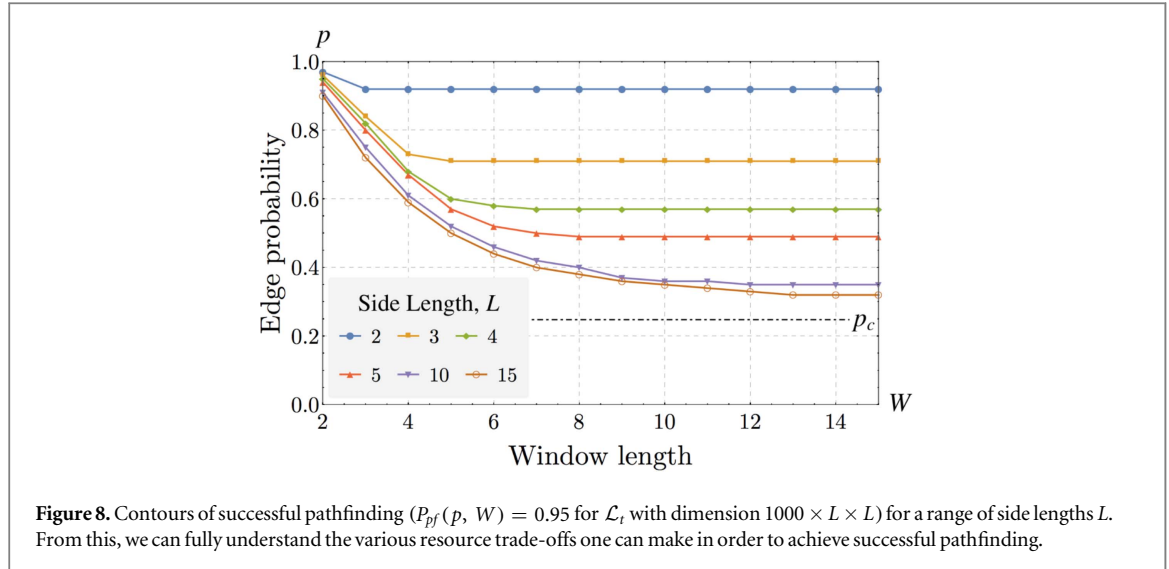


Figure 7. Comparison between the thresholds in LLP success rates P_{pf} and block percolation P_t on instances $1000 \times L \times L$ cubic lattices over a range of L . For pathfinding, depicted by solids lines, a large window size (of $W = 15 \geq W_{\max}(L)$) was chosen to ensure the thresholds found were due to percolation effects, rather than pathfinding’s limited lookahead. By comparison with long-range block percolation, depicted by dashed lines, we can see that $P_{pf} \approx P_t$, confirming that for sufficiently large window lengths, LLP is equivalent to long-range block percolation. Furthermore, we find that within this regime both long-range block percolation and LLP can be approximated as multiple stacked instances of $(15 \times L \times L)$ active block percolation, such that $P_{pf}(p, L_t, 15) \approx P_t(p, L_t) \approx P_t(p, \mathcal{B}_{0,15})^{\frac{1000}{15}}$, as depicted by the dotted lines. Given that simulating LLP is computationally expensive, this *stacked-block* heuristic provides a quick and inexpensive approximation for investigating the performance of LLP on other percolated lattices for LOQC.

The first and most striking feature of these results is the sharp threshold at $p \approx 0.4$ for large W . This clearly identifies the minimum edge probability $p_{\min}(L = 7)$ below which no long-range percolation occurs, and agrees with numerical L_{\min} results depicted in figure 2, showing that $p_{\min}(L = 7) \approx 0.4$. From the argument made in section 4.2, we expect this pathfinding threshold to recreate the standard block percolation threshold of a $1000 \times L \times L$ cubic lattice. We confirm this numerically with figure 7, which depicts LLP and block percolation thresholds found over a range of L , showing LLP reproducing long-range block percolation statistics. Furthermore, we find that percolation statistics found for active blocks can be used to estimate pathfinding performance over long distances. In this simplified *stacked-block* heuristic, we model long-range LLP as $1000/W$ consecutive instances of block percolation, as if adjacently stacked face-to-face in t to form the full block \mathcal{L}_t (without requiring two adjacent blocks’ percolation paths are connected at adjacent faces), such that $P_{pf}(p, W) \approx P_t(p, \mathcal{B}_{0,W})^{\frac{1000}{W}}$. Figure 7 shows that even for large L , this heuristic provides a good estimate for $P_{pf}(p, W)$ and $P_t(p)$ (when $W \geq W_{\max}$).

The second feature we observe is the effect of small window lengths upon pathfinding. For $p = p_{\min}$, we observe a maximum window length $W_{\max}(L, p_{\min}) \approx 10$. As conjectured, $W > W_{\max}$ provides no additional



benefit to pathfinding, whereas for $W < W_{\max}$, the probability of successful LLP is significantly reduced (for fixed p). While it is possible to realize successful pathfinding for $W < W_{\max}(L, p_{\min})$, this can only be achieved by a complementary increase in p .

To fully understand the parameter space for successful pathfinding, we consider contours of $P_{pf} = 0.95$ in p and W for $L = 2, 3, 4, 5, 10$ and 15 , depicted in figure 8. From these results we can also incorporate the effects of L into our previous analysis. As identified by the results of figure 2, an increase in L reduces the minimum edge probability p_{\min} at which long-range percolation occurs, and hence the value of p_{\min} which LLP can succeed. However, whilst an increase in L (for a fixed W) always decreases the required p for successful pathfinding, these gains are most significant when W is also increased, allowing the new $p_{\min}(L)$ to be achieved. Such insights provide us with far greater clarity into the inherent resource trade-offs in a LOQC device.

Finally, we note that even for the largest active blocks considered, $p_{\min}(L = W = 15)$ had yet to approach p_c . This indicates that successful pathfinding is likely to require a lattice with edge probability greater than p_c by some non-insignificant amount. Furthermore, when more sophisticated and computationally expensive pathfinding strategies were simulated, they did not reduce $W_{\max}(L, p_{\min})$, only improving pathfinding in the region of $p > p_{\min}$ and $W < W_{\max}(L, p_{\min})$. For further details on the performance of such strategies, see the Supplementary Materials.

5. Implications for LOQC architectures

Using the results presented in section 4.3 additional clarity can now be given to the resource trade-offs inherent to a realistic LOQC device.

Firstly, generating a lattice with $p > p_c$ is necessary for the reduction of active block size. For p close to p_c , small increases in p will lead to significant resource savings in block size. The success rate of LOQC's boosted fusion gates¹⁰ p_f can be increased from 50% to 75% through the consumption of either a Bell state or four single photons per gate [17, 18]. However, above this first level of boosting, gains in p_f become more marginal at the expense of increasingly costly resource states (which cannot be produced deterministically using linear optics without significant resource overheads). This leads us to believe that it is likely that LOQC will utilize boosted fusion of at least $p_f = 75\%$, from which a choice of active block dimensions, W and L , can be made accordingly. We note that in [10], it was shown that $p_f = 75\%$ greatly exceeds the percolation threshold of $p_c = 62.5\%$. In practise, experimental fusion gate success rates will be reduced by error mechanisms, such as photon loss. However, if this reduction can be sufficiently minimized, our results indicate that small active block sizes can be achieved, thereby reducing overall resource requirements for LOQC.

Secondly, the probability of successful pathfinding affects the accommodation of bond/qubit¹¹ loss for a renormalized lattice. From the perspective of the lattice renormalization, a failure in pathfinding simply represents a missing bond/qubit along the time axis. Thus the quantum error correction (QEC) protocol's ability to deal with bond/qubit loss on the renormalized lattice explicitly determines the required P_{pf} (which adds to all other loss mechanisms). For example, consider the pathfinding requirements for a linear cluster of

¹⁰ Note $p \neq p_f$, as in current proposals multiple fusion operations must succeed for a given edge to be created in the target lattice. Furthermore, failure modes of boosted fusion gates can also maintain connectivity, producing additional connectivity outside the standard percolation model.

¹¹ If a block lacks connectivity to be successfully renormalized, one can choose to represent this either as the loss of individual bonds or an entire qubit.

100 renormalized qubits, with each renormalized block being 10 layers long, such that the dimensions of \mathcal{L}_t are $1000 \times L \times L$. If less than one bond/qubit must be lost per string of 100 renormalized qubits, then we require $P_{pf}(\mathcal{L}_t, W) > 0.99$. However, if more bond/qubit loss can be accommodated, this reduces the required pathfinding probability, thus allowing for a further reduction in L or W .

Finally, we expect the identified resource costs and trade-offs to be somewhat sensitive to our chosen value of P_{pf} , and would expect a reduction in size of the successful long-range pathfinding parameter space (L, W, p) if it were increased (say to 0.99). However, we further expect that the effect of such a difference would be very small and furthermore would decrease¹² as $P_{pf} \rightarrow 1$, and therefore our presented results provide an accurate description of the relevant limited-lookahead phenomenon.

6. Open questions

There are other architectural necessities that must be incorporated to produce a complete model. In this work pathfinding is only considered within the context of producing a single-qubit channel, but in order to produce a renormalized lattice for QEC percolated paths must also be found in y and z . While a renormalization algorithm with optimal scaling is known for 2D [20], none are known for higher-dimension lattices. Additionally, for a realistic device, local pathfinding algorithms must also be designed to reduce the associated computational overheads for finding percolated paths in both y and z (for example, similar to recently proposed cellular automata decoders for QEC [31]).

Also, we do not consider the effects of experimental errors on our pathfinding strategy. It is known that one of the most significant challenges for LOQC is photon loss. The teleportation of quantum information via MBQC in our model assumes that each photon is measured successfully. However, in a physical device some degree of both heralded and unheralded photon loss will undoubtedly occur from active components and memory delay lines. For heralded qubit loss occurring in the lattice generation stage, it is known that the affected qubit's neighbours can be removed from the lattice. With this approach, it was shown in [10] that a loss rate up to 1.5% could be tolerated by the diamond brickwork lattice (with $p_f = 75\%$). Given that for $W \geq W_{\max}(L, p)$ we recover standard percolation statistics, we therefore expect a similar loss tolerance for our pathfinding model. But for an unheralded qubit loss it is not yet known whether it is possible to perform MBQC without an explicit loss-tolerant encoding (such as presented in [32]), especially under the realistic restriction of a fixed order of qubit measurement.

We additionally note that in the context of a LOQC architecture, our approach here is far from optimal. For example, our pathfinding algorithm only considers a single path per qubit channel at any one time. However, for $p \gg p_c$ the number of percolation paths spanning one axis of a $L \times L \times L$ block scales as $\mathcal{O}(L)$, compared to $\mathcal{O}(1)$ for $p > p_c$ close to p_c [33]. It may therefore be possible to utilize these extra paths as *backup* paths to insure against both unheralded photon loss and unforeseen dead ends. This may have the combined effect of both reducing $W_{\min}(L, p)$ and providing loss tolerance, without resulting in an increased susceptibility to accrued Pauli errors (from increased MBQC measurements per single-qubit channel).

Lastly, it remains to extend such pathfinding simulations to candidate lattices for percolated LOQC cluster states. Due to the amorphism, anisotropy and correlations of bond percolation applied to the brickwork diamond lattice presented in [10], a direct mapping of resource costs cannot be made from our results. However, preliminary simulations have shown comparable effects as presented here, suggesting that the presented LLP phenomenon is general to many lattice configurations [34]. Nevertheless, it remains to identify the specific impact of deviations from the standard percolation model as such lattices must also permit resource-efficient LLP in order to be utilized within an LOQC architecture.

7. Conclusions and outlook

Realistic architectures for LOQC must consider the physical constraints of a large-scale device, such as a finite and fixed depth. As such, this work has considered the effect of a finite fixed depth on the creation of a single-qubit channel from a percolated cluster state lattice. We have shown that within this model, a limited-lookahead pathfinding algorithm can be applied to successfully create such a channel and identified resources requirements for successful pathfinding. This suggests that an LOQC architecture with a computational window of $\mathcal{O}(10)$ layers (i.e. clock-cycles of photon production) is sufficient to produce the almost indefinitely large states required for universal quantum computation. However, we also find that these constraints many require percolation-based LOQC architectures to operate above previously-identified minimum resource estimates.

¹² This can be understood by consideration of figure 6. Here we can observe that an increase of P_{pf} from 0.95 to 0.99 only provides a small contraction of the space outlined by the highlighted contour, a difference which clearly decreases as $P_{pf} \rightarrow 1$.

Notably, we find that resource requirements become significant as the cluster state lattice's edge probability approaches its critical threshold. However, this equally implies that even small increases in edge probability (close above the percolation threshold) can provide significant resource savings and allow an LOQC device to operate with surprisingly low fixed depth.

An additional key result of this work is a significant step towards bridging the gap between high- and low-level architectural requirements. When applied to a specific LOQC architectural schema, the model presented here allows direct mapping of high-level architectural resource requirements (such as a qubit channel loss rates) onto low-level device requirements (such as device depth and ancillae resource counts). Once identified, this mapping allows the device's fixed finite depth to be effectively ignored allowing the high-level abstractions required for studying the high-level architecture, such as QEC protocols. Furthermore, by identifying LLP simulation heuristics, the performance of novel candidate lattices for LOQC can be quickly and easily analysed without extensive LLP simulations—a key advantage as architectural models become increasingly sophisticated.

Acknowledgments

The authors thank the anonymous referee for suggesting an additional pathfinding strategy included in the Supplementary Materials (available at stacks.iop.org/qst/3/015005). This work was supported by the UK Engineering and Physical Sciences Research Council (EPSRC). SMS is supported by the Bristol Quantum Engineering Centre for Doctoral Training, EPSRC grant EP/L015730/1. MGS acknowledges support from the University of Bristol. This work was carried out using the computational facilities of the Advanced Computing Research Centre, University of Bristol. All simulation and analysis scripts and datasets presented here are available for download from the Research Data Repository of University of Bristol at data.bris.ac.uk/data/dataset/2wmj58va0tejt23ojmkxk4nu.

ORCID iDs

Sam Morley-Short  <https://orcid.org/0000-0002-4445-734X>

References

- [1] Rudolph T 2017 Why I am optimistic about the silicon-photonics route to quantum computing *APL Photonics* **2** 030901
- [2] Knill E, Laflamme R and Milburn G J 2001 A scheme for efficient quantum computation with linear optics *Nature* **409** 46–52
- [3] Kok P *et al* 2007 Linear optical quantum computing with photonic qubits *Rev. Mod. Phys.* **79** 135–74
- [4] Nielsen M A 2004 Optical quantum computation using cluster states *Phys. Rev. Lett.* **93** 040503
- [5] Browne D E and Rudolph T 2005 Resource-efficient linear optical quantum computation *Phys. Rev. Lett.* **95** 010501
- [6] Varnava M, Browne D E and Rudolph T 2008 How good must single photon sources and detectors be for efficient linear optical quantum computation? *Phys. Rev. Lett.* **100** 060502
- [7] Campbell E T and Fitzsimons J 2009 An introduction to one-way quantum computing in distributed architectures *Int. J. Quantum Inf.* **08** 219–58
- [8] Kieling K, Rudolph T and Eisert J 2007 Percolation, renormalization, and quantum computing with nondeterministic gates *Phys. Rev. Lett.* **99** 130501
- [9] Zaidi H A, Dawson C, van Loock P and Rudolph T 2015 Near-deterministic creation of universal cluster states with probabilistic Bell measurements and three-qubit resource states *Phys. Rev. A* **91** 042301
- [10] Gimeno-Segovia M, Shadbolt P, Browne D E and Rudolph T 2015 From three-photon Greenberger–Horne–Zeilinger states to ballistic universal quantum computation *Phys. Rev. Lett.* **115** 020502
- [11] Li Y, Humphreys P C, Mendoza G J and Benjamin S C 2015 Resource costs for fault-tolerant linear optical quantum computing *Phys. Rev. X* **5** 041007
- [12] Rohde P P 2015 Simple scheme for universal linear-optics quantum computing with constant experimental complexity using fiber loops *Phys. Rev. A* **91** 012306
- [13] Uskov D B *et al* 2015 Resource-efficient generation of linear cluster states by linear optics with postselection *J. Phys. B: At. Mol. Opt. Phys.* **48** 045502
- [14] Gimeno-Segovia M *et al* 2017 Relative multiplexing for minimising switching in linear-optical quantum computing *New J. Phys.* **19** 063013
- [15] Raussendorf R and Briegel H J 2001 A one-way quantum computer *Phys. Rev. Lett.* **86** 5188–91
- [16] Raussendorf R, Browne D E and Briegel H J 2003 Measurement-based quantum computation on cluster states *Phys. Rev. A* **68** 022312
- [17] Grice W P 2011 Arbitrarily complete Bell-state measurement using only linear optical elements *Phys. Rev. A* **84** 042331
- [18] Ewert F and van Loock P 2014 3/4-efficient Bell measurement with passive linear optics and unentangled ancillae *Phys. Rev. Lett.* **113** 140403
- [19] Pant M, Towsley D, Englund D and Guha S 2017 Percolation thresholds for photonic quantum computing arXiv:1701.03775
- [20] Browne D E *et al* 2008 Phase transition of computational power in the resource states for one-way quantum computation *New J. Phys.* **10** 023010
- [21] Silverstone J W, Bonneau D, O'Brien J L and Thompson M G 2016 Silicon quantum photonics *IEEE J. Sel. Top. Quantum Electron.* **22** 390–402
- [22] Gottesman D 1997 Stabilizer codes and quantum error correction *PhD Thesis California Institute of Technology* arXiv:quant-ph/9705052

- [23] Hein M, Eisert J and Briegel H J 2004 Multiparty entanglement in graph states *Phys. Rev. A* **69** 062311
- [24] Hoshen J and Kopelman R 1976 Percolation and cluster distribution: I. Cluster multiple labeling technique and critical concentration algorithm *Phys. Rev. B* **14** 3438–45
- [25] Grimmett G R 1999 *Percolation* (Berlin: Springer)
- [26] Stauffer D and Aharony A 1994 *Introduction to Percolation Theory* (London: Taylor and Francis)
- [27] Newman M E J and Ziff R M 2001 Fast Monte Carlo algorithm for site or bond percolation *Phys. Rev. E* **64** 016706
- [28] Stace T M, Barrett S D and Doherty A C 2009 Thresholds for topological codes in the presence of loss *Phys. Rev. Lett.* **102** 1–4
- [29] Wang J, Zhou Z, Zhang W, Garoni T M and Deng Y 2013 Bond and site percolation in three dimensions *Phys. Rev. E* **87** 052107
- [30] Dijkstra E W 1959 A note on two problems in connexion with graphs *Numer. Math.* **1** 269–71
- [31] Herold M, Campbell E T, Eisert J and Kastoryano M J 2015 Cellular-automaton decoders for topological quantum memories *npj Quantum Inf.* **1** 15010
- [32] Varnava M, Browne D E and Rudolph T 2005 Loss tolerance in one-way quantum computation via counterfactual error correction *Phys. Rev. Lett.* **97** 120501
- [33] Franceschetti M, Dousse O, Tse D N C and Thiran P 2007 Closing the gap in the capacity of wireless networks via percolation theory *IEEE Trans. Inf. Theory* **53** 1009–18
- [34] Morley-Short S *PhD Thesis* University of Bristol in preparation