Graph state data structures

Liming Zhao,[1] Carlos A. Pérez-Delgado,[1] and Joseph F. Fitzsimons[1,2]

[1]*Singapore University of Technology and Design, 20 Dover Drive, Singapore 138682*
[2]*Centre for Quantum Technologies, National University of Singapore, 3 Science Drive 2, Singapore 117543*

Introduction and motivation

The discovery of a polynomial time quantum algorithm for factoring integers [1] sparked a great interest in the study and implementation of quantum computation, and quantum computers. Despite this interest, the number of distinct quantum algorithms has grown relatively slowly. Moreover, the algorithms that do exist are based on an even smaller number of distinct techniques. One feature common to all of these algorithms, however, is while they perform information processing in a quantum manner, the data structures they exploit are essentially classical.

The question then arises as to whether it is possible to develop an intrinsically quantum data structure; one that is not *merely* a superposition of classical data. Here, we answer this in the affirmative. We develop a fully *quantum data structure*, that can represent graphs, based on quantum graph states. Graph states have been studied heavily before, but generally in the context of use as a computational resource for one-way quantum computation[2–7]. Here, we study them as a method to represent graphs. As a data-structure, graph states have no classical analogue: they are not simply a quantum superposition of classical states representing graphs.

Main results

We develop a quantum data structure for representing graphs, based on quantum graph states. We show how to construct the data structure from a classical representation of the graph, how to do basic operations and comparisons on the quantum data structure, and how to read out the contents of the quantum data structure onto a classical representation.

Elementary operations

First, we show how several two-qubit gates change the graph represented by a state in a deterministic manner. Since all quantum operations must be reversible, it is not possible to simply erase or add an edge, but rather operations must *complement* the existence of an edge between two vertices. That is, the complementation operator adds an edge where there wasn't one, and deletes an existing edge.

By applying a single two-qubit operator it is possible to achieve several kinds of edge complementation:

1 Complementation of the edge between vertex A and vertex B.

2 Complementation of edges between vertex A and neighbours of vertex B.

3 Complementation of edges between neighbours of vertex A and neighbours of vertex B.

We show that there are 64 distinct kinds of complementation associated with the choice of a pair of vertices. Some of these operations violate linearity or reversibility, and hence are prohibited in the graph state representation. We fully characterize this set of operations: when an operation is possible we show how to implement it, and when it is impossible we provide a no-go proof.

Constructing graph states

While the construction of graph states has been extensively studied [8–11], many of the previous approaches have not focused on the circuit optimality, but rather on circumventing limitations of certain physical systems. We proposed a method based on two qubit-gates such as Control-Not, Control-Y, Control-Z and $(H \otimes H) \cdot CZ \cdot (H \otimes H)$ gates and greedy algorithm to construct a graph state if we already know the adjacent matrix of a graph.

Graph state data structures

During the construction of a graph-state data structure, one has access to a classical data structure that can be used to accelerate some operations. However, to have an effective data structure, one must be able to operate on the graph states without the need for access to ancillary classical information. Our approach allows for a number of common operations to be performed extremely efficiently by acting on the graph state alone. These include:

1. Adding a vertices, which requires $O(1)$ operations.
2. Duplicating a vertices, which requires $O(1)$ operations.
3. Complementing the neighbourhood of one vertex with another vertex, which requires $O(1)$ operations.
4. Comparison of vertices and subgraphs of constant size in $O(1)$.
5. Comparison of arbitrary subgraphs with $O(n)$ operations.

Finally, we present an efficient algorithm to convert a polynomial number of copies of a given graph state into a classical representation of the graph.

Significance

We develop what is, to the best of our knowledge, one of the first intrinsically quantum data structures. We showed that this data structure allows for some operations to be performed more efficiently than classical data structures. For instance, we get he running time of duplicating a vertex by applying Control-Not gate is $O(1)$. On the other hand, for common classical structures such as an adjacency matrix or an edge list, the best classical algorithm is $O(N)$. These results suggest that some graph problems may be solved faster by using graph states as an intermediate data structure than by using classical data structures.

---

[1] P. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

[2] M. Hein, J. Eisert, and H. J. Briegel. Multiparty entanglement in graph states. *Phys. Rev. A*, 69:062311, Jun 2004.

[3] Daniel Gottesman. Class of quantum error-correcting codes saturating the quantum hamming bound. *Phys. Rev. A*, 54:1862–1868, Sep 1996.

[4] Shiang Yong Looi, Li Yu, Vlad Gheorghiu, and Robert B. Griffiths. Quantum-error-correcting codes using qudit graph states. *Phys. Rev. A*, 78:042303, Oct 2008.

[5] Simon Anders and Hans J. Briegel. Fast simulation of stabilizer circuits using a graph-state representation. *Phys. Rev. A*, 73:022334, Feb 2006.

[6] Robert Raussendorf and Hans J. Briegel. A one-way quantum computer. *Phys. Rev. Lett.*, 86:5188–5191, May 2001.

[7] Dan E Browne and Hans J Briegel. One-way quantum computation-a tutorial introduction. *arXiv preprint quant-ph/0603226*, 2006.

[8] Peter Hyer, Mehdi Mhalla, and Simon Perdrix. Resources required for preparing graph states. In Tetsuo Asano, editor, *Algorithms and Computation*, volume 4288 of *Lecture Notes in Computer Science*, pages 638–649. Springer Berlin Heidelberg, 2006.

[9] Marc Hein, Wolfgang Dür, Jens Eisert, Robert Raussendorf, M Nest, and H-J Briegel. Entanglement in graph states and its applications. *arXiv preprint quant-ph/0602096*, 2006.

[10] L.-M. Duan and R. Raussendorf. Efficient quantum computation with probabilistic quantum gates. *Phys. Rev. Lett.*, 95:080503, Aug 2005.

[11] Michael A. Nielsen. Optical quantum computation using cluster states. *Phys. Rev. Lett.*, 93:040503, Jul 2004.