



Moylett, D., Linden, N., & Montanaro, A. (2017). *Quantum speedup of the Travelling Salesman Problem for bounded-degree graphs*. Poster session presented at Heilbronn Quantum Algorithms Day 2017, Bristol, United Kingdom.

Publisher's PDF, also known as Version of record

[Link to publication record on the Bristol Research Portal](#)  
PDF-document

This is the final presented version of the poster.

## University of Bristol – Bristol Research Portal

### 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/red/research-policy/pure/user-guides/brp-terms/>

# Quantum speedup of the Travelling Salesman Problem for bounded-degree graphs

Dominic J. Moylett<sup>1,2,3</sup>, Noah Linden<sup>4</sup> and Ashley Montanaro<sup>4</sup>

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

<sup>2</sup>Quantum Engineering Centre for Doctoral Training, H. H. Wills Physics Laboratory and Department of Electrical & Electronic Engineering, University of Bristol, BS8 1FD, United Kingdom

<sup>3</sup>Heilbronn Institute for Mathematical Research, University of Bristol, BS8 1SN, United Kingdom

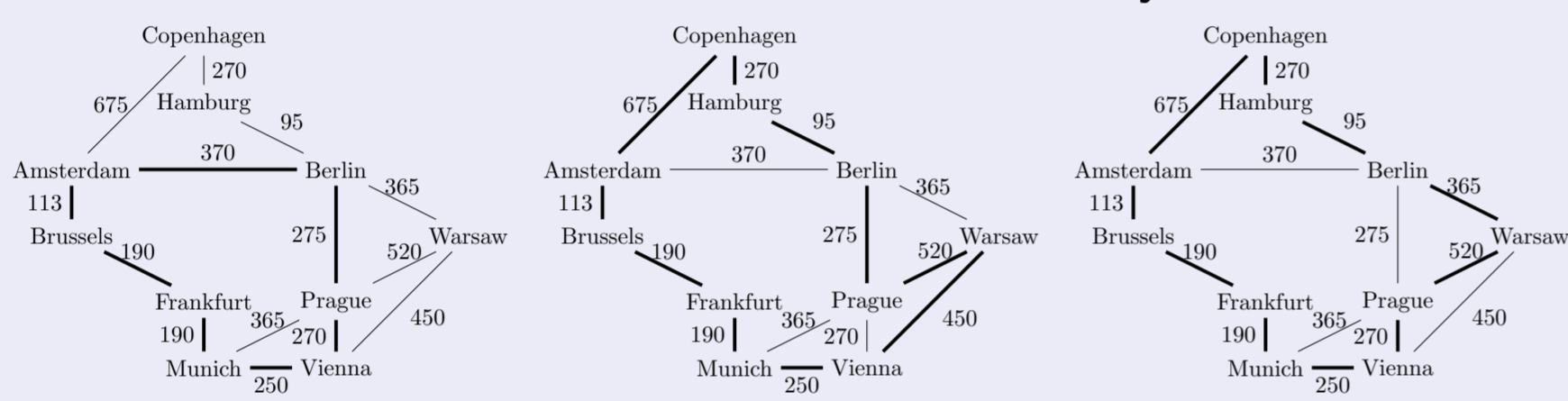
<sup>4</sup>School of Mathematics, University of Bristol, BS8 1TW, United Kingdom

## Main Results

- ▶ We show near-quadratic speedups for the Travelling Salesman Problem (TSP) when the degree of any vertex is at most 4.
- ▶ This is through applying a quantum speedup for backtracking [Mon15] to two TSP algorithms [XN16a,XN16b].
- ▶ We then demonstrate polynomial speedups up to degree-6.
- ▶ See Physical Review A 95(3), 032323 (2017) [arXiv:1612.06203] for further details.

## 1. The Travelling Salesman Problem

- ▶ Let  $G$  be a graph with  $n$  vertices and  $m$  edges.
- ▶ A cycle  $H$  on  $G$  is *Hamiltonian* if it visits every vertex in  $G$ .
- ▶ The TSP is to find the shortest Hamiltonian cycle.



(a) ✗ Not a Hamiltonian cycle. (b) ✗ Not the shortest Hamiltonian cycle. (c) ✓ Solution to the TSP. (Length: 2938 Minutes)

- ▶ The best general classical algorithms take exponential time in  $n$ .

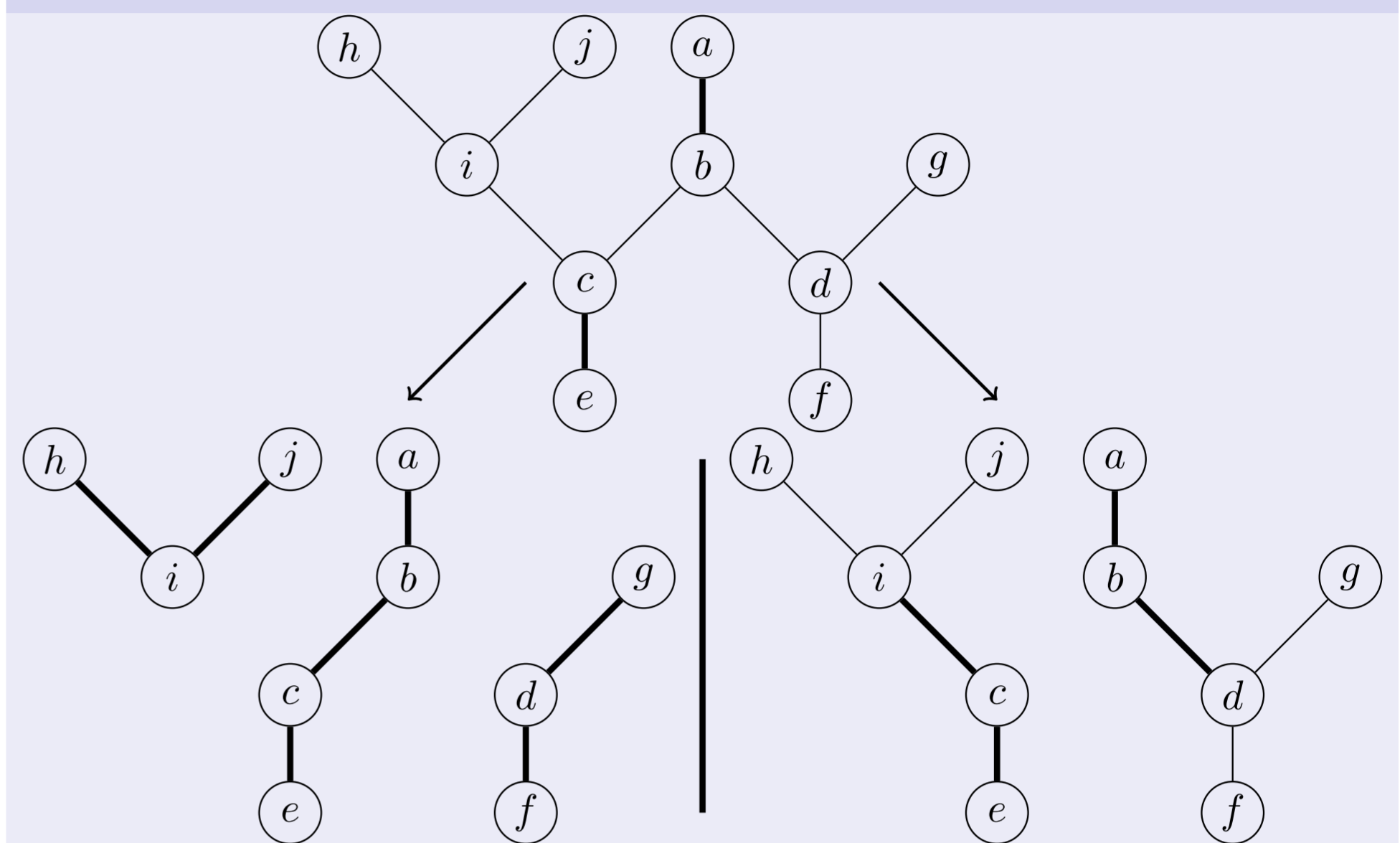
## 2. Backtracking algorithms

- ▶ Backtracking algorithms are a way of solving constraint satisfaction problems.
- ▶ They have two parts:
  1. A predicate, which checks if the constraints are satisfiable;
  2. and a heuristic, which chooses the next variable to assign.
- ▶ When called with a partial assignment, the predicate checks if the constraints are satisfiable. If not, we return.
- ▶ Otherwise, the heuristic picks a variable which we assign a value to and recursively call ourselves with this new partial assignment.
- ▶ Montanaro [Mon15] developed a quantum backtracking algorithm which has a quadratic speedup for finding a solution.

## 3. Quantum speedup for degree-3 graphs

- ▶ Backtracking algorithms can solve the TSP with “forced” edges which we must travel down and “removed” edges which we must avoid travelling on.
- ▶ The predicate checks if a Hamiltonian cycle is possible, and the heuristic selects another edge to force or remove.
- ▶ The best backtracking algorithm on degree-3 graphs runs in  $O^*(2^{3n/10})$  time and polynomial space [XN16a].
- ▶ We apply [Mon15] to this algorithm to find a Hamiltonian cycle, failing to find one when one exists with probability  $\delta$ , in  $O^*(2^{3n/20} \log(1/\delta))$  time, where  $O^*$  hides polynomial factors.
- ▶ We find the shortest Hamiltonian cycle with bounded error (finding a sub-optimal cycle or no cycle) via binary search with  $O(\log L \log \log L)$  overhead, where  $L$  is the longest edge length.

## 4. Example backtracking step



- ▶ Forcing  $bc$ , as shown on the left, means that  $b$  and  $c$  are incident to two forced edges, so  $ci$  and  $bd$  are removed. Now  $d$  and  $i$  are of degree 2, so edges  $df$ ,  $dg$ ,  $hi$  and  $ij$  are forced.
- ▶ Removing  $bc$ , as shown on the right, means that  $b$  and  $c$  are of degree 2, so edges  $bd$  and  $ci$  are now forced.

## 5. Expanding to higher-degree graphs

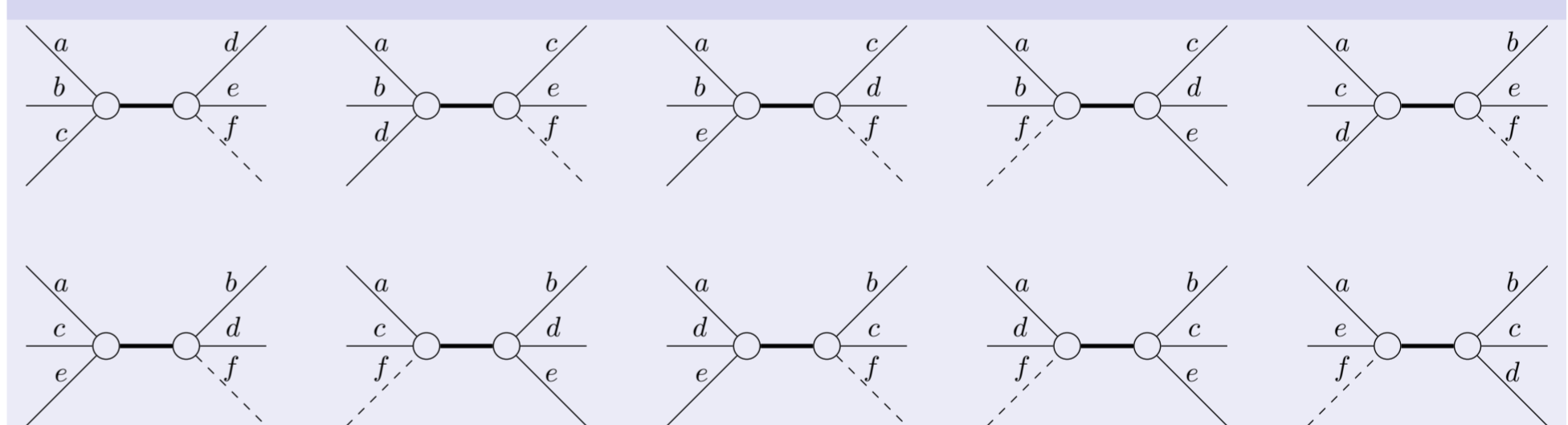


Figure: Ways of splitting a vertex of degree 5 or 6 into two lower-degree vertices.

- ▶ For degree-4 graphs, we apply the same technique to the algorithm of [XN16b] in  $O^*(1.301^n \log L \log \log L)$  time.
- ▶ Other speedups can be found by breaking higher-degree vertices into degree 4 vertices connected by forced edges.
- ▶ We find the shortest way of splitting each vertex via [DH99].
- ▶ For degree-5/6 graphs, there are 10 ways of splitting each vertex, of which 6 will preserve the shortest Hamiltonian cycle. Thus we get an additional  $O((10/6)^{n/2})$  overhead.
- ▶ For degree-7 graphs, this method is slower than classical algorithms for the general TSP [HK62, Bjö14].

## References

- [Bjö14] A. Björklund, SIAM Journal on Computing, 43(1):280–299 (2014)
- [DH99] C. Dürr and P. Høyer, arXiv:quant-ph/9607014 (1999)
- [HK62] M. Held and R. Karp, Journal of the Society for Industrial and Applied Mathematics, 10(1):196–210, (1962)
- [Mon15] A. Montanaro, arXiv:1509.02374 (2015)
- [XN16a] M. Xiao and H. Nagamochi, Algorithmica 74(2):713–741, (2016)
- [XN16b] M. Xiao and H. Nagamochi, Theory of Computing Systems, 58(2):241–272, (2016)