Towards a Quantum Speedup via Applications and Architectures

Alexandra Emily Moylett

A dissertation submitted to the University of Bristol in accordance with the requirements for award of the degree of Doctor of Philosophy in the Faculty of Science.

School of Physics
University of Bristol

May 2020

45,000 words

Abstract

Since their original proposal in the 70s and 80s, quantum computers have evolved from an interesting theoretical concept to a physically realisable technology. This has been particularly exemplified with a recent publication by Arute et al. (*Nature*, 574(7779):505–510, 2019), which has demonstrated a quantum computer solving a problem that is believed to be classically hard. While this is much cause for celebration and interest, the work towards showing what a quantum computer can really do is still yet to come. Arute et al.'s result shows a quantum computer solving a hard problem, but not a useful one.

This is the question we push towards in this thesis: What problem with real-world applications can a quantum computer solve faster than a classical computer? We make contributions towards solving this problem in two ways:

First, an applications-focused approach: We show how a quantum computer can solve the Travelling Salesman Problem on bounded-degree graphs polynomially faster. This is achieved through applying a quantum speedup for Backtracking algorithms to classical algorithms for solving the Travelling Salesman Problem when the degree of the graph is at most 3 or 4. We then obtain further polynomial speedups when the degree of the graph is at most 6, by a combination of reducing to the degree-4 case and quantum minimum finding.

Second, an architecture-focused approach: We consider how photon distinguishability and loss affect the near-term quantum architecture known as Boson Sampling. In doing so, we provide a way of mathematically modelling these imperfections as decoherence in a quantum circuit, via representation theory in first quantisation. We then show how current classical simulation algorithms can be sped up by taking advantage of these imperfections, and suggest what photonic regimes our simulator provides better performance for.

Declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: Alexandra Emily Moylett DATE: Thursday 14th May, 2020

Contents

I Overview 1

1 Introduction 3
1.1 The (eventual) power of quantum computing 4
1.2 What quantum computers can achieve in the near term 4
1.3 Our contributions 6
1.4 A brief note on complexity theory 8

II An application-focused approach: quantum algorithms for the Travelling Salesman Problem 11

2 Preliminary material: Quantum algorithms for NP-Hard problems 13
2.1 P vs NP and NP-Hard problems 14
2.2 Classical algorithms for NP-Hard Problems 16
2.3 Quantum speedups 18

3 Quantum speedup of the Travelling Salesman Problem for bounded-degree graphs 25
3.1 Introduction 25
3.2 Backtracking algorithms for the TSP 27
3.3 Quantum speedups for the Travelling Salesman Problem on bounded-degree graphs 28
3.4 Extending to higher-degree graphs 38
3.5 Conclusion 41

III An architecture-focused approach: Boson Sampling under distinguishability and loss 45

4 Preliminary material: Boson Sampling and the Schur-Weyl Duality 47
4.1 Estimated speedups in practice and the limitations of quantum error correction . 48
4.2 The search for a quantum advantage 49
4.3 Linear optics and Boson Sampling 53
4.4 Experimental achievements 63
4.5 Experimental imperfections in linear optics 64
4.6 Classical simulation algorithms 71
4.7 The Schur-Weyl Duality 74

5 Quantum simulation of partially distinguishable Boson Sampling 83
5.1 Introduction . . . 83
5.2 A quantum circuit for ideal bosonic sampling . . . 84
5.3 A quantum circuit for arbitrarily distinguishable bosonic sampling . . . 87
5.4 Unitary-Unitary Duality . . . 91
5.5 Sampling with loss . . . 93
5.6 Distinguishability and simulateability . . . 95
5.7 Conclusion . . . 99

6 Classically simulating near-term partially-distinguishable and lossy Boson Sampling 101
6.1 Introduction . . . 101
6.2 Limitations of other classical simulation algorithms . . . 102
6.3 Expanding in terms of states . . . 107
6.4 Average case error . . . 110
6.5 Worst case error . . . 113
6.6 Incorporating loss . . . 114
6.7 Empirical errors . . . 116
6.8 Non-uniform loss . . . 120
6.9 Discussion of possible improvements . . . 121
6.10 Conclusion . . . 126

IV Conclusions and open questions 129

7 Conclusion 131
7.1 Open questions . . . 131
7.2 The future . . . 133

List of Figures

2.1 Example backtracking trees 20

3.1 An instance of the recursive step in Eppstein's algorithm 28
3.2 An example of the circuit procedure and branching on a circuit 30
3.3 Some example simplifications used in the Xiao-Nagamochi algorithm 31
3.4 Breaking a vertex of degree 5 or 6 into two lower-degree vertices 39

4.1 Experimental setup of the Mach-Zehnder Interferometer and probability of outcomes over phase θ 56
4.2 Experimental setup of the Hong-Ou-Mandel Dip 57
4.3 Two examples of universal linear-optical interferometers 61
4.4 The probability of seeing the coincident $|1, 1\rangle$ output from a Hong-Ou-Mandel experiment over $\Delta\tau$. 66

5.1 A quantum circuit for Algorithm 5.1 85
5.2 The quantum circuit described in Algorithm 5.2 89
5.3 Circuit diagram illustrating how postselection can be used to 'filter out' distinguishability 93
5.4 Circuit for sampling when k bosons are lost 94

6.1 Highest value of x and η simulable via state or point truncation 117
6.2 Approximate runtime to simulate n-photon Boson Sampling with chosen values of η and x via state or point truncation 119
6.3 Highest value of x and $\eta = 1$ simulable for 90-photon boson sampling at truncation level k 120
6.4 Exact trace distance between $\rho_{n,x}$ and $\rho_{\leq k,x}$ over x for $k < n, n = 2, \ldots, 7$ 122
6.5 Probability of different irreps when $n = 2$ and $n = 3$. 125

List of Tables

3.1 Runtimes of our quantum algorithms for the Travelling Salesman Problem 26
3.2 Runtimes of the quantum speedup of Björklund et al. 42

6.1 Exact equations for $\delta_{\mathrm{tr}}(\rho_{n,x}, \rho{\leq} k, x)$ for $k < n, n = 2, \ldots, 4$ 123

List of Algorithms

2.1 A backtracking algorithm for SAT. 17

3.1 The Xiao-Nagamochi algorithm for solving the TSP on degree-3 graphs. 33
3.2 The predicate function for the Xiao-Nagamochi algorithm for degree-3 graphs. . . 34
3.3 The heuristic function for the Xiao-Nagamochi algorithm for degree-3 graphs. . . 35
3.4 The reduction function for degree-3 graphs. 37

5.1 A quantum circuit for sampling from the same distribution as that produced by indistinguishable bosons in a linear interferometer. 85
5.2 A quantum circuit for sampling from (essentially) the same distribution at that produced by distinguishable bosons in a linear interferometer 88

List of Abbreviations and Acronyms

CSP	:	Constraint Satisfaction Problem
HHL	:	Harrow, Hassidim and Lloyd Algorithm
HOM Dip	:	Hong-Ou-Mandel Dip
IQP Circuit	:	Instantaneous Quantum Polynomial Time Circuit
Irrep	:	Irreducible representation
MZI	:	Mach-Zehnder Interferometer
NISQ	:	Noisy Intermediate-Scale Quantum
QAOA	:	Quantum Approximate Optimisation Algorithm
SAT	:	Boolean Satisfiability
TSP	:	Travelling Salesman Problem
VQE	:	Variational Quantum Eigensolver

Acknowledgements

There are many people without whom the experience of creating this thesis would have been made more challenging, if not impossible. In lieu of a perfect and complete list, which no doubt could be a full text of its own, I offer this imperfect and incomplete substitution.

At a fundamental level, I would like to thank the people who made this PhD possible in the first place: The Quantum Engineering Centre for Doctoral Training (QECDT) for offering me a studentship, the Quantum Engineering Technology Labs (QETLabs) for offering me a desk, and the Heilbronn Institute for Mathematical Research for offering me funding as well as exciting summer work. I also thank my supervisors — Peter Turner and Noah Linden — for keeping me on track, my collaborators — Ashley Montanaro, Raúl García-Patrón and Jelmer Renema — for fruitful work, Tony Short as my Annual Progress advisor and Ryan Mann for helpful comments on this thesis.

I have worked alongside many great fellow researchers during my time. Of particular note are my fellow QECDT students: Daniel Love, Geraint Gough, Henry Semenenko, Jason Mueller, Joe Smith, Lawrence Rosenfeld, Lucio Stefan, Martin Nicolle and Sam Holder. I also thank other theorists, including Lana Mineh, Jake Bulmer, Oliver Thomas, Rachel Chadwick, Stasja Stanisic, Sam Morley-Short, Stephen Piddock, Sam Pallister, Juani Bermejo-Vega, Dominik Hangleiter and Jonas Haferkamp. Thank you all for your interesting discussions.

I thank the many people I have worked with on outreach, in QECDT and QETLabs, but also outside, such as in Pint of Science and with Bristol Doctoral College. I would in particular like to thank Nic Harrigan, Euan Allen, Alasdair Price, Holly Caskie, Jamie Thakrar, Sophie Stephens, Laura Veldenz, Ben Barber and Kate Oliver, for the opportunities they have given me to make a fool of myself in the name of public engagement.

Of course, University is only one part of the rich tapestry that is a PhD life. Outside of University, I would like to thank the many sporting communities that I have become part of during my PhD, including but not limited to: parkrun, GoodGym, LGBT+ Fitness Class at Hamilton House, Project Awesome and Spin City.

My life has also changed in other ways. I would like to thank the wonderful folk at the University of Bristol LGBT+ Society, the Bristol SU Trans Students' Network, transcaf and Off the Record Bristol, for the wonderful support they have given me. And a particular thank you to Iris Dinu, for helping me find my direction in the first place.

Thank you to the many friends I have made or kept over the last 27 years, and the warmest thank you of all to my family for the unconditional love, inspiration and support I have received my entire life. This wouldn't have been possible without you all.

This thesis is dedicated to my grandparents, who were immensely proud of their grandchild's academic efforts. Wherever you all are now, I hope you are just as pleased today.

Part I

Overview

Chapter 1

Introduction

Over the last forty years quantum computers have moved from a theoretically interesting concept to exciting experimental and engineering proposals. We now have many suggested applications which quantum computers can solve faster, from physical and chemical simulations to aid in the development of medicines and fertilizers [1], to machine learning so that companies such as Netflix can offer better recommendations to their customers [2].

Perhaps even more exciting is the fact that these computers have gone from simply being a theoretical exercise to actual physical devices that we can program right here and now. Companies such as IBM and Rigetti allow people to access and program real-world quantum computers from the comfort of their web browser [3, 4]. And they even now exist at a scale where they can solve some problems faster than our best classical computers, as was recently demonstrated by Google [5].

But despite the phenomenal progress made, there is still a long way to go before quantum computers are *useful*. For example, the problem solved by Google's quantum computer is a highly contrived problem with no known applications. Thus, while this result is a significant milestone, there is still work that needs to be done in order to demonstrate the full power of quantum computing.

This thesis focuses on the next step on the roadmap towards truly beneficial quantum computers: Trying to find a useful problem which can be solved faster by a quantum computer in the near future. We shall do this by advancing in two directions: One is application-focused, where we look for real-world problems which quantum computers can solve faster; the other is architecture-focused, where we look at what a near-term quantum computing architecture needs to achieve to outpace classical computers. One can think of these two approaches as "top-down" and "bottom-up" respectively.

In the remainder of this chapter we shall provide more detail on all of these points. First in Section 1.1, we shall discuss some of the best-known applications that sufficiently large quantum computers will be able to solve faster. Next in Section 1.2, we will talk about our aims for what we want a near-term quantum computer to achieve. In Section 1.3, we explain our contributions to this milestone and summarise the following chapters. Finally, we mention some notation that is used throughout this thesis in Section 1.4.

1.1 The (eventual) power of quantum computing

The theory of quantum computing has shown the potential for speedups in a large number of problems, some of which we shall detail below. For further information on these problems, we direct the reader to [6].

The original proposed application of quantum computation is the simulating of physical systems. Indeed, this application was proposed as early as 1981, when Richard Feynman spoke about needing quantum mechanics to simulate nature during a keynote he gave at The California Institute of Technology [7]. It then took 15 more years for a general quantum algorithm to be able to simulate any quantum physical system, proven by Lloyd in 1996 [8]. In the decades since, other quantum simulation algorithms have made significant improvements [9, 10, 11].

Following Feynman's original proposal, the first exponential speedup was given by Deutsch and Jozsa in 1992 [12], who gave a query problem which required an exponential number of classical queries but only one quantum query to solve exactly. However, this was for a contrived problem with no applications. The first quantum algorithm with applications to the real world was by Shor in 1994 [13], who gave an algorithm for factoring integers and computing discrete logarithms in polynomial time. These problems are believed to be classically hard to solve, and efficient computation would lead to being able to break popular encryption schemes[14].

Many other applications have also appeared in the following years. In 1997, Grover [15] showed how quantum computers can be used to find marked items quadratically faster than performing a naïve search. And in 2009, Harrow, Hassidim and Lloyd (HHL) provided the first quantum algorithm for solving linear systems of equations [16]. This lead to many applications, from Support Vector Machines [17] to Recommendation Systems [2], in the field of Quantum Machine Learning [18, 19, 20, 21]. This has also led to the more recent direction of quantum-inspired classical machine learning, where various ideas from quantum algorithms have been applied to classical algorithms via sampling rows and columns of matrices with low rank [22, 23, 24, 25, 26, 27].

1.2 What quantum computers can achieve in the near term

However, several of these problems seem to be far off from offering a benefit in the near term. For example, the largest number factored via Shor's algorithm is 21 [28, 29, 30, 31, 32, 33], far smaller than numbers considered cryptographically relevant. Furthermore, implementations of Shor's algorithm so far have used classical compilation methods which take advantage of the already known solution to simplify the quantum algorithm, even though when taken to extremes this can reduce Shor's algorithm to simply tossing unbiased coins [34]. Other quantum computing techniques designed for near-term architectures, such as adiabatic [35, 36, 37, 38] or variational [39, 40] algorithms, have been able to factor numbers as large as 291,311 and 1,099,551,473,989, respectively. But it is not clear how the performance of these schemes scales [41], and these integers are still smaller than what is used in modern encryption schemes. In comparison, classical computers at the end of 2019 managed to factor RSA-240, which is 240 digits (795 bits) long [42], followed three months later by the factorisation of the 829-bit long RSA-250 [43]. Similarly, classical computers have factored every integer of the form $2^n - 1$ for $1000 \leq n \leq 1200$ [44].

Likewise, experimental implementations of HHL have so far only solved linear systems consisting of 2 variables [45, 46, 47], and related algorithms such as the Quantum Support Vector Machine have only been implemented to successfully distinguish the written characters of "6" and "9" [48]. Again, this is far smaller than the amount of data required to beat classical machine learning algorithms in practice, which are able to process data sets containing millions or even billions of entries [49].

While it is interesting to consider problems which quantum computers will eventually be able to solve faster, there is still a need to motivate building these devices now. Hence, the fundamental question of this thesis is the following: What can a quantum computer do in the near future that a classical computer cannot?

As of 2019 there is arguably a very simple answer to this question: Random Circuit Sampling (RCS). This problem, first specified by Boixo et al. [50], is to sample from the output of a random quantum circuit whose depth is linear in the number of qubits. This problem was proven to be classically hard under certain assumptions [51, 52] — explained in more detail in Chapter 4) — but a quantum computer would be able to solve it by simply implementing the random circuit and making a measurement. And most excitingly, an instance of the problem has now been solved on a quantum computer developed by Google which we believe to be hard to solve classically in a reasonable amount of time [5].

However, there are a number of limitations with this model. First, there are no known applications of RCS. Our interest in the problem is not because of its practical uses, but instead simply because we have strong evidence for the problem being classically hard. Second, Google's device is not especially good at performing RCS. The metric used to verify correctness, a variant of fidelity, is estimated for Google's device as roughly 0.1–0.2%. Google defends this low fidelity by simply reducing their estimate of a classical computer's performance by a factor of 1,000, again emphasising classical hardness over usefulness. And third, there is still a lot of uncertainty and debate as to how quickly a classical computer can perform this problem. Google's paper estimated it would take a state of the art supercomputer 10,000 years [5], whereas a recent preprint by IBM estimates the best current supercomputer could perform the same computation in roughly 2.5 days if used solely for this computation [53]. Even further, a report by Morimae, Takeuchi and Tani suggests that classical computers could simulate Google's device even faster, as the fidelity is so low that even sampling from the uniform distribution could be sufficiently precise [54]. Similar claims have also been made in a recent preprint by Zhou, Stoudenmire and Waintal [55], who have claimed to simulate Google's device on a single core computer in a matter of hours. This was achieved by using Matrix Product States, a classical simulation technique where the limiting factor is the multi-qubit fidelity, rather than the number of qubits or circuit depth.

It is with these limitations in mind that we broadly define what we mean when we refer to a near-term quantum speedup. Ideally, we want to find a problem that meets three requirements. The first is that this problem can be applied to real-world problems, rather than only being interesting because of its hardness. The second is that a quantum computer can be constructed in the next few years that can solve this problem in a reasonable amount of time. And the third is that this problem is infeasible to solve on a classical computer, or will at least take significantly longer than the time required for a quantum computer. We have purposefully left these requirements broad rather than giving specific details. This is in order to allow room for a variety of problems, which might satisfy some of the requirements more than others.

One direction that quickly comes to mind that fits these requirements well is the field of Noisy Intermediate-Scale Quantum (NISQ) devices [56]. When Preskill coined the term at a Keynote Address, a variety of examples were given from quantum simulations to quantum semi-definite programming, as examples of practical problems with known quantum speedups that could be experimentally feasible in the near future. NISQ is often used to refer to two quantum systems in particular: The Variational Quantum Eigensolver (VQE) [57], which estimates eigenstates of some target Hamiltonian H via a combination of quantum state preparation, Hamiltonian evolution and measurement; and the Quantum Approximate Optimisation Algorithm (QAOA)

[58], which uses a combination of alternating unitary rotations and classical optimisation to solve various combinatorial optimisation problems, and was recently proven to offer the same dynamics as universal quantum computation, for correctly chosen Hamiltonian times [59, 60]. Both of these offer potential for meeting our requirements for a quantum speedup and are strong directions of future research, but will not be investigated in this thesis.

1.3 Our contributions

In this thesis, we make some progress towards near term quantum speedups. We shall do this in two directions, laid out below.

1.3.1 An application-focused perspective

Our first direction is an application-focused, or "top-down", approach. This is through taking a problem that cannot be efficiently solved on a classical computer and seeing if a quantum computer can solve it faster.

The problem we focus on in particular is the Travelling Salesman Problem (TSP), where the aim is to find the fastest way of visiting every city on a map. This problem is well-known in Computer Science and Mathematics, and has been considered by academics for over 150 years [61]. As an NP-Hard problem, finding an efficient classical solution would solve the long-standing P vs NP problem, bringing with it a million dollar prize as well as many other significant consequences to the world around us [62, 63]. Being NP-Hard also means that it is highly unlikely for an efficient quantum algorithm to be found, but it does not rule out the possibility that quantum computers can solve this problem faster than classical computers. Our first contribution is to demonstrate this positively, by proving polynomial speedups for the Travelling Salesman Problem in the special case where the graph is of bounded degree. This is achieved by showing how classical algorithms for solving the bounded-degree TSP [64, 65, 66] can be sped up using a quantum algorithm for the general family of classical algorithms known as backtracking algorithms [67].

This approach is directed at satisfying our first and third requirements, that the problem is applicable to the real-world and is difficult to solve classically. We shall not explicitly cover the second requirement, that a near-term quantum computer can solve this efficiently, as part of our contributions, but we will make some comments about other progress that has been made towards this requirement and what can be done to improve things further [68, 69].

1.3.2 An architecture-focused perspective

Our second direction is an architecture-focused, or "bottom-up" approach. Here we will take a near-term architecture for a quantum computer and consider how much more powerful it could be than classical computation.

In particular, we shall consider Boson Sampling, a non-universal model of quantum computation consisting of sampling indistinguishable photons input into a random linear optical interferometer [70, 71]. It was proven by Aaronson and Arkhipov that this architecture is classically infeasible to simulate for sufficiently many photons and a large enough interferometer, assuming certain conjectures related to matrix permanents and computational complexity hold. What was significant about Boson Sampling in particular was that this was a very simple model, with no error correction required, and simply used methods and components in linear optics which were already well-known and understood. This led to significant interest, including many experimental advances [72, 73, 74, 75, 76, 77, 78, 79], as well as architectural variants [80, 81, 82].

However, there are practical issues with Boson Sampling experiments which were left unaddressed in the original result. In particular, how photon distinguishability and loss affect the classical complexity of Boson Sampling. These issues have been studied in recent years, with some exciting results [83, 84, 85, 86, 87]. In this thesis, we make two contributions in this direction. First, we describe a link between Boson Sampling and representation theory, via the Quantum Schur Transform [88, 89, 90]. In doing so, we are able to model distinguishability and loss as different forms of decoherence in a specially-structured quantum circuit. Second, we take this model and apply it to a practically motivated form of distinguishability and loss noted in [84, 85]. We then adapt the best asymptotic classical simulation algorithm for Boson Sampling [91] to take advantage of distinguishability and loss, and in doing so produce a classical algorithm which can better simulate Boson Sampling in near-term experiments under these imperfections.

This direction is aimed more at satisfying our second and third requirements of a quantum speedup, in that Boson Sampling is a problem which can be easily done by a quantum computer, such as a Boson Sampling device, but is classically infeasible. This direction less satisfies the first option, in that there are not any known applications for the standard form of Boson Sampling. However, interferometers used for Boson Sampling have also been used for other applications, such as simulating the vibronic spectra of molecules [92], and Boson Sampling variants have also found applications in both simulations and combinatorics [93, 94, 95].

1.3.3 Structure of this thesis

The remainder of this thesis is laid out as follows. In Part II, we shall focus on the applications perspective. First, in Chapter 2 we explain the significance of NP-Hard problems, general classical algorithms for these problems and their corresponding quantum speedups. In Chapter 3 we give an explicit example of a speedup for an NP-Hard problem, by giving polynomial speedups for the Travelling Salesman Problem for bounded degree graphs. Next, in Part III we switch to the architecture perspective. We start in Chapter 4 by explaining the need for near-term quantum architectures, introducing the theory behind Boson Sampling and its imperfections, and explaining the representation theory of the symmetric and unitary groups. In Chapter 5, we go into detail about the link between Boson Sampling and representation theory, and in doing so explain how distinguishability and loss can be modelled as decoherence in a quantum circuit of particular structure. We then in Chapter 6 use this model to develop a classical simulation algorithm for Boson Sampling under a particular form of these imperfections, and show that this algorithm will be a more effective simulator for near-term devices with 30–50 photons. We conclude with some final points and open questions in Chapter 7.

1.3.4 Previous publications and author contributions

Some of the contents of this thesis has been published previously, or completed in collaboration with others. Permission for reuse of content, where applicable, has been acquired. `arXiv:arxiv_id` is used to refer to manuscripts which have been uploaded to the arXiv e-print repository — `https://arxiv.org/` — where preprints of these manuscripts are freely accessible.

- Parts of Chapter 2 and the main results of Chapter 3 are joint work with Noah Linden and Ashley Montanaro, and has been published as "Quantum speedup of the traveling-salesman problem for bounded-degree graphs", *Physical Review A* **95**, 032323 (2017), copyright American Physics Society. A preprint of this article is freely available at `arXiv:1612.06203`. Note that this article was published under my former name. Chapter 2 is preliminary material with no new results, and sections containing previously published material are stated at the start of the chapter. The initial suggestion of applying quantum speedups to the

Travelling Salesman Problem was proposed by AM and NL. The details and analysis of the quantum algorithm for degree-3 graphs in Section 3.3 was by myself, with some valuable contributions — such as the use of binary search — made by AM and NL. Section 3.4 is my own work under the supervision of AM and NL. The article was written by myself under the supervision of AM and NL.

- Parts of Chapter 4 and the main results of Chapter 5 are joint work with Peter S. Turner, and has been published as "Quantum simulation of partially distinguishable boson sampling", *Physical Review A* **97**, 062329 (2018), copyright American Physics Society. A preprint of this article is freely available at `arXiv:1803.03657`. Chapter 4 is preliminary material with no new results, and sections containing previously published material are stated at the start of the chapter. The initial connection between Boson Sampling and the Schur-Weyl Duality was proposed by PST. The quantum circuits and their analysis in Sections 5.2–5.4 were performed by myself. The use of postselection in Section 5.4 was suggested to myself and PST by Aram Harrow. The initial idea of tracing out particles to simulate loss in Section 5.5 was proposed by PST, with detailed analysis performed by myself. Section 5.6 is my own work under the supervision of PST. The article was written by myself under the supervision of PST.

- Parts of Chapter 4 and the main results of Chapter 6 are joint work with Raúl García-Patrón, Jelmer J. Renema and Peter S. Turner, and has been published as as "Classically simulating near-term partially-distinguishable and lossy boson sampling", *Quantum Science and Technology* **5**, 015001 (2020), copyright Institute of Physics. A preprint of this article is freely available at `arXiv:1907.00022`. Chapter 4 is preliminary material with no new results, and sections containing previously published material are stated at the start of the chapter. The initial idea of applying the work of Chapter 5 to a particular model of distinguishability with the aim of an improved classical simulator was proposed following discussions between myself, RG-P and JJR. Section 6.2 is my own work under the supervision of PST. The analysis in Section 6.3 was performed by myself under the supervision of PST, following a joint initial attempt by myself and RG-P. Analysis in Section 6.4 was carried out by myself under the supervision of PST, with some helpful contributions — such as the expectation of permanents of Gaussian matrices — made by RG-P and JJR. Sections 6.5 and 6.6 are my own work under the supervision of PST. Section 6.7 is my own work under the supervision of RG-P, JJR and PST. Sections 6.8 and 6.9 are my own work under the supervision of PST. The article was written by myself under the supervision of RG-P, JJR and PST.

1.4 A brief note on complexity theory

Both Parts II and III make extensive use of concepts from complexity theory. We summarise these concepts below for the benefit of the reader.

Throughout this thesis we will make extensive use of Big-O (and related) notation to simplify the expression of runtimes. For functions $f, g\colon \mathbb{R}_+ \to \mathbb{R}_+$, we say that

$$f(n) \in O(g(n)) \text{ if } \exists n_0, c \in \mathbb{R} \text{ such that } f(n) \leq cg(n) \forall n \geq n_0, \tag{1.1}$$

and

$$f(n) \in \Omega(g(n)) \text{ if } \exists n_0, c \in \mathbb{R} \text{ such that } f(n) \geq cg(n) \forall n \geq n_0. \tag{1.2}$$

Intuitively, this notation allows us to simplify expressions by ignoring constant factors and focusing only on the most significant bottlenecks in a runtime. We direct the curious reader to [96] for further details. We shall also use O^* to suppress factors which are polynomial in n, the input size, $\tilde{O}$ to suppress factors which are polynomial in $\log n$, $\mathrm{poly}(n)$ to denote terms which are polynomial in n, and $\mathrm{polylog}(n)$ for terms which are polynomial in $\log n$.

A variety of complexity classes are discussed in this thesis. P and NP will be explained in more detail in Section 2.1, but informally: P is the class of problems which a classical computer can solve in polynomial time, and NP the class of problems which a classical computer can verify a solution for in polynomial time. We shall also mention BPP, the class of problems which can be solved probabilistically on a classical computer in polynomial time with probability greater than 2/3; BQP, the class of problems which can be solved on a quantum computer with probability greater than 2/3; and #P, the class of problems around counting the number of solutions to a problem in NP. The last two classes we will mention are NP-Hard and #P-Hard, which are the classes of problems which are at least as hard as any problem in NP and #P, respectively. Further details on these and many other complexity classes can be found at the Complexity Zoo [97].

Part II

An application-focused approach: quantum algorithms for the Travelling Salesman Problem

Chapter 2

Preliminary material: Quantum algorithms for NP-Hard problems

Parts of Sections 2.2.1, 2.2.2, 2.3.3 and 2.3.4 were written by myself under the supervision of Noah Linden and Ashley Montanaro, and published as “Quantum speedup of the traveling-salesman problem for bounded-degree graphs”, *Physical Review A* **95**, 032323 (2017), copyright American Physics Society. A preprint of this article is freely available at `arXiv:1612.06203`. Note that this work was completed and published under my former name. This chapter is preliminary material and contains no original results.

An intuitive starting point for finding a problem that quantum computers could demonstrably outperform classical machines, would be those which are classically intractable. There are many problems to choose from, with perhaps the most famous being the NP-Hard family of problems. Many problems with practical applications fit into this family, and although these problems have not been formally proven to be hard for classical computers, the consequences of them being easy to solve has led to the general consensus that these problems are intractable[1].

The reasons above suggest that these problems might be suitable for a demonstrable quantum advantage. However, the downside is that they are generally believed to be too difficult for quantum computers to solve efficiently as well! Indeed, work by Bennett et al. showed that, relative to an oracle chosen uniformly at random, there is no quantum algorithm that can solve arbitrary NP-Complete problems in $o(2^{n/2})$ time [98]. So it seems unlikely that an exponential speedup is achievable, but what about a polynomial speedup? Although less impressive, a polynomial improvement can still be significant in practice.

The following two chapters will focus on this question. In this chapter we shall survey the world of NP-Hard problems. In particular we shall explore classical and quantum algorithms for exactly solving these problems in general. In Chapter 3, we demonstrate how these quantum speedups can be applied to special instances of the Travelling Salesman Problem, an especially famous NP-Hard problem.

The rest of this chapter is presented as follows. In Section 2.1, we introduce the complexity classes P and NP, explain the significance of the P vs NP problem and discuss the relevance of NP-Hard problems. In Section 2.2, we introduce standard classical algorithms for solving NP-Hard problems exactly in an exponential amount of time. Finally, we present quantum speedups of these approaches in Section 2.3.

[1] For some informal arguments as to why these problems *should* be classically hard, see [63]

2.1 P vs NP and NP-Hard problems

To understand why NP-Hard problems are a good starting point in the hunt for a quantum speedup, we first need to understand why they are so important in the world of complexity theory. This is because they provide a potential solution to one of the famous problems in computer science and mathematics: the P vs NP problem.

Informally, P and NP are classes of decision problems, or alternatively languages. P, short for Polynomial Time, is the set of decision problems that can be *solved* in polynomial time on a classical computer. This means that a language L is in P if and only if there is a polynomial-time classical algorithm which accepts all words $w \in L$ and rejects all words $w \notin L$. NP in comparison, is short for Non-deterministic Polynomial Time, and can be thought of as the set of decision problems which can be *verified* in polynomial time on a classical computer. This means that there is a polynomial-time classical algorithm such that for all words $w \in L$ there exists a certificate c such that the algorithm accepts (w, c), and for all words $w \notin L$ the algorithm rejects (w, c) for all certificates c.

The P vs NP problem is the question of whether or not these two classes are equal. Intuitively it is easy to see that P $\subseteq$ NP: If a polynomial time algorithm exists for solving a decision problem then we can ignore the certificate and simply run the solver to verify the problem in polynomial time. So the main question is whether or not the converse holds: Is P $\supseteq$ NP? It might seem intuitive that there are problems which are easy to verify but not easy to solve, but to this day there is no proof that this is the case. Indeed, solving this problem is now worthy of a million dollar prize courtesy of the Clay Mathematics Institute [62].

So where do NP-Hard problems come in? Well, a potentially good direction for solving the P vs NP problem is to focus on the very hardest problems in NP. NP-Completeness is a way of classifying these problems. More strictly, a language $L \in$ NP is NP-Complete if for any language $L' \in$ NP, there exists a polynomial time classical algorithm which takes any word $w' \in L'$ to a word $w \in L$ and any word $w' \notin L'$ to a word $w \notin L$. Such an algorithm is known as a polynomial time reduction. NP-Hard is a generalisation of this complexity class, consisting of all languages L such that for any language $L' \in$ NP a polynomial time reduction from L' to L exists, but L does not necessarily need to be in NP.

These reductions mean that if an NP-Hard problem requires a polynomial amount of time to solve then so does every problem in NP. To see this, suppose we had a algorithm for solving an NP-Hard problem L in time $T(n) \in \Omega(\text{poly}(n))$, where n is the size of our input. Then we immediately have an algorithm for solving any problem L' in NP in time $T(n) + \text{poly}(n)$: Given w', we run our polynomial time reduction to create a word w, and then run our $T(n)$-time algorithm for deciding if $w \in L$.

This shows why NP-Hard problems are such a strong motivation for the P vs NP problem. Finding a polynomial time algorithm for any NP-Hard problem is sufficient for proving that P $=$ NP. Conversely, if any NP-Complete problem can be proven to not be solvable in polynomial time, then we have proven that P $\neq$ NP. However, both directions of work have proven to be highly non-trivial to solve.

2.1.1 Examples of NP-Hard problems

There are many examples of problems which are NP-Hard. For this summary, we shall give three well-known problems which are often discussed in the theoretical computer science literature and are applicable to which the algorithms discussed later in this chapter. For more examples of NP-Hard problems, we point the reader towards [99, 100], which also includes reductions for these and many other problems.

Problem 2.1 (Boolean Satisfiability (SAT))**.** *Let B be a Boolean formula with variables $\mathbf{x} = (x_1, \ldots, x_n)$. Is there an assignment of $\mathbf{x}$ such that $B(\mathbf{x}) = $ True?*

Boolean Satisfiability was the first example of a problem proven to be NP-Complete, in what is now known as the Cook-Levin Theorem [101]. This is also true for different restrictions of Satisfiability, such as when the Boolean formulae are written in Conjunctive Normal Form with at least three terms in each clause [99]. It is easy to see that a brute force solution, where we try every possible assignment of $\mathbf{x}$, would require $O(2^n)$ time.

The proof that SAT is NP-Complete is non-trivial, and is proven by showing that any polynomial-time non-deterministic algorithm running on some input can be reduced to a Boolean formula. Proving subsequent problems to be NP-Hard is easier however, as the proof just needs to show that a single NP-Hard problem can be reduced to this new problem. This paved the way for many other problems to be proven NP-Hard.

Problem 2.2 (Integer Linear Programming)**.** *Given $A \in \mathbb{Z}^{m \times n}$, $c \in \mathbb{Z}^n$ and $b \in \mathbb{Z}^m$, find $x \in \mathbb{Z}^n$ such that $x_i \geq 0 \forall i$ which maximises $c^T x$ subject to $Ax \leq b$.*

It is worth noting that Integer Linear Programming is considered NP-Hard but not NP-Complete, as it is phrased as an optimisation problem rather than a decision problem. The equivalent decision problem — does there exist an $x \in \mathbb{Z}^n_{\geq}$ that satisfies $Ax \leq b$ — is NP-Complete, with the version where $x \in \{0, 1\}^n$ being one of Karp's 21 NP-Complete problems [99]. As with Boolean Satisfiability, the number of possible solutions in a brute force approach would be exponential in n.

Problem 2.3 (The Travelling Salesman Problem (TSP))**.** *Let $G = (V, E)$ be a graph with n vertices and m weighted edges. Find a minimum-weight cycle which visits every vertex exactly once.*

A solution to the Travelling Salesman Problem is by definition a Hamiltonian cycle for a graph G. Determining whether or not a Hamiltonian cycle exists in a graph was also proven to be NP-Complete by Karp [99]. The number of possible Hamiltonian cycles is related to the number of permutations of vertices, of which there are $n!$. However, some permutations will give identical Hamiltonian cycles, for example the cycle $1 \to 2 \to 3 \to 1$ is equivalent to the cycle $2 \to 3 \to 1 \to 2$, and for undirected graphs both cycles are equivalent to $3 \to 2 \to 1 \to 3$. Removing such duplicates, the number of possible Hamiltonian cycles which need to be considered in a brute force approach is $n!/2n = O((n-1)!)$.

Like Integer Linear Programming, the Travelling Salesman Problem is an optimisation problem rather than a decision problem. The decision variant — does G contains a Hamiltonian cycle of length at most ℓ — is also NP-Complete, and can be seen through the same reasoning as in the preceding paragraph. It is worth noting that if an algorithm exists for the decision version of the problem, an algorithm for the optimisation problem can also be found with a polynomial overhead, by calling the algorithm with varying bounds ℓ chosen in a binary search fashion. Note that this leads to a polynomial overhead in $\log \ell$, meaning that if ℓ is extremely large, i.e. $O\left(2^{2^n}\right)$, then this is an exponential overhead in n. However, this would mean that to even specify ℓ would require an exponentially large number of bits in n, so the algorithm would still be polynomial in the input size. It is tempting to ask whether one can reduce this dependence by multiplying all edge weights by a small value $\epsilon > 0$. However, the algorithm would still require an amount of time polynomial in the number of bits required to estimate ℓ up to sufficient precision, so this does not lead to a saving in runtime.

2.2 Classical algorithms for NP-Hard Problems

We shall now move on to exploring how we can classically solve these NP-Hard problems. We shall focus on exact algorithms, which are required to run in polynomial time classically. It is worth noting that there are classical algorithms for approximating many of these problems in polynomial time; Christofides' Algorithm, for example, provides a Hamiltonian cycle whose length is at most 3/2 of the length of the optimal Hamiltonian cycle in polynomial time [102]. See [100] for a more thorough review of classical algorithms, including approximation algorithms.

In some cases, the best performance we are able to asymptotically achieve is a brute force evaluation. With SAT, for example, there are 2^n possible assignments of n variables, and evaluating a single assignment takes polynomial time. Assuming the Strong Exponential Time Hypothesis holds, which in turn would imply that P $\neq$ NP, any classical algorithm must solve the SAT problem in $\Omega(2^n)$ time [103]. But even when asymptotically brute force approaches are as good as any, in practice there might be many instances that do not require a full evaluation of all possible outcomes. This has led to significant interest in developing faster algorithms for non-worst-case Boolean formulae, with numerous surveys and competitions assessing these approaches [104, 105].

2.2.1 Dynamic programming

Dynamic programming is a method for solving optimisation problems in a recursive fashion. The technique works by computing smaller instances of the problem and storing the result in memory in order to prevent needing to recompute the instance later.

The canonical example of dynamic programming is for computing Fibonacci numbers. A naïve algorithm would compute the n-th Fibonacci numbers by recursively computing the previous two and adding the result together:

$$F(n) = F(n-1) + F(n-2)$$

This algorithm is inefficient as the same values will be computed many times over. For example, $F(n-2)$ will be computed twice, being called by both $F(n)$ and $F(n-1)$. A more efficient method is a "ground-up" approach: First set $F(1) = 1$ and $F(2) = 1$ in memory, and then for each $i \in \{3, \dots, n\}$, compute $F(i)$ using the previously computed answers and save the result in memory.

For NP-Hard problems, dynamic programming has offered a number of results. For many years, the algorithm with the best proven worst-case bounds for the Travelling Salesman Problem was the Held-Karp algorithm [106], which runs in $O(n^2 2^n \log L)$ time and uses $O(n 2^n \log L)$ space, where L is the length of the longest edge. This algorithm uses the fact that for any shortest path, any subpath visiting a subset of vertices on that path must be the shortest path for visiting those vertices. Held and Karp used this to solve the TSP by computing the length of the optimal route for starting at some initial vertex 1, visiting every vertex in a set $S \subseteq V$ and finishing at a vertex $l \in S$. Denoting the length of this optimal route $D(S, l)$, they showed that this distance could be computed as

$$D(S,l) = \begin{cases} c_{1l} & \text{if } S = \{l\} \\ \min_{m \in S \setminus \{l\}} \left[D(S \setminus \{l\}, m) + c_{ml}\right] & \text{otherwise.} \end{cases} \tag{2.1}$$

Solving this relation recursively for $S = V$ would result in iterating over all $O((n-1)!)$ Hamiltonian cycles again, but Held and Karp showed that the relation could be solved in $O(n^2 2^n \operatorname{polylog} L)$ time using dynamic programming, where the $O(\log L)$ overhead comes from the cost of binary arithmetic. Björklund et al. [107] developed on this result, showing that modifications to the Held-Karp algorithm could yield a runtime of

$$O^*((2^{k+1} - 2k - 2)^{n/(k+1)} \log L), \tag{2.2}$$

where k is the largest degree of any vertex in the graph; this bound is strictly less than $O^*(2^n \log L)$ for all fixed k.

2.2.2 Backtracking

Backtracking is a form of recursive algorithm designed for solving Constraint Satisfaction Problems (CSPs). These are problems where the input is a set of variables $x_1, \ldots, x_n$ and the aim is to find an assignment for these variables satisfying constraints $c_1, \ldots, c_n$.

Backtracking works by taking a set of already assigned variables and simplifying the constraints accordingly. We then use a predicate P to check if the constraints are already satisfiable or not under the current assignments. If so, then we are done. Otherwise, we choose an unassigned variable according to some heuristic h, and then recursively call the algorithm on all possible assignments. We can view this algorithm as exploring a tree whose vertices are labelled with partial assignments. The size of the tree determines the worst-case runtime of the algorithm, assuming that there is no assignment that satisfies all the constraints.

```
Function BacktrackSAT($B$, $\tilde{\mathbf{x}}$) is
    Input: A boolean formula $B$, a partial assignment of variables $\tilde{\mathbf{x}}$
    Output: A satisfying assignment or $\emptyset$ if no such assignment exists
    Apply $\tilde{\mathbf{x}}$ to $B$ to get Boolean formula $B'$
    if $B' = True$ then
        Return $\tilde{\mathbf{x}}$
    else if $B' = False$ then
        Return $\emptyset$
    else
        Find smallest $i$ such that $x_i$ is unassigned
        Apply $\tilde{\mathbf{x}}' = \tilde{\mathbf{x}}(x_i = \text{True})$ to $B$ to get Boolean formula $B''$
        if $B'' \neq False$ then
            Return BacktrackSAT($B$, $\tilde{\mathbf{x}}(x_i = True)$);
        else
            Return BacktrackSAT($B$, $\tilde{\mathbf{x}}(x_i = False)$);
        end
    end
end
```

Algorithm 2.1: A backtracking algorithm for SAT.

Algorithm 2.1 gives an example of a backtracking algorithm for Boolean Satisfiability. Suppose we can apply a partial assignment $\tilde{\mathbf{x}}$ to B in order to produce a Boolean formula B' consisting of the remaining unassigned variables. Then our predicate P is simply to apply a partial assignment to B and check if B' is now trivial; if so then we return True or False depending on if B' is True or False, respectively. Otherwise, our predicate P returns Undefined. Our heuristic h is to simply pick an unassigned variable x_i. Asymptotically, this algorithm will require $O^*(2^n)$ time in the worst case, where every possible assignment needs to be tested. However, in practice we might not need to evaluate large portions of the tree, due to finding partial assignments which already succeed or fail to satisfy the Boolean formula. There might also be ways to improve this algorithm in practice, such as by applying polynomial time techniques to simplify B'.

Branch and bound

Branch and bound is a development of backtracking specifically looking at optimisation problems. Rather than proceeding recursively like in backtracking, branch and bound uses a bound function to decide the order in which to evaluate potential solutions, as well as a branch function to decide on how to reduce the solution space.

Branch and bound is a standard approach for Integer Linear Programming. An upper bound can be computed by removing the constraint that the solution vector x needs to be an integer, relaxing the problem to that of a Linear Programming problem which can be solved in polynomial time via, for example, [108]. If x is an integer solution, no solution exists, or the upper bound is worse than our current best solution, we stop exploring that potential solution. Otherwise we branch by choosing an $s = x_i$ in the solution vector which is not an integer, and create two new reduced problems by adding either the constraint that $x_i \leq \lfloor s \rfloor$ or that $x_i \geq \lceil s \rceil$. We can also improve the algorithm even further by adding constraints in the form of cutting planes; such a technique is commonly referred to as branch and cut.

Both branch and bound & branch and cut algorithms have also been developed for the Travelling Salesman Problem [109, 110]. Asymptotic analysis of these algorithms are hard to come by, but these approaches have been found to be the best performing in practice. Indeed, a branch and cut algorithm is responsible for some of the largest solved instances of the Travelling Salesman Problem to date, finding the optimal tour of 85,900 cities in 2005/2006 [111], and later finding the optimal tours of 49,687 UK pubs and 109,399 stars [112].

2.3 Quantum speedups

We shall now summarise some of the research in quantum speedups of the classical algorithms described in this chapter. For a more broad summary of quantum algorithms, we direct the reader to [6].

2.3.1 Quantum search

The most immediate way in which one might try to solve NP-Hard problems is with quantum search, originally developed by Grover [15]. Given an oracle for checking if a solution is correct, Grover search works by querying this oracle over a superposition of different solutions. If there is a total of n solutions, then it is possible to show that Grover search will find a correct solution after $O(\sqrt{n})$ queries, whereas an unstructured classical search would require $O(n)$ queries in the worst case.

Grover search has proven beneficial for a number of quantum algorithms. For instance, it has been shown that if there are m correct solutions then Grover search will succeed after $O(\sqrt{n/m})$ queries. There is also a more general algorithm called amplitude amplification, where if the probability of a quantum algorithm outputting a correct solution is a, then a correct solution can be found after running the quantum algorithm $O(1/\sqrt{a})$ times [113]. It is also possible to find the minimum solution in $O(\sqrt{n})$ queries, by choosing a threshold and periodically updating it as the algorithm runs [114]. Finally, we can also use related methods to traverse a graph until a marked vertex is found, known as a quantum walk [115].

This seems like a reasonable starting place for trying to solve NP-Hard problems. Many of these problems, including the ones given above, are related to either proving a solution exists, or finding a minimum solution. Quantum search algorithms then provide a quadratic speedup over searching for all possible solutions. For example with SAT, our search space is all 2^n possible assignments of n variables, meaning that Grover search would find a satisfying assignment or prove that one does not exist in $O(2^{n/2})$ queries. Assuming the Strong Exponential Time Hypothesis holds, this guarantees a quadratic speedup in worst-case performance.

However, a quadratic speedup over *any* classical algorithm is not necessarily guaranteed. Take the Travelling Salesman Problem for example: The number of possible solutions is on the order of the number of permutations of vertices, of which there are $n!$, meaning that the quantum minimum finding algorithm would find a shortest Hamiltonian cycle in $O(\sqrt{(n-1)!})$ queries. But there are already classical algorithms which are significantly faster, such as the Held-Karp algorithm, which uses $O(n^2 2^n)$ queries. As a result, we need to consider other speedups to gain an improvement over these better classical algorithms.

2.3.2 Dynamic programming

Until recently, it was not known whether or not quantum algorithms would be able to speed up dynamic programming algorithms. This is because of the way in which dynamic programming typically records the solutions to all possible subproblems in memory. Adapting these methods for quantum computers is non-trivial as a result. It is also worth noting that these works came after the work written in this thesis.

The first improvement on dynamic programming algorithms was given by Ambainis et al. [116], which considered the Path in the Hypercube problem. This problem is based around the 2^n Boolean hypercube, which is a graph where each vertex represents an n-bit string and two vertices are adjacent if their Hamming distance is 1. The aim is to find a path from 0^n to 1^n that only uses some subgraph of the Boolean hypercube.

Ambainis et al. provide a quantum algorithm for solving this problem in $O^*(1.817^n)$ time. The fundamental idea of this technique is to use classical dynamic programming to solve subproblems close to 0^n and 1^n, before using quantum search as described previously to find a path between the two subproblems. Allowing the quantum algorithm to solve larger subproblems leads to a faster runtime, but improvements over time start becoming too small to be significant. Ambainis et al. reach the above runtime of $O^*(1.817^n)$ when the quantum algorithm starts running at depth 6. When the subgraph has at most μ^n vertices for $\mu \geq 1.735$, this leads to speedups in other dynamic programming algorithms too.

A quadratic speedup for dynamic programming was later proven by Ronagh [117]. This works by a technique called the Multiplicative Weights Update Method. In this method, there are n experts, each of which advise the algorithm on the next step to take, after which the algorithm updates its weighting of each expert. Initially these experts are evenly weighted, and over time some experts become favoured over others, until the algorithm halts after T iterations for some T. The updating of weights after each iteration is based on computation of some cost vector found by linear programming. A quadratic speedup is achieved by computing this cost vector via quantum minimum finding [114]. This can be applied to dynamic programming via a dual formulation of dynamic programming methods.

2.3.3 Backtracking

Unlike dynamic programming, backtracking seems to hold a structure that more intuitively leads to a quantum speedup. The recursive nature means that each instance of the problem only depends on local results, rather than the global access required for dynamic programming. As a result, speedups in backtracking have been explored for decades.

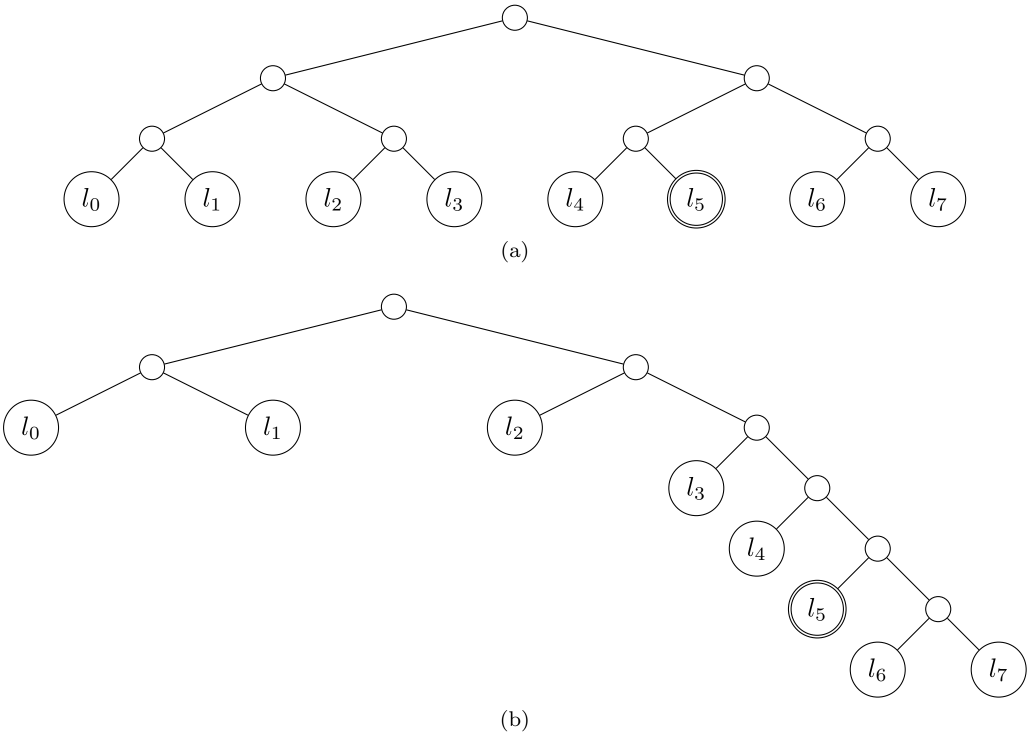


Figure 2.1: Example backtracking trees, where l_5 is a leaf corresponding to a solution to a Constraint Satisfaction Problem: (2.1a) shows an example of a perfectly balanced backtracking tree, where each leaf can be associated with a 3-bit string corresponding to a path to that leaf; (2.1b) on the other hand shows an example of an unbalanced backtracking tree, where specifying a path to a leaf requires 6 bits.

It is first worth considering why Grover search [15] will not necessarily achieve a quadratic speedup over the classical backtracking algorithm. Grover search requires access to a function $f\colon \{0,1\}^n \to \{\text{true}, \text{false}\}$. If there are m results $x \in \{0,1\}^n$ such that $f(x) = \text{true}$, then Grover search will succeed after $O(\sqrt{2^n/m})$ applications of f[15].

To apply Grover search, we would need to access the leaves of the tree, as these are the points where the backtracking algorithm is certain whether or not a solution will be found. Thus, for each integer i, we would need to find a way of determining the i-th leaf l_i in the backtracking tree. In the case of a perfectly balanced tree, such as Fig. 2.1a, where every vertex in the tree is either a leaf or has exactly d branches descending from it, such a problem is easy: Write i in base d and use each digit of i to decide which branch to explore. But not all backtracking trees are perfectly balanced, such as in Fig. 2.1b. In these cases, finding leaf l_i is hard as we cannot be certain which branch leads to that leaf. Some heuristic approaches, by performing amplitude amplification on part of the tree, can produce better speedups for certain trees, but do not provide a quadratic speedup in general.

The first result was by Cerf, Grover and Williams [118]. This algorithm, known as nested quantum search, worked by first applying Grover search to get a set of possible solutions. The recursive calls would then be emulated by using the output of previous Grover search instances as the input to subsequent Grover searches. This recursive nature would continue for all recursive layers of the backtracking tree. Cerf, Grover and Williams demonstrated that on average, if d is the size of the search space considered by the backtracking algorithm, then nested quantum search will find a solution to a constraint satisfaction problem in $O(\sqrt{d^{\alpha}})$, where $\alpha < 1$ is a constant depending on the variables and constraints. However, this was only an average-case speedup, rather than worst-case.

Several attempts at speeding up backtracking algorithms via quantum computation have come since. Farhi and Gutmann [119] showed that some instances of backtracking trees could be solved exponentially faster via a quantum random walk than a classical random walk [119], though these problems could also be solve more efficiently classically. Angelsmark, Dahllöf and Jonsson showed that some constraint satisfaction problems could be reduced to finding a valid certificate from a set of size d^{cn} for $c < 1$, and then using Grover search to find a valid certificate in $O(d^{cn/2})$ queries. Fürer [120] showed that a quadratic speedup could be achieved over the number of leaves in a backtracking tree, but requires an efficient way of indexing each leaf, which might not be possible when the tree is highly unbalanced as in Figure 2.1b.

A worst-case quadratic speedup was later proven by Montanaro [67]. Montanaro's result works by performing a quantum walk on the backtracking tree to find marked vertices, which correspond to assignments that satisfy the constraints. A potential issue with this strategy is that quantum walk algorithms often need to know the full graph at the start of the computation. To avoid this, Montanaro uses a quantum walk by Belovs [121, 122] where steps in the quantum walk only depend on local knowledge, rather than needing to know the entire graph in advance.

One weakness with Belovs' quantum walk is that it is only able to detect the existence of marked vertices in the graph, rather than being able to find the marked vertices. To work around this, Montanaro applies the quantum walk initially to the whole tree to detect a marked vertex, and then applies the walk to subtrees to find subtrees which contain a marked vertex. This is repeated recursively until a marked vertex in the tree is found. As long as the degree of each vertex in the tree is constant, the overhead of this recursion is the maximum depth of the tree, which is $O(n)$. It is worth noting that very recently two preprints have shown how Belovs' quantum walk can be adapted to not only detect the existence of marked vertices but also find them [123, 124].

Theorem 2.1 (Montanaro [67])**.** *Let $\mathcal{A}$ be a backtracking algorithm with predicate P and heuristic h that finds a solution to a constraint satisfaction problem on n variables by exploring a tree of at most T vertices. There is a quantum algorithm which finds a solution to the same problem with failure probability δ with $O(\sqrt{T}n^{3/2}\log n \log(1/\delta))$ uses of P and h.*

The reader familiar with [67] may note that the definition of the set of partial assignments in Montanaro's work also incorporates information about the ordering of assignments to variables. However, it is easy to see from inspection of the algorithm of [67] that removing this information does not affect the stated complexity of the algorithm.

It is worth understanding the limitations of the quantum backtracking algorithm, and why it cannot necessarily speed up all algorithms termed "backtracking algorithms" [67]. First, a requirement for the quantum algorithm is that decisions made in one part of the backtracking tree are independent of results in another part of the tree, which is not true of all classical algorithms, such as constraint recording algorithms [125]. Second, the runtime of the quantum algorithm depends on the size of the entire tree. Thus, to achieve a quadratic speedup over a classical algorithm, the algorithm must explore the whole backtracking tree, instead of stopping after finding the first solution or intelligently skipping branches such as in backjumping [125].

Another limitation of quantum backtracking algorithms is that often there will be a metric $M : \mathcal{D} \to \mathbb{R}$ we want the backtracking algorithm to maximise or minimise while satisfying the other constraints. This is particularly relevant for the TSP, where the aim is to return the shortest Hamiltonian cycle. Classical backtracking algorithms can achieve this by recursively travelling down each branch of the tree to find results $D_1, \ldots, D_d \in \mathcal{D}$ and returning the result that minimises M. The quantum backtracking algorithm cannot perform this; it instead returns a solution selected randomly from the tree that satisfies the constraints. In order to achieve a quantum speedup when finding the result that minimises M, we can modify the original predicate to prune results which are greater than or equal to a given bound. We then repeat the algorithm in a binary search fashion, updating our bound based on whether or not a solution was found. This will find the minimum after repeating the quantum algorithm at most $O(\log M_{max})$ times, where

$$M_{max} = \max\{M(D) : D \in \mathcal{D}, P(D) = \text{true}\}. \tag{2.3}$$

We describe this binary search approach in more detail in Sec. 3.3.2.

We shall conclude this section with a number of improvements made to Montanaro's backtracking algorithm following the work presented in this thesis. The second limitation mentioned above was removed in work by Ambainis and Kokainis [126], using a quantum algorithm they developed for estimating the size of trees in $\tilde{O}(\sqrt{vT})$ steps. Ambainis and Kokainis use this new quantum tree size estimation algorithm to generate a path of the first k vertices visited by the classical backtracking algorithm. To see how this works, suppose we start at the root of our tree. For each of the root's children, we estimate how many vertices are underneath that child. We then generate a path from concatenating all the root's brances, until we reach a point where adding on the next branch would exceed k. We then recursively apply the path generation algorithm to that branch to construct the rest of the length m path.

To apply this to backtracking, Ambainis and Kokainis generate a path of the first 2^i vertices the classical backtracking algorithm visits for some i, and then performs Montanaro's backtracking algorithm on this subtree. This is repeated for increasing values of i until either a marked vertex is found or the entire tree has been visited. This algorithm fails with probability ϵ and runs in

$$O\left(v^{3/2}\sqrt{T}\log^2\frac{v\log T_0}{\epsilon}\right)$$

time, where T_0 is the size of the total tree, T is the size visited by the backtracking algorithm.

Runtime factors of Montanaro's algorithm were improved in subsequent work by Jarret and Wan [127]. This was achieved by noting that the runtime of Belovs' quantum walk algorithm also depends on the effective resistance of a graph, a property of a graph inspired by electrical networks, which is not utilised by Montanaro. Jarret and Wan showed that this quantity can be efficiently approximated on a quantum computer and utilised to achieve a runtime of $O(\sqrt{Tn}\log^4(mn)\log(m/\epsilon))$, where m is the number of satisfying assignments. This leads to an improvement in terms of n as long as m grows sub-exponentially with n.

2.3.4 Adiabatic quantum computing

We conclude this preliminary material by briefly mentioning the role of adiabatic quantum computing and the Quantum Adiabatic Optimisation Algorithm (QAOA) in achieving a quantum speedup for NP-Hard problems. Adiabatic quantum computing is a model of quantum computation where one starts with a quantum system with a known ground state, and then gradually updates the Hamiltonian until the system is one whose ground state is the solution to the problem one is trying to solve. As long as the Hamiltonian evolves slowly enough over time, the adiabatic theorem states that with high probability the system will still be in the ground state at the end of the evolution [128]. The run time of this algorithm is therefore dependent on the minimum distance in energy levels between the ground state and the next excited state at any point in the evolution, known as the spectral gap. This model of quantum computing is very similar to, though not strictly the same as, quantum annealing, which the model used by D-Wave quantum processors [129].

Adiabatic quantum computing has been considered for NP-Hard problems for some time. One example of this is Farhi et al. [130], who gave a quantum annealing algorithm for boolean satisfiability as well as instances where the spectral gap can be estimated and lead to a polynomial runtime. Although known bounds on asymptotic runtimes are limited as it is challenging to get an approximation of the spectral gap for these problems, experiments have been performed on small quantum annealing devices. For instance, Farhi et al. have implemented a quantum annealing algorithm for the NP-Hard problem known as Exact Cover with up to 20 variables, and found the algorithm to perform reasonably well [131]. Other quantum annealing experiments on D-Wave processors for a variety of NP-Hard problems specified by companies were performed by Desimone et al. [132].

There have also been a number of results around applying quantum annealing to find approximate solutions of the Travelling Salesman Problem. Rather than solve the problem purely through quantum annealing, Martoňák, Santoro and Tosatti [133] construct an Ising Hamiltonian for solving the TSP, simplify the Hamiltonian and then use path-integral Monte Carlo [134] to run their model. While no bounds on run time or accuracy were strictly proven, they concluded by comparing their algorithm to simulated annealing via the Metropolis-Hastings algorithm [135] and the Kernighan-Lin algorithm for approximately solving the TSP [136]. Their results showed that quantum annealing could outperform simulated annealing alone, but both could be outperformed by ad hoc algorithms. They also noted that simulated annealing could perform better than in their analysis if combined with local search heuristics [137]. Chen et al. [138] experimentally demonstrated a quantum annealing algorithm for the TSP, using a nuclear-magnetic-resonance quantum simulator to solve the problem for a graph with 4 vertices. Finally, Heim et al. [139] have reported simulations of quantum annealing experiments to solve the TSP, concluding that "analog quantum annealing devices are unlikely to be of interest as TSP solvers in the near future".

2.3.5 Quantum Approximate Optimisation Algorithm

The Quantum Approximate Optimisation Algorithm (QAOA) is a hybrid quantum-classical approach devised for noisy intermediate-scale quantum (NISQ) computaters. The idea behind it is to start with a uniform superposition, apply a sequence of alternating unitary rotations $\prod_{i=1}^{p} U(\theta_i)V(\phi_i)$ for unitaries U and V and angles $\theta_i, \phi_i, i \in [p]$ for some given $p \in \mathbb{N}$, and then measure in the computational basis. From the measurement outcome $|x\rangle$, an objective function $C(x)$ is calculated and a classical optimisation algorithm is used to adjust the rotation angles. This is repeated until the algorithm converges on an optimal $C(x)$.

QAOA has been proposed as a NISQ algorithm for solving NP-Hard problems. It was originally proposed for solving the Max-Cut problem, an NP-Hard problem in graph theory [58]. It was then later used for another NP-Hard problem, known as Max E3LIN2, which is related to solving integer linear equations [140]. For this second case, Farhi, Goldstone and Gutmann showed that even if $p = 1$, meaning that each unitary is only applied once, the result from QAOA was a better approximation of the optimal solution than what was at the time the best-known classical algorithm. However, since then a subsequent classical algorithm has provided an even better approximation [141]. The extent to which QAOA offers a speedup over classical algorithms for NP-Hard problems is therefore left unclear.

Chapter 3

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

3.1 Introduction

This work was completed in collaboration with Noah Linden and Ashley Montanaro, and published as "Quantum speedup of the traveling-salesman problem for bounded-degree graphs", *Physical Review A* **95**, 032323 (2017), copyright American Physics Society. A preprint of this article is freely available at `arXiv:1612.06203`. Note that this work was completed and published under my former name. Details of contributions made by myself are given in Section 1.3.4.

In this chapter we apply known quantum-algorithmic techniques to accelerate a sequence of classical TSP algorithms for the important special case of bounded-degree graphs. We say that a graph G is degree-k if the degree of any vertex in G is at most k. Although a sub-instance of the general Travelling Salesman Problem, this restriction is still NP-Hard for graphs of degree at least[1] 3 [142]. In fact, NP-Hardness even holds in highly restrictive cases such as when the graph is not only degree-3 but also planar [143].

A recent line of research has produced a sequence of classical algorithms which improve on the $O^*(2^n \log L)$ runtime of the general Held-Karp algorithm in this setting. First, Eppstein presented algorithms which solve the TSP on degree-3 graphs in time $O^*(2^{n/3} \log L) \approx O^*(1.260^n \log L)$, and on degree-4 graphs in time $O^*((27/4)^{n/3} \log L) \approx O^*(1.890^n \log L)$ [64]. The algorithms are based on the standard classical technique of *backtracking*, an approach where a tree of partial solutions is explored to find a complete solution to a problem (see Section 2.2.2 for an introduction to this technique). Following subsequent improvements [144, 145], the best classical runtimes known for algorithms based on this general approach are $O^*(1.232^n \log L)$ for degree-3 graphs [65], and $O^*(1.692^n \log L)$ for degree-4 graphs [66], in each case due to Xiao and Nagamochi. All of these algorithms use polynomial space in n.

An algorithm of Bodlaender et al. [146] achieves a faster runtime of $O^*(1.219^n \log L)$ for solving the TSP in degree-3 graphs, which is the best known; however, this algorithm uses exponential space. Similarly, an algorithm of Cygan et al. [147] solves the TSP in unweighted

[1] Note that the Travelling Salesman Problem on any graph of degree strictly less than 3 is trivial to solve, as the only such graphs that contain a Hamiltonian cycle are the cycle graphs C_n.

Degree	Quantum	Classical (poly space)	Classical (exp space)
3	$O^*(1.110^n \operatorname{polylog} L)$	$O^*(1.232^n \log L)$ [65]	$O^*(1.219^n \log L)$ [146]
4	$O^*(1.301^n \operatorname{polylog} L)$	$O^*(1.692^n \log L)$ [66], $O^*(1.657^n L)$ [148]	$O^*(1.588^n \log L)$ [147]
5, 6	$O^*(1.680^n \operatorname{polylog} L)$	$O^*(1.657^n L)$ [148]	—

Table 3.1: Runtimes of our quantum algorithms for a graph of n vertices with maximum edge cost L, compared with the best classical algorithms known.

degree-4 graphs in $O^*(1.588^n \log L)$ time and exponential space. Both of these algorithms use an approach known as cut-and-count, and a quantum speedup is not known for either algorithm.

In the case where we have an upper bound L on the maximum edge cost in the graph, Björklund [148] gave a randomised algorithm which solves the TSP on arbitrary graphs in $O^*(1.657^n L)$ time and polynomial space, which is an improvement on the runtime of the Xiao-Nagamochi algorithm for degree-4 graphs when L is subexponential in n. Again, the techniques used in this algorithm do not seem obviously amenable to quantum speedup.

Here we use the quantum speedup for backtracking described in Section 2.3.3 to speed up the algorithms of Xiao and Nagamochi in order to find Hamiltonian cycles shorter than a given upper bound, if such cycles do exist. We run this algorithm several times, using binary search to specify what our upper bound should be, to find the shortest Hamiltonian cycle and solve the Travelling Salesman Problem. In doing so, we achieve a near-quadratic reduction in the runtimes.

Theorem 3.1. *There are bounded-error quantum algorithms which solve the TSP on degree-3 graphs in time $O^*(1.110^n \log^2 L \log\log L)$ and on degree-4 graphs in time $O^*(1.301^n \log^2 L \log\log L)$, where L is the maximum edge cost. The algorithms use* $\operatorname{poly}(n)\log L$ *space.*

In this result and elsewhere in the chapter, "bounded-error" means that the probability that the algorithm either doesn't find a Hamiltonian cycle when one exists or returns a non-optimal Hamiltonian cycle is at most 1/3. This failure probability can be reduced to δ, for arbitrary $\delta > 0$, by repeating the algorithm $O(\log 1/\delta)$ times. Also here and throughout, log denotes log base 2. Note that the time complexity of our algorithms has some dependence on L, the largest edge cost in the input graph. However, this dependence is quite mild. For any graph whose edge costs are specified by w bits, $L \leq 2^w$. Thus terms of the form $\operatorname{polylog}(L)$ are at most polynomial in the input size.

Next, we show that degree-5 and degree-6 graphs can be dealt with via a randomised reduction to the degree-4 case.

Theorem 3.2. *There is a bounded-error quantum algorithm which solves the TSP on degree-5 and degree-6 graphs in time $O^*(1.680^n \log^2 L \log\log L)$. The algorithm uses* $\operatorname{poly}(n)\log L$ *space.*

We summarise our results in Table 3.1.

The rest of this chapter is as follows. In Section 3.1.1, we introduce related work on quantum speedups for the Travelling Salesman problem when the graphs are of bounded degree. Then, in Section 3.3, we describe how this technique can be used to accelerate classical algorithms of Xiao and Nagamochi for graphs of degree at most 4 [65, 66]. In Section 3.4, we extend this approach to graphs of degree at most 6.

3.1.1 Related work

Surprisingly little work has been done on quantum algorithms for the TSP. Dörn [149] proposed a quantum speedup for the TSP for degree-3 graphs by applying amplitude amplification [150] and quantum minimum finding [114] to Eppstein's algorithm, and stated a quadratic reduction in the runtime. However, we were not able to reproduce this result (see Section 3.2 for a discussion).

Very recently, Mandrà, Guerreschi and Aspuru-Guzik [151] developed a quantum algorithm for finding a Hamiltonian cycle in time $O(2^{(k-2)n/4})$ in a graph where *every* vertex has degree k. Their approach reduces the problem to an Occupation problem, which they solve via a backtracking process accelerated by the quantum backtracking algorithm [67]. The bounds obtained from their algorithm are $O(1.189^n)$ for $k = 3$ and $O(1.414^n)$ for $k = 4$, in each case a bit slower than the runtimes of our algorithms; for $k \geq 5$, their algorithm has a slower runtime than Björklund's classical algorithm [148].

3.2 Backtracking algorithms for the TSP

The intuition behind why backtracking is a useful technique for solving the TSP is that we can attempt to build up a Hamiltonian cycle by determining for each edge in the graph whether it should be included in the cycle ("forced"), or deleted from the graph. As we add more edges to the cycle, we may either find a contradiction (e.g. produce a non-Hamiltonian cycle) or reduce the graph to a special case that can be handled efficiently (e.g. a collection of disjoint cycles of four unforced edges [64]). We can also apply a number of simplifications which can be found in polynomial time. For example, if a vertex is of degree 2, then both edges incident to that vertex must be in the Hamiltonian cycle. Similarly if a vertex of degree 3 is incident to two edges already included in the Hamiltonian cycle, the third edge must not be in the Hamiltonian cycle and can therefore be removed. And once a vertex of degree 2 is incident to two forced edges, both edges can be reduced to a single forced edge. There are also more technical reductions described in Section 3.3.1. These can allow us to prune the backtracking tree substantially.

To analyse the performance of backtracking algorithms for the TSP, a problem size measure is often defined that is at least 0 and at most n (e.g. the number of vertices minus the number of forced edges). Note that if there are more than n forced edges then it is impossible to form a Hamiltonian cycle that includes every forced edge, so the number of forced edges is at most n. At the start of the backtracking algorithm, there are no forced edges so the problem size is n. Each step of the backtracking algorithm reduces the problem size until the size is 0, at which point either the n forced edges form a Hamiltonian cycle or a Hamiltonian cycle that includes every forced edge cannot be found. A quasiconvex programming problem can be developed based on how the backtracking algorithm reduces the problem size. Solving this quasiconvex problem determines the number of recursive calls the backtracking algorithm needs to make before the problem size has been reduced to 0. This is a runtime for the algorithm in terms of the problem size, which can be re-written in terms of n due to the problem size being at most n.

It was proposed by Dörn [149] that amplitude amplification could be applied to speed up the runtime of Eppstein's algorithm for the TSP on degree-3 graphs [64] from $O^*(2^{n/3}\log L)$ to $O^*(2^{n/6}\log L)$. Amplitude amplification can be used in this setting by associating a bit-string with each sequence of choices of whether to force or delete an edge, and searching over bit-strings to find the shortest valid Hamiltonian cycle. However, as suggested by the discussion in Section 2.3.3, a difficulty with this approach is that some branches of the recursion, as shown in Figure 3.1, only reduce the problem size by 2 (as measured by the number of vertices n, minus the number of forced edges). The longest branch of the recursion can, as a result, be more than $n/3$ levels deep. In the worst case, this depth could be as large as $n/2$ levels. Specifying the

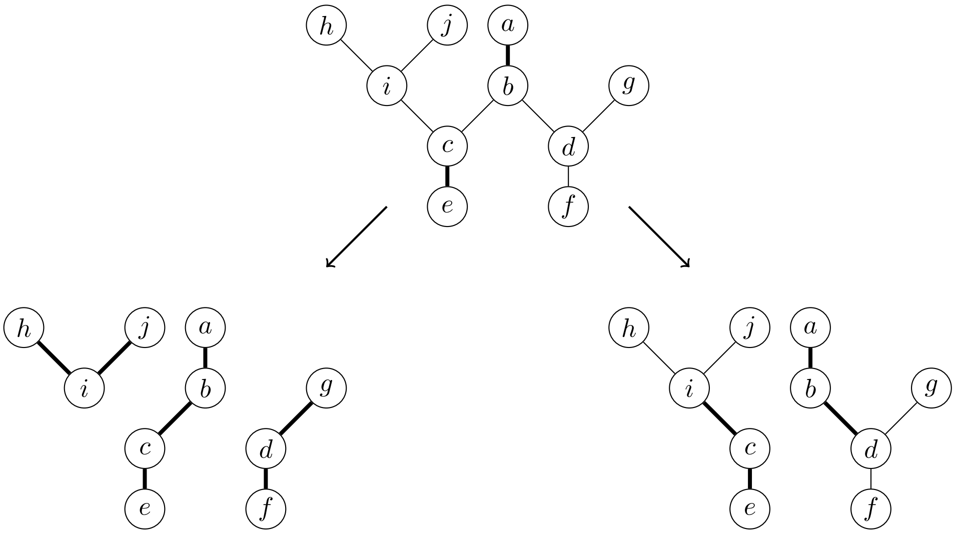


Figure 3.1: An instance of the recursive step in Eppstein's backtracking algorithm for the TSP [64] for a subgraph of a larger graph G, with forced edges displayed in bold and branching on edge bc. If we force bc, then b and c are both incident to two forced edges, so bd and ci cannot be part of the Hamiltonian cycle and can be removed from the graph. After these edges are removed, vertices i and d are both of degree 2, so in order to reach those vertices the edges hi, ij, df and dg must also be included in the Hamiltonian cycle. So forcing bc has overall added five edges to the Hamiltonian cycle. On the other hand, if we remove edge bc, we find that b and c are vertices of degree 2, so edges bd and ci must be part of the Hamiltonian cycle. Thus we have only added two more edges to the Hamiltonian cycle.

input to the checking function f could then require up to $n/2$ bits, giving a search space of size $O(2^{n/2})$. Under these conditions, searching for the solution via amplitude amplification could require up to $O^*(2^{n/4}\log L)$ time in the worst case. To yield a better runtime, we must take more of an advantage of the structure of our search space to avoid instances which will definitely not succeed.

The same issue with amplitude amplification applies to other classical algorithms for the TSP which are based on backtracking [65, 66]. In the case of the Xiao-Nagamochi algorithm for degree-3 graphs, although the overall runtime bound proven for the problem means that the number of vertices in the tree is $O(2^{3n/10})$, several of the branching vectors used in their analysis have branches that reduce the problem size by less than 10/3, leading to a branch in the tree that could be more than $3n/10$ levels deep.

3.3 Quantum speedups for the Travelling Salesman Problem on bounded-degree graphs

Our algorithms are based on applying the quantum algorithm for backtracking (Theorem 2.1) to Xiao and Nagamochi's algorithm for solving the TSP for degree-3 graphs [65]. Before describing our algorithms, we need to introduce some terminology from [65] and describe their original algorithm. The algorithm, and its analysis, are somewhat involved, so we omit details wherever possible.

3.3.1 The algorithm of Xiao and Nagamochi

A graph G is k-edge connected if there are k edge-disjoint paths between every pair of vertices. An edge in G is said to be forced if it must be included in the final tour, and unforced otherwise. The set of forced edges is denoted F, and the set of unforced edges is denoted U. An induced subgraph of unforced edges which is maximal and connected is called a U-component. If a U-component is just a single vertex, then that U-component is trivial. A maximal sequence $\mathcal{C}$ of edges in a U-component H is called a circuit if either:

- $\mathcal{C} = \{xy\}$ and there are three edge-disjoint paths from x to y,
- or $\mathcal{C} = \{c_0, c_1, \ldots, c_{m-1}\}$ such that for $0 \leq i < m-1$, there is a subgraph B_i of H such that the only two unforced edges incident to B_i are c_i and c_{i+1}.

A circuit is reducible if subgraph B_i for some i is incident to only two edges. In order for B_i to be reached, both edges incident to B_i need to be forced. Forcing one edge in the circuit then means that the other edges can be either forced or removed. The polynomial time and space process by Xiao and Nagamochi to reduce circuits, by forcing and removing alternating edges in the circuit, is known as the *circuit procedure* [65]. An example of the circuit procedure is shown in Figure 3.2, applied to a circuit of six vertices. In this case, there are two possible ways this circuit could be reduced, by forcing or removing alternating edges in the graph. In order to find the shortest Hamiltonian cycle in the graph, both reductions of this circuit must be considered and the shortest Hamiltonian cycles of each case compared. We refer to this process as “branching on a circuit”, and will describe it more formally later in this section.

Note that each edge can be in at most one circuit. If two distinct circuits $\mathcal{C}, \mathcal{C}'$ shared an edge e_i, then there are two possibilities. The first is that there is a subgraph B_i incident to unforced edges $e_i \in \mathcal{C} \cap \mathcal{C}', e_{i+1} \in \mathcal{C} - \mathcal{C}', e_j \in \mathcal{C}' - \mathcal{C}$. In this case, B_i is incident to more than two unforced edges, so neither $\mathcal{C}$ nor $\mathcal{C}'$ are circuits, which is a contradiction.

The second is that there is some edge e_i which is incident to distinct subgraphs B_i, B_i' related to $\mathcal{C}, \mathcal{C}'$, respectively. Circuits are maximal sequences, so it cannot be the case that B_i is a subgraph of B_i', otherwise $\mathcal{C}' \subseteq \mathcal{C}$. Now we consider the subgraphs $B_i \cap B_i'$ and $B_i - B_i'$, which must be connected by unforced edges as they are both subgraphs of B_i. These unforced edges are incident to B_i', which is a contradiction as they are not part of $\mathcal{C}'$.

Let X be a subgraph. We define $\text{cut}(X)$ to be the set of edges that connect X to the rest of the graph. If $|\text{cut}(X)| = 3$, then we say that X is 3-cut reducible. It was shown by Xiao and Nagamochi [65] that, if X is 3-cut reducible, X can be replaced with a single vertex of degree 3 with outgoing edges weighted such that the length of the shortest Hamiltonian cycle is preserved. The simplest example of this reduction is when X is a triangle graph, as shown in Figure 3.3a. In this case, a triangle with edge weights a, b, c incident to edges with weights x, y, z can be reduced to a single vertex by noting that to visit every vertex via edge x (resp. y, z), we also need to travel via edge c (resp. a, b).

The definition of 4-cut reducible is more complex. Let X be a subgraph such that $\text{cut}(X) \subseteq F$ and $|\text{cut}(X)| = 4$. A solution to the TSP would have to partition X into two disjoint paths such that every vertex in X is in one of the two paths. If x_1, x_2, x_3 and x_4 are the four vertices in X incident to the four edges in $\text{cut}(X)$, then there are three ways these paths could start and end:

- $x_1 \leftrightarrow x_2$ and $x_3 \leftrightarrow x_4$,

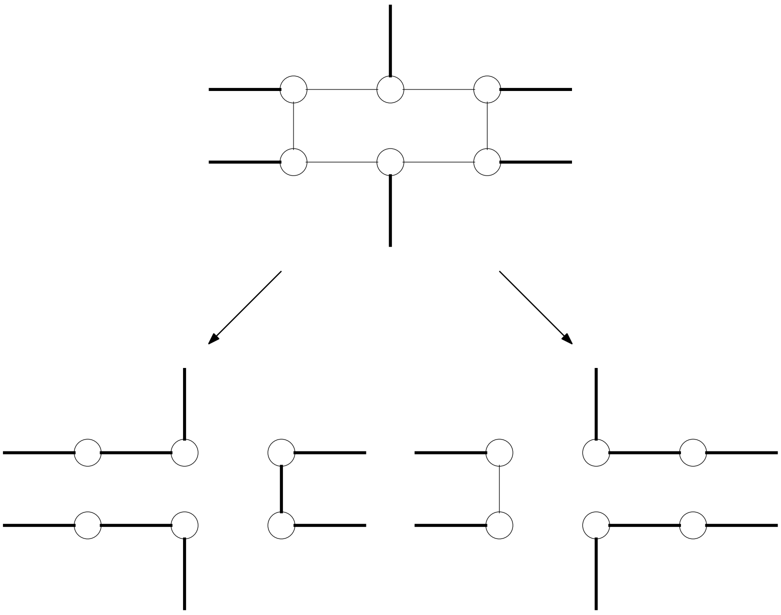


Figure 3.2: An example of the circuit procedure, and branching on a circuit, applied to a circuit of six vertices.

- $x_1 \leftrightarrow x_3$ and $x_2 \leftrightarrow x_4$,
- or $x_1 \leftrightarrow x_4$ and $x_2 \leftrightarrow x_3$.

If all three cases are not possible then a Hamiltonian cycle cannot be found. We say that X is 4-cut reducible if, for at least one of these cases, it is impossible to create two disjoint paths in X that include all vertices in X. If X is 4-cut reducible then we can reduce X to either a pair of forced paths (which can in turn be reduced to a pair of forced edges) or to a cycle of four unforced edges. An example of a 4-cut reducible graph with six vertices is shown in Figure 3.3b. In this case, note that the only possible way of visiting every vertex is to force the paths $x_1 \leftrightarrow y_1 \leftrightarrow x_2$ and $x_3 \leftrightarrow y_2 \leftrightarrow x_4$.

Xiao and Nagamochi defined a polynomial time and space process for applying the above reductions for 3-cut and 4-cut reducible graphs, known as 3/4-*cut reduction* [65].

A set of edges $\{e_i\}$ are *parallel* if they are incident to the same vertices (note that here we implicitly let G be a multigraph; these may be produced in intermediate steps of the algorithm). If there are only two vertices in the graph, then the TSP can be solved directly by forcing the shortest two edges. Otherwise if at least one of the edges is not forced, then we can reduce the problem by removing the longer unforced edges until the vertices are only adjacent via one edge. This is the process Xiao and Nagamochi refer to as *eliminating parallel edges* [65]. An example of this process is shown in Figure 3.3c, where we are able to reduce two parallel edges to a single edge.

Finally, a graph is said to satisfy the parity condition if every U-component is incident to an even number of forced edges and for every circuit $\mathcal{C}$, an even number of the corresponding subgraphs B_i satisfy that $|\text{cut}(B_i) \cap F|$ is odd. An example of a circuit which does not satisfy the parity condition is given in Figure 3.3d. Note that this circuit is incident to five forced edges, so no possible assignment of edges in this circuit will produce a valid Hamiltonian cycle.

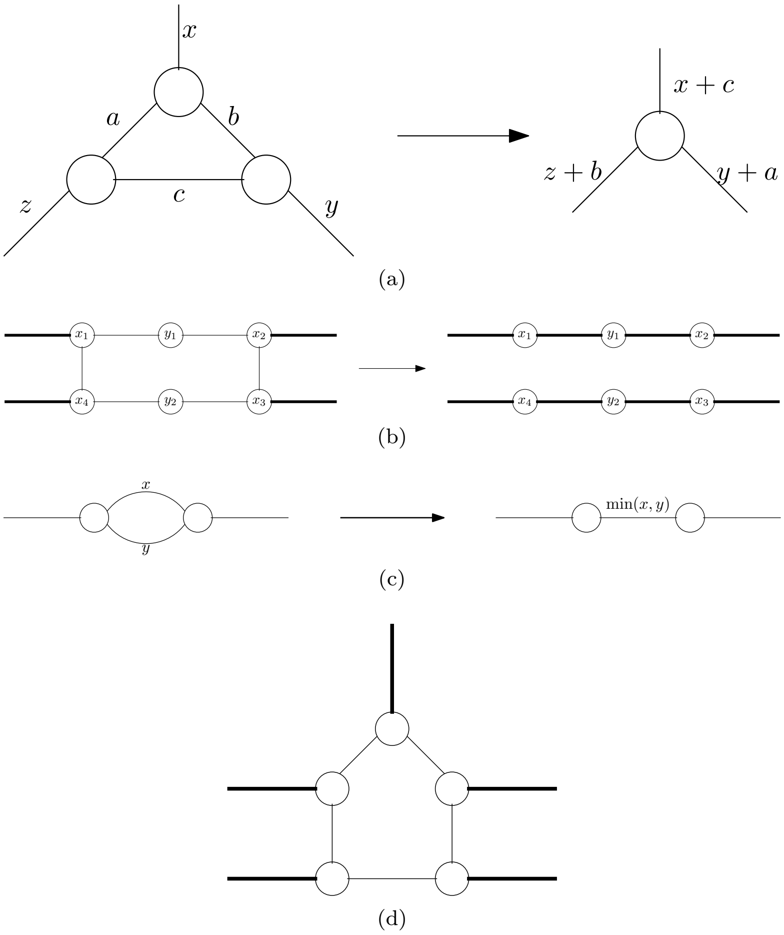


Figure 3.3: Some example simplifications used in the Xiao-Nagamochi algorithm. 3.3a: An example of a 3-cut reducible graph being reduced to a single point. 3.3b: An example of a 4-cut reducible graph being reduced to a pair of forced paths. 3.3c: An example of parallel edges being reduced to a single edge. 3.3d: An example of a circuit which fails the parity condition.

We are now ready to describe Xiao and Nagamochi's algorithm. The algorithm takes as input a graph $G = (V, E)$ and a set of forced edges $F \subseteq E$ and returns the length of the shortest Hamiltonian cycle in G containing all the edges in F, if one exists.

The algorithm is based on four subroutines: *eliminating parallel edges*, the *3/4-cut reduction*, *selecting a good circuit* and the *circuit procedure*, as well as the following lemma.

Lemma 3.1 (Eppstein [64]). *If every U-component in a graph G is trivial or a component of a 4-cycle, then a minimum cost tour can be found in polynomial time.*

One example of a graph G and set of forced edges F which satisfies the above lemma is when $G \setminus F$ is a collection of disjoint cycles of length 4. Lemma 3.1 explains that for such G and F, the Travelling Salesman Problem can be efficiently solved in polynomial time on a classical computer.

We will not define the subroutines here in any detail; for our purposes, it is sufficient to assume that they all run in polynomial time and space. The circuit procedure for a circuit $\mathcal{C}$ begins by either adding an edge $e \in \mathcal{C}$ to F or deleting it from the graph, then performing some other operations. "Branching on a circuit $\mathcal{C}$ at edge $e \in \mathcal{C}$" means generating two new instances from the current instance by applying each of these two variants of the circuit procedure starting with e. Note that this branching on a circuit step is where the exponential runtime of the algorithm comes from, as we are needing to solve two instances of the problem and compare them. This is the step that we are primarily concerned with improving via the quantum backtracking algorithm. An example of branching on a circuit is given in Figure 3.2.

The Xiao-Nagamochi algorithm is described in Algorithm 3.1, reproduced from [65]. Lines 2 and 3 check that the existence of a Hamiltonian cycle is not ruled out, by ensuring that that there are at least two disjoint paths between any pair of vertices and that the graph satisfies the parity condition. Lines 4 and 5 reduce any reducible circuit by initially forcing one edge and then alternately removing and forcing edges. Lines 6 & 7 remove any parallel edges from the graph, and lines 8 & 9 remove any circuits of three edges as well as setting up circuits of four edges so that all edges incident to them are forced. Lines 10–12 are the recursive step, branching on a good circuit by either forcing or removing an edge in the circuit and then applying the circuit procedure. The algorithm continues these recursive calls until it either finds a Hamiltonian cycle or $G \setminus F$ is a collection of single vertices and cycles of length 4, all of which are disjoint from one another, at which point the problem can be solved in polynomial time via lines 14 and 15.

Xiao and Nagamochi looked at how the steps of the algorithm, and in particular the branching step, reduced the size of the problem for different graph structures. From this they derived a quasiconvex program corresponding to 19 branching vectors, each describing how the problem size is reduced at the branching step in different circumstances. Analysis of this quasiconvex program showed that the algorithm runs in $O^*(2^{3n/10} \log L)$ time and polynomial space [65].

3.3.2 Quantum speedup of the Xiao-Nagamochi algorithm

Here we describe how we apply the quantum backtracking algorithm to the Xiao-Nagamochi algorithm. At a high level, we implement the Xiao-Nagamochi algorithm as predicate and heuristic oracles for the quantum backtracking algorithm. These are the steps that will be run on the quantum computer. The quantum backtracking algorithm does not necessarily return the shortest Hamiltonian cycle, but instead returns a randomly selected Hamiltonian cycle that it found. Adding constraints on the length of the Hamiltonian cycles to our predicate and running the quantum backtracking algorithm multiple times will allow us to find a solution to the TSP. Each run of the backtracking algorithm to find a Hamiltonian cycle of acceptable length is run on the quantum computer, with the range of acceptable lengths chosen classically via binary search.

```
Function TSP3($G$, $F$) is
    Input: A graph $G = (V, E)$, a set of forced edges $F \subseteq E$
    Output: The length of the shortest Hamiltonian cycle which includes all edges in $F$
    if $G$ is not 2-edge-connected or the instance violates the parity condition then
        Return $\infty$
    else if there is a reducible circuit $\mathcal{C}$ then
        Return TSP3($G'$, $F'$) for an instance $(G', F')$ obtained by applying the circuit
          procedure on $\mathcal{C}$ started by adding a reducible edge in $\mathcal{C}$ to $F$
    else if there is a pair of parallel edges then
        Return TSP3($G'$, $F'$) for an instance $(G', F')$ obtained by applying the reduction
          rule of eliminating parallel edges
    else if there is a 3/4-cut reducible subgraph $X$ containing at most eight vertices then
        Return TSP3($G'$, $F'$) for an instance $(G', F')$ obtained by applying the 3/4-cut
          reduction on $X$
    else if there is a $U$-component $H$ that is neither trivial nor a 4-cycle then
        Select a good circuit $\mathcal{C}$ in $H$
        Return $\min\{$TSP3($G_1$, $F_1$), TSP3($G_2$, $F_2$)$\}$, where $(G_1, F_1)$ and $(G_2, F_2)$ are the
          two resulting instances after branching on $\mathcal{C}$
    else
        /* each $U$-component of the graph is trivial or a 4-cycle          */
        Solve the problem directly in polynomial time by Lemma 3.1
        Return the cost of an optimal tour
    end
end
```

Algorithm 3.1: The Xiao-Nagamochi algorithm for solving the TSP on degree-3 graphs.

The first step towards applying the quantum backtracking algorithm is to define the set of partial assignments. A partial assignment will be a list of edges in G ordered by when they are assigned in the backtracking algorithm and paired with whether the assignment was to force or remove the edge. The assignment is denoted $A \in (\{1, \dots, m\}, \{\text{force}, \text{remove}\})^j$, where $j \leq m$. We have $m \leq 3n/2$ as G is degree-3.

The quantum approach to backtracking requires us to define a predicate `P3` and heuristic `h3`, described by Algorithms 3.2 and 3.3, respectively, and each taking as input a partial assignment. Our predicate and heuristic make use of a reduction function, introduced in [65], as a subroutine; this function is denoted `Reduce3` and described as Algorithm 3.4 in Section 3.3.3. However it may be worth noting that the algorithm uses the original graph G, and partial assignments of it at each stage.

Firstly, we describe the predicate in Algorithm 3.2. Lines 3 & 4 match lines 2 and 3 of Xiao and Nagamochi's algorithm. Lines 5 and 6 are where the same conditions are met as in lines 14 and 15 of Xiao and Nagamochi's algorithm, where a shortest length Hamiltonian cycle is guaranteed to exist and can be found in polynomial time classically via Lemma 3.1. The rest of Algorithm 3.2 continues the branching process, which together with how the circuit is picked by `h3` and the use of the circuit procedure in `Reduce3` matches the branching step of Xiao and Nagamochi.

```
1  Function P3($A$) is
       Input: A partial assignment $A = ((e_1, A_1), \dots, (e_j, A_j))$ describing edges that have
              been forced or removed
       Output: True, false, or indeterminate depending on if a Hamiltonian cycle can be
               found
2      Using the partial assignment $A$, run Reduce3($G$, $F$) to get $(G', F')$
3      if $G$ is not 2-edge-connected or fails the parity condition then
4          Return false
5      else if Every $U$-component in $G'$ is either trivial or a 4-cycle then
6          Return true
7      else
8          Return indeterminate
9      end
10 end
```

Algorithm 3.2: The predicate function for the Xiao-Nagamochi algorithm for degree-3 graphs.

The heuristic is described in Algorithm 3.3, taking as input a partial assignment $A = ((e_1, A_1), \dots, (e_j, A_j))$ of the edges of G. Here we apply the same branching strategy as Xiao and Nagamochi's algorithm, by selecting the next circuit to branch on and picking an edge in that circuit. If the reduced version of the graph results in `h3` picking an edge corresponding to multiple edges in the original graph, line 4 ensures that we only return one of these edges to the backtracking algorithm, as the reduction function will ensure that every edge in the original graph corresponding to an edge in the reduced graph will be consistently forced or removed. The rest of the circuit will be forced or removed by line 26 of the reduction function (Algorithm 3.4).

We can now apply the backtracking algorithm (Theorem 2.1) to `P3` and `h3` to find a Hamiltonian cycle. At the end of the algorithm, we will receive either the information that no assignment was found, or a partial assignment. By applying the reduction steps and the partial assignments, we can reconstruct the graph at the moment our quantum algorithm terminated, which will give a graph such that every U-component is either trivial or a 4-cycle. We then construct

```
Function h3($A$) is
    Input: A partial assignment $A = ((e_1, A_1), \ldots, (e_j, A_j))$ describing edges that have
           been forced or removed
    Output: The next edge to force or remove from $G$
    Using the partial assignment $A$, run Reduce3($G$, $F$) to get $(G', F')$
    Select a $U$-component in $G'$ that is neither trivial nor a cycle of length 4. Select a
      circuit $\mathcal{C}$ in that component that fits the criteria of a "good" circuit [65], then select
      an edge $e'_i \in \mathcal{C}$
    Return an edge in $G$ corresponding to $e'_i$
    /* if there is more than one edge corresponding to $e'_i$, we can choose
       one arbitrarily                                                        */
end
```

Algorithm 3.3: The heuristic function for the Xiao-Nagamochi algorithm for degree-3 graphs.

and return the full Hamiltonian cycle in polynomial time using step 6 of Xiao and Nagamochi's algorithm [65].

To solve the TSP, we need to find the shortest Hamiltonian cycle. This can be done as follows. First, we run the backtracking algorithm. If the backtracking algorithm does not return a Hamiltonian cycle then we report that no Hamiltonian cycle was found. Otherwise after receiving Hamiltonian cycle Γ with length L_Γ, we create variables $\ell \leftarrow 0$ & $u \leftarrow L_\Gamma$ and modify the predicate to return false if

$$\sum_{e_{i,j} \in F} c_{ij} \geq \lceil (\ell + u)/2 \rceil. \tag{3.1}$$

If no cycle is found after running the algorithm again, we set $\ell \leftarrow \lceil (\ell + u)/2 \rceil$ and repeat. Otherwise, upon receiving Hamiltonian cycle Γ' with total cost $L_{\Gamma'}$, we set $u \leftarrow L_{\Gamma'}$ and repeat. We continue repeating until ℓ and u converge, at which point we return the Hamiltonian cycle found by the algorithm. The scenario that will give the longest runtime is when the shortest cycle is found during the first run of the backtracking algorithm: The backtracking algorithm will fail to find a Hamiltonian cycle shorter than $\lceil (\ell + u)/2 \rceil$, update ℓ and repeat until ℓ and u converge. In this case, this algorithm matches a binary search. So the number of repetitions of the backtracking algorithm required to return the shortest Hamiltonian cycle is at most $O(\log L')$, where

$$L' = \sum_{i=1}^{n} \max\{c_{ij} : j \in \{1, \ldots, n\}\} \tag{3.2}$$

is an upper bound on the total cost of any Hamiltonian cycle in the graph. Note that we can bound this quantity in terms of the longest edge weight:

$$\begin{aligned} \log L' &\leq \log(nL) && (3.3) \\ &= \log n + \log L && (3.4) \\ &= O^*(\log L). && (3.5) \end{aligned}$$

In order for the overall algorithm to successfully find the shortest Hamiltonian cycle we need every run of the backtracking algorithm to successfully determine whether or not a Hamiltonian cycle exists in the specified range. By use of a union bound we can see that if the probability of the backtracking algorithm failing is δ, the probability of the overall algorithm failing is at most $\delta \log L'$. Therefore, choosing δ to be at most $O(1/\log L')$ is sufficient to ensure a bounded probability of the overall algorithm failing.

3.3.3 The reduction function

Finally, we describe the reduction function, which takes the original graph G and partial assignment A, and applies the partial assignment to this graph in order to reduce it to a smaller graph G' with forced edges F'. This reduction might mean that forcing or removing a single edge in G' would be akin to forcing several edges in G. For example, let X be a 3-reducible subgraph of at most 8 vertices with $\mathrm{cut}(X) = \{ax_1, bx_2, cx_3\}$ for vertices $x_1, x_2, x_3 \in V(X)$. The 3/4-cut reduction reduces X to a single vertex $x \in G'$ with edges ax, bx, cx. If the edges ax and bx are forced, this is equivalent to forcing every edge in $\Pi \cup \{ax_1, bx_2\}$, where Π is the shortest path that starts at x_1, visits every vertex in X exactly once, and ends at x_2. As we need to solve the problem in terms of the overall graph G and not the reduced graph G', our assigned variables need to correspond to edges in G. To do this, our `h3` function described in Sec. 3.3.2 includes a step where if the edge selected in G' corresponds to multiple edges in G, we simply select one of the corresponding edges in G to return. Likewise, if the next edge in our partial assignment is one of several edges in G corresponding to a single edge in G', we apply the same assignment to all of the other corresponding edges in G.

The reduction function is described in Algorithm 3.4, using reductions and procedures from Xiao and Nagamochi [65]. This function applies the edge assignments made so far, as well as any possible reductions at each step. Lines 5 and 6 case recreate lines 4 and 5 from Xiao and Nagamochi's original algorithm by applying the circuit procedure where possible. Steps 7 and 8 recreate steps 6 and 7 of the original algorithm by applying the reduction of parallel edges. And steps 9 & 10 case recreate steps 8 and 9 of the original algorithm via the 3/4-cut reduction. We then apply the next step of the branching that has been performed so far, to ensure that the order in which the edges are forced is the same as in the classical algorithm, followed by branching on a circuit at edge e_i via the circuit procedure. Finally, we check whether or not the graph can be reduced further by running the reduction steps again for edge j.

One might ask if an edge could be part of two circuits, in which case our algorithm would fail as it would not be able to reduce the circuit. However, as discussed in Sec. 3.3.1, any edge can only be part of at most one circuit.

3.3.4 Analysis

All procedures in the reduction algorithm can be completed in polynomial time in n and $\log L$ [65]. All of these steps also reduce the size of a problem by at least a constant amount, so only a polynomial number of these steps are needed. Step 2(b) is constant time and step 2(c) can be run in polynomial time as the circuit is now reducible. All steps are only repeated $O(m)$ times, so the whole reduction algorithm runs in polynomial time in terms of m.

The remaining steps in the heuristic subroutine run in polynomial time as searching for a good circuit in a component can be done in polynomial time [65]. Likewise, remaining steps in the predicate function involve looking for certain structures in the graph that can be found in polynomial time. As a result, the runtimes for the `P3` and `h3` functions are both polynomial in m.

By Theorem 2.1, the number of calls to `P3` and `h3` we make in order to find a Hamiltonian cycle with failure probability δ is $O(\sqrt{T}\,\mathrm{poly}(m)\log(1/\delta))$, where T is the size of the backtracking tree, which in our case is equal to the number of times the Xiao-Nagamochi algorithm branches

```
1  Function Reduce3($G$, $F$) is
       Input: A graph $G = (V, E)$, a partial assignment of edges $A$
       Output: A reduced graph $G'$ and set of forced edges $F'$
2      Create a copy of the graph $G' \leftarrow G$ and set of forced edges $F' \leftarrow \emptyset$
3      for $i = 1, \dots, j$ do
4          repeat
5              if $G'$ contains a reducible circuit $\mathcal{C}$ then
6                  Apply the circuit procedure to $\mathcal{C}$
7              else if $G'$ contains parallel edges then
8                  Apply the reduction rule of eliminating parallel edges
9              else if $G'$ contains a subgraph $X$ of at most 8 vertices such that $X$ is 3/4-cut
                reducible then
10                 Apply the 3/4-cut reduction to $X$
11             end
12         until None of the cases apply
13         if $A_i = force$ then
14             if $e_i$ is in a set of edges corresponding to a single reduced edge in $e'_i \in G'$ then
15                 $F' \leftarrow F' \cup \{e'_i\}$
16             else
17                 $F' \leftarrow F' \cup \{e_i\}$
18             end
19         else
               /* $A_i$ = remove                                                  */
20             if $e_i$ is in a set of edges corresponding to a single reduced edge in $e'_i \in G'$ then
21                 $G' \leftarrow G' - \{e'_i\}$
22             else
23                 $G' \leftarrow G' - \{e_i\}$
24             end
25         end
26         Apply the circuit procedure to the rest of the circuit containing edge $e_i$
27     end
28     Repeat lines 4–12 until no further reductions can be applied
29     Return $(G', F')$
30 end
```

Algorithm 3.4: The reduction function for degree-3 graphs.

on a circuit. `P3` and `h3` both run in polynomial time and as a result can be included in the $\mathrm{poly}(m)$ term of the runtime. Because $m \leq 3n/2$, the polynomial term in this bound is also polynomial in terms of n.

The behaviour of the `P3` and `h3` subroutines is designed to reproduce the behaviour of Xiao and Nagamochi's TSP3 algorithm [65]. It is shown in [65, Theorem 1] that this algorithm is correct, runs in time $O^*(2^{3n/10})$ and uses polynomial space. As the runtime of the TSP3 algorithm is an upper bound on the number of branching steps it makes, the algorithm branches on a circuit $O^*(2^{3n/10})$ times. Therefore, the quantum backtracking algorithm finds a Hamiltonian cycle, if one exists, with failure probability at most δ in time $O^*(2^{3n/20} \log L \log(1/\delta)) \approx O^*(1.110^n \log L \log(1/\delta))$ and polynomial space.

Finding the shortest Hamiltonian cycle requires repeating the algorithm $O(\log L')$ times, where L' is given in Equation 3.2. As mentioned in Section 3.3.2, for all runs succeed with high probability it is sufficient for the failure probability δ of each run to be at most $O(1/(\log L'))$. From this we obtain the following result, proving the first part of Theorem 3.1:

Theorem 3.3. *There is a bounded-error quantum algorithm which solves the TSP on degree-3 graphs in time $O^*(1.110^n \log^2 L \log\log L)$, where L is the maximum edge cost. The algorithm uses* $\mathrm{poly}(n)$ *space.*

3.4 Extending to higher-degree graphs

We next consider degree-k graphs for $k \geq 4$. We start with degree-4 graphs by applying the quantum backtracking algorithm to another algorithm by Xiao and Nagamochi [66]. We then extend this approach to graphs of higher degree by reducing the problem to degree-4 graphs.

3.4.1 Degree-4 graphs

Here we will show the following, which is the second part of Theorem 3.1:

Theorem 3.4. *There is a bounded-error quantum algorithm which solves the TSP for degree-4 graphs in time $O^*(1.301^n \log^2 L \log\log L)$, where L is the maximum edge cost. The algorithm uses* $\mathrm{poly}(n)$ *space.*

As the argument is very similar to the degree-3 case, we only sketch the proof.

Proof sketch. Xiao and Nagamochi's algorithm for degree-4 graphs works in a similar way to their algorithm for degree-3 graphs. Indeed, the predicate function considers largely the same cases as in Algorithm 3.2:

Lemma 3.2 (Xiao and Nagamochi [66])**.** *Let $G = (V, E)$ be a graph and $F \subseteq E$ a set of forced edges. There is no Hamiltonian cycle that visits every edge in F if:*

1. *G is not 2-edge-connected;*

2. *a vertex v is incident to more than 2 edges in F;*

3. *(V, F) contains a non-Hamiltonian cycle;*

4. *or there exists a U-component incident to an odd number of edges in F.*

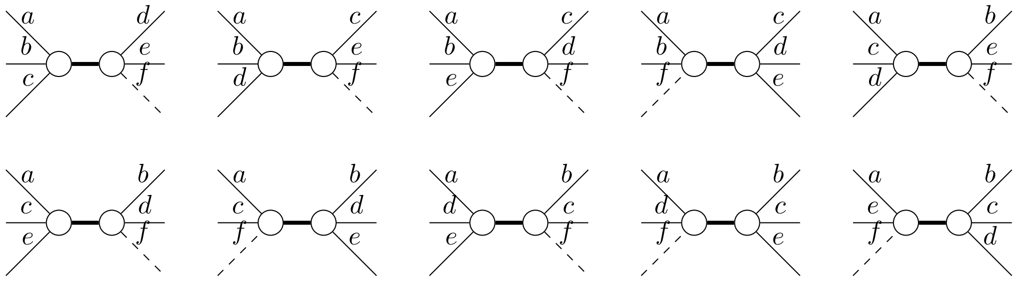


Figure 3.4: Breaking a vertex of degree 5 or 6 into two lower-degree vertices. In the degree-5 case, dashed edge f is not present and the vertex is split into one vertex of degree 3 and another of degree 4 connected by a forced edge in bold. In the degree-6 case, dashed edge f is present and the vertex is split into two vertices of degree 4 connected by a forced edge. If edges a and b are included in the original graph's shortest Hamiltonian cycle, then they must not be adjacent to one another in the final graph. This holds in six of the ten ways of splitting the vertex.

Cases 1 and 4 are already considered in Algorithm 3.2, with case 4 forming part of the parity condition. The algorithm also only returns an definite solution in the same way as the degree 3 case, via Lemma 3.1. As a result, our new predicate only needs to care about cases 2 and 3, both of which can be checked in polynomial time.

The reductions utilised are also simple to observe and related to those used in Algorithm 3.4. The first two are removing unforced edges incident to any vertex v where v is incident to two forced edges, and forcing any edges incident to a vertex of degree 2, both of which are similar to what is already performed by the circuit procedure. Likewise, the fourth reduction process is reducing triangles in the graph to a single vertex, akin to the 3-cut reduction from before.

The third reduction process is more involved, but can still be computed in polynomial time. This reduction checks for each edge e if it is a bridge, an edge that would split a U-component H in the graph into two disconnected subgraphs H_1 and H_2. If e is a bridge, then it is either forced or removed depending on if H_1 is incident to and odd or even number of forced edges, respectfully. If H is incident to an even number of forced edges, then so are the subgraphs H_1 and H_2. Note that this might leave H_2 disconnected from the rest of the graph, but this would violate the 2-edge-connected rule and therefore be rejected by the predicate.

The heuristic is more involved than previously, but essentially consists of selecting a degree-4 vertex v incident to a forced edge and branching on an edge incident to that vertex. If no such vertex exists then we select a degree-3 vertex v incident to a forced edge and branch on an edge incident to a vertex at most two edges away from v. We shall omit details on how the branching edge is chosen for simplicity, but note that these conditions can be computed in polynomial time.

We apply the quantum backtracking algorithm as before, finding a Hamiltonian cycle with failure probability δ in $O^*(1.301^n \log L \log(1/\delta))$ time. We then use binary search to find the shortest Hamiltonian cycle after $O(\log L)$ repetitions of the algorithm, rejecting if the total length of the forced edges is above a given threshold. To achieve overall failure probability 1/3, the algorithm runs in $O^*(1.301^n \log^2 L \log\log L)$ time. □

3.4.2 Degree-5 and degree-6 graphs

To deal with degree-5 and degree-6 graphs, we reduce them to the degree-4 case. The complexity of the two cases turns out to be the same; however, for clarity we consider each case separately.

Theorem 3.5. *There is a bounded-error quantum algorithm which solves the TSP for degree-5 graphs in time $O^*(1.680^n \log^2 L \log\log L)$.*

Proof. Our algorithm works by splitting each vertex of degree 5 into one vertex of degree 3 and another of degree 4 connected by a forced edge. The forced edges can be included in our quantum algorithm by modifying step 1 of the reduction function so that F' contains all the forced edges created by splitting a vertex of degree-5 into two vertices connected by a forced edge. Once all degree-5 vertices are split this way, we run the degree-4 algorithm. It is intuitive to think that this splitting of the vertices could increase the runtime complexity of the degree-4 algorithm, due to n being larger. However, the addition of a forced edge incident to every new vertex means that we do not need to create more branches in the backtracking tree in order to include the new vertex in the Hamiltonian cycle. As a result, the time complexity of the degree-4 algorithm will remain the same.

There are 10 unique ways of splitting a vertex of degree 5 into one vertex of degree 3 and another of degree 4 connected by a forced edge. These ten ways of splitting the vertex are shown in Fig. 3.4 for a vertex incident to edges a, b, c, d, e. Without loss of generality, let a and b be the two edges which are part of the Hamiltonian cycle. In order for a and b to also be part of the Hamiltonian cycle in the degree-4 graph produced, a and b cannot be adjacent to one another. Looking at Fig. 3.4, the split is successful in six of the ten ways of splitting the vertex.

If there are f vertices of degree 5, then there are 10^f possible ways of splitting all such vertices, of which 6^f will give the correct solution to the TSP. We can apply Dürr and Høyer's quantum algorithm for finding the minimum [114] to find a splitting that leads to a shortest Hamiltonian cycle, or reporting if no cycle exists, after $O((10/6)^{f/2})$ repeated calls to the degree-4 algorithm. To ensure that the failure probability of the whole algorithm is at most 1/3, we need to reduce the failure probability of the degree-4 algorithm to $O((10/6)^{-f/2})$, which can be achieved by repeating it $O(f)$ times and returning the minimum-length tour found. The overall runtime is thus

$$O^*\left(\left(\frac{10}{6}\right)^{\frac{f}{2}} 1.301^n \log L \log\log L\right) \tag{3.6}$$

$$= O^*(1.680^n \log^2 L \log\log L). \tag{3.7}$$

□

It is also possible to split a vertex of degree 5 into three vertices of degree 3 connected by two forced edges. There are 15 ways of performing this splitting, of which 6 will succeed. Applying the degree-3 algorithm to these reduced graphs finds a runtime of

$$O^*\left(\left(\frac{15}{6}\right)^{\frac{f}{2}} 1.110^n \log L \log\log L\right) \tag{3.8}$$

$$= O^*(1.754^n \log^2 L \log\log L) \tag{3.9}$$

which performs worse than Theorem 3.5. We next turn to degree-6 graphs, for which the argument is very similar.

Theorem 3.6. *There is a quantum algorithm which solves the TSP for degree-*6 *graphs with failure probability* 1/3 *in time $O^*(1.680^n \log^2 L \log\log L)$.*

Proof. We can extend the idea of Theorem 3.5 to degree-6 graphs by splitting vertices of degree 6 into two vertices of degree 4 connected by a forced edge. Because the degree of both new vertices is 4, there are $\binom{6}{3}/2 = 10$ unique ways of partitioning the edges, of which 4 will fail. We show this in Fig. 3.4 by including the dashed edge f as the sixth edge. The overall runtime is the same as the degree-5 case. □

3.4.3 Degree-7 graphs

We next consider extending the algorithm to degree-7 graphs by partitioning degree-7 vertices into one of degree 5 and another of degree 4, connected by a forced edge. We can split a vertex of degree 7 into a vertex of degree 4 and another of degree 5 in $\binom{7}{4} = 35$ ways, of which $\binom{7-2}{4-2} + \binom{7-2}{3-2} = 15$ will not preserve the shortest Hamiltonian cycle. We then use the same process as for the degree-5 and degree-6 case, halting after $O((35/20)^{k/2})$ iterations and returning either the shortest Hamiltonian cycle found or reporting if no Hamiltonian cycle exists. From this, our overall runtime is

$$O^*\left(\left(\frac{35}{20}\right)^{k/2} 1.680^n \log^2 L \log\log L\right) \tag{3.10}$$

$$=O^*(2.222^n \log L \log\log L). \tag{3.11}$$

This is the point where we no longer see a quantum speedup over the fastest classical algorithms using this approach, as classical algorithms such as those of Held-Karp [106] and Björklund et al. [107] run in $O^*(2^n)$ and $O^*(1.984^n)$ time, respectively.

3.5 Conclusion

In this work we have given examples of where quantum algorithms can provide speedups for the Travelling Salesman Problem in the NP-hard case of bounded-degree graphs. In particular, we are able to offer polynomial speedups for graphs of up to degree 4. For degree-5 and degree-6 graphs, the algorithm of [148] is polynomially faster in terms of n compared to our degree-5 and degree-6 algorithms. However, our algorithm is exponentially faster in terms of L. Therefore our algorithm offers an improved runtime in these cases when L is exponentially large in terms of n. Furthermore, the quantum computing space requirements of all our algorithms are polynomial.

3.5.1 Subsequent work

Since the initial publication of the main results in this work a number of other results along these lines have arisen independently [116, 117, 69]. We shall now discuss each of these results and how they compare to ours.

Ambainis et al. [116] used the work discussed in Section 2.3.2 to develop a related algorithm for solving the Travelling Salesman Problem, as well as similar NP-Hard problems. This algorithm works by using classical computation to compute all ways of partitioning the set of vertices into two sets of size k and $n - k$ for some k and solving the TSP in these subspaces. The quantum subroutine then uses Grover search over all these subproblem solutions to find pairings that give an optimal solution to the TSP. Ambainis et al. found that the optimal runtime is when $k = n\alpha/4$ for $\alpha \approx 0.055362$, which gives a runtime of $O^*(1.728^n \log L)$. This algorithm can be generalised to other dynamic programming algorithms as well, by generalising from the set of vertices to more general sets with some cost function.

This algorithm gives a speedup over classical algorithms such as the Held-Karp algorithm, which runs in $O(n^2 2^n \log L)$ time [106]. However, it does not perform better than our algorithm

Degree k	Quantum	Classical (exp space)	Classical (poly space)
5	$O^*(1.390^n L^4)$	$O^*(1.932^n \log L)$	$O^*(1.989^n \operatorname{poly}(L))$
6	$O^*(1.403^n L^4)$	$O^*(1.967^n \log L)$	$O^*(1.996^n \operatorname{poly}(L))$
7	$O^*(1.409^n L^4)$	$O^*(1.984^n \log L)$	$O^*(1.998 \operatorname{poly}(L))$
8	$O^*(1.411^n L^4)$	$O^*(1.991^n \log L)$	$O^*(1.999 \operatorname{poly}(L))$

Table 3.2: Runtimes of the quantum speedup achieved by applying the quantum speedup of dynamic programming by Ronagh to the exponential space bounded-degree TSP algorithm of Björklund et al. Classical runtimes given in Björklund et al. [107].

for degree below 7, nor does it perform faster than Björklund's $O*(1.657^n L)$-time algorithm, assuming that L is small [148]. Another cost with Ambainis et al.'s algorithm is that the algorithm requires an exponentially large amount of space, in order to store all optimal subpaths of length $n(1-\alpha)/4$ in a quantum random access memory.

Ronagh's speedup for dynamic programming was also applied to the Travelling Salesman Problem [117], leading to a runtime of $O^*(L^4 2^{n/2})$. This algorithm does perform better then our original algorithm, with a runtime of $O^*(2^{n/2}L^4)$, though only for degree-5 and degree-6 graphs, and even then only when the edge weights of the graph are bounded by poly(n). Furthermore, it is possible to adapt Ronagh's algorithm to provide an even better speedup for the bounded-degree case, by speeding up the exponential space algorithm of Björklund et al. [107]. Björklund et al. use the Held-Karp algorithm with an additional check to ensure that S is transient with endpoint l. This means that S is connected, the starting point of our cycle s is in S, and for every vertex v which is not a neighbour of our start point s or l, if $v \in S$ then so are at least two neighbours of v, and if $v \notin S$ then so are at least two neighbours of v. Any prefix of a Hamiltonian cycle must satisfy this property, and checking whether or not a set is transient with endpoint l can be done using depth-first search in polynomial time. Applying Ronagh's quadratic speedup would yield a runtime of

$$O^*\left(L^4\left(2^{k+1}-2k-2\right)^{n/(2k+2)}\right). \tag{3.12}$$

Runtimes for small instances of k are given in Table 3.2, with comparative runtimes from the exponential space and polynomial space algorithms by Björklund et al. [107]. It is worth noting that the quantum runtime is strictly less than $O(2^{n/2}L^4)$ for all constant k. Compared to the Björklund algorithm [148], which has a run time of $O(1.657^n L)$, this algorithm is polynomially faster in terms of n at the cost of being polynomially slower in terms of L. Also note that for $L = \operatorname{poly}(n)$, this runtime is better than the runtimes given in Theorem 3.2.

We will discuss the algorithm of Ge and Dunjko more in Section 4.1, but for the meantime note that it has a runtime of at best $O^*(2^{n/4})$, which matches the runtime of Dörn [149] given in Section 3.2 but is slower than our result.

Another relevant result that came out subsequent to the work published in this chapter is a quantum speedup for branch-and-bound algorithms by Montanaro [152]. This algorithm works by applying the quantum backtracking algorithm with a predicate that rejects branches whose cost bound exceeds a cut-off c, and then updating c via binary search. As a result, this algorithm can be seen as a generalisation of our algorithms for degree-3 and degree-4 graphs. Montanaro showed that this algorithm runs in $\tilde{O}(\sqrt{T}n^{3/2}\log c)$ time, where T is the size of the tree visited in order to find a minimal solution, n is the number of variables and c is maximal cost of a solution. In particular, Montanaro gave an example problem of searching for the ground state in most instances of the Sherrington-Kirkpatrick spin glass in $O(2^{0.226n})$ time, which is considerably faster than any speedup from Grover search.

3.5.2 Open questions

There are several interesting further questions with this research. One is the question of whether or not these algorithms can offer speedups for degree-7 graphs and beyond. On the one hand, in principle any improvements to the classical backtracking algorithms in lower-degree cases will naturally lead to an improvement in the higher-degree cases, due to the reductions. On the other hand, there is a large jump in performance required to move from $O^*(2.222^n \operatorname{polylog} L)$ to $O^*(1.984^n \log L)$, and an even larger one to beat the $O^*(1.728^n \log L)$ runtime of Ambainis et al. [116]. And even if such an improvement is made, there will eventually be a degree at which our current algorithm will not perform better, due to the exponential overhead required for the reduction.

Another interesting question is what other classical algorithms for these problems can be sped up. There are several algorithms which could be interesting in this regard. One example is cut and count algorithms which use a combination of Monte-Carlo and dynamic programming [148, 146, 147]. Both of these subroutines have quantum speedups [153, 117], but it is unclear if they can both be applied to these algorithms. Another example are branch and bound & branch and cut algorithms such [109, 110], which, as mentioned in Section 2.2.2, offer the best current classical algorithm for solving the Travelling Salesman Problem in practice [111]. Again, quantum speedups exist for these algorithms [152]. However, performance of these algorithms is evaluated by implementing and benchmarking the algorithms, rather than by analytical means. To determine speedups available would require a (logical) gate and qubit decomposition of the quantum algorithm, which is the main challenge.

Part III

An architecture-focused approach: Boson Sampling under distinguishability and loss

Chapter 4

Preliminary material: Boson Sampling and the Schur-Weyl Duality

Parts of Sections 4.3.3, 4.3.4, 4.5.1, 4.7.2 and 4.7.3 were written by myself under the supervision of Peter S. Turner, and published as "Quantum simulation of partially distinguishable boson sampling", *Physical Review A* **97**, 062329 (2018), copyright American Physics Society. Similarly, parts of Section 4.6 were written by myself under the supervision of Raúl García-Patrón, Jelmer J. Renema and Peter S. Turner, and published as "Classically simulating near-term partially-distinguishable and lossy boson sampling", *Quantum Science and Technology* **5**, 015001 (2020), copyright Institute of Physics. Preprints of these articles are freely available at `arXiv:1803.03657` and `arXiv:1907.00022`, respectively. This chapter is preliminary material and contains no original results.

In the previous part of this thesis we have considered how quantum computers can offer speedups for computationally hard problems, with a particular focus on the NP-Hard Travelling Salesman Problem. But these speedups have focused on asymptotic runtimes, rather than considering how much of a speedup can be achieved in real life. This is particularly crucial when considering the issue of quantum error correction, which is a necessity for these algorithms to work in the real world.

In this part, we shift gears to focus on an architecture-driven approach. We will focus in particular on Boson Sampling, an example of a near-term quantum architecture which is of interest due to its classical hardness rather than due to any practical applications. In Chapter 5, we formalise a link between Boson Sampling and the representation theory of the symmetric and unitary groups. In doing so, we show that Boson Sampling can be modelled in first quantisation, or particle-based picture, as sampling from a particular structure of quantum circuit, and how practical issues can be modelled as decoherence in these circuits. Then in Chapter 6, we model particular kinds of imperfections in first quantisation, and show how this can lead to simpler classical simulation algorithms which we estimate will perform faster when simulating near-term devices.

The rest of this chapter is laid out as follows. We start by explaining in Section 4.1 why this change in directions is necessary, by discussing practical limitations with universal quantum computation, particularly error correction. Next we introduce the topic of Quantum Advantage in Section 4.2, where the focus is on non-universal models of quantum computation which seem classically hard to simulate. In Section 4.3, we discuss linear optics and introduce the problem of Boson Sampling, an example of a Quantum Advantage proposal with strong connections to representation theory. This architecture will be the focus of this part. Experimental achievements are given in Section 4.4, as well as limitations with larger experiments and how these lead to classical simulations in Sections 4.5 and 4.6. We will conclude this chapter summarising some ideas in representation theory, in particular the Quantum Schur Transform, in Section 4.7. Although the switch to representation theory at the end of this chapter might seem at odds with the rest of this chapter, we will show in Chapters 5 and 6 how these two subjects relate to one another.

4.1 Estimated speedups in practice and the limitations of quantum error correction

At the same time as pursuing theoretical speedups through algorithms such as those described in Chapters 2 and 3, it is worth questioning the extent to which these algorithms can provide speedups in real-world scenarios. Ideally we want a situation where a quantum computer would be able to solve a problem with relevant applications significantly faster than a classical computer.

The question of whether or not NP-Hard problems can fit this scenario was considered by Campbell, Khurana and Montanaro [154], who looked at estimating the resources required for Grover Search and Montanaro's backtracking algorithm when applied to the problems of boolean satisfiability and graph colouring. Campbell, Khurana and Montanaro provided a gate decomposition for these two algorithms when running for random problem instances under different assumptions about the quantum hardware, from realistic scenarios to more optimistic ones. These estimated runtimes were then compared with the best classical solvers for boolean satisfiability and graph colouring. From this, Campbell, Khurana and Montanaro came up with the largest problem sizes that these models could solve in a day, and showed that a speedup could potentially be achieved: For SAT, the estimated improvement in the optimistic scenario was as much as $100,000$ faster, and a $10,000$ times speedup was estimated for graph colouring [154].

However, there is also a cost that comes with running these algorithms for the largest problem sizes: how much error correction is required to reliably solve the problem. This was estimated using a gate decomposition of Clifford gates as well as either the single qubit T gate or the three-qubit Toffoli gate as non-Clifford operations which can provide universal quantum computation. The idea is that one can use the surface code to implement any Clifford gates in a fault-tolerant fashion, and then the non-Clifford gates can be implemented by preparing a particular quantum state, called a magic state, and then using that state in the rest of the Clifford operations. This means that the only significant overhead is in the preparation of these magic states, which is done via a purification technique where less ideal magic states are used to construct more ideal states. The qubits required to produce these states are known as the factory qubits.

Unfortunately the result of Campbell, Khurana and Montanaro shows that the number of magic states required in these instances is significant [154]. For the 10^5 speedup mentioned earlier, a total of 10^{19} Toffoli gates are required, corresponding to 10^{12} factory qubits. Similarly for graph colouring, the 10^4 speedup requires on the order of 10^{20} T or Toffoli gates, and 10^{12} factory qubits. What is even more concerning is that implementing so many Toffoli or T gates also requires a significant amount of classical processing: Campbell, Khurana and Montanaro showed that the classical processing required to implement 10^{20} Toffoli gates was on the order of 10^8 processor days even for specialised electronics such as application-specific integrated circuits. For a standard CPU, this overhead could be as large as 10^{16} processor days. Such a large overhead means that any quantum advantage from these techniques would immediately be lost.

A potential workaround for this is adapting quantum algorithms to near-term architectures. There are a small number of results in this area, particularly when the number of logical qubits required is small. Dunjko, Ge and Cirac [68] showed that a polynomial (at most quadratic) speedup for Satisfiability can be shown for a quantum computer of arbitrary size. This was achieved by first using classical backtracking to reduce the Boolean formula to problem instances small enough that they can be run on the quantum computer, and then applying Grover Search to obtain a quadratic speedup for these smaller instances. Ge and Dunjko [69] later developed a general framework based on this idea, and showed how it can be applied to find Hamiltonian cycles in bounded-degree graphs, achieving a polynomial speedup over Eppstein's algorithm [64]. However these algorithms still require logical qubits, and therefore might use a large number of physical qubits to operate in practice. It is also worth noting that this speedup is not at most a quadratic one over Eppstein's algorithm. This is because the approach of Ge and Dunjko uses Grover Search, which searches over the space of all possible solutions. The quantum backtracking algorithm on the other hand searches over an asymptotically smaller space, as it is able to see earlier on if several possible solutions will all fail or all succeed. See Section 2.3.3 for further details.

4.2 The search for a quantum advantage

So if fault tolerant quantum computing is not currently an option, this begs the question of what is achievable without it, and in particular what can be performed exponentially faster than classical computers in spite of having little or no error correction.

This is a concept that has gone through a few different names, most notably "quantum computational supremacy." In this chapter and throughout the rest of this thesis, we shall use the term "quantum advantage" to refer to this area, and refer the reader to [155, 156] for further discussion of the issues surrounding the use of "supremacy." Here, the emphasis is on finding a model of quantum computation that is hard to classically simulate. Such a quantum computer need not be universal, or even have practical applications. However, it should be provably hard to simulate classically, under reasonable assumptions, and with little to no error correction required.

For simplicity, we shall not detail the formal proofs of different quantum advantage problems, and instead note a general structure of the proofs of hardness. These problems are typically described as sampling problems, where the aim is to produce an output from (approximately) the same probability distribution as the quantum computer. For these proofs to work, we require that approximating one of these probabilities is #P-Hard. If this is true, then a result of Stockmeyer [157] shows that classically being able to sample from this exact distribution leads to a algorithm that can approximate these probabilities up to a multiplicative error. This would imply a model of computer which can approximately solve #P-Hard problems, which in turn would lead to the collapse of the Polynomial Hierarchy to the third level by Toda's Theorem [158]. This consequence is similar to, though not as strong as, P = NP, and is considered equally unlikely.

This proves that if sampling from the exact distribution can be done classically, then the Polynomial Hierarchy collapses to the third level. However, we often want to go further than this claim, and show that to even sample from a distribution which is approximately equal to the target distribution is hard.

Before explaining how to prove the hardness of approximate sampling, it is worth discussing what it means to sample from an approximate distribution in the first place. For this, we use the total variation distance, which is defined between two probability distributions P and Q over a finite set of outcomes Ω as half the L_1 distance:

$$\Delta(P,Q) \colon = \frac{1}{2}\sum_{\omega\in\Omega}|P(\omega)-Q(\omega)|. \tag{4.1}$$

Statistically, the total variation distance is connected to hypothesis testing. Suppose we have some samples from a distribution, and we want to deterministically decide whether these samples are from P or Q. Define ϵ_P as our false positive rate, the probability that we output P when the distribution was in fact Q, and ϵ_Q as our false negative rate, the probability that we output Q when the distribution was in fact P. The sum of these error rates can be bounded as $\epsilon_P + \epsilon_Q \geq 1 - \Delta(P,Q)$ [159]. This motivates our use of this distance for approximate sampling, as it limits our ability to determine whether our samples have come from a true quantum device or a classical simulation.

Another distance which is also used particularly for quantum states is the trace distance, which is defined between two states ρ and σ as

$$\delta_{\mathrm{Tr}}(\rho,\sigma) \colon = \frac{1}{2}\operatorname{Tr}\left[\sqrt{(\rho-\sigma)^\dagger(\rho-\sigma)}\right]. \tag{4.2}$$

The trace distance can also be defined as the maximum total variation distance between any two states when a measurement is applied:

$$\delta_{\mathrm{Tr}}(\rho,\sigma) \colon = \max_E \Delta(E(\rho),E(\sigma)), \tag{4.3}$$

where E is taken over all positive operator valued measures. This is better known as the data processing inequality. As a result, it is often beneficial to use the trace distance as an upper bound for the total variation distance. We can also note that this distance is convex:

$$\delta_{\mathrm{Tr}}(\rho, p_\sigma\sigma + p_\tau\tau) \leq p_\sigma\delta_{\mathrm{Tr}}(\rho,\sigma) + p_\tau\delta_{\mathrm{Tr}}(\rho,\tau), \tag{4.4}$$

where $p_\sigma + p_\tau = 1$.

As well as understanding how we measure the distance between two distributions, it is worth discussing the size of the error. Let $p \in [0,1]$ be our target probability for some outcome, and $\tilde{p} \in [0,1]$ be its approximation. For some $\epsilon > 0$, we say that p and $\tilde{p}$ are approximately equal up to additive error ϵ if $|p-\tilde{p}| \leq \epsilon$, and approximately equal up to multiplicative error ϵ if $|p-\tilde{p}| \leq p\epsilon$. It is easy to see that multiplicative error immediately implies additive error as well, but additive error does not necessarily imply multiplicative error.

We are now ready to define approximate sampling. We say that a target distribution P can be approximately sampled from efficiently if there is a probability distribution $\tilde{P}$ which can be sampled from in polynomial time and is approximately equal to P in total variation distance up to additive error ϵ for some $\epsilon > 0$:

$$\Delta(P,\tilde{P}) \leq \epsilon. \tag{4.5}$$

To prove that even approximately sampling is hard, we require two further assumptions, in order to reduce sampling from the target distribution to sampling from the approximate distribution. The first is the distribution must anticoncentrate; this means that the distribution is largely spread out rather than having peaks[1]. The second is that approximating the probability of a random outcome up to multiplicative error must be hard, not just the worst-case outcome. Using these two assumptions, we can reduce from an exact sampling problem to an approximate

[1] Note however that the distribution cannot be too far spread out, otherwise it becomes close to the uniform distribution which can be sampled classically.

one, and then use the argument above to show that even approximately sampling must be hard for the Polynomial Hierarchy to not collapse. Intuitively, these assumptions ensure that a polynomial-time classical simulator cannot simply sample from the peaks of the distribution (by anticoncentration), nor can it just sample from the outcomes whose probabilities are easy to approximate (average-case hardness).

As we shall see in Sections 4.2.1 and 4.3.4, each of the problems have more or less the same argument as that stated above, though what is proven and what remains conjecture tends to vary. It has been proven for some sampling problems that their distributions do anticoncentrate, and it has been proven for some problems that exactly computing the probability of a random outcome is #P-hard. What has not currently been proven for any quantum advantage problem is that it is #P-hard to approximate the probability of a random outcome, though we have some idea of what techniques will not lead to a successful proof. Aaronson and Chen [160] proved two details, regarding the use of oracles[2] in such a proof. The first is that a proof cannot be relative to arbitrary oracles, as there exists an oracle relative to which the complexity classes of classical probabilistic sampling and quantum sampling are equivalent, yet the Polynomial Hierarchy does not collapse. The second is that if the proof is restricted to using an oracle in P with a polynomial advice string that only depends on the size of the input, then in order for the Polynomial Hierarchy to collapse we need to assume that either classical and quantum sampling complexity are not equivalent, in which case an oracle is not necessary, or NP $\not\subseteq$ BPP, where BPP is the class of what can be solved in polynomial time probabilistically with bounded error. This second assumption is a closely related to a standard assumption in cryptography: That there are functions which are easy to compute but hard to invert, also known as one-way functions.

Note that while these proofs rule out polynomial classical simulations of Quantum Advantage problems, they do not state the point at what size these problems become intractable. This requires a more fine-grained approach to the computational complexity conjectures. Dalzell et al. [161, 162] gave such an approach, devising conjectures similar to the Strong Exponential Time Hypothesis mentioned in Chapter 2, and providing fine-grained lower bounds by showing that a classical simulator running in a certain time with polynomial multiplicative error could violate these conjectures. From this, Dalzell et al. estimate the largest such instances that a modern supercomputer could classically simulate in a century, based on number of floating point operations required. The requirement of simulating up to multiplicative error is strong, but some very recent work by Morimae and Tamaki has given similar results for certain problems with additive error [163].

Another interesting question, though one that will not be explored in great detail in this thesis, is the question of verifying a quantum advantage. That is, given a collection of samples from some distribution, how do we check that this is the (known) target distribution we want, instead of some other (classical) distribution. This is a challenge, primarily for the same reasons that it is hard to simulate these problems in the first place: a combination of having many probabilities close to uniform and the fact that even estimating one probability is exponentially hard. Indeed, Hangleiter et al. [164] showed that for any sufficiently flat distribution, verifying the distribution purely from samples and a description of the target distribution must require exponentially many samples.

[2] An oracle in complexity theory is a process that an algorithm can query. The overall runtime is the overhead of the algorithm, plus the number of oracle queries multiplied by the oracle's runtime for each query.

4.2.1 Example problems

We will now describe some example problems related to Quantum Advantage, and explain what is currently known about them.

Problem 4.1 (Instantaneous Quantum Polynomial Time (IQP) Circuit Sampling)**.** *Let C be an n-qubit polynomial time quantum circuit consisting only of nearest-neighbour controlled phase gates in a 2D lattice. Sample from the distribution corresponding to measuring the state*

$$H^{\otimes n} C H^{\otimes n} |0\rangle^{\otimes n} \tag{4.6}$$

in the computational basis.

IQP circuits were originally proposed by Shepherd and Bremner [165, 166], where some applications were proven and the circuits were conjectured to be classically hard. This was followed up in 2011, when Bremner, Jozsa and Shepherd proved that efficient classical simulation led to the collapse of the Polynomial Hierarchy, assuming anticoncentration and average-case hardness conjectures [167]. The anticoncentration conjecture was proven to hold by Bremner, Montanaro and Shepherd [168].

It is worth noting that these IQP circuits required controlled phase gates between arbitrary qubits. The more restricted form defined above was given by Bremner, Montanaro and Shepherd in 2017 [169], showing that a circuit of $O(\sqrt{n} \log n)$ depth implemented on a 2D lattice anticoncentrates and fits the hardness arguments above. They also considered noise in such a circuit, and gave two further results: One, that an average IQP circuit with ϵ depolarising noise applied to each qubit can be simulated with accuracy δ in time polynomial in n; and two, that a simple form of error correcting code can manage to retain classical hardness even when faced with depolarising noise.

Problem 4.2 (Random Circuit Sampling)**.** *Let C be a random circuit of depth $O(n)$ which consists of random 1- and 2-qubit operations from a universal gate set applied to nearest neighbours on a 2D lattice. Sample from the distribution corresponding to measuring the state $C|0\rangle^{\otimes n}$ in the computational basis.*

Random Circuit Sampling is a model first proposed by Boixo et al. [50], motivated by the superconducting quantum computing architecture developed by Google. In their result, Boixo et al. argue that it is computationally hard to classical sample from this distribution under anticoncentration and average-case hardness assumptions, used numerical simulations to give reason to believe that $O(\sqrt{n})$-depth circuits are sufficient for the distribution to anticoncentrate, and proposed a verification metric called the cross entropy difference. Since their publication, the anticoncentration conjecture was proven to hold by Hangleiter et al. [51], and exactly computing the probability of an outcome was proven to be #P-hard on average by Bouland et al. [52]. Most recently, Random Circuit Sampling has been the centre of interest due to a publication by Arute et al. who implemented the problem on Google's 54 qubit quantum processor, claiming that this is now of a size that cannot be simulated by classical computers [5]. This was verified by estimating a property known as the cross-entropy fidelity, a metric proposed by Boixo et al. [50] and later proven hard to classically spoof by Aaronson and Gunn [170]. However, it is worth noting that there is some rebuttal from Pednault et al. at IBM [53], who claim that given sole use of the Summit supercomputer at Oak Ridge National Laboratory classical simulations could be achieved in two and a half days, and argue that this isn't sufficient for the claim of quantum advantage. The main difference between the classical simulation methods considered by Arute et al. and Pednault et al. is the tradeoff between time and space. Arute et al. consider using a polynomial space method, which they believe requires approximately 10,000 years as a result. Pednault et al. on the other hand propose an algorithm with exponentially large space, essentially storing a full description of the quantum state in 64PiB (1PiB = 2^{50} bytes) of disk storage, with most of the 2.5 days runtime spent writing to and reading from storage. Likewise, recent reports and manuscripts suggest that classical simulators can be improved even further by taking advantage of the low fidelity of Google's quantum processor [54, 55].

4.3 Linear optics and Boson Sampling

So far in this chapter we have motivated the need for pratical sampling experiments which can offer a quantum advantage. We now move to explaining the problem of interest for this part of the thesis: Boson Sampling. We will start by explaining the theory of linear optics and some simple example interferometers, before introducing Boson Sampling and discussing its classical complexity, as well as briefly summarising some variants which use non-linear photonic inputs.

4.3.1 Single photons and linear optical components

Boson Sampling experiments can be developed from the use of single photons, simple linear optical components and single photon detectors. We shall now summarise how such experiments can be described in both second and first quantisation. For a more thorough understanding of these concepts we direct the interested reader to [171, 172].

Second quantisation

We shall start with second quantisation, as the more common and natural way of describing bosonic systems. The bosonic Fock space is a direct sum of symmetric tensors:

$$F(H) = \oplus_{n=0}^{\infty} \mathrm{Sym}(H^{\otimes n}), \tag{4.7}$$

where H is a Hilbert space, and Sym denotes the symmetric subspace. Note that $H^{\otimes 0}$ is the Hilbert space spanned by the vacuum state $|0\rangle$.

An m-mode bosonic Fock state is written as an m-dimensional complex vector $|S_1, S_2, \ldots, S_m\rangle$, where S_i is the number of photons occupying mode i. The number of photons in mode i can be decremented or incremented by annihilation and creation operations a_i and $a_i^\dagger$, respectively:

$$a_i \left|S_1, \ldots, S_i, \ldots, S_m\right\rangle = \sqrt{S_i} \left|S_1, \ldots, S_i - 1, \ldots, S_m\right\rangle, \tag{4.8}$$

$$a_i \left|S_1, \ldots, S_i, \ldots, S_m\right\rangle = \sqrt{S_i + 1} \left|S_1, \ldots, S_i + 1, \ldots, S_m\right\rangle. \tag{4.9}$$

We also have the commutation relations $[a_i^\dagger, a_j^\dagger] = [a_i, a_j] = 0$ and $[a_i^\dagger, a_j] = \delta_{ij}$.

A particular Fock occupation can therefore be described by creation operators acting on the vacuum state:

$$\left|S_1, \ldots, S_m\right\rangle = \prod_{i=0}^{m} \frac{(a_i^\dagger)^{S_i}}{\sqrt{S_i!}} \left|0^m\right\rangle, \tag{4.10}$$

where we have used the shorthand notation

$$|i^j\rangle = |\overbrace{i, i, i, \ldots, i}^{j \text{ copies}}\rangle. \tag{4.11}$$

We shall use these creation and annihilation operators to describe how we apply optical components to act on the initial state.

There are two basic linear optical components we require to implement Boson Sampling. The first is a phase shifter, which induces a phase on a spatial mode. A phase shifter on mode j induces:

$$a_j^\dagger \to e^{i\theta} a_j^\dagger. \tag{4.12}$$

Physically, a phase shifter can be implemented as a change of refractive index, which can speed up or slow down the transmission of light.

The second component is a beam splitter. This component acts as a partial mirror between two spatial modes, transmitting some light and reflecting the remainder. A beam splitter on spatial modes j and k with reflectivity r, r' and transmittance t, t' acts as the following:

$$a_j^\dagger \to t a_j^\dagger + r' a_k^\dagger, \tag{4.13}$$

$$a_k^\dagger \to r a_j^\dagger + t' a_k^\dagger, \tag{4.14}$$

where we have the constraints that $|r| = |r'|, |t| = |t'|, |r|^2 + |t|^2 = 1$ and $r^* t' + r' t^* = r^* t + r' t'^* = 0$. Of particular note is the 50-50 beam splitter, where exactly half the light is reflected and the other half is transmitted. In this case, $r = r' = 1/\sqrt{2}$ and $t = t' = i/\sqrt{2}$. Note the $\pi/2$ phase difference between reflection and transmission, which is important for the interferometers we shall be discussing in Section 4.3.2.

Finally, we need to measure the number of photons in each spatial mode. To do this, we can use the photon number measurement $n_i = a_i^\dagger a_i$. It can be seen that this operator is Hermitian and its eigenvectors are the Fock basis for mode i, with eigenvalues depending on the number of photons occupying mode i. For example, if there are j photons in mode i we see that

$$\begin{align}
n_i |j\rangle &= a_i^\dagger a_i |j\rangle \tag{4.15}\\
&= \sqrt{j} a_i^\dagger |j-1\rangle \tag{4.16}\\
&= \sqrt{j}\sqrt{j-1+1}\, |j-1+1\rangle \tag{4.17}\\
&= j |j\rangle . \tag{4.18}
\end{align}$$

An alternative form of photon detection is to measure using threshold detectors. Rather than counting the exact number of photons in spatial mode i, these detectors project into the subspaces of $|0_i\rangle\langle 0_i|$ and $\sum_{j=1}^{\infty} |j_i\rangle\langle j_i|$. While these detectors are less refined as they only measure if any photons exist in a spatial mode rather than how many photons exist, they are more readily available and often have other better properties such as higher detection efficiencies and lower dark count rates [173, 174].

Any linear optical interferometer can be composed from the above components, as we shall see in Section 4.3.5 [175, 176, 177].

First quantisation

Equivalently, we can describe these effects in the first quantisation, or particle-based, picture. In this picture, our input is a state in the symmetric subspace of $\mathbb{C}^{m \times n}$. This means that each of our n particles is represented as am m-dimensional qudit, and the overall state has to be symmetric as no single particle can be uniquely identified. For example, the Fock state $|2, 0\rangle$ corresponds to the first quantised state $|11\rangle$, whereas the Fock state $|1,1\rangle$ corresponds to $(|12\rangle + |21\rangle)/\sqrt{2}$. In general, the Fock state $|S_1, \ldots, S_m\rangle$ corresponds to

$$\frac{1}{\sqrt{n! \prod_{i=1}^{m} S_i!}} \left(\sum_{\sigma \in \mathrm{S}_n} \sigma \bigotimes_{j=1}^{m} |j\rangle^{\otimes S_j} \right), \tag{4.19}$$

where S_n is the symmetric group, or the group of permutations of n objects. Linear optical components can now be applied to each particle individually as an $m \times m$ unitary matrix. For example, the action of a θ-phase shifter applied to one mode in a two-mode interferometer can be written as

$$\begin{pmatrix} e^{i\theta} & 0 \\ 0 & 1 \end{pmatrix}. \tag{4.20}$$

The action of a beam splitter with reflectivity r, r' and transmittance t, t' can be written as

$$\begin{pmatrix} t & r' \\ r & t' \end{pmatrix}, \tag{4.21}$$

with the same restraints as in first quantisation. Of particular note, the 50-50 beam splitter has unitary matrix

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{i}{\sqrt{2}} \\ \frac{i}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}. \tag{4.22}$$

Finally, a photon number counting measurement simply consists of measuring each qudit in the computational basis and counting the number of qudits in each state $j \in \{1, \ldots, m\}$ for number resolving detectors, or simply if any qudit is in state j for threshold detectors.

4.3.2 Example linear optical interferometers

We shall now discuss some example interferometers, and their resulting effects, using the notation described above. This will eventually lead us to talk about Boson Sampling in Section 4.3.3.

One-photon-two-modes: The Mach-Zehnder Interferometer

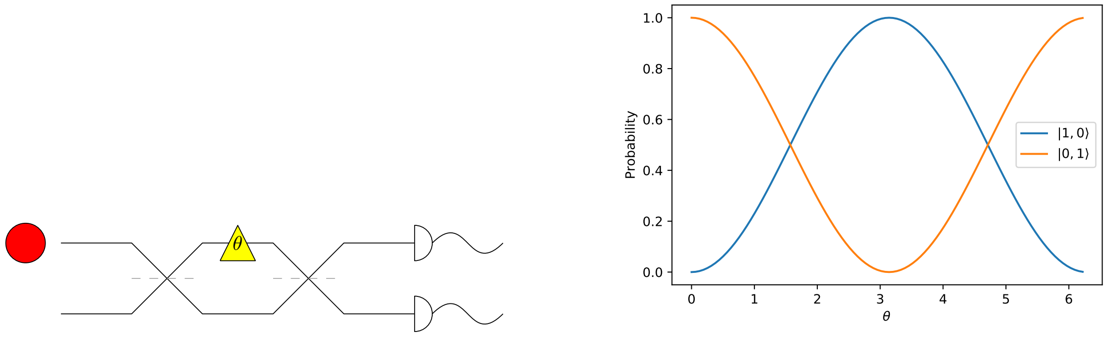
A Mach-Zehnder Interferometer (MZI) is a demonstration of how a single photon can interfere with itself [178, 179]. This experiment uses a single photon in one of two modes, and consists of a 50-50 beam splitter across both modes, followed by a phase shifter on a single mode, and finally a second 50-50 beam splitter. A diagram of this experiment can be seen in Figure 4.1b.

In second quantisation, we can describe the MZI as follows, labelling our spatial modes 1 and 2. We start with a single photon in mode 1 $|1,0\rangle = a_1^\dagger |0,0\rangle$. After our first beam splitter we have the state $(a_1^\dagger + i a_2^\dagger) |0,0\rangle / \sqrt{2} = (|1,0\rangle + i\,|0,1\rangle)/\sqrt{2}$.

We next apply a phase shifter to spatial mode 1, giving $(e^{i\theta} a_1^\dagger + i a_2^\dagger) |0,0\rangle / \sqrt{2}$. Finally, we apply our second beam splitter and find that

$$\frac{1}{\sqrt{2}}(e^{i\theta} a_1^\dagger + i a_2^\dagger) |0,0\rangle \to \frac{1}{2}(e^{i\theta}(a_1^\dagger + i a_2^\dagger) + i(i a_1^\dagger + a_2^\dagger)) |0,0\rangle \tag{4.23}$$

$$= \frac{1}{2}((e^{i\theta} - 1) a_1^\dagger + i(e^{i\theta} + 1) a_2^\dagger) |0,0\rangle . \tag{4.24}$$



(a)

(b)

Figure 4.1: The Mach-Zehnder Interferometer, a demonstration of single-photon interference. 4.1a: An experimental diagram, consisting of a single photon (red circle) in one of two spatial modes (solid black lines), a phase shifter (yellow triangle containing θ) surrounded by two 50-50 beam splitters (grey dashed lines) and single photon detectors (white semi-circles). 4.1b: The Probability of measuring $|0,1\rangle$ (orange) and $|1,0\rangle$ (blue) from a Mach-Zehnder Interferometer over θ.

Adjusting our phase θ determines what output we measure: For $\theta = 0$, the photon will be in mode 2; for $\theta = \pi/2$, the photon will be in mode 1, and for other values of θ the photon will be measured randomly in one of the two modes. How the probabilities change over θ is given in Figure 4.1b.

In first quantisation, we see the same outcome. Our input is $|1\rangle$, and applying the first beam splitter leads to the state $(|1\rangle + i\,|2\rangle)/\sqrt{2}$. The phase shifter on mode 1 results in the state $(e^{i\theta}\,|1\rangle + i\,|2\rangle)/\sqrt{2}$, and the second beam splitter leaves us with the state

$$
\begin{aligned}
|\psi\rangle &= \frac{1}{2}\left(e^{i\theta}\left(|1\rangle + i\,|2\rangle\right) + i\left(i\,|1\rangle + |2\rangle\right)\right) &\quad (4.25)\\
&= \frac{1}{2}\left(\left(e^{i\theta} - 1\right)|1\rangle + i\left(e^{i\theta} + 1\right)|2\rangle\right). &\quad (4.26)
\end{aligned}
$$

Again, we see that adjusting θ will determine which spatial mode we detect a photon in.

Two-photons-two-modes: The Hong-Ou-Mandel Dip

The Hong-Ou-Mandel (HOM) Dip is the earliest demonstration of multi-photon interference [180]. It features two spatial modes, each starting with a single photon, and consists of simply a 50-50 beam splitter across both modes.

In second quantisation, we start with the Fock state $|1,1\rangle = a_1^\dagger a_2^\dagger\,|0,0\rangle$. Applying the beam splitter gives us

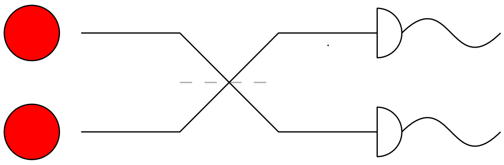


Figure 4.2: An experimental diagram for the Hong-Ou-Mandel Dip, consisting of two indistinguishable photons (red circles) in two spatial modes (solid black lines) interacting with a single 50-50 beam splitter (grey dashed line) before being measured by single photon detectors (white semi-circles).

$$\begin{aligned}
a_1^\dagger a_2^\dagger \left|0,0\right\rangle &\to \frac{1}{2}(a_1^\dagger + i a_2^\dagger)(i a_1^\dagger + a_2^\dagger)\left|0,0\right\rangle && (4.27)\\
&= \frac{1}{2}\left(i(a_1^\dagger)^2 + a_1^\dagger a_2^\dagger - a_2^\dagger a_1^\dagger + i(a_2^\dagger)^2\right)\left|0,0\right\rangle && (4.28)\\
&= \frac{1}{2}\left(i(a_1^\dagger)^2 + a_1^\dagger a_2^\dagger - a_1^\dagger a_2^\dagger + i(a_2^\dagger)^2\right)\left|0,0\right\rangle && (4.29)\\
&= \frac{i\left((a_1^\dagger)^2 + (a_2^\dagger)^2\right)}{2}\left|0,0\right\rangle && (4.30)\\
&= \frac{i(\left|2,0\right\rangle + \left|0,2\right\rangle)}{\sqrt{2}}. && (4.31)
\end{aligned}$$

After measurement, we will find that both photons will always be detected in the same (random) spatial mode. This example has a strong connection to distinguishability, and we will return to it in Section 4.5.1.

In first quantisation, we start with the state $(\left|12\right\rangle + \left|21\right\rangle)/\sqrt{2}$. The beam splitter acts on each particle individually, giving

$$\begin{aligned}
\left|\psi\right\rangle &= \frac{1}{2\sqrt{2}}\left(\left(\left|1\right\rangle + i\left|2\right\rangle\right)\left(i\left|1\right\rangle + \left|2\right\rangle\right) + \left(i\left|1\right\rangle + \left|2\right\rangle\right)\left(\left|1\right\rangle + i\left|2\right\rangle\right)\right) && (4.32)\\
&= \frac{1}{2\sqrt{2}}\left(2i\left|11\right\rangle + (1+i^2)\left|12\right\rangle + (i^2+1)\left|21\right\rangle + 2i\left|22\right\rangle\right) && (4.33)\\
&= \frac{i(\left|11\right\rangle + \left|22\right\rangle)}{\sqrt{2}}. && (4.34)
\end{aligned}$$

Again, we find that only bunching is observed at the output.

4.3.3 Bosonic sampling

We now define the ideal probability distribution of indistinguishable single bosons interacting on a linear interferometer. We'll refer to this as bosonic sampling, as it's a bit more general than Aaronson and Arkhipov's Boson Sampling problem as we describe below. The input is $U \in \mathrm{U}(m)$, an $m \times m$ unitary matrix which describes an m-mode linear interferometer, and $S = (S_1, S_2, \ldots, S_m)$ with $\sum_{i=1}^m S_i = n$, an ordered list of integers that corresponds to an n-boson, m-mode occupation describing the input state with S_i bosons in mode i. For this problem, we will assume that photon measurement is performed using number-resolving detectors. Although theory on the complexity of bosonic sampling with threshold detectors is limited, for the specific problem of Boson Sampling the two methods of detection can be taken as being equivalent due to the Bosonic Birthday Paradox, which we shall discuss further in Section 4.3.4. Note also that the hardness of variants of Boson Sampling with threshold detectors has been considered, in particular with Gaussian Boson Sampling [181], which we shall discuss further in Section 4.3.6.

Given an output occupation S', define the $n \times n$ (not necessarily unitary) matrix $U_{S',S}$ as that formed by first taking S'_i copies of row i of U in order to create an $m \times n$ matrix, from which we then take S_j copies of column j. We can then define $\mathcal{D}_{U,S}$, the probability distribution for measuring an n-boson m-mode occupation S' for interferometer U and input state S, as

$$\Pr_{\mathcal{D}_{U,S}}[S'] = \frac{|\operatorname{per}(U_{S',S})|^2}{\prod_{i=1}^{m} S'_i! S_i!}, \tag{4.35}$$

where per is the matrix permanent, defined as

$$\operatorname{per}(M) = \sum_{\sigma \in \mathrm{S}_n} \prod_{i=0}^{n} M_{i,\sigma(i)}. \tag{4.36}$$

This relationship between linear optics and matrix permanents was originally found by Scheel and Buhmann [182], and later proven by Aaronson and Arkhipov using a different approach [70, 71].

In a photonics experiment, this setting is described in terms of creation operators $a_i^\dagger$ for a photon in mode i. The initial state is then

$$|S\rangle = \prod_{i=0}^{m} \frac{(a_i^\dagger)^{S_i}}{\prod_{j=2}^{S_i} \sqrt{j}} |0^m\rangle. \tag{4.37}$$

The evolution of the photonic state induced by a linear optical interferometer implementing U can then be expressed as $a_i^\dagger \mapsto \sum_{j=0}^{m} U_{i,j} a_j^\dagger$. Thus single boson states evolve under linear interferometry just as an m dimensional qudit does under a unitary gate U (sometimes called unary encoding). This suggests how quantum circuits simulating photonics might be constructed, as we'll see.

It is easy to see that both of the two examples discussed in Section 4.3.2 can be described in this picture as well. For the Mach-Zehnder Interferometer, we find that the unitary matrix describing our interferometer is

$$U = \frac{1}{2} \begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix} \begin{pmatrix} e^{i\theta} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix} \tag{4.38}$$

$$= \frac{1}{2} \begin{pmatrix} e^{i\theta} & i \\ ie^{i\theta} & 1 \end{pmatrix} \begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix} \tag{4.39}$$

$$= \frac{1}{2} \begin{pmatrix} e^{i\theta} - 1 & i(e^{i\theta} + 1) \\ i(e^{i\theta} + 1) & 1 - e^{i\theta} \end{pmatrix}. \tag{4.40}$$

Assuming our photon is initially in mode 1, the relevant matrices describing our outcomes are the 1×1 matrices $M_{1,1} = e^{i\theta} - 1$ and $M_{1,2} = i(e^{i\theta} + 1)$. For 1×1 matrices the permanent is just the single element, and so we find the probabilities are $|e^{i\theta} - 1|^2$ and $|e^{i\theta} + 1|^2$, giving us the expected outcome of controlling the photon output based on θ.

For the Hong-Ou-Mandel Dip, we note that the relevant unitary matrix is the matrix we have for a beam splitter. We then note that the probability of seeing outcome occupation $|1,1\rangle$ is

$$\left|\text{per}\begin{pmatrix}\frac{1}{\sqrt{2}} & \frac{i}{\sqrt{2}}\\ \frac{i}{\sqrt{2}} & \frac{1}{\sqrt{2}}\end{pmatrix}\right|^2 = \frac{|1+i^2|^2}{4} \tag{4.41}$$

$$= 0. \tag{4.42}$$

Similarly we find that the outcomes for $|2,0\rangle$ and $|0,2\rangle$ are

$$\frac{\left|\text{per}\begin{pmatrix}\frac{1}{\sqrt{2}} & \frac{i}{\sqrt{2}}\\ \frac{1}{\sqrt{2}} & \frac{i}{\sqrt{2}}\end{pmatrix}\right|^2}{2!} = \frac{|2i|^2}{8} \tag{4.43}$$

$$= \frac{1}{2}, \tag{4.44}$$

and

$$\frac{\left|\text{per}\begin{pmatrix}\frac{i}{\sqrt{2}} & \frac{1}{\sqrt{2}}\\ \frac{i}{\sqrt{2}} & \frac{1}{\sqrt{2}}\end{pmatrix}\right|^2}{2!} = \frac{|2i|^2}{8} \tag{4.45}$$

$$= \frac{1}{2}, \tag{4.46}$$

respectfully. This matches the expected outcome statistics of the coincident output never occurring and the two other possible outcomes happening purely at random.

4.3.4 The computational complexity of Boson Sampling

The problem known as Boson Sampling is that of sampling from the probability distribution described in Section 4.3.3 under certain conditions [70, 71]. First, the input occupation is specified as $|1^n 0^{m-n}\rangle = \prod_{i=1}^n a_i^\dagger |0\rangle$. Second, the interferometer U is sampled Haar randomly from $\mathrm{U}(m)$ with $m = O(n^2)$. And finally, the sampled output is anti-bunched, meaning that there is at most one photon in each spatial mode. Such a photon occupation is also referred to as coincident.

Problem 4.3 (Boson Sampling)**.** *Let U be an m-mode linear optical interferometer sampled Haar-randomly from $U(m)$. Sample from the distribution corresponding to measuring U acting on $|1^n 0^{m-n}\rangle$ in the (anti-bunched) Fock basis, where $m = O(n^2)$.*

It was proven by Aaronson and Arkhipov that if there was a polynomial time classical algorithm for sampling from this distribution, then $\mathrm{P}^{\#\mathrm{P}} = \mathrm{BPP}^{\mathrm{NP}}$ and the Polynomial Hierarchy would collapse to the third level [70, 71]. This is done using the fact that the permanent of a matrix is #P-hard, even to approximate [183, 184]. Aaronson and Arkhipov also showed the same result for sampling from any approximate distribution up to ϵ away in total variation distance from Boson Sampling, assuming the conjectures from before.

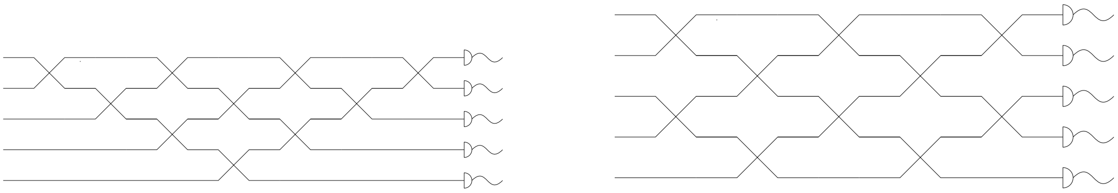
In comparison to the other quantum advantage problems, the anticoncentration conjecture has not been proven for Boson Sampling. It is also not known if it is #P-hard to approximate the permanent of a random complex matrix, however it has been proven that exactly computing the permanent of a Gaussian matrix is #P-hard [70, 71].

It is worth discussing the three conditions specified at the start of this section. These conditions are required because of a result by Aaronson and Arkhipov that sampling n rows and columns from an $m \times m$ unitary matrix is close in variation distance to an $n \times n$ Gaussian matrix, meaning a matrix whose elements are complex values independently sampled from the Gaussian distribution. Ensuring our photonics setup satisfies these conditions therefore means that the probability of an outcome is proportional to computing the permanent of an approximately Gaussian matrix. Note that the best bound for ensuring the matrix is Gaussian actually requires $m = \Omega(n^5 \log n)$, but Aaronson and Arkhipov conjecture that this bound can be improved to $m = O(n^2)$.

The third condition, that our photons are anti-bunched, also provides another experimental convenience. This is because we are now able to use threshold detectors rather than photon number resolving detectors, which are more easily available and often have other better properties [185, 174]. Although it looks like this condition relies on postselection and could therefore be costly in practice, it can be satisfied with high probability if $m = O(n^2)$ and our unitary U is Haar random. This can be seen in two steps: First by showing that over Haar random unitary matrices U the expected probability of any outcome is uniform; and second by using a counting argument to show that over the uniform distribution of n-photon m-mode occupations, if $m = O(n^2)$ then the number of cases where our photons are not bunched is significantly larger than the number of cases where multiple photons occupy the same output mode. This phenomenon is known as the Bosonic Birthday Paradox [186]. It is worth noting that just because photons need to occupy different spatial modes at the output does not mean that they do not cross paths during the experiment. Indeed, if the experiment was arranged such that no photons ever crossed paths, then such an experiment would be trivial to simulate classically, by simply simulating each photon travelling through the interferometer individually.

Regarding the question of verifying Boson Sampling, there have been some interesting results. Gogolin et al. [187] argued that if a verifier does not know the underlying distribution, and is therefore only using the samples obtained to verify the distribution, then Boson Sampling requires an exponentially large number of samples to differentiate from the uniform distribution. This was refuted by Aaronson and Arkhipov [188], who argued that this definition of a verifier is poor, akin to arguing that factoring an integer into its prime factors $N = pq$ should not be verified by multiplying p and q together, or that a Hamiltonian cycle in a graph shouldn't be verified by ensuring every vertex is visited exactly once. Aaronson and Arkhipov then develop further on this result, first showing that Boson Sampling is far from the uniform distribution in total variation distance, and then providing an explicit estimator which can discriminate Boson Sampling from the uniform distribution. But there are other classical distributions, notably Boson Sampling with fully distinguishable photons, which the algorithm fails to distinguish. This was resolved in work by Spagnolo et al. [189], who experimentally implemented the uniform discriminator and developed another algorithm which could differentiate ideal Boson Sampling from Boson Sampling with fully distinguishable photons. Agresti et al. [190] later used the K-means clustering algorithm from machine learning, with a three-photon Boson Sampling experiment providing the target data, to discriminate between indistinguishable and distinguishable Boson Sampling as well. Agresti et al. then used classical Boson Sampling simulators (see Section 4.6) to test how effective their discriminator was using the same original test data to distinguish experiments with up to 25 photons across up to 625 modes, and found that in most cases 10^4 samples were sufficient to distinguish the two distributions.

As we shall see in Section 4.4, distinguishing from the two distributions above is a common technique for verifying Boson Sampling experiments today [76, 191, 78, 79]. However, it is still



(a) (b)

Figure 4.3: Two examples of universal linear-optical interferometers by (4.3a) Reck et al. [176] and (4.3b) Clements et al. [177]. Each point where two spatial modes overlap is a Mach-Zehnder interferometer, as described in Section 4.3.2.

unclear what properties can be used to fully distinguish Boson Sampling from other distributions which can be classically simulated. One proposal by Carolan et al. [192] was to use photon bunching to characterise effects, but this was quickly refuted by Tichy et al. [193], who gave a Monte Carlo algorithm which could output a distribution with the same properties. Walschaers et al. [194, 195] later provided more refined statistical features from random matrix theory which can discriminate against all models discussed so far for $m < n^{5.1}$. Giordani et al. [196] demonstrate this experimentally, and use machine learning to identify what characteristics give way to these features. However, even with these approaches as m and n increase the distributions start to become more alike.

4.3.5 Universal linear optical interferometers

One of the requirements of Boson Sampling is that the unitary U should be Haar-random. Choosing a unitary purely at random might seem like a challenge at first, but there are examples of linear optical interferometers which are fully reprogrammable. We will give two examples, from which any unitary can be implemented by reprogramming the phase shifters.

The first of these schemes was realised mathematically by Hurwitz, and later rediscovered by Reck et al. [175, 176]. This scheme uses a sequence of Mach-Zehnder Interferometers arranged in a pyramid-like structure, shown in Figure 4.3a. The universality of this scheme is proven recursively, by decomposing an $m \times m$ unitary matrix into an $(m-1) \times (m-1)$ unitary matrix accompanied by m two-mode transformations in the m-th mode. These additional transformations can be implemented as MZIs. However, a potential weakness with this interferometer is that it is of very high depth; a photon starting and finishing in mode 1 might interact with up to $2m-3$ beam splitters. This could result in experiments being impractically large, and also result in issues such as photon loss, something we will discuss further in Section 4.5.2.

These issues were addressed in a later interferometer designed by Clements et al. [177]. This design is similarly constructed out of MZIs, arranged in a chequerboard-like design alternating between pairs of modes, shown in Figure 4.3b. The universality of this design is proven by decomposing the diagonals along either side of U into two-mode transformations. This design has depth m, lower than that of the Hurwitz scheme above. It is also easy to see that a lower depth scheme is impossible, as a photon in mode 1 would then not be able to reach mode m.

For the specific case of Boson Sampling, Aaronson and Arkhipov showed that the circuit depth required can be improved even further [71, 70]. This is because the only input we provide to U is the Fock state $|1^n 0^{m-n}\rangle$. It is therefore sufficient for us to consider some interferometer $\tilde{U}$ which, when applied to $|1^n 0^{m-n}\rangle$, produces the same state as U. Aaronson and Arkhipov describe such an interferometer on $m+n$ modes which uses $O(mn)$ components and has depth

$O(n \log m)$, which is an improvement over the Hurwitz and Clements schemes if $n = O(m^2)$. Note that a crucial assumption underlying this proof is that swapping mode i with mode $m+i$ is a constant-depth operation. For 2D photonic circuits, where spatial mode i is only coupled with modes $i-1$ and $i+1$, such an operation would require $O(m)$ depth and therefore lose all advantage. This setup also requires that we only apply U to the input state $|1^n 0^{m-n}\rangle$, which does not hold for the variants of Boson Sampling which we shall discuss in Section 4.3.6.

4.3.6 Boson Sampling variants

Since originally being proposed, a number of Boson Sampling variants have arisen to varying degrees of success, which aim to fix experimental challenges in vanilla Boson Sampling. We give two particular examples below which have been experimentally realised.

Problem 4.4 (Scattershot Boson Sampling)**.** *Let U be an m-mode linear optical interferometer. Sample from the distribution corresponding to measuring U acting on $|\bar{S}\rangle$ in the (anti-bunched) Fock basis, where $m = O(n^2)$ and conditioned on sampling $\bar{S}$ uniformly at random from the $\binom{m}{n}$ ways of choosing n out of m input modes.*

One of the largest issues with Boson Sampling is generating the input state $|1^n 0^{m-n}\rangle$. This is because we do not currently have deterministic single photon sources [173]. If a single photon source outputs a photon with probability p, n independent sources will all generate single photons with probability p^n, so we will have to wait an exponentially long time just to generate enough photons.

The idea of Scattershot Boson Sampling is to increase the number of probabilistic photon sources and condition the distribution on which sources output a photon [80, 81]. This can be implemented experimentally by using heralded sources, which use nonlinear optical effects such as Spontaneous Parametric Down Conversion to generate a pair of photons of different wavelengths [197]. By measuring one of these photons, we know that a pair of photons were generated in this spatial mode and therefore we can input the other photon to our interferometer. Now if there are n heralded sources each firing with probability p, then we expect on average np photons to be input to our interferometer.

It is worth noting that with heralded photon sources there is also a probability that more than one pair of photons is generated by a single source. Although it is not fully clear how the computational hardness is affected when multiple photons are generated in the same mode, it is known that photons generated in the same spatial mode do not interfere with each other. We can avoid this scenario by limiting the probability of a successful herald. Christ and Silberhorn estimate that a heralding probability of 25% is optimal for achieving the best success rate while also minimising multi-photon emissions, and recommend active switching to improve heralding rates even further [198].

Problem 4.5 (Gaussian Boson Sampling). *Let U be an m-mode linear optical interferometer. Sample from the distribution corresponding to measuring U acting on $|\zeta^n 0^{m-n}\rangle$ in the (anti-bunched) Fock basis, where $m = O(n^2)$ and $|\zeta\rangle$ is a single mode squeezed vacuum state.*

Gaussian Boson Sampling is another proposed way of working around the need for single photon sources [82]. The idea is to start with $k \approx n$ single mode squeezed states, each generated from the $|0\rangle$ Fock state. These squeezed states are then input into an interferometer and measured in the Fock basis.

Hamilton et al. [82] used arguments akin to Aaronson and Arkhipov to argue that Gaussian Boson Sampling is computationally hard. The main difference between the two is that the probability of a particular output from Gaussian Boson Sampling is proportional to another

#P-Hard function called the Hafnian of a $2n \times 2n$ matrix, rather than the permanent of an $n \times n$ matrix. From this starting point the rest of the argument follows through. Another intuitive way of seeing this is in Aaronson, courtesy of García-Patrón [80], by noting that Scattershot Boson Sampling can be reduced to Gaussian Boson Sampling. To do this, we simply embed the photon pair sources into the Gaussian Boson Sampling experiment by generating two-mode entangled squeezed states and using them as input.

Like Boson Sampling, a few modifications have been made to Gaussian Boson Sampling over time to make it more practical. Kruse et al. [199] considered the effects of multiple photons in the same output mode, as well as what happens if squeezed coherent states are used as input to our interferometer, rather than simply the squeezed vacuum state. Note that Boson Sampling using coherent states without any squeezing can be simulated in polynomial time [83]. And Quesada, Arrazola and Killoran [181] considered the effect of using threshold detectors in Gaussian Boson Sampling. To do this, they derived a similar function to the Hafnian, called the Torontonian[3]. Quesada, Arrazola and Killoran showed that computational hardness was preserved, but could only show this in the regime where the probability of seeing multiple photons in the same output mode was small, effectively reducing the hardness of Gaussian Boson Sampling with threshold detectors to the hardness of standard Gaussian Boson Sampling.

Another unexpected benefit of Gaussian Boson Sampling is that there are some applications demonstrated. Huh, et al. [93] showed that Gaussian Boson Sampling could be used to simulate the vibronic spectra of molecules. It has also recently been that graphs can be encoded into Gaussian Boson Sampling experiments to solve various problems including Graph Isomorphism and finding dense subgraphs [94, 95, 201], though it is unclear how these techniques fare against the best classical algorithms [41].

It is worth noting that measuring in the Fock basis is crucial for computational hardness in Gaussian Boson Sampling. In particular, if the measurement is instead a Homodyne measurement for measuring Gaussian states, the resulting distribution is classically simulable in polynomial time [202].

4.4 Experimental achievements

Since originally being proposed in 2010, Boson Sampling has taken the interest of a number of quantum optics research groups, each eager to demonstrate a quantum advantage through this process. We provide some of the results of said groups below.

The earliest experimental demonstrations were shown in 2013, by four different groups independently [72, 73, 74, 75]. These results were published simultaneously, with two publications in *Nature Photonics* and two in *Science*. All four results used reprogrammable designs on integrated photonic circuits to implement the unitary transformation. Crespi et al. [75] and Tillmann et al. [74] both demonstrated Boson Sampling with up to three photons across five modes. Broome et al. [72] demonstrated three photons across six spatial modes, while Spring et al. [73] demonstrated four photons across six modes. Crespi et al., Broome et al., and Spring et al., verified their experimental results by computing the distance between their experimental distribution and the theoretical distribution [73, 74, 75], while Tillman et al. verified their experiment by estimating the fidelity [74].

Later in 2015, an experiment by Carolan et al. [76] demonstrated experimental Boson Sampling with six photons across six modes. This was implemented on a fully reprogrammable

[3] The naming of this function is a reference to two things: First that the three authors work for Xanadu, a quantum computing startup in Toronto; and second, that the Hafnian was named by Eduardo Caianiello after visiting the city of Copenhagen (Hafnia in Latin) [200].

integrated silica circuit. One caveat is that the authors used the input Fock state $|3,3,0,0,0,0\rangle$, rather than the traditional form of having single photons in each mode. The authors justify this in order to use the verification approach of [192], though as discussed in Section 4.3.4 this verification approach has limitations of its own. This device was later also used by Sparrow, Martín-López et al. [92] with two-mode squeezed inputs and adaptive feedback to simulate the vibronic structure of a variety of 4-atom molecules, hinting at a potential application for bosonic sampling.

The first Boson Sampling demonstration with five single photons was performed in 2017 by Wang et al. [77]. Wang et al. used a triangular array of beam splitters bonded together to create a structure smaller than regular bulk optics but larger than integrated silicon or silica chips, featuring 9 input and output modes. Wang et al. state that many interferometers can be generated from this chip by adjusting the polarisation of photons, but acknowledge that the structure itself is not universal. Boson Sampling is verified in this experiment by checking against the uniform and fully distinguishable distributions.

Following this came a few results on up to five photons for Boson Sampling variants. Zhong et al. [191] demonstrated Scattershot Boson Sampling with up to five photons across twelve modes, using an interferometer constructed by six trapezoidal quartz blocks covered in film and fused together. Then, Paesani et al. [78] demonstrated four photon Scattershot, Gaussian and vanilla Boson Sampling on a single integrated silicon chip, by simply adjusting which state they input to the chip. Finally, Zhong et al. [203] demonstrated Gaussian Boson Sampling with up to five photons, using the same interferometer as in [191]. All of the standard and Scattershot Boson Sampling experiments were validated by comparison to distinguishable and uniform sampling. In the case of Gaussian Boson Sampling, comparisons were also made to thermal states, non-squeezed coherent states and two-mode squeezed states, as well as demonstrating applications related to molecular simulations and graph theory [78, 203].

The most recent experimental demonstration of Boson Sampling was performed in 2019 by Wang et al. [79]. This experiment used a 3D integrated photonic circuit with 60 spatial modes. Wang et al. managed to demonstrate standard Boson Sampling with up to 10 photons, as well as Boson Sampling under loss with up to 20 input photons and 14 output photons, both of which were verified by comparison to uniform and distinguishable photon distributions. To date this is the largest demonstration of Boson Sampling publicly announced.

4.5 Experimental imperfections in linear optics

We shall now discuss two imperfections that are particularly common in Boson Sampling. The first is distinguishability, when one or more factors make bosons distinct from one another. The second, loss, is when photons which are generated at input are not detected at the output. We shall discuss some ways in which these issues occur, how they affect the probability distributions, and what is known about their computational complexity. In Section 4.6, we shall discuss some of the classical simulation algorithms that have been developed surrounding these issues.

4.5.1 Distinguishability

One of the points made when discussing Boson Sampling is that the photons need to be indistinguishable. This means that if we permuted the photons in some way, it would be impossible for someone else to identify which permutation was applied.

A number of internal characteristics about the photons could make them distinguishable. The most visually intuitive one is wavelength: A red photon is definitely distinct from a green photon. But other aspects can also distinguish photons, such as their polarisation, or what time they were generated. The time generation point is particularly important following the discussion in Section 4.3.6 about the challenges of single photon sources [173]: Not only do all of these single photon sources need to emit a photon, but all of them need to emit a single photon *at the exact same time* [70]. We can model distinguishability by introducing additional modes to describe the internal state. Throughout this thesis, we will use spatial or "System" modes to describe where a photon is, and internal or "Label" modes to describe its distinguishable characteristics.

Before we consider Boson Sampling with distinguishability, we will first return to the Hong-Ou-Mandel dip from Section 4.3.2 [180]. This time, we will consider one red photon generated in mode 1, and one photon generated in mode 2 which is red with amplitude $v \in [0,1]$ and green with amplitude $d = \pm\sqrt{1-v^2}$. We can represent this in second quantisation by introducing additional indices to our creation and annihilation operators: $a^\dagger_{i,R}$ and $a^\dagger_{i,G}$ to indicate a red or green photon in mode i, respectively. Our input is now

$$a^\dagger_{1,R}\left(va^\dagger_{2,R} + da^\dagger_{2,G}\right)|0_R, 0_G, 0_R, 0_G\rangle\,. \tag{4.47}$$

A beam splitter only affects System modes, and is invariant over Label modes. After applying the beam splitter our photons are in the state

$$\begin{aligned} a^\dagger_{1,R}\left(va^\dagger_{2,R} + da^\dagger_{2,G}\right)|0_R, 0_G, 0_R, 0_G\rangle \to\ & \frac{1}{2}\left(a^\dagger_{1,R} + ia^\dagger_{2,R}\right) \\ & \times \left(v\left(ia^\dagger_{1,R} + a^\dagger_{2,R}\right) + d\left(ia^\dagger_{1,G} + a^\dagger_{2,G}\right)\right)|0_R, 0_G, 0_R, 0_G\rangle \end{aligned} \tag{4.48}$$

$$\begin{aligned} &= \frac{v}{2}\left(i(a^\dagger_{1,R})^2 + a^\dagger_{1,R}a^\dagger_{2,R} - a^\dagger_{2,R}a^\dagger_{1,R} + i(a^\dagger_{2,R})^2\right) \\ &\quad \times |0_R, 0_G, 0_R, 0_G\rangle \\ &\quad + \frac{d}{2}\left(ia^\dagger_{1,R}a^\dagger_{1,G} + a^\dagger_{1,R}a^\dagger_{2,G} - a^\dagger_{2,R}a^\dagger_{1,G} + ia^\dagger_{2,R}a^\dagger_{2,G}\right) \\ &\quad \times |0_R, 0_G, 0_R, 0_G\rangle \\ &= \frac{v}{2}\left(i(a^\dagger_{1,R})^2 + i(a^\dagger_{2,R})^2\right)|0_R, 0_G, 0_R, 0_G\rangle \\ &\quad + \frac{d}{2}\left(ia^\dagger_{1,R}a^\dagger_{1,G} + a^\dagger_{1,R}a^\dagger_{2,G} - a^\dagger_{2,R}a^\dagger_{1,G} + ia^\dagger_{2,R}a^\dagger_{2,G}\right) \\ &\quad \times |0_R, 0_G, 0_R, 0_G\rangle \\ &= \frac{v}{2}\left(i(a^\dagger_{1,R})^2 + i(a^\dagger_{2,R})^2\right)|0_R, 0_G, 0_R, 0_G\rangle \\ &\quad + \frac{d}{2}\left(ia^\dagger_{1,R}a^\dagger_{1,G} + a^\dagger_{1,R}a^\dagger_{2,G} - a^\dagger_{2,R}a^\dagger_{1,G} + ia^\dagger_{2,R}a^\dagger_{2,G}\right) \\ &\quad \times |0_R, 0_G, 0_R, 0_G\rangle \end{aligned} \tag{4.49}$$

$$. \tag{4.50}$$

Note that in the final line, the terms $a^\dagger_{1,R}a^\dagger_{2,G}$ and $a^\dagger_{2,R}a^\dagger_{1,G}$ do not cancel out, due to the different creation operators from the fact that the photons are different colours. We apply our creation operators to reach the final state

$$\begin{aligned} &\frac{1}{2}\left(v\left(i\sqrt{2}\,|2_R, 0_G, 0_R, 0_G\rangle + i\sqrt{2}\,|0_R, 0_G, 2_R, 0_G\rangle\right)\right) \\ &+ \frac{1}{2}\left(d\left(i\,|1_R, 1_G, 0_R, 0_G\rangle + |1_R, 0_G, 0_R, 1_G\rangle - |0_R, 1_G, 1_R, 0_G\rangle + i\,|0_R, 0_G, 1_R, 1_G\rangle\right)\right). \end{aligned} \tag{4.51}$$

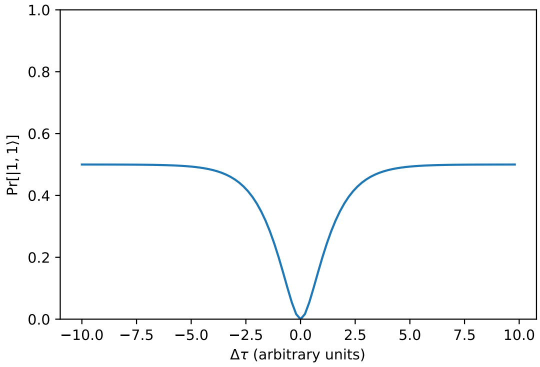


Figure 4.4: The probability of seeing the coincident $|1,1\rangle$ output from a Hong-Ou-Mandel experiment over $\Delta\tau$.

If we only measure what spatial mode our photons are in and discard the wavelength, we will find that we are in the Fock state $|2,0\rangle$ with probability $v^2/2 + d^2/4$ and likewise for $|0,2\rangle$, but we'll also find ourselves in the state $|1,1\rangle$ with probability $d^2/2$. Noting that $d^2 = 1 - v^2$ and that $v = e^{-|\Delta\tau|}$ for some $\Delta\tau \in \mathbb{R}$, it is easy to see that adjusting $\Delta\tau$ determines how likely we are to see coincidences at the output: if $\Delta\tau = 0$, meaning that our photons are completely indistinguishable, we don't see any coincidences at all; as $\Delta\tau \to \pm\infty$, corresponding to completely distinguishable photons, then we see coincidences with probability $1/2$. This is where the "dip" in Hong-Ou-Mandel Dip comes from: as our photons become more indistinguishable we see a drop in coincidences. This is plotted visually in Figure 4.4. The extent to which we see a dip is known as the visibility of a Hong-Ou-Mandel dip, hence v, and is a standard technique when characterising linear optics experiments to determine how distinguishable pairs of photons are. Physically, $\Delta\tau$ corresponds to the difference between our photons, for example if there is a difference in path length between the photon sources and the beamsplitter.

In first quantisation, we can likewise realise this interference by introducing a second register noting what Label mode our photons are in. Our state is now

$$|\psi\rangle = \frac{1}{\sqrt{2}}\left(v(|12\rangle + |21\rangle)\,|RR\rangle + d(|12\rangle\,|RG\rangle + |21\rangle\,|GR\rangle)\right). \tag{4.52}$$

Our unitary U will act as $U \otimes U$ on the spatial modes and $I \otimes I$ on the Label modes. Thus, our state evolves to

$$\begin{aligned} U^{\otimes 2} \otimes I^{\otimes 2}\,|\psi\rangle = & \frac{v}{2\sqrt{2}}\big((2i\,|11\rangle + (1+i^2)\,|12\rangle + (i^2+1)\,|21\rangle + 2i\,|22\rangle)\,|RR\rangle\big) \\ & + \frac{d}{2\sqrt{2}}\left(i\,|11\rangle\,(|RG\rangle + |GR\rangle) + i\,|22\rangle\,(|RG\rangle + |GR\rangle)\right) \\ & + \frac{d}{2\sqrt{2}}\left(|12\rangle\,(|RG\rangle - |GR\rangle) + |21\rangle\,(|GR\rangle - |RG\rangle)\right) \qquad (4.53) \\ = & \frac{iv}{\sqrt{2}}\left((|11\rangle + |22\rangle)\,|RR\rangle\right) \\ & + \frac{d}{2\sqrt{2}}\left(i(|11\rangle + |22\rangle)(|RG\rangle + |GR\rangle) + (|12\rangle - |21\rangle)(|RG\rangle - |GR\rangle)\right). \qquad (4.54) \end{aligned}$$

Again, we see that coincidences occur with probability $d^2/2$, matching the first quantisation result.

A rich library of theoretical work has been developed on the topic of Boson Sampling under distinguishability in recent years [204, 205, 206, 207, 208]. Here, we shall follow the notation of Tichy [206], but note that this model of distinguishability is equivalent to several of the other models also mentioned. In Tichy's model, each boson is assigned an "internal" state $|\Phi_i\rangle$, where $i = 1, \cdots, m$. Thus, in our terminology, every boson in System mode i has a Label state given by Φ_i. An $m \times m$ Gram matrix $\mathcal{S}_{i,j} = \langle \Phi_i | \Phi_j \rangle$ is used to discribe the indistinguishability between photons generated in mode i and photons generated in mode j, noting that any pair of photons generated in the same mode are indistinguishable from one another. Under Tichy's model, the probability of outcome $|S'\rangle$ is

$$\Pr[S'] = \frac{1}{\prod_{i=1}^{m} S_i! S_i'!} \sum_{\tau,\tau' \in \mathrm{S}_n} \prod_{k=1}^{n} U_{s_k', s_{\tau(k)}} U^*_{s_k', s_{\tau'(k)}} \mathcal{S}_{\tau'(k),\tau(k)}. \tag{4.55}$$

By noting that $\sum_\tau \prod_i M_{i,\tau(i)} = \mathrm{per}(M)$, we can rewrite the sum over τ into a matrix permanent and see that

$$\Pr[S'] = \frac{1}{\prod_{i=1}^{m} S_i! S_i'!} \sum_{\sigma \in \mathrm{S}_n} \prod_{i=1}^{n} \mathcal{S}_{i,\sigma(i)} \, \mathrm{per}(U_{S',S} * U^*_{S',\sigma(S)}), \tag{4.56}$$

where $*$ denotes element-wise multiplication, M^* is the matrix whose elements are complex conjugates of M, and $\sigma(S)$ means the elements of S permuted by σ. Again, the Hong-Ou-Mandel Dip can be visualised in this instance by noting that $\mathcal{S}_{1,2} = \mathcal{S}_{2,1} = v$:

$$\begin{aligned}
\Pr[(1,1)] &= \mathrm{per}\begin{pmatrix} \frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{2}} & \frac{i}{\sqrt{2}} \times \frac{-i}{\sqrt{2}} \\ \frac{i}{\sqrt{2}} \times \frac{-i}{\sqrt{2}} & \frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{2}} \end{pmatrix} + v^2\, \mathrm{per}\begin{pmatrix} \frac{1}{\sqrt{2}} \times \frac{-i}{\sqrt{2}} & \frac{i}{\sqrt{2}} \times \frac{1}{\sqrt{2}} \\ \frac{i}{\sqrt{2}} \times \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \times \frac{-i}{\sqrt{2}} \end{pmatrix} && (4.57)\\
&= \mathrm{per}\begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix} + v^2\, \mathrm{per}\begin{pmatrix} \frac{-i}{2} & \frac{i}{2} \\ \frac{i}{2} & \frac{-i}{2} \end{pmatrix} && (4.58)\\
&= \frac{2 - 2v^2}{4} && (4.59)\\
&= \frac{1 - v^2}{2} && (4.60)\\
&= \frac{d^2}{2}, && (4.61)
\end{aligned}$$

matching the result discussed above.

So what can be done to work around distinguishability? A few options are available. Shchesnovich used average mutual fidelity to give an upper bound on the complexity, stating that a sampling experiment is more powerful when the single-photon mode mismatch scales as $O(n^{-3/2})$ for n photons [209]. Rohde and Ralph [210] suggest applying filtering to discard photons which are too distinguishable. This does end up creating loss as a result, trading one imperfection for another, but this can be tolerable to some extent, as we shall see in Section 4.5.2 [210, 211]. Another option, known as Multi-Boson Correlation Sampling [212, 213, 214, 215, 216], is to measure across some of these characteristics in such a way that the photons end up essentially behaving indistinguishably. Laibacher and Tamma used an example where n photons were of different colours, and showed that a suitable time interval and polarisation could be chosen such that at the detectors the photons appeared indistinguishable. This was later demonstrated experimentally by Wang et al. [217], as well as a subsequent experiment by Orre et al. [218] where frequency-resolving detectors were used to demonstrate Boson Sampling with photons generated at different time.

Finally, we note there are a number of results that link distinguishability in linear optics to representation theory [219, 220, 221, 208, 222]. We will introduce representation theory in Section 4.7, and explain its link to Boson Sampling in more detail in Chapter 5.

4.5.2 Loss

Now we turn to photon loss, where we know that some photons were generated at the input but not detected at the output. There are a wide variety of reasons for why loss might occur, including but not limited to coupling between optical components, two-photon absorption, Rayleigh scattering and detector inefficiencies [223, 224]. Note that unlike distinguishability, here we will discuss two different ways of modelling photon loss, based on first and second quantisation, respectively.

First, we shall look at the model of photon loss in second quantisation, otherwise known as the “beam splitter” model [86, 83, 87]. In this model, loss is induced by adding “lost” modes $a^\dagger_{\mathrm{L},i}$ to our experiment. These additional modes are coupled to spatial modes in our experiment by a single beam splitter with transmission $\sqrt{\eta}$ for some $\eta \in [0,1]$

$$a_1^\dagger \to \sqrt{\eta} a_1^\dagger + \sqrt{1-\eta} a^\dagger_{\mathrm{L},i} \tag{4.62}$$

such that a photon which starts in this spatial mode ends up in our lost mode with probability $1-\eta$. The fact that only a single beam splitter couples this additional mode to a spatial mode in our experiment is required to ensure that any photons which are lost do not reappear later in the experiment. Finally, these additional spatial modes are traced out to induce the photon loss

$$\begin{aligned}
\mathrm{Tr}_{\mathrm{L}}[a_1^\dagger \left|0, 0_{\mathrm{L}}\right\rangle] &\to \mathrm{Tr}_{\mathrm{L}}[(\sqrt{\eta} a_1^\dagger + \sqrt{1-\eta} a_{\mathrm{L}}^\dagger) \left|0, 0_{\mathrm{L}}\right\rangle] && (4.63)\\
&= \mathrm{Tr}_{\mathrm{L}}[\sqrt{\eta} a_1^\dagger \left|0, 0_{\mathrm{L}}\right\rangle + \sqrt{1-\eta} a_{\mathrm{L}}^\dagger \left|0, 0_{\mathrm{L}}\right\rangle] && (4.64)\\
&= \mathrm{Tr}_{\mathrm{L}}[\sqrt{\eta} \left|1, 0_{\mathrm{L}}\right\rangle + \sqrt{1-\eta} \left|0, 1_{\mathrm{L}}\right\rangle] && (4.65)\\
&= \eta \left|1\right\rangle \left\langle 1\right| + (1-\eta) \left|0\right\rangle \left\langle 0\right| . && (4.66)
\end{aligned}$$

At first glance, it might seem like this model of loss might require an arbitrarily large number of additional spatial modes, due to the requirement that each loss-inducing beam splitter couples to an additional spatial mode. However, note that if we have two loss-inducing beam splitters in sequence with survival probabilities η_1 and η_2, respectively, then this is equivalent to a singe loss-inducing beam splitter with overall survival probability $\eta_1\eta_2$. This reduces the number of additional modes required to a number polynomial in terms of how many optical components are in our experiment. As shown in Section 4.3.5, this number is polynomial in m and n. It is in fact possible to reduce this number of additional modes even further, in cases where the loss-inducing components can commute with the optical components in an experiment. As we will see in Section 4.6.3, this is often used to approximate a lossy linear optical interferometer as a single layer of loss followed by a less lossy interferometer.

We now move to the model of photon loss in first quantisation. Note that the beam splitter model considered above is inconvenient in first quantisation due to the fact that it is tracing out spatial modes, which is not easily done in first quantisation. Likewise, this next model, which can be seen as tracing out particles, is inconvenient in second quantisation.

Consider a Boson Sampling experiment which starts with n photons, of which only k are detected. Crucially, not only does loss reduce the number of photons, but also we do not know which of the photons have been lost. In first quantisation, this is equivalent to performing Boson Sampling with input given by the mixed state

$$\rho = \sum_{\substack{K \subseteq S \\ |K|=k}} p_K \rho_K, \tag{4.67}$$

where ρ_K describes a photonic input state with k photons whose spatial modes are specified by K. When this loss is uniform, meaning that $p_K = 1/\binom{n}{k} \forall K$, this is equivalent to tracing out $n-k$ particles [86]. This leads to the probability distribution forming a convex sum over different subsets of input photons

$$\Pr[K'] = \sum_{\substack{K \subseteq S \\ |K|=k}} p_K |\operatorname{per}(U_{K',K})|^2, \tag{4.68}$$

where p_K is the probability of the photons which started in input modes K surviving, with $\sum_K p_K = 1$.

One known complexity result is by Aaronson and Brod [211], who considered Boson Sampling with uniform loss. Aaronson and Brod showed that in this model, Boson Sampling remains classically hard if $k = n - \ell$ for some constant ℓ. This was proven by showing that if someone could estimate a convex sum of squared permanents of submatrices of an $(n) \times (n)$ matrix, then one can estimate the squared permanent of an $(n-\ell) \times (n-\ell)$ matrix with bounded probability. However, this only works if a constant number of photons are lost, regardless of how large n is. No hardness proofs exist even for the case of a constant fraction of photons being lost.

Another workaround for loss was suggested by Latmiral, Spagnolo and Sciarrino [225]. This proposal considers Scattershot Boson Sampling under photon loss. This is a sensible proposal, as the point of Scattershot Boson Sampling is to use a large number of photon sources to produce a small number of photons at the output. However, one weakness is that there is now a chance of a photon not being heralded, as the photon measured to detect its presence might be lost. Latmiral, Spagnolo and Sciarrino propose working around this by placing a shutter on each input mode, such that only heralded photons can enter the interferometer. From analysis of the distribution under this new architecture, Latmiral, Spagnolo and Sciarrino gave bounds for how long it would take to simulate lossy Scattershot Boson Sampling, tough it is important to note that the classical algorithm used was to compute the probability of every single outcome and then sample over the full probability distribution. For this naïve sampling strategy, Latmiral, Spagnolo and Sciarrino showed that as few as 7 photons across 50 modes would be sufficient for a quantum advantage, but there are far more efficient classical algorithms as we will see in Section 4.6.

Two papers have experimentally explored Boson Sampling under loss. The first, in 2018 by Wang et al. [226], considers five, six and seven photon Boson Sampling with one and two photons lost. The second, as mentioned in Section 4.4, was performed in 2019 by Wang et al. [79], and demonstrates Boson Sampling with 20 input photons of which up to 14 survive.

4.5.3 Other imperfections

Before we move on to classical simulations, we will briefly mention here other imperfections that might occur or be considered in a Boson Sampling experiment, as well as what results are known for these models.

Errors in the interferometer implemented

It is not necessarily possible to implement a random unitary U perfectly in an interferometer. Even when using universal linear optical interferometers such as those mentioned in Section 4.3.5, an interferometer might require us to apply a phase shift that our electronic components are not fine-grained enough to implement. There may also be other component imperfections, such as a beam splitter which is not perfectly balanced.

The complexity of Boson Sampling when our interferometer is not implemented perfectly was first considered by Leverrier and Garcáa-Patrón [227]. Let U be our ideal unitary, and $\tilde{U}$ be the unitary actually implemented by our interferometer. Leverrier and Garcáa-Patrón showed that it is necessary for each optical component to have fidelity $1 - O(1/n^2)$ in order for the probability of an outcome S' under $\tilde{U}$ to be approximately close to the probability of the same outcome under U. Arkhipov [228] improved this result, showing that it is sufficient for each optical component to be implemented up to $O(\epsilon/n^2 \log m)$ error in order for the L_1 distance between the distributions generated by U and $\tilde{U}$ to be at most ϵ. Note that this result by Arkhipov uses the $O(n \log m)$-depth interferometer mentioned in Section 4.3.5; for other universal interferometers [175, 176, 177], a smaller component-wise error might be required.

It is interesting to note that a constant error in each optical component is not sufficient for Boson Sampling, and that instead the component-wise error must decrease as m and n increase. Arkhipov argues that this is unsurprising; each photon interacting with the interferometer induces some error, and as the size of the unitary grows so too must the number of optical components.

Low depth interferometers

Loss is more likely to occur in Boson Sampling if the interferometer has a high depth. An intuitive question to ask in this case is what can be done with a Boson Sampling circuit of reduced depth? A hardness proof for exact sampling was given Brod [229], showing that even four layers of beam splitters was sufficient to ensure hardness in the exact case. Classical algorithms for these instances were later devised, by taking advantage of the low depth to only simulate interference between photons which could actually cross paths [230, 231]. Note that these classical simulation algorithms inherently assume the photons are not all generated adjacently from each other.

Dark counts

Dark counts occur when a detector registers a photon even though one was not present. This is most often due to thermal noise within the single photon detector, meaning that even if the detector was in complete darkness it would still claim that photons were present.

Dark counts themselves are not much of a challenge in Boson Sampling experiments; superconducting nanowire single photon detectors have sufficiently low dark counts that false clicks can be noticed easily and the experiment discarded [232]. However, where they become challenging is when loss is also involved: It is very difficult to differentiate between a photon not being lost and a photon being lost combined with a false click at the detector. The intersection of these two imperfections was considered by Shchesnovich [233], who showed that both dark counts and loss can be considered as producing additive errors with respect to m and n.

Gaussian noise

A Gaussian noise model was proposed as a way of considering errors in Boson Sampling by Kalai and Kindler [234]. In this model, the probability of an outcome S' is still related to the permanent of a matrix, but now the matrix in question is $\sqrt{1-\epsilon}U_{S',S} + \sqrt{\epsilon}X$ for some Gaussian matrix X that acts as our noise. Kalai and Kindler suggest that for $\epsilon = \omega(1/n)$, vanilla Boson Sampling and Boson Sampling under Gaussian noise are already uncorrelated, and furthermore for $\epsilon = \Omega(1)$, that the probability of an outcome of Boson Sampling under Gaussian noise can be approximated by up to constant error.

4.6 Classical simulation algorithms

Having introduced distinguishability and loss, we are now ready to discuss classical simulation algorithms for these imperfections. We will start by explaining classical algorithms for vanilla Boson Sampling, which were the original classical simulators devised. We will then discuss Boson Sampling under fully distinguishable photons, as well as Boson Sampling under various models of loss. Finally, we shall consider the classical simulation algorithms of [84, 85], which were the first known classical algorithms to consider both distinguishability and loss.

4.6.1 Simulation algorithms for ideal Boson Sampling

Boson Sampling under ideal conditions (lossless indistinguishable single photons) is intractable for sufficiently large n. Until recently the only classical simulation method explicitly known was to compute the entire probability distribution before taking a sample, though it was widely believed that more efficient, albeit still exponential time, approaches existed. A brute force method cannot scale, due to both the number of possible outcomes and the complexity of computing even one $n \times n$ complex matrix permanent.

Two major results gave the first explicit classical simulation strategies which were faster than brute-force sampling. The first, by Neville et al. [235], demonstrated that Boson Sampling experiments with up to 30 photons could be simulated on a single laptop, and suggests that a supercomputer could handle up to 50 photons. This was achieved by starting with the classical distribution of n distinguishable photons, and then using Metropolised Independence Sampling to adapt the distribution to that of ideal Boson Sampling. The second result, by Clifford & Clifford [91], gave a classical algorithm for exact Boson Sampling and runs in $O(n^2 2^n + mn^2)$. In practice, the time required is equivalent to computing two $n \times n$ matrix permanents, plus a polynomial overhead. This is through a combination of optimizations, particularly computing marginal probabilities and sampling via the chain rule.

4.6.2 A simulation algorithm for fully distinguishable Boson Sampling

In the case where the n input photons are fully distinguishable, a simple polynomial time algorithm exists [188]. In this case, there is no photon interference, so photons can be sampled individually. This is done by taking a photon which starts in mode i, and sampling output mode j with probability $|U_{j,i}|^2$. Repeating for all photons gives us the complete sample in $O(mn)$ time.

4.6.3 Simulation algorithms for Boson Sampling with loss

Another common source of imperfections in linear optics is that of photon loss, which arises through a number of different means. Indeed, any large-scale demonstration of Boson Sampling is bound to face photon loss, and therefore needs to take such issues into account. Some results have already shown instances where hardness is still retained, such as when only a constant number of photons are lost [211, 226].

Neville et al. compared the classical simulation of their approach to a Boson Sampling experiment where any photon loss was considered a rejected experiment [235]. Novel classical simulations for Boson Sampling under loss have also been considered by use of classically simulable states such as thermal [83] or separable [86] states.

There has also been some consideration of how classical simulations can be generalised to non-uniform loss. This usually means photon loss that is dependent on the number of optical components, with each component having transmission probability τ. Classical simulation methods can be generalised to this model by identifying a layer of uniform losses from the circuit, followed a non-uniform lossy circuit which can be simulated classically through the use of additional modes for lost photons [83, 86]. These results showed that Boson Sampling under non-uniform loss can be classically simulated as long as every photon encounters at least $O(\log n)$ components. More recently, Brod and Oszmaniec developed on these methods to give a polynomial-time algorithm in the case where some photons experience little or no loss while others experience significant amounts of loss, by extracting losses into a layer of non-uniform loss and simulating via a generalisation of the Clifford & Clifford algorithm [87].

4.6.4 Simulation under multiple imperfections

In [84, 85], Renema et al. consider the Tichy model of inter-photon distinguishability described in Section 4.5.1. As mentioned previously, the probability distribution of arbitrarily distinguishable bosons in a coincident occupation S' is modelled as

$$\Pr[S'] = \sum_{\sigma \in \mathrm{S}_n} \prod_{i=1}^{n} \mathcal{S}_{i,\sigma(i)} \operatorname{per}(M * M^*_{1,\sigma}), \tag{4.69}$$

where $\mathcal{S}$ is the same matrix describing the distinguishability as in the previous section, M is a matrix defined by the rows and columns of our interferometer U selected based on our photon output S' and input S, $M^*_{1,\sigma}$ is the conjugate matrix with the identity permutation applied to rows and permutation σ applied to columns, and $*$ denotes element-wise multiplication. They further restrict to a model where the indistinguishability overlap is defined by a single parameter $\mathcal{S}_{i,j} = x + (1-x)\delta_{i,j}, x \in [0, 1]$. Physically, this corresponds to a situation where all photons have equal amounts of distinguishability, independent of which spatial mode they are generated in. It is worth noting that more complicated causes of distinguishability, such as photons produced by a quantum dot where distinguishability depends, among other things, on what time a photon is generated, might require more complex models than simply a single parameter [236]. Renema et al. argue informally at the end of [84] that techniques described here can also be applied to more general cases.

The sum over permutations can be ordered based on how many *fixed points* a permutation has, giving

$$\Pr[S'] = \sum_{j=0}^{n} \sum_{\sigma^j} x^j \operatorname{per}(M * M^*_{1,\sigma}), \tag{4.70}$$

where σ^j denotes permutations which have $n - j$ elements as fixed points. Each permanent can be broken down via the Laplace expansion into a sum of a complex matrix permanent multiplied by a positive matrix permanent:

$$\Pr[S'] = \sum_{j=0}^{n} \sum_{\sigma^j} x^j \sum_{\substack{J' \leq S' \\ |J'|=j}} \operatorname{per}(M_{J',1} * M^*_{J',\sigma_p}) \operatorname{per}(|M_{\bar{J}',\sigma_u}|^2), \tag{4.71}$$

where J' is an occupation with j photons. Here we are now choosing submatrices of M, with J' representing the $\binom{n}{j}$ possible combinations of rows from M, $\bar{J}'$ representing the remaining rows, and σ_p and σ_u representing permuted and unpermuted elements of σ respectively. The $J' \leq S'$ notation is used to indicate that J' is a Fock state such that $J_i \leq S'_i \, \forall i \in [m]$.

The classical simulation method used truncates the number of non-fixed points in a permutation as being at most k, with the remainder of the probability treated as an error margin. It is important to note while these approximations are real, they are not necessarily positive. This is due to the truncation, where positive higher order terms which would have corrected the probability to be positive are now missing from the approximation. To correct this, any negative approximations are rounded up to 0. These probabilities are then used to train a Metropolised Independence Sampler, akin to the technique of [235]. Training this sampler requires approximating a number of probabilities dependent on the underlying distribution, each of which involves computing $O(n^{2k})$ permanents of $k \times k$ complex matrices, and the same number of permanents of $(n-k) \times (n-k)$ matrices with non-negative entries. The permanents of $k \times k$ complex matrices can be computed classically in $O(k2^k)$ time via Ryser's algorithm, and the permanents of matrices with non-negative entries can be approximated up to multiplicative error in polynomial time [237, 238]. As long as k is independent of n, this means that there is a polynomial runtime.

To work out a suitable value k, define coefficients c_j as

$$c_j = \sum_{\sigma^j} \sum_{\substack{J' \leq S' \\ |J'|=j}} \operatorname{per}(M_{J',1} * M^*_{J',\sigma_p}) \operatorname{per}(|M_{\bar{J}',\sigma_u}|^2). \tag{4.72}$$

Assuming the matrices are Gaussian, the variance of each permanent can be bounded as

$$\operatorname{Var}[\operatorname{per}(M_{J',1} * M^*_{J',\sigma_p})] = \frac{j!}{m^{2j}}, \tag{4.73}$$

and

$$\operatorname{Var}[\operatorname{per}(|M_{\bar{J}',\sigma_u}|^2)] < \frac{(n-j)!}{m^{2(n-j)}} \sum_{l=0}^{n-j} \frac{1}{l!}. \tag{4.74}$$

This leads to two key results. The first is that the variance of c_j tends towards a constant value:

$$\operatorname{Var}[c_j] < \left(\frac{n!}{m^n}\right)^2 \frac{1}{e} \sum_{l=0}^{n-j} \frac{1}{l!} \tag{4.75}$$

$$\to \left(\frac{n!}{m^n}\right)^2 \text{ as } n \to \infty, \tag{4.76}$$

and the second is that the covariance for different values of j is zero:

$$\operatorname{Cov}[c_j, c'_j] = 0 \, \forall \, j \neq j'. \tag{4.77}$$

From this one can approximate the variance of the error as a geometric series, which as $n \to \infty$ tends towards the inequality

$$\operatorname{Var}[\Delta \Pr[S']] = \operatorname{Var}[\Pr[S'] - \Pr_k[S']] \tag{4.78}$$

$$= \sum_{j=k+1}^{n} x^{2j} \operatorname{Var}[c_j] \tag{4.79}$$

$$< \left(\frac{n!}{m^n}\right)^2 \left(\frac{x^{2(k+1)}}{1-x^2}\right), \tag{4.80}$$

where P_k is the probability distribution when truncated at $j \leq k$ for some $k < n$.

Finally one can use a Markov inequality to show that if the variance of the error is of the form $(n!/m^n)^2\epsilon^2$, the average error of the simulation is at most ϵ [85]. Crucially, this value of ϵ is only dependent on x and k and no longer dependent on n. This means that for any value of x, one can choose a suitable value of k to achieve a required error ϵ, and run a classical simulation in time polynomial in n.

In [85], the method described above was adapted to consider uniform loss as well as distinguishability, showing that the same result can be found, with the only difference being that x is now replaced by $\alpha = \sqrt{\eta}x$, where η is the probability of each individual photon surviving. Crucially, this result demonstrated that Boson Sampling where a constant fraction of photons were lost can be simulated in $O(\ell^{2k}k2^k)$, where ℓ is the number of photons which survive and k is only dependent on the constant ℓ/n, distinguishability x, and the desired accuracy of the simulation. This can be expanded to classically simulating Boson Sampling under uniform loss by sampling ℓ from the binomial distribution before sampling output photons, which offers a runtime of $O(n^{2k}k2^k)$.

Although this runtime is polynomial in terms of n and can therefore be considered asymptotically efficient, it might not be classically simulable in practice. There are three main contributions to this: First, the algorithm is reliant on Metropolised Independence Sampling, which potentially requires many probabilities to be approximated per sample. Second, approximating each probability requires $O(n^{2k})$ permanents of $k \times k$ matrices, which even for small k could be a large number of permanents. And third, approximating each probability requires $O(n^{2k})$ permanents of $(n-k) \times (n-k)$ matrices with non-negative elements. Although approximating permanents of matrices with non-negative elements can be achieved in polynomial time, classical algorithms still have a runtime ranging from $O((n-k)^4 \log(n-k))$ to $O((n-k)^7 \log^4(n-k))$, depending on the sparsity of the matrix [238]. These issues are the main points to address in order to achieve a practical classical algorithm for Boson Sampling. The Clifford & Clifford algorithm could help to alleviate these issues, but there is a challenge due to the fact that the approximation used in Renema et al. does not correspond to a bosonic state. This in turn leads to negative probabilities, which are not clear how to correct for in the Clifford & Clifford algorithm.

4.7 The Schur-Weyl Duality

We'll now turn to the Schur-Weyl Duality, a theorem related to the irreducible representations of the symmetric and unitary groups. Although this seems like a sudden jump in subject, we emphasise to the reader that the connection between linear optics and Boson Sampling as described earlier in this chapter and the Schur-Weyl Duality will be explained in Chapters 5 and 6.

We will start by introducing representation theory, before explaining the Schur-Weyl Duality in detail. At the end of this section we shall consider its connection to quantum information, and in Chapter 5 how it is linked to Boson Sampling specifically. For further reading we direct the reader to [239, 89].

4.7.1 Representation theory

Let G be a group and X a set. We define an action of G on X as a map $\Phi\colon G \times X \to X$ such that the action of the identity element $e \in G$ is the identity map, $\Phi_e(x) = \Phi(e, x) = x$, and the following relation holds:

$$\Phi_{g_1}(\Phi_{g_2}(x)) = \Phi_{g_1 g_2}(x). \tag{4.81}$$

One example is the action of the permutation group S_n on the set of numbers from 1 to n, defined as $\Phi_\sigma(i) = \sigma^{-1}(i)$. Note that Φ need not be injective with respect to G, so another trivial example is the identity action, which is $\Phi_g(x) = x \forall g \in G$.

A representation is defined as a vector space V combined with an action $\Phi : G \times V \to V$ with the additional constraint that Φ is a linear map over V:

$$\Phi(g, u + v) = \Phi(g, u) + \Phi(g, v). \tag{4.82}$$

Now let $W \subseteq V$ be a vector subspace. We say that W is invariant with respect to the action of $\{\Phi_g : g \in G\}$ if

$$\Phi_g(w) \in W \forall g \in G, w \in W. \tag{4.83}$$

Two trivial example of invariant subspaces are the full vector space $W = V$ and the empty vector space $W = \{0\}$. We call a vector subspace W invariant if the above holds and W is not equal to V or $\{0\}$.

This finally brings us to the topic of irreducible representations. A representation V, Φ of G is irreducible, also known as an irrep, if it contains no proper invariant subspaces. If V does contain a proper invariant subspace, then we say that V is reducible.

4.7.2 Symmetric and unitary groups

The main interest of this thesis is the representation theory of the unitary group $\mathrm{U}(m)$, and its action on the space of $(\mathbb{C}^m)^{\otimes n}$. The intuition for linking this to Boson Sampling is that each vector in $\mathbb{C}^m$ is referring to a photon which could be in one of m modes, and the action of $U \in \mathrm{U}(m)$ will be our interferometer. As we shall see from the Schur-Weyl Duality, the irreducible representations for $\mathrm{U}(m)$ are intimately related to those of the symmetric group S_n.

For the tensor space of $(\mathbb{C}^m)^{\otimes n}$, the actions of the symmetric and unitary groups on a tensor $v_1 \otimes \cdots \otimes v_n \in (\mathbb{C}^m)^{\otimes n}$ are explicitly as follows. For a permutation $\sigma \in \mathrm{S}_n$, the action permutes the tensor factors,

$$\Phi_\sigma(v_1 \otimes \cdots \otimes v_n) = v_{\sigma^{-1}(1)} \otimes \cdots \otimes v_{\sigma^{-1}(n)}, \tag{4.84}$$

and for a unitary matrix $U \in \mathrm{U}(m)$, the action is the n-fold tensor product $\Phi_U = U^{\otimes n}$.

It is a simple exercise to see that these two actions commute:

$$\begin{aligned} \Phi_U(\Phi_\sigma(v_1 \otimes \cdots \otimes v_n)) &= \Phi_U(v_{\sigma^{-1}(1)} \otimes \cdots \otimes v_{\sigma^{-1}(n)}) && (4.85)\\ &= (Uv_{\sigma^{-1}(1)}) \otimes \cdots \otimes (Uv_{\sigma^{-1}(n)}) && (4.86)\\ &= \Phi_\sigma((Uv_1) \otimes \cdots \otimes (Uv_n)) && (4.87)\\ &= \Phi_\sigma(\Phi_U(v_1 \otimes \cdots \otimes v_n)). && (4.88) \end{aligned}$$

The Schur-Weyl Duality describes how $(\mathbb{C}^m)^{\otimes n}$ can be decomposed into irreps for the direct product group $\mathrm{U}(m) \times \mathrm{S}_n$.

Theorem 4.1 (Schur-Weyl Duality [240]). *The complex vector space of* $(\mathbb{C}^m)^{\otimes n}$ *decomposes into irreducible subspaces*

$$(\mathbb{C}^m)^{\otimes n} \simeq \bigoplus_{\lambda \vdash n} \mathbb{C}^{\{\lambda\}} \otimes \mathbb{C}^{(\lambda)}, \tag{4.89}$$

where $\mathbb{C}^{\{\lambda\}}$ *carries irrep* $\{\lambda\}$ *of* $U(m)$ *and* $\mathbb{C}^{(\lambda)}$ *carries irrep* (λ) *of* S_n*, and* $\simeq$ *indicates a change of basis is involved. The dimension of irrep* (λ) *can be viewed as the multiplicity of irrep* $\{\lambda\}$*, and vice versa.*

It is common to label irreps of both of these groups by ordered partitions $\lambda = (\lambda_1, \lambda_2, \cdots, \lambda_m)$ of n such that $\lambda_i \geq \lambda_{i+1}$ and $\sum_{i=1}^{m} \lambda_i = n$. We usually suppress zeros in this notation, so for example the totally symmetric irrep $\lambda = (n, 0, \ldots, 0)$ is written (n). The number of nonzero λ_i is called the length of the partition, $\ell(\lambda)$, and only partitions with $\ell(\lambda) \leq m$ occur, which we will assume in all of our expressions that follow. Similarly, the width of the partition is $w(\lambda) = \lambda_1$, the largest partition in λ.

4.7.3 The Quantum Schur Transform

The Quantum Schur Transform realises the basis change of The Schur-Weyl Duality as a quantum circuit. This circuit takes as input a state in the basis $(\mathbb{C}^m)^{\otimes n}$, and decomposes it into a basis of states labelled by which irrep λ they are in.

The canonical example of the Quantum Schur Transform is when $m = n = 2$. In this case, the only partitions available are $\lambda = (2, 0)$ and $\lambda = (1, 1)$. Our representation labelled by $\lambda = (2, 0)$ is defined by the basis states $|00\rangle$, $|11\rangle$ and $(|01\rangle + |10\rangle)/\sqrt{2}$, while our $(1, 1)$ irrep is simply the state $(|01\rangle - |10\rangle)/\sqrt{2}$. This is equivalent to the triplet and singlet states of two electrons and their corresponding description in angular momentum notation:

$$|j = 1, m = 1\rangle = |00\rangle\,, \tag{4.90}$$

$$|j = 1, m = 0\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}}, \tag{4.91}$$

$$|j = 1, m = -1\rangle = |11\rangle\,, \tag{4.92}$$

$$|j = 0, m = 0\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}}. \tag{4.93}$$

This can generalise further to describe n qubits, noting that $j = (\lambda_1 - \lambda_2)/2$. The Quantum Schur Transform is even more general, offering a mapping for any collection of m-level qudits. Given a state $|\Psi\rangle \in (\mathbb{C}^m)^{\otimes n}$ written in the computational basis, the Quantum Schur Transform, which we denote as the quantum circuit W, performs the transformation

$$W|\Psi\rangle = \sum_{\lambda \vdash n} \sum_{q_\lambda} \sum_{p_\lambda} C^{\lambda}_{q_\lambda, p_\lambda} |\lambda\rangle |q_\lambda\rangle |p_\lambda\rangle, \tag{4.94}$$

where λ indexes the irrep, q_λ and p_λ index bases of irreps $\{\lambda\}$ and (λ) respectively, and $C^{\lambda}_{q_\lambda, p_\lambda}$ is a generalised Clebsch-Gordan coefficient. For interferometry applications, the unitary action of $\mathrm{U}(m)$ in this basis is

$$U : |\lambda\rangle |q_\lambda\rangle |p\rangle \to |\lambda\rangle |U \cdot q_\lambda\rangle |p\rangle := |\lambda\rangle \left(\sum_{q'_\lambda} U^{\lambda}_{q_\lambda, q'_\lambda} |q'_\lambda\rangle \right) |p\rangle, \tag{4.95}$$

where U^{λ} is the irreducible unitary matrix corresponding to $U \in \mathrm{U}(m)$. Equivalently, one can write q_λ as a photon occupation $\underline{n} \in F(\mathbb{C}^m)$ and inner multiplicity $r_{\underline{n}} \in \mathbb{N}$.

Although this notation will largely not be used in this thesis, we will briefly describe how the basis states of $\{\lambda\}$ and (λ) are denoted. An equivalent way of writing irrep λ, rather than using partitions, is to use Young diagrams. These are diagrams of n boxes across m rows such that the number of boxes in row i is at most the number of boxes in row $i - 1$. A simple mapping from partitions λ to Young diagrams is then that row i of the diagram has λ_i boxes. Using $m = n = 3$ as an example, the possible irreps are as follows:

$$(3) \equiv \ydiagram{3}, \tag{4.96}$$

$$(2,1) \equiv \ydiagram{2,1}, \tag{4.97}$$

$$(1,1,1) \equiv \ydiagram{1,1,1}. \tag{4.98}$$

The advantage of using Young diagrams is that they also give convenient notation for the basis states of $\{\lambda\}$ and (λ). For $\{\lambda\}$, our unitary irreps, we use the Gelfand-Tsetlin basis. In this basis, we denote basis states by semistandard Young tableaux, which are Young diagrams with each box filled with an integer from $1, \ldots, m$ such that each row is non-decreasing and each column is strictly ascending. The Gelfand-Tsetlin bases for $m = n = 3$ for instance are the following:

$$\begin{aligned} \mathbb{C}^{\{(3)\}} =& \text{span}\left\{ \left|\ytableaushort{111}\right\rangle, \left|\ytableaushort{112}\right\rangle, \left|\ytableaushort{113}\right\rangle, \left|\ytableaushort{122}\right\rangle, \left|\ytableaushort{123}\right\rangle, \right. & (4.99)\\ & \left. \left|\ytableaushort{133}\right\rangle, \left|\ytableaushort{222}\right\rangle, \left|\ytableaushort{223}\right\rangle, \left|\ytableaushort{233}\right\rangle, \left|\ytableaushort{333}\right\rangle \right\}, & (4.100) \end{aligned}$$

$$\mathbb{C}^{\{(2,1)\}} = \text{span}\left\{ \left|\ytableaushort{11,2}\right\rangle, \left|\ytableaushort{12,2}\right\rangle, \left|\ytableaushort{13,2}\right\rangle, \left|\ytableaushort{11,3}\right\rangle, \left|\ytableaushort{12,3}\right\rangle, \left|\ytableaushort{13,3}\right\rangle, \left|\ytableaushort{22,3}\right\rangle, \left|\ytableaushort{23,3}\right\rangle \right\}, \tag{4.101}$$

$$\mathbb{C}^{\{(1,1,1)\}} = \text{span}\left\{ \left|\ytableaushort{1,2,3}\right\rangle \right\}. \tag{4.102}$$

To index the basis states of (λ), we can use the Young-Yagamouchi basis. This basis denotes states by standard Young tableaux, where each box is filled with numbers from 1 to n such that each integer occurs exactly once and all rows and columns are strictly ascending. The Young-Yagamouchi bases for $m = n = 3$ for instance are the following:

$$((3)) = \left\{ \left|\ytableaushort{123}\right\rangle \right\}, \tag{4.103}$$

$$((2,1)) = \left\{ \left|\ytableaushort{12,3}\right\rangle, \left|\ytableaushort{13,2}\right\rangle \right\}, \tag{4.104}$$

$$((1,1,1)) = \left\{ \left|\ytableaushort{1,2,3}\right\rangle \right\}. \tag{4.105}$$

Note that for the fully symmetric (3) and antisymmetric (1, 1, 1) irreps there is only one state, but for $\lambda = (2, 1)$ there are two basis states. This creates a multiplicity, leading to two equivalent bases for the (2, 1) irrep of the unitary group, with permutations mapping from one irrep to the other. The total dimensionality across these basis representations is

$$1 \times 10 + 2 \times 8 + 1 \times 1 = 27, \tag{4.106}$$

which is the same size as our computational basis $(\mathbb{C}^3)^{\otimes 3}$.

Of course, we have not stated explicitly how these basis states map to the standard basis. We shall omit these details for simplicity, but simply note that algorithms for explicitly computing these bases do exist, by starting with a highest-weighted vector in the basis and applying lowering operators to find lower-weight vectors. Explicit algorithms can be found in, for example, [241, 242].

Efficient quantum circuits

The first demonstration of an efficient quantum circuit was for $m = 2$, proven in 2004 and later published in 2006 by Bacon, Chuang and Harrow [88]. This circuit worked by decomposing the Schur Transform into a sequence of n Clebsch-Gordan Transforms, each coupling pairs of qubits together. The individual Clebsch-Gordan transformations could then be implemented using two Controlled-Not gates and a single doubly-controlled rotation gate in the y-direction, which can be implemented up to precision ϵ in $O(\text{polylog}(1/\epsilon))$ gates [243, 244]. Thus, the overall runtime of this circuit was $O(n\,\text{polylog}(1/\epsilon))$.

This result was subsequently generalised to arbitrary m in Aram Harrow's PhD thesis [89, 90]. Now the Schur transform was decomposed into Clebsch-Gordan Transforms applied to pairs of qudits. The m-dimensional Clebsch-Gordan Transformation could be implemented efficiently by recursively applying the $(m-1)$-dimensional Clebsch-Gordan Transform followed by a reduced Wigner Transform. This Wigner Transform can, like the Clebsch-Gordan Transform for $m = 2$, be implemented as two Controlled-Not Operations followed by a doubly-controlled rotation, the angles for which can be computed efficiently and for which an efficient circuit can be decomposed [244]. As a result, this circuit can be implemented in polynomial time in m, n and $\log(1/\epsilon)$. Harrow also briefly mentions in a footnote that if $m \gg n$, then the runtime can be reduced to polynomial in $\log m$ by mapping to a state in $(\mathbb{C}^n)^{\otimes n}$, though this is not formally proven.

Recently there has been some interest in making this quantum circuit more efficient. Kirby and Strauch [245] went into the original Bacon, Chuang and Harrow circuit for $m = 2$ in more detail in order to offer a more explicit analysis of what the quantum computer needed to do, from which they provided a number of simplifications. The result was a quantum circuit which requires $O(n^4 \log(n/\epsilon))$ operators using a Clifford+T gate set. They then extended this algorithm to the qudit setting, where they were able to find a circuit which uses $O(m^{1+p} n^{3m} \log^p(mn/\epsilon))$ operations from any universal set of quantum gates, where $p \approx 3.97$. Note that the number of gates grows superexponentially with m, due to the n^m term, so while this is efficient for a constant m, we shall see that this circuit does not benefit our work. Another recent result by Krovi [246] gives an explicit quantum circuit which is polynomial in $n, \log m$ and $\log(1/\epsilon)$. Krovi's circuit works by first block-diagonalising the state into permutation modules, and then block-diagonalising each permutation module into irreps. Krovi argues that this circuit can be thought of as a dual of Harrow's, as it uses the representation theory of the symmetric group, whereas Harrow's circuit uses that of the unitary group.

Applications

With efficient circuits for the Quantum Schur Transform now described, we will briefly discuss some example applications. Further applications can be found in Aram Harrow's PhD Thesis [89].

Many applications exist which simply take advantage of measuring the irreps of a quantum state. One example is estimating the spectrum of a quantum state ρ from n copies of the state $\rho^{\otimes n}$. Vidal et al. [247] used irreps of the symmetric group to find optimal measurements, that is, measurements which allow us to learn the most information about ρ, when we only have a fixed number of copies of ρ. Later, Keyl and Werner [248] gave an approximator from measuring the irreps of a state, and showed that the probability of an error decreased exponentially with n.

Another such application is quantum state compression, the task of encoding a large quantum state in a smaller one. Hayashi and Matsumoto [249, 250] gave a scheme for compressing an arbitrary quantum state by weakly measuring the irrep λ and then applying an encoding process. This scheme is variable-length, meaning that the output was not guaranteed to be bounded but the scheme would ensure that no information was lost in decoding. Hayashi and Matsumoto also showed that the probability of this scheme returning an encoded state whose size was larger than a particular bound (also known as an overflow) is optimal in terms of its exponent.

One final use, proven by Bartlett, Rudolph and Spekkens[251], is quantum communication. One issue with communicating quantum states is that the receiver might not have the same reference frame as the sender, and so the two parties might not be working in the same basis. Bartlett, Rudolph and Spekkens gave a work-around for this, by encoding information in basis states of the symmetric group's irrep. A difference in reference frames will mean some unitary acting as $U^{\otimes n}$ on the communicated state, but this will only affect the irrep of the unitary group and not that of the symmetric group where the data is actually encoded.

4.7.4 Computational complexity

We shall conclude this chapter with a discussion of known results regarding the computational complexity of the Quantum Schur Transform. In particular, we shall discuss Quantum Schur Sampling [252], a generalisation of another model called Permutational Quantum Computing (PQC) [253], both of which have been proven to be classically simulable in many cases [254]. We will also finish with some discussion about immanants, which are a generalisation of the permanent and determinant.

We shall start by describing Permutational Quantum Computing, which was introduced by Jordan in 2010 [253]. This model starts with an n-qubit computational basis state $|x\rangle$. The Quantum Schur Transform maps this state into the angular momentum basis, with spins coupled sequentially. This means that the spin of qubit $|x_1\rangle$ is coupled with the spin of qubit $|x_2\rangle$, the result is then coupled with the spin of $|x_3\rangle$, the result of which is then coupled with the spin of $|x_4\rangle$, and so on up to $|x_n\rangle$. After this transform is complete, a permutation is applied to reorder the spins, followed by the inverse Schur transform resulting in a new computational basis state $|x'\rangle$, which is then measured in the computational basis to produce a sample. Although computational hardness for this model was not formally proven, unlike Boson Sampling and the other models considered in Section 4.2.1, Jordan gave some reasoning for why it seems reasonable to believe that a polynomial time classical algorithm would not exist. Jordan did this by providing PQC algorithms for approximating elements of irreps of the symmetric group, as well as showing that PQC can approximately simulate certain models of Topoligical Quantum Field Theory [253]. At the time of Jordan's publication, no polynomial time algorithm existed for either problem.

Quantum Schur Sampling is a generalisation of Permutational Quantum Computing, first proposed by Havlíček and Strelchuk, and name due to its similar structure to Quantum Fourier Sampling proposed by Fefferman and Umans [255]. In this model, rather than restricting to permutations, the circuit is free to perform any reversible classical computation on the spin qubits. The result is similar in structure to that of Shor's Algorithm, consisting of some kind of quantum transform — Fourier for Shor's algorithm, Schur for Quantum Schur Sampling — a polynomial-time classical circuit, and then the inverse transform and measurement. This model does not offer any (known) benefit over Permutational Quantum Computing; in fact it was first proposed not with the intention of showing further benefits, but instead to show a more general class of quantum circuit that could still be classically simulated.

It has only been in recent years that Quantum Schur Sampling has been shown to be classically simulable. This was proven as a result of two papers. The first, by Havlíček and Strelchuk [252], disproved a conjecture of Jordan upon which the computational hardness of PQC was reliant: that the transition amplitudes of a PQC circuit cannot be approximated to within additive error in polynomial time. This was disproven by showing that states output from applying the Quantum Schur Transform to a computational basis state were Computationally Tractable (CT). This is a term coined by Van den Nest [256] meaning that the following tasks can be performed classically in polynomial time up to polynomial multiplicative error: first, that the amplitudes of a state can be approximated; and second, that a sample can be output from a measurement in the computational basis. With this proven for Quantum Schur Transform states, Havlíček and Strelchuk used a result by Van den Nest that unitary transformations which simply map between states in the same CT basis, such as permutations or reversible classical circuits in this case, will also produce a CT state. Finally, they used one more result of Van den Nest which is that the transition amplitude of two CT states can be approximated in polynomial time on a classical computer, thus rendering Jordan's conjecture false.

However, while the main claim to hardness had been proven wrong, this still left open the possibility that Quantum Schur Sampling was classically intractable. This is because while a polynomial time algorithm existed for computing the probability of an outcome, which is simply the square of the its amplitude, there might be an exponentially large number of probabilities to approximate before a sample can be obtained. This was resolved by Havlíček, Strelchuk and Temme, who gave a polynomial time classical algorithm for sampling when the output distribution is approximately t-sparse, meaning that it can be approximated by a distribution with t elements with non-zero probability up to accuracy ϵ. This is based on an algorithm by Schwarz and Van den Nest [257], and follows three steps: First, they use the Computational Tractability result of [252] to show that approximate the marginal probabilities; second, they use an algorithm by Kushilevitz and Mansour to show that a subset of outcomes with high probabilities in the output distribution can be found efficiently; and third, they sample that sampling from this distribution of high-probability outcomes. Havlíček, Strelchuk and Temme showed that this high-probability distribution is at most 6ϵ away from the Quantum Schur Sampling distribution in total variational distance and can be done in time polynomial in n, t and $1/\epsilon$. But how sparse is the output of a Quantum Schur Sampling circuit? Havlíček, Strelchuk and Temme ran some numerical simulations exploring the output of PQC circuits with up to $n = 10$ qubits, and showed that in these small cases all except 0.1% of tested cases on 10 qubits could be approximated by a $2n^2$-sparse distribution up to accuracy $1/n$. Although this was not a formal proof of how many distributions are approximately sparse, it gives some intuition as to how common these sparse distributions might be.

It is also worth discussing the actions of the symmetric and unitary groups on these different representations, and how they affect the classical complexity of computing the amplitudes. For this, we shall discuss a generalisation of the matrix permanent, and known results on its complexity.

Let λ be an irreducible representation, and σ be a permutation. Define

$$\chi_\lambda(\sigma) \colon = \mathrm{Tr}\left[\Phi_\sigma\right] \tag{4.107}$$

as the characteristic of σ in representation λ. Note that for any $\sigma' = \tau\sigma\tau^{-1}$ for some $\tau \in \mathrm{S}_n$, $\chi_\lambda(\sigma) = \chi_\lambda(\sigma')$, meaning that σ and σ' are in the same conjugacy class:

$$\begin{aligned}
\chi_\lambda(\sigma') &= \mathrm{Tr}\left[\Phi_{\sigma'}\right] && (4.108)\\
&= \mathrm{Tr}\left[\Phi_{\tau\sigma\tau^{-1}}\right] && (4.109)\\
&= \mathrm{Tr}\left[\Phi_\tau \Phi_\sigma \Phi_{\tau^{-1}}\right] && (4.110)\\
&= \mathrm{Tr}\left[\Phi_\sigma \Phi_{\tau^{-1}} \Phi_\tau\right] && (4.111)\\
&= \mathrm{Tr}\left[\Phi_\sigma \Phi_{\tau^{-1}\tau}\right] && (4.112)\\
&= \mathrm{Tr}\left[\Phi_\sigma\right] && (4.113)\\
&= \chi_\lambda(\sigma). && (4.114)
\end{aligned}$$

The map χ_λ for all classes of S_n is defined as the character of λ. Note that computing χ_λ is #P-Hard [258].

We define the immanant of an $n \times n$ matrix M as the polynomial

$$\mathrm{imm}_\lambda(M) \colon = \sum_{\sigma \in \mathrm{S}_n} \chi_\lambda(\sigma) \prod_{i=1}^{n} M_{i,\sigma(i)}. \tag{4.115}$$

Note that when $\lambda = (n)$, we have $\chi_\lambda(\sigma) = 1 \forall \sigma \in \mathrm{S}_n$, which gives the permanent as defined in Equation (4.36). Similarly for $\lambda = (1^n)$, $\chi_\lambda(\sigma) = \mathrm{sgn}(\sigma) \forall \sigma \in \mathrm{S}_n$, giving the determinant:

$$\det(M) \colon = \sum_{\sigma \in \mathrm{S}_n} \mathrm{sgn}(\sigma) \prod_{i=1}^{n} M_{i,\sigma(i)}. \tag{4.116}$$

So what is known about the computational complexity of these immanants then? The best-known results are for the permanent and determinant, which are #P-Hard and computable in polynomial time, respectively [183, 71, 259]. But other hardness results are also known [260, 261, 262, 263]. For example, Bürgisser [264] showed that for $w = \mathrm{poly}(n)$, computation of imm_λ is #P-Hard when $\lambda = (w, 1^{n-w})$, known as a hook diagram, or when $\lambda = (w^{n/w})$ if $w|n$, known as a rectangular diagram with n/w rows.

Chapter 5

Quantum simulation of partially distinguishable Boson Sampling

5.1 Introduction

This work was completed in collaboration with Peter S. Turner, and published as "Quantum simulation of partially distinguishable boson sampling", *Physical Review A* **97**, 062329 (2018), copyright American Physics Society. A preprint of this article is freely available at `arXiv:1803.03657`. Details of contributions made by myself are given in Section 1.3.4.

In this chapter we explore the problem of sampling from a collection of n partially distinguishable single bosons interacting on a m-mode interferometer from the opposite direction, that is, from the perspective of quantum simulation. We do so by observing that ideal Boson Sampling is equivalent to sampling from the totally symmetric representation of the unitary group, and that partial distinguishability generalises this to the problem of sampling from arbitrary irreducible representations. We show that quantum circuits for the Schur transform [90] can therefore be used to give a polynomial time quantum algorithm for sampling from the same probability distribution as bosons emerging from a linear interferometer, regardless of distinguishability.

Although it is generally accepted that quantum computers can perform Boson Sampling efficiently, there are few places where such algorithms are actually described explicitly. An example of such a method for the ideal (indistinguishable) case is by Aaronson and Arkhipov [71], using a technique by Reck et al. [176] to decompose the unitary matrix U into a sequence of $O(m^2)$ optical elements, each implemented via the Solovay-Kitaev theorem [244]. Here we show explicitly how nonideal linear optics can be viewed as a quantum computation, allowing a wider range of realistic experimental situations to be considered. Our approach shows that while the ideal case is intimately related to the symmetric representation of the unitary group through matrix permanents, in the nonideal case all representations play a role.

The chapter is structured as follows. We provide a simple quantum circuit for ideal indistinguishable photon sampling in Section 5.2, before introducing the full circuit for sampling with distinguishable particles in Section 5.3. In Section 5.4, we provide some further explanation as to why these circuits work, via what is known as Unitary-Unitary Duality. Following this result, we discuss a few interesting consequences: in Section 5.4.1, we show how postselection can be used with this circuit to sample from the indistinguishable distribution when given a distinguishable input; in Section 5.5, we consider how the circuit can be used to simulate Boson Sampling when photons are lost; and in Section 5.6, we consider the multipartite entanglement of the output in the partially distinguishable case.

5.2 A quantum circuit for ideal bosonic sampling

Here we describe a quantum circuit for bosonic sampling when the bosons are perfectly indistinguishable (and free from other errors such as loss, which we'll discuss later on). This circuit samples with accuracy $\delta + \epsilon$ and runs in polynomial time and space in terms of m, n, $\log(\delta^{-1})$ & $\log(\epsilon^{-1})$. Here and throughout the paper, δ describes the precision with which we are able to approximate the Schur transform via the Bacon-Chuang-Harrow circuit, and ϵ the accuracy to which we can approximate the unitary matrix U via the Solovay-Kitaev theorem [244, 243]. Note that although the Solovay-Kitaev construction can involve exponential resources in terms of m, this can be avoided by first performing a Hurwitz (or Reck) decomposition into smaller unitaries [175, 176, 265].

The goal of the circuit is to sample from the totally symmetric subspace of $(\mathbb{C}^m)^{\otimes n}$, where the interferometer $U \in \mathrm{U}(m)$ acts on the totally symmetric irrep of the unitary group given by $\{\lambda = (n)\}$. In order to construct symmetrised states given the input occupation S, we use the (inverse) Schur transform. The Schur circuit W specifies irreps of $\mathrm{U}(m)$ in the Gelfand-Zeitlin (GZ) basis, so we need a way to map between these states and occupations. We can do this via the pattern weight $\nu = (\nu_1, \cdots, \nu_d)$, which can be related to a GZ pattern for any irrep [241]. For the fully symmetric irrep, the pattern weight is unique for each GZ state and there is a particularly simple mapping from occupations to symmetric GZ states in this case, namely $\nu_i = S_i$ [266]; this has also been referred to as a quantum analog of a classical "type" [89]. Thus, we have an efficient way to identify an input occupation S with a GZ basis state $q_{(n)}$.

We can now see how a circuit for indistinguishable boson sampling would work. Given an input occupation S, we prepare the corresponding state $|q_{(n)}\rangle$ of the q-register. To use the inverse Schur transform, we append to this input state a quantum register for the irrep $|(n)\rangle$, and another for the symmetric group index $|p_{(n)}\rangle$. Note that there is only one possible state for the $p_{(n)}$ register, because the fully symmetric irrep of the symmetric group is one dimensional; thus $p_{(n)} = 1$ always. The inverse Schur transformation $W^\dagger$ takes this state to a symmetric state of n qudits in $(\mathbb{C}^m)^{\otimes n}$. In this tensor space, we now need only apply the interferometer matrix U to each qudit in parallel as the circuit $U^{\otimes n}$. This can be done with accuracy ϵ in $O(\log^c(1/\epsilon))$ time via the Solovay-Kitaev theorem [244, 243]. Finally, we apply the Schur transform again and measure the q-register to get a sample $q'_{(n)}$, from which we can easily compute the pattern weight to get an output occupation S'. A complete version of the quantum circuit for Boson Sampling is given in Algorithm 5.1, as well as a circuit description in Figure 5.1.

We can demonstrate correctness by showing that this distribution does indeed match the one we have for sampling from indistinguishable bosons. We start with the input occupation S. After mapping this to a unitary irrep state $|q_{(n)}\rangle$ and applying $W^\dagger$, we end up with the symmetrized state

$$W^\dagger \, |\lambda = (n)\rangle \, |q_{(n)}\rangle \, |p_{(n)} = 1\rangle = \frac{1}{\sqrt{n! \prod_{i=1}^m S_i!}} \sum_{\sigma \in \mathrm{S}_n} \sigma |s\rangle, \tag{5.1}$$

where $|s\rangle$ can be chosen to be any computational basis state with occupation S, that is, with S_i of the qudits in state i. Arguing similarly for the output S', we see that the probability of the

```
input  : A matrix $U \in \mathrm{U}(m)$ and
         an $n$-boson $m$-mode occupation $S$.
output: An $n$-boson $m$-mode occupation $S'$.
Map $S$ to $q$-register basis index $q_{(n)}$
Prepare input state $|\lambda = (n)\rangle|q_{(n)}\rangle|p_{(n)} = 1\rangle$
Apply $W^\dagger$, producing a state $|\Psi\rangle \in (\mathbb{C}^m)^{\otimes n}$
Synthesize $U$ via Solovay-Kitaev
Execute $U$ on each qudit in parallel, implementing $U^{\otimes n}$
Apply $W$, producing state $|(n)\rangle|U \cdot q_{(n)}\rangle|1\rangle$
Measure the $q$-register to obtain a sample $q'_{(n)}$
Map $q'_{(n)}$ to an occupation $S'$
return $S'$
```

Algorithm 5.1: A quantum circuit for sampling from the same distribution as that produced by indistinguishable bosons in a linear interferometer.

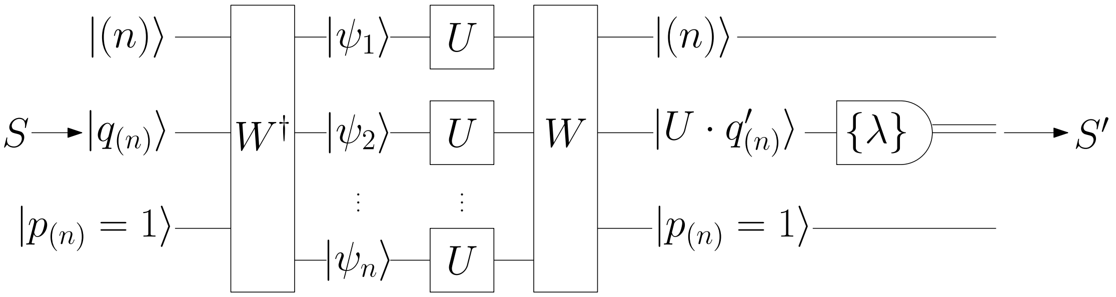


Figure 5.1: A quantum circuit for Algorithm 5.1. Note that the measurement of the q-register returns a string that we associate to a GZ basis state.

algorithm outputting $q'_{(n)}$ given inputs U and S is

$$
\begin{aligned}
\Pr[q'_{(n)}] &= |\langle (n), q'_{(n)}, 1|WU^{\otimes n}W^{\dagger}|(n), q_{(n)}, 1\rangle|^2 &(5.2)\\
&= \left| \frac{1}{n!\sqrt{\prod_{i=1}^{m} S'_i! S_i!}} \sum_{\sigma,\sigma'\in \mathrm{S}_n} \langle s'|\sigma'^{\dagger} U^{\otimes n} \sigma |s\rangle \right|^2 &(5.3)\\
&= \frac{1}{\prod_{i=1}^{m} S'_i! S_i!} \left| \frac{1}{n!} \sum_{\sigma,\sigma'\in \mathrm{S}_n} \bigotimes_{k=1}^{n} \langle s'_{\sigma'^{-1}(k)}|U|s_{\sigma^{-1}(k)}\rangle \right|^2 &(5.4)\\
&= \frac{1}{\prod_{i=1}^{m} S'_i! S_i!} \left| \frac{1}{n!} \sum_{\sigma,\sigma'\in \mathrm{S}_n} \prod_{k=1}^{n} U_{s'_k, s_{\sigma^{-1}(\sigma'(k))}} \right|^2 &(5.5)\\
&= \frac{1}{\prod_{i=1}^{m} S'_i! S_i!} \left| \sum_{\tau\in \mathrm{S}_n} \prod_{k=1}^{n} U_{s'_k, s_{\tau(k)}} \right|^2 &(5.6)\\
&= \frac{|\operatorname{per}(U_{S',S})|^2}{\prod_{i=1}^{m} S'_i! S_i!}. &(5.7)
\end{aligned}
$$

Thus the output probability distribution matches the one given in Equation (4.35). We also see that the Schur-Weyl Duality implies

$$
U^{(n)}_{q_{(n)},q'_{(n)}} = \frac{\operatorname{per}(U_{S',S})}{\sqrt{\prod_{i=1}^{m} S'_i! S_i!}}. \tag{5.8}
$$

That is, the totally symmetric representation of the unitary group can be constructed from permanents of $U_{S',S}$ matrices [267].

As for the complexity of this circuit, the mapping from bosons to $q_{(n)}$ states and back can be done in polynomial time and space in terms of n [266], the Quantum Schur Transform takes polynomial time and space in terms of m, n & $\log(1/\epsilon)$ and the Solovay-Kitaev theorem allows U to be implemented in polynomial time and space. From this and the earlier points discussed in this section, we find that Theorem 5.1 holds.

Theorem 5.1. *Algorithm 5.1 performs ideal bosonic sampling with approximation $\delta + \epsilon$ in polynomial time and space in terms of m, n, $\log(\delta^{-1})$ and $\log(\epsilon^{-1})$.*

We observe that this circuit could be simplified in several ways. Firstly, the entire Schur transform is not required because in the ideal case the problem is confined to the symmetric irrep only. As we will see, in the non-ideal (distinguishable) case other irreps can occur.

Another simplification that we'll use in the next section is the following observation. If rather than applying step 5 onwards in Algorithm 5.1 we simply measure the registers in the computational basis and return the result rewritten as an occupation, we also end up with a distribution given by the permanents. The probability of measuring a particular computational basis state $|s'\rangle \in (\mathbb{C}^m)^{\otimes n}$ is

$$\begin{aligned}
\Pr[s'] &= \frac{1}{n! \prod_{i=1}^m (S_i!)} \left| \langle s'| U^{\otimes n} \sum_{\sigma \in \mathrm{S}_n} \sigma |s\rangle \right|^2 && (5.9)\\
&= \frac{1}{n! \prod_{i=1}^m (S_i!)} \left| \sum_{\sigma \in \mathrm{S}_N} \bigotimes_{k=1}^n \langle s'_k | U | s_{\sigma^{-1}(k)} \rangle \right|^2 && (5.10)\\
&= \frac{\left| \sum_{\sigma \in \mathrm{S}_n} \prod_{k=1}^n U_{s'_k, s_{\sigma^{-1}(k)}} \right|^2}{n! \prod_{i=1}^m (S_i!)} && (5.11)\\
&= \frac{|\operatorname{per}(U_{S',S})|^2}{n! \prod_{j=1}^m (S_i!)}. && (5.12)
\end{aligned}$$

The probability of measuring an occupation S' is equal to summing over the probabilities of all states $|s'\rangle$ of pattern weight S', of which there are $n!/\prod_{i=1}^m (S'_i!)$. We will consider both versions of this circuit in subsequent sections on sampling from distinguishable bosonic distributions.

5.3 A quantum circuit for arbitrarily distinguishable bosonic sampling

We now turn to the question of sampling from a distribution of partially distinguishable bosons, (again with no loss). Distinguishability is modelled as correlation between the modes of the bosons' 'System' degrees of freedom, and new modes corresponding to 'Label' degrees of freedom. In order to accommodate the possibility of all n bosons being completely distinguishable, the number of Label modes must be n so that each boson can be correlated to a unique Label. Thus there are now a total of mn modes in the problem. Physically we can think of the System degree of freedom as the spatial modes available to the bosons, and the Label as, say, temporal modes — however, the model is completely general.

On the aggregate Hilbert space we have the same setup as the ideal case, but now by tracing out the Label register we see that distinguishability can lead to decoherence of the System qudits. We assume that an interferometer implementing an $m \times m$ unitary matrix acts only on the m System modes, while the Label remains unchanged. In this model, as well as receiving a unitary matrix U as input, we also receive an $m \times n$ occupation T which describes how many bosons are in System mode i and Label mode j. This can be described in terms of creation operators as

$$|T\rangle = \prod_{i=1}^m \prod_{j=1}^n \frac{(a^\dagger_{i,j})^{T_{i,j}}}{\sqrt{T_{i,j}!}} |0\rangle. \tag{5.13}$$

Since the Labels are assumed to be unaffected by the interferometer, the creation operators evolve as $a^\dagger_{i,j} \to \sum_{k=1}^m U_{i,k} a^\dagger_{k,j}$.

Our technique for handling distinguishable bosons is similar to the ideal case where we consider the symmetric irrep $\{(n)\}$ of the unitary group. However, the introduction of the Label degree of freedom means that we no longer map onto the $\{(n)\}$ irrep of $\mathrm{U}(m)$. Instead we must map onto the $\{(n)\}$ irrep of the aggregate unitary group $\mathrm{U}(mn)$, since it is now the total state of m System modes and n Label modes that must be symmetrised.

When we apply $W^\dagger$ to the input, we find the same Young symmetrizer as before, but now output a state $|\Psi\rangle \in (\mathbb{C}^m \otimes \mathbb{C}^n)^{\otimes n}$. We can think of this as the ideal case but now with each qudit being mn dimensional. Furthermore each System-Label qudit can be viewed as bipartite, with an m-dimensional qudit describing the System degree of freedom and another n-dimensional qudit describing the Label. We can therefore split the n System-Label qudits into two registers, with the interferometer action and boson detection taking place on only the System register, while the Label register 'eavesdrops'. A complete description of the circuit is given in Algorithm 5.2, with a circuit diagram given in Figure 5.2.

```
input : A matrix $U \in \mathrm{U}(m)$ and
        an $n$-boson $mn$-mode occupation $T$.
output: An $n$-boson $m$-mode occupation $S'$.
1 Map $T$ to $q$-register basis index $q_{(n)}$ (for $\mathrm{U}(mn)$)
2 Prepare input state $|\lambda = (n)\rangle|q_{(n)}\rangle|p_{(n)} = 1\rangle$
3 Apply $W^\dagger$, producing a state $|\Psi\rangle \in (\mathbb{C}^{m\times n})^{\otimes n}$
4 Rearrange into two (possibly entangled) quantum registers
    $|\Psi_{\mathrm{Sys}}\rangle = |\psi_{\mathrm{Sys},1}\rangle \dots |\psi_{\mathrm{Sys},n}\rangle \in (\mathbb{C}^m)^{\otimes n}$ and $|\Psi_{\mathrm{Lab}}\rangle = |\psi_{\mathrm{Lab},1}\rangle \dots |\psi_{\mathrm{Lab},n}\rangle \in (\mathbb{C}^n)^{\otimes n}$
5 Synthesize $U$ via Solovay-Kitaev
6 Execute $U$ on the $|\Psi_{\mathrm{Sys}}\rangle$ qudits in parallel, implementing $U^{\otimes n} \otimes \mathbb{1}_{\mathrm{Lab}}$
7 Measure the System in the computational basis to obtain a sample $s'$
8 Map $s'$ to an occupation $S'$ ($S'_i = \#$ of qudits in state $1 \leq i \leq m$)
9 return $S'$
```

Algorithm 5.2: A quantum circuit for sampling from (essentially) the same distribution at that produced by distinguishable bosons in a linear interferometer. In order to sample from exactly the same distribution, instead of step 7 one could transform back to the Schur basis by applying W on the System and sample the q-register, or one could perform some post-processing as discussed at the end of the previous section.

To see that this distribution matches that of partially distinguishable bosons, we will compare with the results of Tichy [206] discussed in Section 4.5.1. In order to connect our model with this model of distinguishability, one simply needs to take superpositions of mn-mode input occupations T in such a way as to realise the Label states Φ_i. This is always possible since the space of internal states, $\mathrm{span}\{\Phi_i\}_{i=1}^m$, can always be embedded in the Label space $(\mathbb{C}^n)^{\otimes n}$. For example, consider two bosons in two System modes where one boson is in System mode 1 and has internal state $|1\rangle$ (corresponding to Label mode 1), and the other boson is in System mode 2 and has internal state $|\Phi_2\rangle = \alpha|1\rangle + \beta|2\rangle$. This is represented as the following superposition of System-Label occupations:

$$\alpha\,|1_{1,R}, 0_{1,G}, 1_{2,R}, 0_{2,G}\rangle + \beta\,|1_{1,R}, 0_{1,G}, 0_{2,R}, 1_{2,G}\rangle\,. \tag{5.14}$$

Given a distinguishability matrix $\mathcal{S}_{k,l}$, in this way we can prepare a corresponding superposition of occupations T at step 2 of the algorithm. (We can in fact consider more general partially distinguishable situations of bosons with different Label states in the same System mode.)

After step 3 of the algorithm, the state can be written as

$$|\Psi\rangle_{\mathrm{Sys,Lab}} = \frac{1}{\sqrt{n!\prod_{i=1}^m S_i!}} \sum_{\sigma\in \mathrm{S}_n} \sigma|s\rangle\sigma \bigotimes_{k=1}^n |\Phi_{s_k}\rangle \tag{5.15}$$

$$= \frac{1}{\sqrt{n!\prod_{i=1}^m S_i!}} \sum_{\sigma\in \mathrm{S}_n} \sigma|s\rangle \bigotimes_{k=1}^n |\Phi_{s_{\sigma^{-1}(k)}}\rangle \tag{5.16}$$

$$\tag{5.17}$$

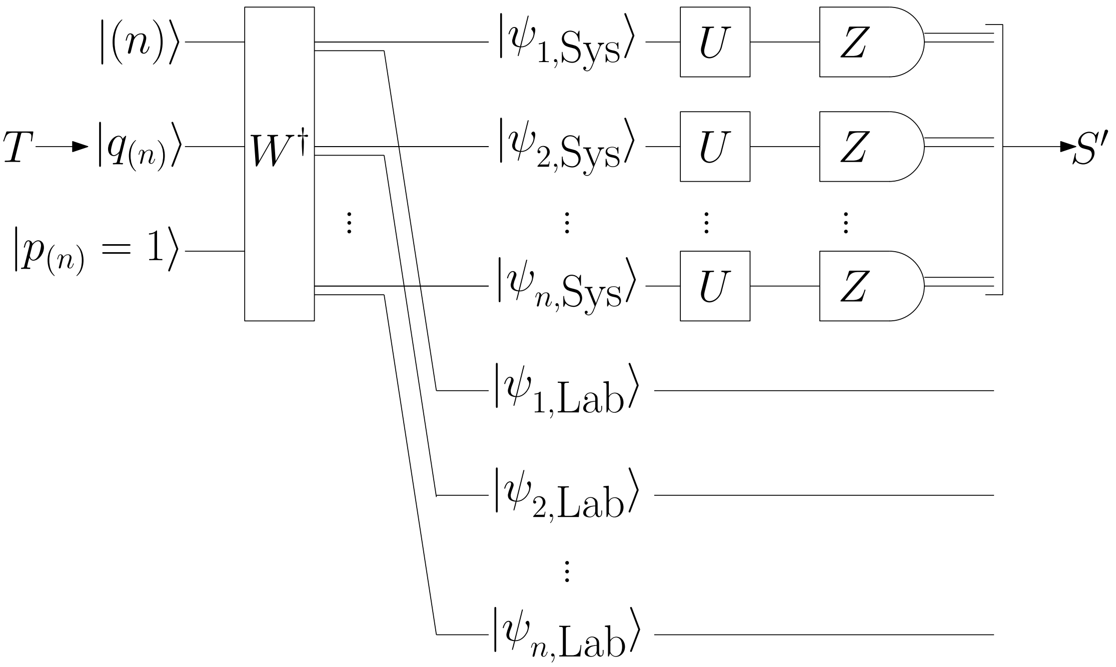

$|(n)\rangle$
$T \rightarrow |q_{(n)}\rangle$
$|p_{(n)} = 1\rangle$
$W^\dagger$
$|\psi_{1,\mathrm{Sys}}\rangle$
$|\psi_{2,\mathrm{Sys}}\rangle$
$|\psi_{n,\mathrm{Sys}}\rangle$
U
Z
S'
$|\psi_{1,\mathrm{Lab}}\rangle$
$|\psi_{2,\mathrm{Lab}}\rangle$
$|\psi_{n,\mathrm{Lab}}\rangle$

Figure 5.2: The quantum circuit described in Algorithm 5.2. For simplicity we forego the second Schur transform as discussed in the previous section and measure in the computational (Z) basis. Note that we only sample from the System qudits, effectively tracing out the Label qudits.

where $|s\rangle = \bigotimes_{k=1}^{n} |s_k\rangle$ and $\bigotimes_{k=1}^{n} |\Phi_{s_k}\rangle$ are quantum registers describing the System and Label. Tracing out the Label register yields the reduced density matrix

$$\begin{aligned} \rho_{\mathrm{Sys}} &= \mathrm{Tr}_{\mathrm{Lab}}[|\Psi\rangle_{\mathrm{Sys,Lab}}\langle\Psi|] \qquad (5.18)\\ &= \frac{1}{n!\prod_{i=1}^{m} S_i!} \sum_{\sigma,\sigma'\in \mathrm{S}_n} \sigma|s\rangle\langle s|\sigma'^{\dagger} \prod_{k=1}^{n} \langle \Phi_{s_{\sigma'^{-1}(k)}} | \Phi_{s_{\sigma^{-1}(k)}}\rangle \qquad (5.19)\\ &= \frac{1}{n!\prod_{i=1}^{m} S_i!} \sum_{\sigma,\sigma'\in \mathrm{S}_n} \sigma|s\rangle\langle s|\sigma'^{\dagger} \prod_{k=1}^{n} \mathcal{S}_{\sigma'^{-1}(k),\sigma^{-1}(k)}. \qquad (5.20)\end{aligned}$$

When we apply the action of the interferometer U on the m-dimensional System qudits, the probability of measuring a state $|s'\rangle$ after step 7 is

$$\begin{aligned} \Pr[|s'\rangle] &= \mathrm{Tr}[|s'\rangle\langle s'|U^{\otimes n}\rho_{\mathrm{Sys}}(U^{\dagger})^{\otimes n}] \qquad (5.21)\\ &= \frac{\sum_{\sigma,\sigma'\in \mathrm{S}_n} \langle s'|U^{\otimes n}\sigma|s\rangle\langle s|\sigma'^{\dagger}(U^{\dagger})^{\otimes n}|s'\rangle \prod_{k=1}^{n} \mathcal{S}_{\sigma'^{-1}(k),\sigma^{-1}(k)}}{n!\prod_{i=1}^{m} S_i!} \qquad (5.22)\\ &= \frac{\sum_{\sigma,\sigma'\in \mathrm{S}_n} \prod_{k=1}^{n} \langle s'_k|U|s_{\sigma^{-1}(k)}\rangle\langle s_{\sigma'^{-1}(k)}|U^{\dagger}|s'_k\rangle \mathcal{S}_{\sigma'^{-1}(k),\sigma^{-1}(k)}}{n!\prod_{i=1}^{m} S_i!} \qquad (5.23)\\ &= \frac{\sum_{\tau,\tau'\in \mathrm{S}_n} \prod_{k=1}^{n} U_{s'_k,s_{\tau(k)}} U^{*}_{s'_k,s_{\tau'(k)}} \mathcal{S}_{\tau'(k),\tau(k)}}{n!\prod_{i=1}^{m} S_i!}. \qquad (5.24)\end{aligned}$$

Up to a factor, this is the desired probability distribution of Equation (4.55). As discussed at the end of the last section, this factor could be handled either by applying a second Schur transform on the System and sampling the q-register, or by classical post processing.

Counting resources goes much the same as it did in the ideal case, though now we have mn-dimensional qudits that are made up of pairs of m- and n-dimensional qudits. Separating these System and Label registers in step 3 can be done with polynomial resources, as can the unitary transformation on the input q-register that prepares the input state of arbitrary distinguishability. From this and the points above, we find that Theorem 5.2 holds.

Theorem 5.2. *Algorithm 5.2 samples from the distinguishable bosonic distribution with approximation $\delta + \epsilon$ when the distinguishability of the input bosons is known. The circuit runs in polynomial time and space in terms of $m, n, \log(\delta^{-1})$ and $\log(\epsilon^{-1})$.*

5.3.1 Complete (in)distinguishability

The two extreme cases of completely indistinguishability and complete distinguishability are of interest. For completely indistinguishable bosons, the Label for each is the same (call it $|1\rangle$), and after the (inverse) Schur transform step we have

$$\frac{1}{\sqrt{n!\prod_{i=1}^{m} S_i!}} \sum_{\sigma\in \mathrm{S}_n} \sigma|s\rangle|1\rangle^{\otimes n}, \qquad (5.25)$$

It is clear that the Label register is separable from the System register, and tracing out the Label yields the same situation as the ideal case in Equation (5.1), as it should.

In the completely distinguishable case, each boson has a different unique Label, correlated to a unique System mode (note $m \geq n$ in this case). This implies that the occupation T has a single 1 in each of n rows and columns, and zeros elsewhere. The (inverse) Schur transformed state has System and Label registers s and l that are completely correlated sequences of length n with no repetitions; if we choose to order the bases $123\cdots n$ then we have the state

$$\frac{1}{\sqrt{n!}}\sum_{\sigma\in\mathrm{S}_n}\sigma|s\rangle\sigma|l\rangle=\frac{1}{\sqrt{n!}}\sum_{\sigma\in\mathrm{S}_n}|\sigma^{-1}(1)\sigma^{-1}(2)\cdots\sigma^{-1}(n)\rangle|\sigma^{-1}(1)\sigma^{-1}(2)\cdots\sigma^{-1}(n)\rangle. \tag{5.26}$$

We see that this is maximally entangled on the "coincident" subspace of states with a single boson in each mode. Tracing out the Label yields the reduced System state

$$\rho_{\mathrm{Sys}}=\frac{1}{n!}\sum_{\sigma,\sigma'\in\mathrm{S}_n}\sigma|s\rangle\langle s|\sigma'^{\dagger}\,\langle l|\sigma'^{\dagger}\sigma|l\rangle \tag{5.27}$$

$$=\frac{1}{n!}\sum_{\sigma\in\mathrm{S}_n}\sigma|s\rangle\langle s|\sigma^{\dagger}, \tag{5.28}$$

which follows because the Label overlap is only nonzero if $\sigma'^{\dagger}\sigma=\mathbb{1}\Rightarrow\sigma'=\sigma$ due to the fact that l has no repetitions. After the action of U, the probability of measuring $|s'\rangle\in(\mathbb{C}^m)^{\otimes n}$ is

$$\Pr[|s'\rangle]=\mathrm{Tr}[|s'\rangle\langle s'|U^{\otimes n}\rho_{\mathrm{Sys}}U^{\dagger\otimes n}] \tag{5.29}$$

$$=\frac{1}{n!}\langle s'|U^{\otimes n}\left(\sum_{\sigma\in\mathrm{S}_n}\sigma|s\rangle\langle s|\sigma^{\dagger}\right)U^{\dagger\otimes n}|s'\rangle \tag{5.30}$$

$$=\frac{1}{n!}\sum_{\sigma\in\mathrm{S}_n}\prod_{k=1}^{n}|U_{s'_k,s_{\sigma^{-1}(k)}}|^2 \tag{5.31}$$

$$=\frac{\mathrm{per}(|U_{S',S}|^2)}{n!}, \tag{5.32}$$

where we've defined $|U_{S',S}|^2$ as the elementwise square of the absolute value. We can find the probability of returning occupation S' by summing up the probabilities of all $n!/\prod_{i=1}^{m}(S'_i!)$ states of pattern weight s', giving

$$\Pr[S']=\frac{\mathrm{per}(|U_{S',S}|^2)}{\prod_{i=1}^{m}S'_i!}, \tag{5.33}$$

which agrees with the (classical) probability distribution for sampling with distinguishable bosons [188].

5.4 Unitary-Unitary Duality

The preceding shows how the Schur transform gives a map between second quantised occupation states and first quantised single particle states via symmetrisation. The complication added by distinguishability is that each single particle becomes bipartite, with a System and Label degree of freedom. As shown above, distinguishability arises as correlations between the System and Label registers of the circuit in Figure 5.2. It turns out that independently transforming the System and Label registers back into the Schur basis can give a Schmidt decomposition for these correlated states (see Figure 5.3). This can be seen as a consequence of the following duality [268, 240].

Theorem 5.3 (Unitary-Unitary Duality). *The totally symmetric irrep of* $\mathrm{U}(md)$ *can be decomposed into irreps of* $\mathrm{U}(m)\times\mathrm{U}(d)$ *as*

$$(\mathbb{C}^m\otimes\mathbb{C}^d)^{(n)}\cong\bigoplus_{\lambda\vdash n}\mathbb{C}^{\{\lambda\}_m}\otimes\mathbb{C}^{\{\lambda\}_d}, \tag{5.34}$$

where $\{\lambda\}_m$ *indicates an irrep* λ *of* $\mathrm{U}(m)$*, similarly for* $\{\lambda\}_d$*, and* λ *runs over all partitions of* n *consistent with both* m *and* d*.*

This can be proven by Schur decomposing the System and Label registers, each of which, by the Schur-Weyl Duality, will have good permutation symmetry quantum numbers. The question is then which linear combinations of tensor products of such states are totally symmetric, and the answer turns out to be only states of the form (suppressing normalisation) [269]

$$|\lambda, q_\lambda, {q'}_\lambda\rangle_{\mathrm{SysLab}} = \sum_{p_\lambda} |\lambda, q_\lambda, p_\lambda\rangle_{\mathrm{Sys}} \, |\lambda, {q'}_\lambda, p_\lambda\rangle_{\mathrm{Lab}} \,. \tag{5.35}$$

Thus a basis for the totally symmetric irrep of the System-Label Hilbert space consists of states of this form, leading to the decomposition in Equation (5.34) and to a Schmidt decomposition of totally symmetric (second quantised) System-Label states.

Because the Schur transformations on each register are local to the System and Label, entanglement across this bipartition is unchanged. For example, the completely distinguishable state that from Equation (5.26) is seen to have Schmidt rank $n!$ in the computational basis, and therefore must be of the form

$$\frac{1}{\sqrt{n!}} \sum_{\lambda \vdash n} \sum_{q_\lambda^{\mathrm{coin}}} \sum_{p_\lambda} |\lambda, q_\lambda, p_\lambda\rangle_{\mathrm{Sys}} |\lambda, q_\lambda, p_\lambda\rangle_{\mathrm{Lab}}, \tag{5.36}$$

where the sum over $q_\lambda^{\mathrm{coin}}$ is taken over the coincident subspace, that is, the irrep basis states with pattern weight $(11\cdots 1)$. That the dimensions of these spaces are the same can be shown combinatorially and follows from the theorem. This shows that although in the ideal case only the symmetric subspace is in play and therefore the full Schur transform is overkill, for the distinguishable case all irreps λ can play a role.

5.4.1 Postselection of ideal bosonic sampling

Although Unitary-Unitary Duality can be demonstrated in this model by implementing local Schur transforms before measuring, in both the ideal and distinguishable case circuits considered previously, it was argued that this was not necessary; it was enough to measure in the computational basis after the unitary transformation was implemented and post-process. An interesting observation is that given a distinguishable input, by performing the second Schur transform on the System it becomes possible to use postselection to sample from the indistinguishable distribution. Of course, this comes at the cost of throwing away a lot of bad samples.

To achieve this postselective filtering, we need to ensure that we only sample the System from the fully symmetric irrep of $\mathrm{U}(m)$. This is done by measuring the irrep register $|\lambda\rangle_{\mathrm{Sys}}$ and waiting for the outcome $\lambda_{\mathrm{Sys}} = (n)$. After postselection, the amplitudes of the System q-register $|U \cdot q_{(n)}\rangle$ are given by Equation 5.8, which give the same probability distribution as sampling indistinguishable bosons. Following the arguments above, the circuit remains efficient since the added Schur transform can be implemented efficiently.

More generally, such a postselected quantum circuit could sample from any irrep λ of $\mathrm{U}(m)$. All we need to achieve this is to ensure that the input state has support in the irrep we wish to sample from, and postselect on being in that irrep. A dimension counting argument shows that the completely distinguishable input discussed above has support in all irreps [222], and so could be used for this purpose.

It is worth noting however that the probability of this postselection succeeding can be extremely small. For the worst case, when we have the fully distinguishable state, the probability of postselecting on the fully symmetric irrep is $1/n!$, thus we expect on average $O(n!)$ attempts before postselection is successful.

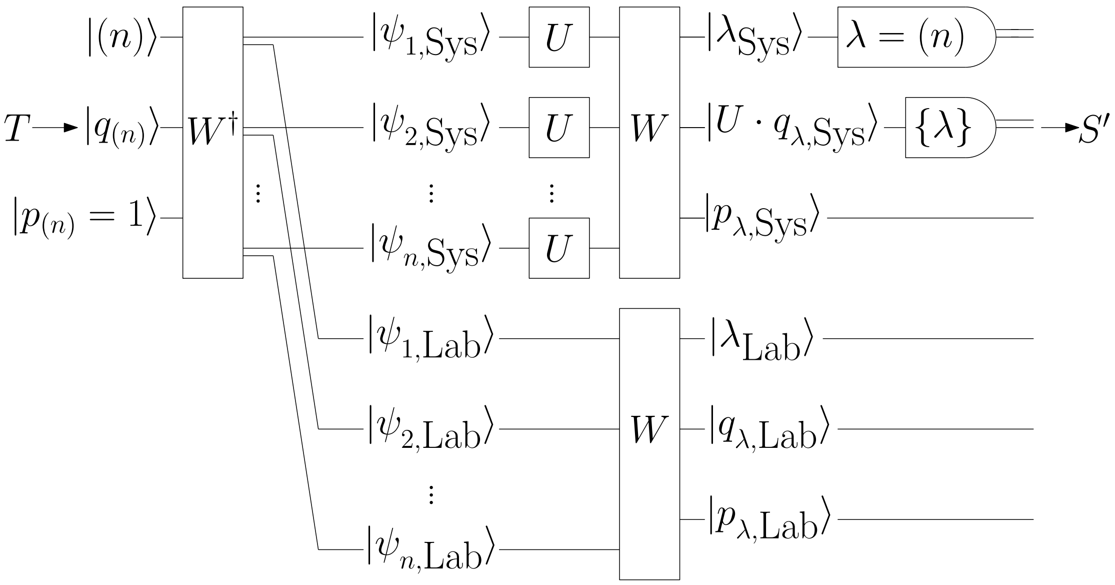

$|(n)\rangle$
T
$|q_{(n)}\rangle$
$|p_{(n)} = 1\rangle$
$W^\dagger$
$|\psi_{1,\mathrm{Sys}}\rangle$
$|\psi_{2,\mathrm{Sys}}\rangle$
$|\psi_{n,\mathrm{Sys}}\rangle$
U
W
$|\lambda_{\mathrm{Sys}}\rangle$
$\lambda = (n)$
$|U \cdot q_{\lambda,\mathrm{Sys}}\rangle$
$\{\lambda\}$
S'
$|p_{\lambda,\mathrm{Sys}}\rangle$
$|\psi_{1,\mathrm{Lab}}\rangle$
$|\psi_{2,\mathrm{Lab}}\rangle$
$|\psi_{n,\mathrm{Lab}}\rangle$
$|\lambda_{\mathrm{Lab}}\rangle$
$|q_{\lambda,\mathrm{Lab}}\rangle$
$|p_{\lambda,\mathrm{Lab}}\rangle$

Figure 5.3: Circuit diagram illustrating how postselection can be used to 'filter out' distinguishability. Note that the postselection measurement of the System λ-register is in the irrep basis, while that of the System q-register is in the GZ basis. The second Schur transform on the Label register is not necessary, but illustrates the Unitary-Unitary Duality.

5.5 Sampling with loss

Another serious practical difficulty with linear optics is the loss of photons through unwanted scattering processes. In this section, we discuss how the loss model developed by Aaronson and Brod [211] can be simulated. In their model, $n + k$ bosons are generated as occupation S^0, k of which are lost before they reach the interferometer. As we don't know which bosons were lost, Aaronson and Brod take the average over the set of all n-boson occupations which are consistent with S^0, denoted $\bar{\Lambda}_{m,S^0,n}$. The result can be shown by tracing out any choice of k qudits in the ideal case, as shown in Figure 5.4. This fits with the model of loss in first quantisation described in Section 4.5.2.

Theorem 5.4. *Let $|\psi\rangle$ be the state after step 2 of Algorithm 5.1 with $(n + k)$-boson input state S^0, and $\bar{\Lambda}_{m,S^0,n}$ be the set of all n-boson occupations which are consistent with S^0. If k qudits of $|\psi\rangle$ are traced out before continuing with the algorithm, the final probability distribution of output occupations S', denoted $\mathcal{D}_U$, is*

$$\Pr_{\mathcal{D}_U}[S'] = \frac{1}{\binom{n+k}{k}} \sum_{S \in \bar{\Lambda}_{m,S^0,n}} |\mathrm{per}(U_{S',S})|^2 \prod_{i=1}^{m} \frac{\binom{S_i^0}{S_i}}{S_i'! S_i!}. \tag{5.37}$$

Proof. The state $|\psi\rangle$ can be written as the density matrix

$$\rho = |\psi\rangle\langle\psi| = \frac{1}{(n+k)! \prod_{i=1}^{m} S_i^0!} \sum_{\sigma \in \mathrm{S}_{n+k}} \sigma|s^0\rangle \sum_{\sigma' \in \mathrm{S}_{n+k}} \langle s^0|\sigma'^\dagger, \tag{5.38}$$

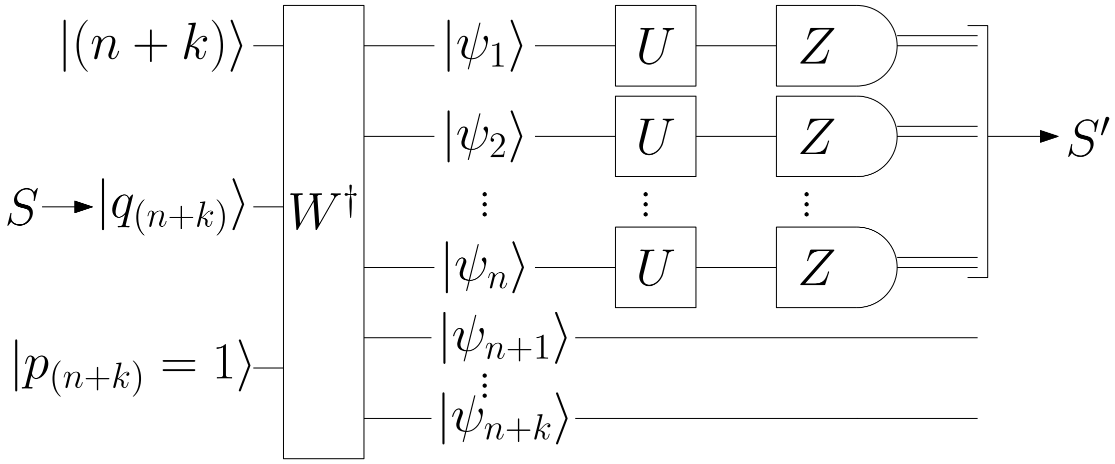


Figure 5.4: Circuit for sampling when k bosons are lost. Here, we ignore k qudits of the System register, tracing them out with the Label register. As with Figure 5.2, measurements are in the computational basis.

where $|s^0\rangle$ is any state consistent with the input state occupation S^0. This state is symmetric, so the choice of which qudits to trace out is moot. Choosing the last k qudits, the reduced density matrix for the remaining n particles is

$$\rho_n = \frac{1}{(n+k)!\prod_{i=1}^m S_i^0!}\sum_{\sigma,\sigma'\in\mathrm{S}_{n+k}}\bigotimes_{l=1}^{n}|s^0_{\sigma^{-1}(l)}\rangle\langle s^0_{\sigma'^{-1}(l)}|\bigotimes_{l=n+1}^{n+k}\langle s^0_{\sigma'^{-1}(l)}|s^0_{\sigma^{-1}(l)}\rangle \tag{5.39}$$

$$= \frac{1}{(n+k)!\prod_{i=1}^m S_i^0!}\sum_{\substack{S\subseteq\{s_1^0,\dots,s_{n+k}^0\}\\ \bar{S}=\{s_1^0,\dots,s_{n+k}^0\}\setminus S\\ |S|=n}}\sum_{\sigma,\sigma'\in\mathrm{S}_n}\sigma|s\rangle\langle s|\sigma'^\dagger\sum_{\tau,\tau'\in\mathrm{S}_k}\langle\bar{s}|\tau'^\dagger\tau|\bar{s}\rangle \tag{5.40}$$

$$= \frac{1}{(n+k)!\prod_{i=1}^m S_i^0!}\sum_{\substack{S\in\Lambda_{m,S^0,n}\\ \bar{S}=\{s_1^0,\dots,s_{n+k}^0\}\setminus S}}\prod_{i=1}^m\left(\frac{S_i^0!}{S_i!(S_i^0-S_i)!}\right)^2\sum_{\sigma,\sigma'\in\mathrm{S}_n}\sigma|s\rangle\langle s|\sigma'^\dagger\sum_{\tau,\tau'\in\mathrm{S}_k}\langle\bar{s}|\tau'^\dagger\tau|\bar{s}\rangle \tag{5.41}$$

$$= \frac{k!}{(n+k)!}\sum_{S\in\Lambda_{m,S^0,n}}\frac{\prod_{i=1}^m S_i^0!}{(\prod_{j=1}^m S_j!)^2\prod_{i=1}^m(S_i^0-S_i)!}\sum_{\sigma,\sigma'\in\mathrm{S}_n}\sigma|s\rangle\langle s|\sigma'^\dagger \tag{5.42}$$

$$= \frac{1}{\binom{n+k}{n}}\sum_{S\in\Lambda_{m,S^0,n}}\frac{\prod_{i=1}^m\binom{S_i^0}{S_i}}{n!\prod_{j=1}^m S_j!}\sum_{\sigma,\sigma'\in\mathrm{S}_n}\sigma|s\rangle\langle s|\sigma'^\dagger, \tag{5.43}$$

where now $|s\rangle$ (resp. $|\bar{s}\rangle$) is any state consistent with the occupation S (resp. $\bar{S}$). In this calculation we first break the qudits into multisets of S and $\bar{S}$ with respective sizes n and k, and permute each multiset individually, which is done in Equation (5.40). Note that $S\subseteq\{s_1^0,\dots,s_{n+k}^0\}$ such that $|S|=n$ implies that $S\in\Lambda_{m,S^0,n}$ defined above, so we can sum over $\Lambda_{m,S^0,n}$. However, doing so will ignore duplicates of S we had when considering multisets included in $\{s_1^0,\dots,s_{n+k}^0\}$, which need to be acconted for. The total number of duplicate terms is the number of permutations $\sigma, \sigma' \in \mathrm{S}_{n+k}$ for which $|s^0\rangle$ is invariant, of which there are $(\prod_{i=1}^m S_i^0!)^2$. The permutations $\sigma, \sigma' \in \mathrm{S}_n$ and $\tau, \tau' \in \mathrm{S}_k$ mean that $(\prod_{i=1}^m S_i!(S_i^0 - S_i)!)^2$ duplicates are already accounted for. Putting these two points together, we get the factor seen in Equation (5.41). Finally in Equation (5.42), we take the inner product, noting that $\sum_{\tau,\tau' \in \mathrm{S}_k} \langle \bar{s}|\tau'^{\dagger}\tau|\bar{s}\rangle = k! \prod_{i=1}^m (S_i^0 - S_i)!$.

Applying the unitary transformation U and measuring in the computational basis, we find that the calculation of the probability of measuring a state $|s'\rangle \in (\mathbb{C}^m)^{\otimes n}$ goes through much as in the previous sections. Applying the same methods as before, we have

$$\Pr[|s'\rangle] \tag{5.44}$$

$$= \mathrm{Tr}[|s'\rangle\langle s'|(U)^{\otimes n}\rho_n(U^{\dagger})^{\otimes n}] \tag{5.45}$$

$$= \frac{1}{\binom{n+k}{n}} \sum_{S \in \bar{\Lambda}_{m,n+k,n}} \frac{\prod_{i=1}^m \binom{S_i^0}{S_i}}{n! \prod_{j=1}^m S_j!} \left| \langle s'|U^{\otimes n} \left(\sum_{\sigma \in \mathrm{S}_n} \sigma|s\rangle \right) \right|^2 \tag{5.46}$$

$$= \frac{1}{\binom{n+k}{n}} \sum_{S \in \bar{\Lambda}_{m,n+k,n}} |\mathrm{per}(U_{S',S})|^2 \prod_{i=1}^m \frac{\binom{S_i^0}{S_i}}{n! S_i!}. \tag{5.47}$$

To find the probability of sampling occupation S', we add together the probabilities for all computational basis states $|s'\rangle$ that map to S', of which there are $n!/\prod_{i=1}^m S_i'!$. This gives us the desired probability distribution. □

Combining loss with distinguishability can be simulated by splitting the remaining n qudits in the state given by Equation (5.43) into System and Label registers, and tracing out the Label. This would result in similar averages over the lossless cases described in the sections above.

5.6 Distinguishability and simulateability

Our model of distinguishability as correlations with the Label register gives an explicit decoherence model for the computation on the System register. A natural question is to ask at what point this decoherence renders the quantum computation classically simulateable. There is a large amount of literature surrounding classical simulation of mixed state quantum computing [270, 271, 272], and the role of entanglement [273, 274, 275, 276, 277]. We've already seen that when the input is completely distinguishable, the output distribution is given by the permanents of positive matrices, Equation (5.33), which can be approximated in polynomial time [278]. This efficient permanent approximation method can be used with Clifford and Clifford's algorithm [91] to produce a polynomial runtime for approximate sampling. Another method for efficiently simulating fully distinguishable photons is to simulate each photon going through the interferometer individually [188, 235].

The discussion up to this point shows that results on the classical simulation of the Schur transform would allow us to answer this question, but general results along these lines are to the best of our knowledge not available. As mentioned above, one way to approach the question is to consider the multipartite entanglement properties of the mixed state of the System that results after tracing out the Label.

Without a specific noise model, there are several mixed states we could consider; an obvious one is a mixture of the ideal and completely indistinguishable states

$$\rho_\epsilon = \epsilon \left(\frac{\sum_{\sigma \in \mathrm{S}_n} \sigma|s\rangle}{\sqrt{n!}} \right) \left(\frac{\sum_{\tau \in \mathrm{S}_n} \langle s|\tau^{\dagger}}{\sqrt{n!}} \right) + (1-\epsilon) \left(\frac{\sum_{\sigma \in \mathrm{S}_n} \sigma|s\rangle\langle s|\sigma^{\dagger}}{n!} \right). \tag{5.48}$$

For simplicity, we shall only consider coincident photon states, and leave analysis of more general photon states as an open question. Equation (5.26) tells us that the completely indistinguishable state is maximally mixed on the coincident subspace, and it has been shown that states of the form $\epsilon |\psi\rangle \langle\psi| + (1-\epsilon)\mathbb{1}/d$ are separable for sufficiently small ϵ [279, 280], where $\mathbb{1}/d$ is the completely mixed state on the entire space (the tensor product $(\mathbb{C}^m)^{\otimes n}$). It is therefore tempting to conclude that there is a measurable set of states near the completely distinguishable state that are separable. However, the completely distinguishable state only has support on the coincident subspace, which is not a tensor product, and so these results cannot be applied directly.

We can in fact show that for any $\epsilon > 0$ the reduced System state of Equation (5.48) is entangled, in that it fails the partial transpose criterion [281]. This is similar to results that show a mixture of the totally antisymmetric state and the projector on the symmetric subspace are entangled for two qudits [282].

To demonstrate entanglement, we shall use the generalised partial transpose criterion [281], which is a method for testing if a mixed state $\rho = \sum_{i,j} \rho_{i,j} |i\rangle \otimes \langle j|$ is entangled. Define the trace norm of ρ as $\|\rho\| \colon = \operatorname{Tr} \sqrt{\rho^\dagger \rho}$. We denote a row transposition as $\rho^{T_r} = \sum_{i,j} \rho_{i,j} \langle i| \otimes \langle j|$, and a column transposition as $\rho^{T_c} = \sum_{i,j} \rho_{i,j} |i\rangle \otimes |j\rangle$. The generalised partial transpose criterion states that ρ is separable if for any transposition $y \subset \{r_A, c_A, \dots, r_Z, c_Z\}$ of rows or columns corresponding to subsystems of ρ, $\|\rho^{T_y}\| \leq 1$. In our case, it is sufficient to only transpose the rows and columns corresponding to the first particle.

First, we rewrite ρ_ϵ in the form

$$\rho_\epsilon = \frac{1}{n!}\left((1-\epsilon) \sum_{\sigma \in \mathrm{S}_n} \sigma|s\rangle\langle s|\sigma^\dagger + \epsilon \sum_{\sigma,\tau \in \mathrm{S}_n} \sigma|s\rangle\langle s|\tau^\dagger \right) \tag{5.49}$$

$$= \frac{1}{n!}\left((1-\epsilon) \sum_{\sigma \in \mathrm{S}_n} \sigma|s\rangle\langle s|\sigma^\dagger + \epsilon \sum_{\substack{\sigma,\tau \in \mathrm{S}_n \\ \sigma^{-1}(1)=\tau^{-1}(1)}} \sigma|s\rangle\langle s|\tau^\dagger + \epsilon \sum_{\substack{\sigma,\tau \in \mathrm{S}_n \\ \sigma^{-1}(1)\neq\tau^{-1}(1)}} \sigma|s\rangle\langle s|\tau^\dagger \right), \tag{5.50}$$

where we have separated the sum over σ and τ into two sums, based on whether or not $\sigma^{-1}(1) = \tau^{-1}(1)$. Transposing the first qudit will leave the first of these sums invariant, while always affecting the second. The resulting state after this transposition is

$$\rho_\epsilon^{T_1} = \frac{1}{n!}\left((1-\epsilon) \sum_{\sigma \in \mathrm{S}_n} \sigma|s\rangle\langle s|\sigma^\dagger + \epsilon \sum_{\substack{\sigma,\tau \in \mathrm{S}_n \\ \sigma^{-1}(1)=\tau^{-1}(1)}} \sigma|s\rangle\langle s|\tau^\dagger \right.$$
$$\left. + \epsilon \sum_{\substack{\sigma,\tau \in \mathrm{S}_n \\ \sigma^{-1}(1)\neq\tau^{-1}(1)}} |s_{\tau^{-1}(1)}\rangle\langle s_{\sigma^{-1}(1)}| \bigotimes_{i=2}^{n} |s_{\sigma^{-1}(i)}\rangle\langle s_{\tau^{-1}(i)}| \right). \tag{5.51}$$

To work out the trace norm of this density matrix, we need to multiply it by its transpose and take the square root. We note that of the three summations that make up $\rho_\epsilon^{T_1}$, the first two have no overlap with the third. This is because of the transposition, which as noted earlier only affects the third summation. For every term in the third summation, there is some $i \in \{2, \dots, n\}$ such that $|s_{\tau^{-1}(1)}\rangle = |s_{\sigma^{-1}(i)}\rangle$. As a result, any term from the other two summations with permutation χ either has $\langle s_{\chi^{-1}(1)}|s_{\tau^{-1}(1)}\rangle = 0$ or $\langle s_{\chi^{-1}(i)}|s_{\sigma^{-1}(i)}\rangle = 0$. Therefore we only need to consider the product of the first two sums with each other and product of the third summation with itself.

Note that this relies on our photons being coincident, to ensure that $\langle s_i|s_j\rangle = \delta_{i,j}\forall i,j \in [n]$. For more general bosonic states, where multiple photons might occupy the same spatial mode, this simplification would not hold and we might need to consider more complicated analysis. We discuss some consequences of more general photonic states at the end of this section.

Using this knowledge, we compute $\rho_\epsilon^{T_1}\rho_\epsilon^{T_1\dagger}$ as

$$\begin{aligned}
\rho_\epsilon^{T_1}\rho_\epsilon^{T_1\dagger} = \frac{1}{(n!)^2}\Bigg(& (1-\epsilon)^2 \sum_{\sigma\in \mathrm{S}_n} \sigma|s\rangle\langle s|\sigma^\dagger + 2(1-\epsilon)\epsilon \sum_{\substack{\sigma,\tau\in \mathrm{S}_n\\ \sigma^{-1}(1)=\tau^{-1}(1)}} \sigma|s\rangle\langle s|\tau^\dagger \\
& +\epsilon^2 \sum_{\substack{\sigma,\tau,\upsilon,\chi\in \mathrm{S}_n\\ \sigma^{-1}(1)=\tau^{-1}(1)\\ \upsilon^{-1}(1)=\chi^{-1}(1)}} \sigma|s\rangle\langle s|\tau^\dagger\chi|s\rangle\langle s|\upsilon^\dagger \\
& +\epsilon \sum_{\substack{\sigma,\tau,\upsilon,\chi\in \mathrm{S}_n\\ \sigma^{-1}(1)\neq\tau^{-1}(1)\\ \upsilon^{-1}(1)\neq\chi^{-1}(1)}} |s_{\tau^{-1}(1)}\rangle\langle s_{\sigma^{-1}(1)}|s_{\upsilon^{-1}(1)}\rangle\langle s_{\chi^{-1}(1)}| \\
& \bigotimes_{i=2}^{n} |s_{\sigma^{-1}(i)}\rangle\langle s_{\tau^{-1}(i)}|s_{\chi^{-1}(i)}\rangle\langle s_{\upsilon^{-1}(i)}| \Bigg)
\end{aligned} \tag{5.52}$$

$$\begin{aligned}
= \frac{1}{(n!)^2}\Bigg(& (1-\epsilon)^2 \sum_{\sigma\in \mathrm{S}_n} \sigma|s\rangle\langle s|\sigma^\dagger + 2(1-\epsilon)\epsilon \sum_{\substack{\sigma,\tau\in \mathrm{S}_n\\ \sigma^{-1}(1)=\tau^{-1}(1)}} \sigma|s\rangle\langle s|\tau^\dagger \\
& +\epsilon^2(n-1)! \sum_{\substack{\sigma,\tau\in \mathrm{S}_n\\ \sigma^{-1}(1)=\tau^{-1}(1)}} \sigma|s\rangle\langle s|\tau^\dagger \\
& +\epsilon \sum_{\substack{\sigma,\tau,\upsilon\in \mathrm{S}_n\\ \sigma^{-1}(1)\neq\tau^{-1}(1)\\ \sigma^{-1}(1)=\upsilon^{-1}(1)}} |s_{\tau^{-1}(1)}\rangle\langle s_{\tau^{-1}(1)}| \bigotimes_{i=2}^{n} |s_{\sigma^{-1}(i)}\rangle\langle s_{\upsilon^{-1}(i)}| \Bigg).
\end{aligned} \tag{5.53}$$

The square root of this matrix can be written as

$$\sqrt{\rho_\epsilon^{T_1}\rho_\epsilon^{T_1\dagger}} = \frac{1}{n!}\left((1-\epsilon)\sum_{\sigma\in \mathrm{S}_n}\sigma|s\rangle\langle s|\sigma^\dagger + \epsilon\sum_{\substack{\sigma,\tau\in\mathrm{S}_n\\ \sigma^{-1}(1)=\tau^{-1}(1)}}\sigma|s\rangle\langle s|\tau^\dagger\right.$$
$$\left.+\frac{\epsilon}{(n-1)!}\sum_{\substack{\sigma,\tau,\upsilon\in\mathrm{S}_n\\ \sigma^{-1}(1)\neq\tau^{-1}(1)\\ \sigma^{-1}(1)=\upsilon^{-1}(1)}}|s_{\tau^{-1}(1)}\rangle\langle s_{\tau^{-1}(1)}|\bigotimes_{i=2}^{n}|s_{\sigma^{-1}(i)}\rangle\langle s_{\upsilon^{-1}(i)}|\right). \tag{5.54}$$

From this we can work out the trace norm as

$$\|\rho_\epsilon^{T_1}\|_* = \mathrm{Tr}\left[\sqrt{\rho_\epsilon^{T_1}\rho_\epsilon^{T_1\dagger}}\right] \tag{5.55}$$
$$= \frac{1}{n!}\left((1-\epsilon)n! + \epsilon n! + \frac{\epsilon}{(n-1)!}(n-1)n!(n-1)!\right) \tag{5.56}$$
$$= 1-\epsilon+\epsilon+\epsilon(n-1) \tag{5.57}$$
$$= 1+\epsilon(n-1). \tag{5.58}$$

Here, the $(1-\epsilon)n!$ and $\epsilon n!$ terms come from the trace of the first two terms. The third term comes from Equation (5.54), the trace of which one can think of as the number of ways we can pick permutations $\sigma,\tau\in\mathrm{S}_n$ such that $\sigma^{-1}(1)\neq\tau^{-1}(1)$. This can be worked out by choosing any permutation $\sigma\in\mathrm{S}_n$, and constructing τ by first choosing $\tau^{-1}(1)\neq\sigma^{-1}(1)$ and choosing $\tau^{-1}(i)$ for $i>1$ to be a permutation in S_{n-1}. Thus the overall number is $(n-1)n!(n-1)!$.

We can see that the trace norm of $\rho_\epsilon^{T_1}$ therefore fails the generalised partial transposition criterion for separability for all $n>1$ and $\epsilon>0$. Note that this does not imply that efficiently sampling from unitary actions on such states is not classically possible for any nonzero ϵ, merely that techniques used for simulating separable states cannot be used for exact sampling in this case. Indeed, as we shall see in Section 6.2.2, Boson Sampling with this input state can be *approximately* simulated using separable states.

As stated earlier, this relies on our initial state being n coincident photons. One example of where this does not hold for particular input occupations is the photonic state corresponding to n photons all generated in the same spatial mode. Let i be the spatial mode in question. In this case note that the Label register is completely separable from the System register regardless of distinguishability

$$\frac{1}{\sqrt{n!}}\sum_{\sigma\in\mathrm{S}_n}\sigma\,|i\rangle^{\otimes n}\,\sigma\,|l\rangle = |i\rangle^{\otimes n}\left(\frac{1}{\sqrt{n!}}\sum_{\sigma\in\mathrm{S}_n}\sigma\,|l\rangle\right). \tag{5.59}$$

As a result, tracing out the Label register will leave the state $|i\rangle^{\otimes n}$, which is a product state. Thus an equivalent state given in Equation (5.48), a mixture of the fully indistinguishable and fully distinguishable photonic states, would be separable for any choice of ϵ. It is therefore likely that entanglement in first quantisation depends on both indistinguishability and photon occupation. We leave more general analysis as a direction for further research.

5.7 Conclusion

We have described how to use the Schur transform to perform a quantum simulation of bosonic sampling when the bosons are arbitrarily distinguishable. These results make it clear that ideal n boson, m mode linear interferometry is equivalent to a transversal n qudit quantum circuit, with the constraint that the input must be totally symmetric — that is, the ordering of the qudits must be erased. Moreover, we can introduce nonideal aspects into the quantum simulation by tracing out qudits (loss), or introducing ancillas and entanglement (distinguishability). A recently released paper focusing on the issue of loss in more detail makes similar connections [86].

5.7.1 Open questions

A broad aim of future research along this approach is to better understand how the computational complexity of sampling from photons changes as photons become more distinguishable. By better understanding of how these intermediate levels of distinguishability link to representation theory, the hope is that it will be easier to find either classical algorithms for these cases similarly to Clifford and Clifford [91], or in finding reductions to sampling immanants [264] akin to Aaronson and Arkhipov's focused on permanents. Indeed, recent work by Havlíček and Strelchuk has demonstrated the potential for the use of the Schur transform in understanding the complexity of sampling problems [252]. We shall start to make some progress towards this understanding in Chapter 6.

Chapter 6

Classically simulating near-term partially-distinguishable and lossy Boson Sampling

6.1 Introduction

This work was completed in collaboration with Raúl García-Patrón, Jelmer J. Renema and Peter S. Turner, and published as "Classically simulating near-term partially-distinguishable and lossy boson sampling", *Quantum Science and Technology* **5**, 015001 (2020), copyright Institute of Physics. A preprint of this article is freely available at `arXiv:1907.00022`. Details of contributions made by myself are given in Section 1.3.4.

As we saw in Chapter 4, recent classical simulations have been expanded to consider practical issues such as photon distinguishability, based on a rich collection of theoretical work [220, 212, 209, 204, 205, 206, 283]. Renema et al. [84] demonstrated that Boson Sampling with partially-distinguishable photons can be simulated in time which grows polynomially with n, which was later expanded to consider loss as well [85]. However, the runtime might still not be efficient in practice, as the polynomial can be large. There is also a further disadvantage in that the error bounds are the average case for a random linear optical interferometer, meaning that there could be interferometers for which the algorithm performs significantly worse. A significant improvement could be achieved through adapting the method of Clifford & Clifford to this algorithm, but there are challenges with this approach.

Here we consider the cost of classically simulating Boson Sampling when the photons are partially distinguishable or lossy. We look at the same model of distinguishability as considered in [84, 85], and use techniques for modelling photon distinguishability in first quantization as described in Chapter 5 to show that this is akin to choosing the indistinguishable photons of a Boson Sampling experiment via the binomial distribution. We combine this with the well-studied model of uniform loss, where each photon independently survives with probability η. Under this model, the probability of how many photons survive overall also follows a binomial distribution. This gives rise to a method which is able to naturally apply the Clifford & Clifford algrithm and take advantage of its efficiency. This algorithm also offers a worst-case error bound for *any* linear optical interferometer, rather than simply offering a small error on average for a Haar random interferometer. Although this approach only offers a polynomial improvement compared to the runtime for ideal Boson Sampling (unlike the exponential improvement shown in [84, 85]) we use analytical bounds to show that for photon numbers of experimental interest our algorithm can make a significant improvement over alternative approaches.

This chapter is laid out as follows. We start in Section 6.2 by discussing some ideas which were previously considered by us but had limitations with their feasibility. Our main results start in Section 6.3, we show what the Renema et al. [84, 85] model of distinguishability looks like in first quantization, and provide an alternative classical simulation. In Section 6.4, we consider average error bounds for a Haar-random unitary interferometer, via the methods explained in Section 4.6.4. In Section 6.5, we improve this bound to a worst-case error bound, by computing an upper bound for the trace distance between our approximation and the model. In Section 6.6, we expand these results to consider uniform loss, and show how distinguishability and loss relate to each other. In Section 6.7, we explore these error bounds for experimentally interesting numbers of photons, and show that there are some cases where our algorithm offers an improvement. Finally, we briefly consider *non*-uniform loss, where loss is a function of the number of optical components, and use the methods of [83, 86] to show that classical simulations with non-uniform loss also become easier when distinguishability is introduced. We conclude with some possible improvements to the work in Section 6.9, and some open research questions in Section 6.10.

6.2 Limitations of other classical simulation algorithms

Before disucssing our classical algorithm, we shall look at other classical potential classical simulation algorithms and explain some of their limitations when applied to Boson Sampling under distinguishability. In particular, we shall discuss adapting the methods of [252, 254] and [86] to Boson Sampling under distinguishability and loss. It is worth noting that these ideas do not contribute to our main results, which begin in Section 6.3.

6.2.1 Classical Tractability

We shall start by discussing the issue with employing the Classical Tractability method of Van den Nest [256], similarly to the technique used by Havlíček and Strelchuk [252]. There are some promising initial results for using this technique to approximate the transition amplitudes of imperfect Boson Sampling, but it very quickly becomes clear that this is not efficiently possible in general.

Recall from Section 4.7.4 that a state is Classically Tractable (CT) if we can efficiently approximate its amplitudes and measure the state in the computational basis up to polynomial multiplicative error [256]. It is easy to see in this case that the input state to Boson Sampling in first quantisation is a CT state, as the only basis states with non-zero amplitude are those which are permutations of $[n]$, and those states have uniform amplitudes. More formally, the amplitude of a state $|a\rangle \in (\mathbb{C}^m)^{\otimes n}$ is

$$\psi_a = \begin{cases} 1/\sqrt{n!} & \text{if } \exists \sigma \in \mathrm{S}_n \text{ s.t. } \sigma(a_i) = i \forall i \in [n] \\ 0 & \textit{otherwise} \end{cases} . \tag{6.1}$$

Similarly, a measurement in the computational basis can be efficiently sampled by sampling a permutation $\sigma \in \mathrm{S}_n$ uniformly at random and outputting $\sigma(1), \sigma(2), \ldots, \sigma(n)$. Thus our input state meets the requirements for Computational Tractability.

This seems promising, as the entanglement was believed to be the hard part of Section 5.6, and now this provides a way of classically simulating that step. However, we shall see that in this setting the action of the interferometer leads to classical simulation being hard.

In order to approximate the probability of an outcome, the operation we apply to the input state needs to be Efficiently Computable Sparse (ECS). From Van den Nest [256], this means that each row and column of the matrix has at most poly(n) non-zero elements and each element of the matrix can be approximated to polynomial multiplicative error in polynomial time. In our case, this operator is $U^{\otimes n}$.

Each element of this operator can be efficiently approximated as a product of different elements of U. However, the matrix is very large, with m^n elements in each row and column. This is especially poor in the case where $m = O(n^2)$, believed to be the regime required for Boson Sampling to provide a quantum advantage.

There are some potentially positive directions here. If there are a small number of non-zero elements in each row and column of U, then many of the elements in $U^{\otimes n}$ would also be zero, and then the matrix could be interpreted as ECS. This could be true if, for example, the photonic circuit is low depth. Another potential direction is to use this method to simulate Boson Sampling under photon loss, which would reduce the exponent. However, it is unclear how either of these methods would lead to a faster simulation than ones that are already known.

6.2.2 Distance from the fully distinguishable state

In this section we shall consider limitations with simply using a separable state. This is inspired by the work on Oszmaniec and Brod [86], who showed that with sufficient amounts of loss the Boson Sampling input state is arbitrarily close to a known separable state. However, it is also similar to the technique used by Deshpande et al. [230] and Maskara et al. [231], who used the input state of fully distinguishable particles to classically simulate Boson Sampling with low-depth circuits.

The most natural state to use to classically simulate Boson Sampling under distinguishability is the fully distinguishable state

$$\rho_D = \frac{1}{n!} \sum_{\sigma \in \mathrm{S}_n} \sigma \left|s\right\rangle \left\langle s\right| \sigma^\dagger, \tag{6.2}$$

where $\left|s\right\rangle = \bigotimes_{i=1}^n \left|i\right\rangle$. We shall also use

$$\rho_I = \frac{1}{n!} \sum_{\sigma,\sigma' \in \mathrm{S}_n} \sigma \left|s\right\rangle \left\langle s\right| \sigma'^\dagger \tag{6.3}$$

to indicate the fully indistinguishable input state.

As already noted in Section 4.6.2, the state ρ_D is completely separable, and an interferometer acting on it can be classically simulated in polynomial time. In this section, we shall compute the trace distance between this state and various other Boson Sampling input states which could be interesting. For this, note that the trace distance between mixed states ρ, ρ' is defined as

$$\delta_{\mathrm{Tr}}(\rho,\rho') = \mathrm{Tr}\left(\sqrt{(\rho-\rho')(\rho-\rho')^\dagger}\right). \tag{6.4}$$

We start by subtracting ρ_I from ρ_D, which gives the matrix

$$\rho_I - \rho_D = \frac{1}{n!}\left(\sum_{\sigma,\sigma'\in \mathrm{S}_n} \sigma\left|s\right\rangle\left\langle s\right|\sigma'^{\dagger} - \sum_{\sigma\in \mathrm{S}_n} \sigma\left|s\right\rangle\left\langle s\right|\sigma^{\dagger}\right) \tag{6.5}$$

$$= \frac{1}{n!}\left(\sum_{\substack{\sigma,\sigma'\in \mathrm{S}_n\\ \sigma\neq\sigma'}} \sigma\left|s\right\rangle\left\langle s\right|\sigma'^{\dagger}\right). \tag{6.6}$$

Multiplying this matrix by its Hermitian adjoint gives the positive semidefinite matrix

$$(\rho_I - \rho_D)(\rho_I - \rho_D)^{\dagger} = \frac{1}{(n!)^2}\left(\sum_{\substack{\sigma,\sigma'\in \mathrm{S}_n\\ \sigma\neq\sigma'}} \sigma\left|s\right\rangle\left\langle s\right|\sigma'^{\dagger}\right)\left(\sum_{\substack{\tau,\tau'\in \mathrm{S}_n\\ \tau\neq\tau'}} \tau'\left|s\right\rangle\left\langle s\right|\tau^{\dagger}\right) \tag{6.7}$$

$$= \frac{1}{(n!)^2}\left((n!-2)\sum_{\sigma,\tau\in \mathrm{S}_n} \sigma\left|s\right\rangle\left\langle s\right|\tau^{\dagger} + \sum_{\sigma\in \mathrm{S}_n} \sigma\left|s\right\rangle\left\langle s\right|\sigma^{\dagger}\right). \tag{6.8}$$

As the above matrix is positive semidefinite, we can find its square root as

$$\sqrt{(\rho_I - \rho_D)(\rho_I - \rho_D)^{\dagger}} = \frac{1}{n!}\left(\left(\frac{n!-2}{n!}\right)\sum_{\sigma,\tau\in \mathrm{S}_n} \sigma\left|s\right\rangle\left\langle s\right|\tau^{\dagger} + \sum_{\sigma\in \mathrm{S}_n} \sigma\left|s\right\rangle\left\langle s\right|\sigma^{\dagger}\right). \tag{6.9}$$

From this we can work out the trace distance between the two states to be

$$\delta_{\mathrm{tr}}(\rho_I, \rho_D) = \frac{1}{2\times n!}\left(\left(\frac{n!-2}{n!}\right)n! + n!\right) \tag{6.10}$$

$$= \frac{1}{2}\left(\frac{n!-2}{n!} + 1\right) \tag{6.11}$$

$$= \frac{2\times n! - 2}{2\times n!} \tag{6.12}$$

$$= 1 - \frac{1}{n!}. \tag{6.13}$$

We next consider the singly distinguishable state, as considered by Stanisic and Turner [222]. If we denote the distinguishable photon as the photon in mode i, the first quantisation state looks like

$$\rho_{SD} = \frac{1}{n!}\sum_{\substack{\sigma,\sigma'\in \mathrm{S}_n\\ \sigma^{-1}(i)=\sigma'^{-1}(i)}} \sigma\left|s\right\rangle\left\langle s\right|\sigma'^{\dagger}.$$

Following the same structure as with fully indistinguishable photons, we find that subtracting the two matrices gives

$$\rho_{SD} - \rho_D = \frac{1}{n!}\left(\sum_{\substack{\sigma,\sigma'\in \mathrm{S}_n\\ \sigma^{-1}(i)=\sigma'^{-1}(i)}} \sigma\left|s\right\rangle\left\langle s\right|\sigma'^{\dagger} - \sum_{\sigma\in \mathrm{S}_n}\sigma\left|s\right\rangle\left\langle s\right|\sigma^{\dagger}\right) \tag{6.14}$$

$$= \frac{1}{n!}\left(\sum_{\substack{\sigma,\sigma'\in \mathrm{S}_n\\ \sigma^{-1}(i)=\sigma'^{-1}(i)\\ \sigma\neq\sigma'}} \sigma\left|s\right\rangle\left\langle s\right|\sigma'^{\dagger}\right). \tag{6.15}$$

Multiplying the matrix by its adjoint gives the positive semidefinite matrix

$$(\rho_{SD} - \rho_D)(\rho_{SD} - \rho_D)^{\dagger} = \frac{1}{(n!)^2}\left(\sum_{\substack{\sigma,\sigma'\in \mathrm{S}_n\\ \sigma^{-1}(i)=\sigma'^{-1}(i)\\ \sigma\neq\sigma'}} \sigma\left|s\right\rangle\left\langle s\right|\sigma'^{\dagger}\right)\left(\sum_{\substack{\tau,\tau'\in \mathrm{S}_n\\ \tau^{-1}(i)=\tau'^{-1}(i)\\ \tau\neq\tau'}} \tau'\left|s\right\rangle\left\langle s\right|\tau^{\dagger}\right) \tag{6.16}$$

$$= \frac{1}{(n!)^2}\left(((n-1)!-2)\sum_{\substack{\sigma,\tau\in \mathrm{S}_n\\ \sigma^{-1}(i)=\tau^{-1}(i)}} \sigma\left|s\right\rangle\left\langle s\right|\tau^{\dagger} + \sum_{\sigma\in \mathrm{S}_n}\sigma\left|s\right\rangle\left\langle s\right|\sigma^{\dagger}\right). \tag{6.17}$$

We can then work out the square root of this positive semidefinite matrix as the positive matrix

$$\sqrt{(\rho_{SD} - \rho_D)(\rho_{SD} - \rho_D)^{\dagger}} = \frac{1}{n!}\left(\left(\frac{(n-1)!-2}{(n-1)!}\right)\sum_{\substack{\sigma,\tau\in \mathrm{S}_n\\ \sigma^{-1}(i)=\tau^{-1}(i)}} \sigma\left|s\right\rangle\left\langle s\right|\tau^{\dagger} + \sum_{\sigma\in \mathrm{S}_n}\sigma\left|s\right\rangle\left\langle s\right|\sigma^{\dagger}\right). \tag{6.18}$$

Finally, we work out the trace of this matrix, and find the trace distance to be

$$\delta_{\mathrm{tr}}(\rho_{SD}, \rho_D) = \frac{1}{2\times n!}\left(\left(\frac{(n-1)!-2}{(n-1)!}\right)n! + n!\right) \tag{6.19}$$

$$= \frac{1}{2}\left(\frac{(n-1)!-2}{(n-1)!} + 1\right) \tag{6.20}$$

$$= \frac{2\times(n-1)!-2}{2\times(n-1)!} \tag{6.21}$$

$$= 1 - \frac{1}{(n-1)!}. \tag{6.22}$$

An intuitive question to ask at this point is to make sure this distance still applies after applying the Schur transform again, thus matching the method used by Stanisic and Turner [222]. The fully distinguishable state in this picture is

$$\rho_D = \frac{1}{n!}\sum_{\lambda\vdash n}\sum_{r_\lambda}\sum_{p_\lambda} |\lambda, p_\lambda, \underline{1}, r_\lambda\rangle\langle\lambda, p_\lambda, \underline{1}, r_\lambda|, \tag{6.23}$$

where we have written our unitary irrep basis state as a coincident occupation number $\underline{1} = (1^n)$ and multiplicity r_λ. Depending on choice of Schur basis, we can describe the Singly Distinguishable state as

$$\rho_{SD} = \frac{1}{n}\left(|(n), 1, \underline{1}, 1\rangle\,\langle(n), 1, \underline{1}, 1| + \sum_p |(n-1,1), p, \underline{1}, 1\rangle\,\langle(n-1,1), p, \underline{1}, 1|\right). \tag{6.24}$$

The difference between these two states is

$$\begin{aligned}\rho_{SD} - \rho_D = {}& \left(\frac{1}{n} - \frac{1}{n!}\right)\left(|(n), 1, \underline{1}, 1\rangle\,\langle(n), 1, \underline{1}, 1| + \sum_p |(n-1,1), p, \underline{1}, 1\rangle\,\langle(n-1,1), p, \underline{1}, 1|\right)\\ &+ \frac{1}{n!}\sum_{\substack{\lambda\vdash n\\ \ell(\lambda)>1}}\sum_{r_\lambda>1}\sum_{p_\lambda} |\lambda, p_\lambda, \underline{1}, r_\lambda\rangle\langle\lambda, p_\lambda, \underline{1}, r_\lambda| \qquad (6.25)\\ = {}& \frac{(n-1)!-1}{n!}\left(|(n), 1, \underline{1}, 1\rangle\,\langle(n), 1, \underline{1}, 1| + \sum_p |(n-1,1), p, \underline{1}, 1\rangle\,\langle(n-1,1), p, \underline{1}, 1|\right)\\ &+ \frac{1}{n!}\sum_{\substack{\lambda\vdash n\\ \ell(\lambda)>1}}\sum_{r_\lambda>1}\sum_{p_\lambda} |\lambda, p_\lambda, \underline{1}, r_\lambda\rangle\langle\lambda, p_\lambda, \underline{1}, r_\lambda|. \qquad (6.26)\end{aligned}$$

This matrix is diagonal and all non-zero elements are positive, meaning that $\sqrt{(\rho_{SD} - \rho_D)^2} = \rho_{SD} - \rho_D$. The trace distance thus ends up being

$$\begin{aligned}\delta_{\mathrm{tr}}(\rho_{SD}, \rho_D) &= \frac{1}{2}\,\mathrm{Tr}[\rho_{SD}, \rho_D] & (6.27)\\ &= \frac{1}{2}\left(\frac{(n-1)!-1}{n!} \times n + \frac{n! - n}{n!}\right) & (6.28)\\ &= \frac{1}{2}\left(2 \times \frac{n! - n}{n!}\right) & (6.29)\\ &= \frac{n! - n}{n!} & (6.30)\\ &= 1 - \frac{1}{(n-1)!}. & (6.31)\end{aligned}$$

Therefore either technique produces the same trace distance.

We can easily generalise the two results above to a case where k photons are fully indistinguishable, and the remaining $n - k$ are fully distinguishable. Now we specify that $\sigma^{-1}(i) = \sigma'^{-1}(i) \forall i \in \bar{K}$, where $\bar{K}$ is the set of $n - k$ distinguishable photons. The only other change is that the $(n-1)!$ terms become $k!$. The final trace distance will be $1-1/k!$. Even more generally, it seems likely that if we have two states with indistinguishable photons defined by set K and K', then the trace distance would equal $|K \cap K'|!/|K \cap K'|!$, though this has not been formally proven.

As can be seen from the above, for k indistinguishable photons and $n-k$ photons which are fully distinguishable from every other photon, we have the trace distance δ_{tr} decreasing as $1-1/k!$.

The property which the trace distance is useful for is that it is an upper bound of the total variation distance. According to Oszmaniec and Brod, this is helpful for loss as the trace distance tends to 0 as more photons are lost. But even when only two photons are indistinguishable, we still have a trace distance of 1/2, and this error only increases with the number of indistinguishable photons.

We conclude this section by considering a somewhat more positive case: the state ρ_ϵ, as described in Equation 5.48 from Section 5.6:

$$\rho_\epsilon = (1-\epsilon)\rho_D + \epsilon\rho_I. \tag{6.32}$$

The trace distance between this state and the fully distinguishable state is

$$\begin{aligned}\delta_{\text{tr}}(\rho_\epsilon, \rho_D) &\leq (1-\epsilon)\delta_{\text{tr}}(\rho_D, \rho_D) + \delta_{\text{tr}}(\epsilon\rho_I, \rho_D) &(6.33)\\ &= \epsilon - \frac{\epsilon}{n!}, &(6.34)\end{aligned}$$

where we have used the fact that the trace distance in convex. Noting that $\lim_{n\to\infty} \epsilon/n! = 0$, we can conclude that as $n \to \infty$ this state tends to ϵ. As our interferometer U cannot increase the trace distance, and the state ρ_D is classically simulable, then Boson Sampling with this state could be approximately simulated. This is an interesting change from the other cases, but not particularly surprising, given that this state is cooked up to essentially be the fully distinguishable state, with only ϵ weighting towards an ideal Boson Sampling state.

6.3 Expanding in terms of states

We now begin explaining the main results of this chapter, by considering the results of Renema et al. [84, 85] in first quantisation. First, recall from Section 4.6.4 that the case considered by Renema et al. is one where pairs of photons overlap with probability x, leading to a distribution where the probability of outcome photon occupation S' is

$$\Pr[S'] = \sum_{j=0}^{n} \sum_{\sigma^j} x^j \operatorname{per}(M * M^*_{1,\sigma}), \tag{6.35}$$

where the sum over σ^j means the sum of all permutations with j fixed points, M is a matrix constructed from rows and colums of our interferometer $U \in \mathrm{U}(m)$, and $M^*_{1,\sigma}$ is M with columns permuted by σ. Renema et al. approximate this probability as a sum of simpler matrix permanents which can be approximated in polynomial time.

We will now introduce a new expression for partially distinguishable particles. We start by writing the input state of n photons, with pairwise distinguishability parameter x as in the previous section, in first quantization

$$\rho_{n,x} = \frac{1}{n!}\left(\sum_{\sigma,\sigma'\in \mathrm{S}_n} \sigma \left|s\right\rangle \left\langle s\right| \sigma' x^{n-\sigma\cdot\sigma'}\right), \tag{6.36}$$

where we have used $\sigma \cdot \sigma'$ to denote the number of places where permutations σ and σ' match. For reference, the expansion of [84, 85] is carried out by identifying σ and σ' that match for a fixed set of i points:

$$\rho_{n,x} = \frac{1}{n!}\left(\sum_{i=0}^{n} x^i \sum_{\substack{\sigma,\sigma' \in \mathrm{S}_n \\ \exists I \subseteq [n] \text{ s.t. } |I|=i \\ \sigma^{-1}(j) \neq \sigma'^{-1}(j) \forall j \in I \\ \sigma^{-1}(j) = \sigma'^{-1}(j) \forall j \notin I}} \sigma \left|s\right\rangle \left\langle s\right| \sigma'^{\dagger}\right). \tag{6.37}$$

Note here that the terms in the sum over permutations do not correspond to physical states. This can be seen by the fact that for $i \neq 0$ this summation has no elements along the diagonal of the density matrix, as σ and σ' need to differ in *exactly* i places.

We instead look at an alternative expansion, in order to decompose the model into a linear combination of physical states:

$$\rho_{n,x} = \frac{1}{n!}\left(\sum_{i=0}^{n} p_i \sum_{\substack{\sigma,\sigma' \in \mathrm{S}_n \\ \exists I \subseteq [n] \text{ s.t. } |I|=i \\ \sigma^{-1}(j) = \sigma'^{-1}(j) \forall j \notin I}} \sigma \left|s\right\rangle \left\langle s\right| \sigma'^{\dagger}\right) \tag{6.38}$$

$$= \sum_{i=0}^{n} p_i \sum_{\substack{I \subseteq [n] \\ |I|=i}} \left(\frac{1}{n!} \sum_{\substack{\sigma,\sigma' \in \mathrm{S}_n \\ \sigma^{-1}(j) = \sigma'^{-1}(j) \forall j \notin I}} \sigma \left|s\right\rangle \left\langle s\right| \sigma'^{\dagger}\right) \tag{6.39}$$

$$= \sum_{i=0}^{n} p_i \sum_{\substack{I \subseteq [n] \\ |I|=i}} \rho_I, \tag{6.40}$$

where ρ_I is the state where photons in modes $j \in I$ are fully indistinguishable from each other, all other photons are fully distinguishable, and $0 \leq p_i \leq 1$ is a coefficient dependent on x and n determining the probability of a state with i indistinguishable single photons.

Note that unlike Equation (6.37), where permutations must differ in exactly i points, in Equation (6.38) we allow permutations to differ in *at most* i points. This means that elements closer to and along the diagonal of the density matrix are also part of this summation, and this means that each $\sigma, \sigma' \in \mathrm{S}_n$ term forms a valid density matrix.

Already we can see how a classical simulation might work — if we are able to sample p_i efficiently, then we can choose ρ_I by selecting i photons uniformly at random to be indistinguishable. These i photons can be classically simulated using Clifford & Clifford [91], while the remaining $n - i$ photons are treated as fully distinguishable photons, each of which can be simulated individually in polynomial time [188, 235].

6.3.1 The p_i are binomially distributed

Here, we will show that the coefficients p_i follow the binomial distribution

$$p_i = x^i (1-x)^{n-i}. \tag{6.41}$$

To see that the matrix elements of Equation (6.40) with p_i binomially distributed equal those of Equation (6.37), consider σ, σ' which differ at points in the set I, where $|I| = i$; the coefficient here should be x^i. Contributing to this element of the density matrix will be the state ρ_I, as well as other states $\rho_{I'}$, where $I \subseteq I'$. The number of such sets I' is $\binom{n-i}{i'-i}$, as it is equivalent to choosing i' from n elements when i elements have already been chosen. The corresponding matrix element is

$$
\begin{aligned}
&\frac{1}{n!}\left(\sum_{i'=i}^{n} x^{i'}(1-x)^{n-i'}\binom{n-i}{i'-i}\right) && (6.42)\\
&= \frac{1}{n!}\left(x^i \sum_{i'=i}^{n} x^{i'-i}(1-x)^{n-i'}\binom{n-i}{i'-i}\right) && (6.43)\\
&= \frac{1}{n!}\left(x^i \sum_{i'=0}^{n-i} x^{i'}(1-x)^{n-i-i'}\binom{n-i}{i'}\right) && (6.44)\\
&= \frac{x^i(x+1-x)^{n-i}}{n!} && (6.45)\\
&= \frac{x^i}{n!}. && (6.46)
\end{aligned}
$$

It is not hard to see that the state is normalised

$$
\begin{aligned}
\mathrm{Tr}[\rho_{n,x}] &= \sum_{i=0}^{n} x^i(1-x)^{n-i} \sum_{\substack{I\subseteq[n]\\|I|=i}} \mathrm{Tr}[\rho_I] && (6.47)\\
&= \sum_{i=0}^{n} x^i(1-x)^{n-i} \sum_{\substack{I\subseteq[n]\\|I|=i}} 1 && (6.48)\\
&= \sum_{i=0}^{n} x^i(1-x)^{n-i}\binom{n}{i} && (6.49)\\
&= (x+1-x)^n && (6.50)\\
&= 1. && (6.51)
\end{aligned}
$$

Thus this model of fixed pairwise distinguishability can be written as an expansion in terms of valid states, where indistinguishable photons are drawn from a binomial distribution.

6.3.2 Classical simulation

We can now see explicitly how a simulation for Boson Sampling with distinguishable photons would work. First, we sample an integer $i \in [n]$ according to the Binomial distribution with coefficients n and x. Next, we sample a subset I of the photons uniformly at random from the $\binom{n}{i}$ possible subsets of size i. These are the indistinguishable photons of our simulation, which we simulate using Clifford and Clifford in $O(i2^i + \mathrm{poly}(i, m))$ time. The remaining $n - i$ photons are considered to be distinguishable. Rather than needing to compute the output probabilities of these photons collectively, which could take between $O(n-i)^4 \log(n-i)$ and $O(n-i)^7 \log^4(n-i)$ time via permanents of matrices with non-negative entries [238], we can instead sample each distinguishable photon individually. To do so, we take a distinguishable photon in mode a, and compute the probability of this photon being measured in mode b as $|U_{b,a}|^2$. Thus we can compute all output probabilities and obtain a sample for a single distinguishable photon in $O(m)$ time, meaning that we can obtain a sample for all $n-i$ distinguishable photons in $O(m(n-i))$ time [188, 235].

The run time is dominated by the time taken to sample our indistinguishable photons, which can be as large as $O(n2^n + \mathrm{poly}(n,m))$ if we are unlucky. By truncating our binomial sampling up to some level k, we can simulate Boson Sampling up to some level of error. The extent of this error will be the focus of Secs. 6.4 & 6.5.

6.4 Average case error

We can use the same strategies used in [84, 85] to derive an error bound for Boson Sampling via state truncation for a Haar-random interferometer. We shall do this by considering the expected total variation distance between our approximation and the model for partial distinguishability for a Gaussian matrix. This is given by

$$\mathbb{E}[\Delta P] = \mathbb{E}\left[\frac{1}{2}\sum_{S'} |\Pr[S'] - \Pr_k[S']|\right] \tag{6.52}$$

$$= \frac{1}{2}\sum_{S'} \mathbb{E}\,|\Pr[S'] - \Pr_k[S']|, \tag{6.53}$$

where $\Pr_k$ is the probability distribution truncated at k indistinguishable photons via our approximation. For a specific outcome S', we can expand the right hand side to

$$\mathbb{E}\left[|\Pr[S'] - \Pr_k[S']|\right] = \mathbb{E}\left|\sum_{i=0}^{k}(p_i - p_i')\sum_{\substack{I\subseteq[n]\\|I|=i}} P_I[S'] + \sum_{i=k+1}^{n} p_i \sum_{\substack{I\subseteq[n]\\|I|=i}} P_I[S']\right|, \tag{6.54}$$

where P_I is now the probability distribution with indistinguishable photons defined by set I, and $p_i' = p_i/(\sum_{i=0}^{k} x^i(1-x)^{n-i}\binom{n}{i})$ is the normalised version of the p_i coefficients defined in Section 6.3. Note that P_I is the distribution arising from state ρ_I in Section 6.3. We can use the triangle inequality to bound this value to

$$\mathbb{E}\left[|\Pr[S'] - \Pr_k[S']|\right] \leq \mathbb{E}\left|\sum_{i=0}^{k}(p_i - p_i')\sum_{\substack{I\subseteq[n]\\|I|=i}} P_I[S']\right| + \mathbb{E}\left|\sum_{i=k+1}^{n} p_i \sum_{\substack{I\subseteq[n]\\|I|=i}} P_I[S']\right| \tag{6.55}$$

$$= \mathbb{E}[\Delta P_{\leq k}] + \mathbb{E}[\Delta P_{>k}], \tag{6.56}$$

where we have introduced variables $\Delta P_{\leq k}$ and $\Delta P_{>k}$ for convenience. We shall consider the expected values of these terms for a Gaussian matrix separately, starting with the latter. Using the Laplace expansion, we find that

$$\mathbb{E}[\Delta P_{>k}] = \sum_{i=k+1}^{n} p_i \sum_{\substack{I\subseteq[n]\\|I|=i}} \mathbb{E}\left[P_I[S']\right], \tag{6.57}$$

$$= \sum_{i=k+1}^{n} p_i \sum_{\substack{I\subseteq[n]\\|I|=i}} \sum_{\substack{J\leq S'\\|J|=i}} \mathbb{E}\left[|\operatorname{per}(U_{I,J})|^2 \operatorname{per}(|U_{\bar{I},\bar{J}}|^2)\right], \tag{6.58}$$

where $U_{I,J}$ is a matrix defined from our interferometer U by selecting columns according to I and rows according to J, and $|U_{\bar{I}\bar{J}}|^2$ is a matrix whose elements are the absolute values squared of $U_{\bar{I},\bar{J}}$.

We next need to consider the expected values of the matrix permanents in Equation (6.58) for a Haar random unitary. To do this, we shall assume the matrix describing our interferometer is Gaussian. This allows us to assume that each entry of $U_{I,J}$ and $|U_{\bar{I},\bar{J}}|^2$ is independent, and that the two matrices are independent of each other. Starting with $\mathrm{per}(|U_{\bar{I},\bar{J}}|^2)$, we note that this is a Gaussian matrix of size $(n-i)\times(n-i)$, and each entry is the square of two independent Gaussians, meaning that each entry of $|U_{\bar{I},\bar{J}}|^2$ has expected value $1/m$. From this, we can calculate the expected value as

$$\mathbb{E}[\mathrm{per}(|U_{\bar{I},\bar{J}}|^2)] = \frac{(n-i)!}{m^{n-i}}. \tag{6.59}$$

For $U_{I,J}$, we note that each element of $U_{I,J}$ is an independent Gaussian entry, with mean value 0 due to symmetry, and second order moment $\mathbb{E}[|U_{i,j}|^2] = 1/m$. The second order moment for the permanent can then be calculated using the same methods as in [71]:

$$\begin{aligned}
\mathbb{E}[|\,\mathrm{per}(U_{I,J})|^2] &= \mathbb{E}\left[\sum_{\sigma,\sigma'\in \mathrm{S}_i}\prod_{l=1}^{n}(U_{I,J})_{l,\sigma(l)}(U^*_{I,J})_{l,\sigma'(l)}\right] && (6.60)\\
&= \mathbb{E}\left[\sum_{\sigma\in \mathrm{S}_i}\prod_{l=1}^{n}|(U_{I,J})_{l,\sigma(l)}|^2\right] && (6.61)\\
&= \sum_{\sigma\in \mathrm{S}_i}\prod_{l=1}^{n}\mathbb{E}\left[|U_{l,\sigma(l)}|^2\right] && (6.62)\\
&= \frac{i!}{m^i}. && (6.63)
\end{aligned}$$

Because the two matrices are independent [84], we can express the expected value of their product as

$$\begin{aligned}
\mathbb{E}\left[|\,\mathrm{per}(U_{I,J})|^2\,\mathrm{per}(|U_{\bar{I},\bar{J}}|^2)\right] &= \frac{(n-i)!}{m^{n-i}}\times\frac{i!}{m^i} && (6.64)\\
&= \frac{(n-i)!i!}{m^n}. && (6.65)
\end{aligned}$$

Plugging this into Equation (6.58), we find that

$$\begin{aligned}
\mathbb{E}[\Delta P_{>k}] &= \sum_{i=k+1}^{n} p_i \sum_{\substack{I\subseteq[n]\\|I|=i}}\sum_{\substack{J\leq S'\\|J|=j}}\frac{(n-i)!i!}{m^n} && (6.66)\\
&= \sum_{i=k+1}^{n} p_i \binom{n}{i}^2 \frac{(n-i)!i!}{m^n} && (6.67)\\
&= \frac{n!}{m^n}\sum_{i=k+1}^{n} p_i\binom{n}{i} && (6.68)\\
&= \frac{n!}{m^n}\sum_{i=k+1}^{n} x^i(1-x)^{n-i}\binom{n}{i}, && (6.69)
\end{aligned}$$

where first we use the fact that since our input and output are both collision-free, there are $\binom{n}{i}$ ways of choosing I and J, then apply cancellation, and finally substitute the values of p_i. This gives our error bound for terms not in our approximation.

Next we shall consider the term $\Delta P_{\leq k}$. First we can note that $p_i' \geq p_i \forall i \leq k$, due to the normalisation of p_i'. Thus we can rewrite this term as

$$\mathbb{E}[\Delta P_{\leq k}] = \sum_{i=0}^{k} (p_i' - p_i) \sum_{\substack{I \subseteq [n] \\ |I|=i}} \mathbb{E}\left[P_I[S']\right]. \tag{6.70}$$

Using the same techniques for computing the Laplace expansion and calculating the expected value of permanents of Gaussian matrices, we can show that

$$\mathbb{E}[\Delta P_{\leq k}] = \sum_{i=0}^{k} (p_i' - p_i) \binom{n}{i}^2 \frac{(n-i)!i!}{m^n} \tag{6.71}$$

$$= \frac{n!}{m^n} \sum_{i=0}^{k} (p_i' - p_i) \binom{n}{i}. \tag{6.72}$$

Next we expand p_i':

$$\sum_{i=0}^{k} p_i' \binom{n}{i} = \frac{\sum_{i=0}^{k} x^i (1-x)^{n-i} \binom{n}{i}}{\sum_{i=0}^{k} x^i (1-x)^{n-i} \binom{n}{i}} \tag{6.73}$$

$$= 1, \tag{6.74}$$

and use this expansion as well as the value of p_i to calculate

$$\mathbb{E}[\Delta P_{\leq k}] = \frac{n!}{m^n} \left(1 - \sum_{i=0}^{k} x^i (1-x)^{n-i} \binom{n}{i} \right) \tag{6.75}$$

$$= \frac{n!}{m^n} \sum_{i=k+1}^{n} x^i (1-x)^{n-i} \binom{n}{i}. \tag{6.76}$$

We plug our values for Equations 6.72 & 6.76 into Equation 6.56 to bound our error for a single outcome as

$$\mathbb{E}\left[|\Pr[S'] - \Pr_k[S']|\right] \leq 2 \frac{n!}{m^n} \sum_{i=k+1}^{n} x^i (1-x)^{n-i} \binom{n}{i}. \tag{6.77}$$

Finally, we use these values and sum over all collision-free S', of which there are $\binom{m}{n} \approx m^n/n!$ to bound the expectation of our total variation distance for a Haar-random unitary as

$$\mathbb{E}[\Delta P] \leq \frac{1}{2} \sum_{S'} 2 \frac{n!}{m^n} \sum_{i=k+1}^{n} x^i (1-x)^{n-i} \binom{n}{i} \tag{6.78}$$

$$\approx \frac{m^n}{n!} \frac{n!}{m^n} \sum_{i=k+1}^{n} x^i (1-x)^{n-i} \binom{n}{i} \tag{6.79}$$

$$= \sum_{i=k+1}^{n} x^i (1-x)^{n-i} \binom{n}{i}. \tag{6.80}$$

It seems like there should be some room for improvement in this bound. In particular, the use of the triangle inequality suggests that there might be more precise approximations of the expected distance.

6.5 Worst case error

We shall now use a different technique to show that the above Haar average case error bound matches the worst-case error for any linear-optical interferometer. We do so by finding an upper bound for the trace distance between our ideal partially distinguishable state and the approximation that results from truncating at some k, the size of the largest indistinguishable set of particles. As the trace distance is an upper bound for any POVM measurement, we know that this will provide an upper bound for the difference in distribution produced by any interferometer.

Denoting our truncated state

$$\rho_{\leq k,x} = \frac{\sum_{i=0}^{k} x^i (1-x)^{n-i} \sum_{\substack{I\subseteq[n] \\ |I|=i}} \rho_I}{\sum_{i=0}^{k} x^i (1-x)^{n-i} \binom{n}{i}}, \tag{6.81}$$

where the denominator is a normalising factor, and similarly

$$\rho_{>k,x} = \frac{\sum_{i=k+1}^{n} x^i (1-x)^{n-i} \sum_{\substack{I\subseteq[n] \\ |I|=i}} \rho_I}{\sum_{i=k+1}^{n} x^i (1-x)^{n-i} \binom{n}{i}}. \tag{6.82}$$

We can now estimate the trace distance by rewriting $\rho_{n,x}$ as

$$\rho_{n,x} = \sum_{i=0}^{k} x^i (1-x)^{n-i} \binom{n}{i} \rho_{\leq k,x} + \sum_{i=k+1}^{n} x^i (1-x)^{n-i} \binom{n}{i} \rho_{>k,x}, \tag{6.83}$$

which, by convexity, gives

$$\delta_{\mathrm{tr}}(\rho_{n,x}, \rho_{\leq k,x}) \leq \sum_{i=k+1}^{n} x^i (1-x)^{n-i} \binom{n}{i}. \tag{6.84}$$

If we want this bound to be small, we can use techniques like those used for working out the tails of the binomial distribution. For example, it is known that for $nx < k < n$ that

$$\delta_{\mathrm{tr}}(\rho_{n,x}, \rho_{\leq k,x}) \leq \exp\left(-nD\left(\frac{k}{n}||x\right)\right), \tag{6.85}$$

where $D(k/n||x)$ is the relative entropy between coins with bias k/n and x, respectively [284]. Choosing a value of $k = n\alpha$ for $\alpha > x$ will give an error bound of $\exp(-nD(\alpha||x))$. Thus, choosing such a value for k would give an error that decreases as n increases, albeit at the cost of needing to increase k linearly with n. While this is still an exponential time algorithm, it would offer a polynomial speedup over the Clifford & Clifford method for Boson Sampling with indistinguishable photons.

If we are only interested in simulating up to a constant error, we can obtain a slightly better runtime. To do this, we use Hoeffding's inequality to find that

$$\delta_{\mathrm{tr}}(\rho_{n,x}, \rho_{\leq k,x}) \leq \exp\left(-2\frac{(k-nx)^2}{n}\right). \tag{6.86}$$

We can note that this value is at most ϵ if

$$k \geq nx + \sqrt{\frac{n\ln(1/\epsilon)}{2}}. \tag{6.87}$$

Our overall runtime is then $O(k^2 2^k + k^2 m)$. Asymptotically, this gives us a lower value of k than choosing $n\alpha$.

Finally, we note that the trace distance is only dependent on the initial states and not the measurement outcomes. As a result, this error bound also applies in the case where the output is not collision free.

6.6 Incorporating loss

We now consider how to adapt this simulation to Boson Sampling under uniform loss. We shall assume that each photon survives with probability η.

As discussed in Section 5.5, it was shown that the initial state for Boson Sampling with a fixed number of lost photons can be represented in the first quantisation as the initial state

$$\frac{1}{\binom{n}{\ell}} \sum_{\substack{L\subseteq[n]\\|L|=\ell}} \frac{1}{\ell!} \sum_{\sigma,\sigma'\in \mathrm{S}_n} \sigma \left|s_L\right\rangle \left\langle s_L\right| \sigma'^{\dagger}, \tag{6.88}$$

where s_L is the state where photons in the subset L of the original input photons have survived. In order to generalise this to uniform loss, we append $n-\ell$ "lost" photons in an additional spatial mode (single particle state 0) which isn't affected by the interferometer:

$$\left(\frac{1}{\binom{n}{\ell}} \sum_{\substack{L\subseteq[n]\\|L|=\ell}} \frac{1}{\ell!} \sum_{\sigma,\sigma'\in \mathrm{S}_n} \sigma \left|s\right\rangle \left\langle s\right| \sigma'^{\dagger}\right) \otimes \left(\left|0\right\rangle\left\langle 0\right|\right)^{\otimes n-\ell}. \tag{6.89}$$

Note that in the same way that it doesn't matter which particles are traced out when initially applying the loss, it similarly doesn't matter which particles are replaced with the $\left|0\right\rangle\left\langle 0\right|$ state. Uniform loss matches that of choosing which subset of photons survive according to the binomial distribution [86, 85]. We can combine this model with the distinguishability model of Section

6.3, giving

$$\begin{aligned}\rho_{n,\eta,x} &= \sum_{\ell=0}^{n} \eta^{\ell}(1-\eta)^{n-\ell} \sum_{\substack{L\subseteq[n]\\|L|=\ell}} \sum_{i=0}^{\ell} x^i(1-x)^{\ell-i} \\ &\quad \times \sum_{\substack{I\subseteq L\\|I|=i}} \left(\frac{1}{\ell!} \sum_{\substack{\sigma,\sigma'\in \mathrm{S}_n\\ \sigma^{-1}(j)=\sigma'^{-1}(j)\\ \forall j\notin I}} \sigma\left|s\right\rangle\left\langle s\right|\sigma'^{\dagger} \right) \otimes (\left|0\right\rangle\left\langle 0\right|)^{\otimes n-\ell} \qquad (6.90)\\ &= \sum_{\ell=0}^{n} \eta^{\ell}(1-\eta)^{n-\ell} \sum_{\substack{L\subseteq[n]\\|L|=\ell}} \sum_{i=0}^{\ell} x^i(1-x)^{\ell-i} \sum_{\substack{I\subseteq L\\|I|=i}} \rho_{L,I}. \qquad (6.91)\end{aligned}$$

We can now see how our classical simulation for Boson Sampling under distinguishability can be adapted to accommodate loss as well. First, we choose a subset of photons L to indicate the photons that were not lost. From this subset, we choose another subset of photons $I \subseteq L$ to indicate the indistinguishable photons, which are simulated via the Clifford & Clifford algorithm. The photons in $L \setminus I$ are all distinguishable photons, and can be simulated classically as before. The classical complexity of this algorithm depends on the number of indistinguishable photons we choose. As in Section 6.3, by truncating this to be some maximum size k we can get an algorithm that runs in $O(k2^k + \mathrm{poly}(k,m))$ time.

To understand the precision of this algorithm, we first note that if $|L| \leq k$, then we can classically simulate any number of indistinguishable photons within our desired runtime. As a result, we only need to truncate when $|L| > k$, and only need to do so up to $|I| \leq k$.

As with Section 6.4, we start by considering the error bound for a random interferometer. We note that in cases where at most k photons survive our approximation is exact, so these outcomes do not contribute to our total variation distance. For the remainder, we see that

$$\mathbb{E}[\Delta P] = \sum_{\ell=k+1}^{n} \eta^{\ell}(1-\eta)^{n-\ell} \sum_{\substack{L\subseteq[n]\\|L|=\ell}} \mathbb{E}[\Delta P_L], \qquad (6.92)$$

where ΔP_L denotes the error of our simulation with photons in input modes denoted by L. Using our bound in Section 6.4 as well as the rule of conditional binomial distributions (see below), we can bound this as

$$\mathbb{E}[\Delta P] \leq \sum_{i=k+1}^{n} (\eta x)^i (1-\eta x)^{n-i} \binom{n}{i}. \qquad (6.93)$$

Again, an improvement over the use of the triangle inequality can potentially lead to an improvement in this bound.

For the worst-case error, we construct analogous states to those in Section 6.5:

$$\rho_{\leq k,\eta,x} = \frac{\sum_{\ell=0}^{n} \eta^{\ell}(1-\eta)^{n-\ell} \sum_{\substack{L\subseteq[n]\\|L|=\ell}} \sum_{i=0}^{\min(\ell,k)} x^i(1-x)^{\ell-i} \sum_{\substack{I\subseteq L\\|I|=i}} \rho_{L,I}}{\sum_{\ell=0}^{n} \eta^{\ell}(1-\eta)^{n-\ell}\binom{n}{\ell} \sum_{i=0}^{\min(\ell,k)} x^i(1-x)^{\ell-i}\binom{\ell}{i}}, \qquad (6.94)$$

$$\rho_{>k,\eta,x} = \frac{\sum_{\ell=k+1}^{n} \eta^{\ell}(1-\eta)^{n-\ell} \sum_{\substack{L\subseteq[n]\\|L|=\ell}} \sum_{i=k+1}^{\ell} x^i(1-x)^{\ell-i} \sum_{\substack{I\subseteq L\\|I|=i}} \rho_{L,I}}{\sum_{\ell=k+1}^{n} \eta^{\ell}(1-\eta)^{n-\ell}\binom{n}{\ell} \sum_{i=k+1}^{\ell} x^i(1-x)^{\ell-i}\binom{\ell}{i}}, \qquad (6.95)$$

and note that $\rho_{n,\eta,x}$ is a linear combination of these states. As a result, the worst-case error of this simulation, using the convex properties of the trace distance, can be bounded as

$$\begin{aligned}\delta_{\mathrm{tr}}(\rho_{n,\eta,x}, \rho_{\leq k,\eta,x}) &\leq \sum_{\ell=k+1}^{n} \eta^{\ell}(1-\eta)^{n-\ell}\binom{n}{\ell} \times \sum_{i=k+1}^{\ell} x^{i}(1-x)^{\ell-i}\binom{\ell}{i} &(6.96)\\ &= \sum_{i=k+1}^{n} (\eta x)^{i}(1-\eta x)^{n-i}\binom{n}{i}, &(6.97)\end{aligned}$$

where in the second line we have used the rule of conditional binomial distributions. Using the same result as used for Equation (6.85), we can bound the error to a value decreasing in n by setting $k = n\beta$ for $\beta > \eta x$ [284].

Similarly, we can use Hoeffding's inequality as in Equation 6.86 to show that for a constant error it suffices to have

$$k \geq n\eta x + \sqrt{\frac{n\ln(1/\epsilon)}{2}}. \tag{6.98}$$

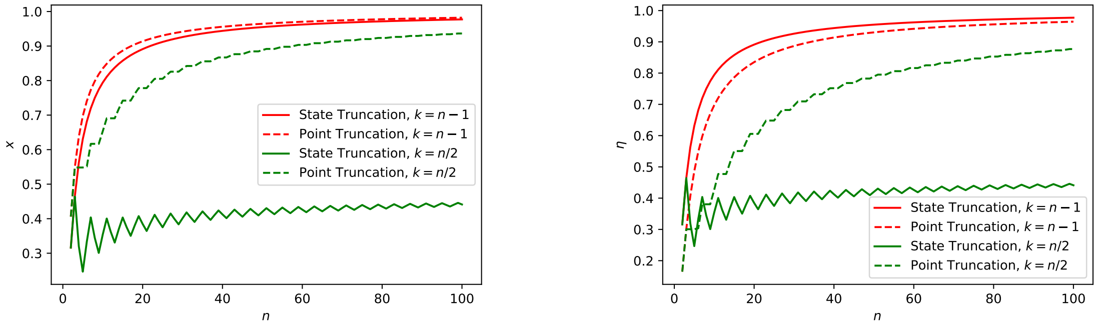
This shows a relationship between distinguishability and loss similarly to, but not exactly the same, as the one found in [85]: the more distinguishable photons are, the more we can classically simulate photon loss, and vice versa. It is remarkable to see that these two algorithms have a different dependence on x and η: while state truncation depends on ηx, point truncation depends on ηx^2. It is not immediately clear where this difference comes from, and we leave it as an open question.

6.7 Empirical errors

A natural question at this point is how to assess the performance of this new approach over that of [84, 85]. It is not immediately clear how to find a fair comparison, as each approach has its own strengths and weaknesses. Truncating based on fixed points has the benefit of the error asymptotically tending towards a constant as n increases, which means that k can be chosen independently of n and does not need to increase. But this comes at the cost of a potentially large, albeit polynomial, runtime of at least $O(n^{2k}k2^k(n-k)^4\log(n-k))$ [84, 238]. Truncating based on states, on the other hand, provides a significant improvement in runtime based on k, and is able to run in $O(2^k + \mathrm{poly}(k,m,n))$ time, but at the cost of k increasing linearly with n for constant error.

We therefore consider a variety of comparisons. In Section 6.7.1, we start by considering the highest value of x and η simulable by each approach when given the same values of n and k for a fixed error rate. We then introduce the runtime for each algorithm in Section 6.7.2, by comparing how fast they can simulate particular values of x and η for increasing n. Finally in Section 6.7.3, we compare the highest value of x and η simulable by both algorithms for a 90-photon Boson Sampling experiment, where k is varying but under the condition that the algorithms have similar run times. The motivation for this is that 90 photons has been suggested as strict upper bound for what is achievable using classical computation [162].

Before we go further, we make a few observations on the calculations of error bounds and runtimes used in this section. Rather than using the asymptotic error bounds for fixed point truncation, which assume $n \to \infty$, we have used bounds for finite n. This provides an improvement in the error of up to $1/\sqrt{e}$. For the runtime of fixed point truncation, we explicitly calculate $\sum_{i=0}^{k}\binom{n}{i}R(n,n-i)i2^i(n-i)^4\log(n-i)$, where $R(n,n-i) = \binom{n}{i}\lceil i!/e \rfloor$ is the number



(a)

(b)

Figure 6.1: Highest value (6.1a) of x when $\eta = 1$ and (6.1b) of η when $x = 1$ simulable via state (solid) or point (dashed) truncation up to 10% error ($\epsilon = 0.1$). The number of photons, n, is varying, with k chosen as either $k = n - 1$ (red) or $k = n/2$ (green). The oscillatory behaviour is due to rounding $k = n/2$ when n is odd.

of permutations with $n - i$ fixed points, and we have assumed that computing every permanent of the $(n - i) \times (n - i)$ matrices with non-negative entries requires $O((n - i)^4 \log(n - i))$ time [1]. For Metropolised Independence Sampling, we choose the number of probabilities to approximate via state truncation as 100, which matches currently used “burn in” and “thinning” times[235], though it is worth noting that this number could be different depending on the distribution of partially-distinguishable and lossy bosons. For the runtime of state truncation, we use $2k2^k + mk(k-1)/2 + m(n-k)$, where we have used the fact that the first term is approximately equivalent to computing two matrix permanents, the second term is the polynomial overhead of the Clifford & Clifford approach, and the final term is the polynomial overhead of sampling the fully distinguishable photons [91]. For these calculations, we have also assumed that there are $m = n^2$ spatial modes [2].

Finally, we note that in Secs. 6.7.1 and 6.7.3, we only consider distinguishability and loss separately, by comparing the highest value of x simulable in a lossless system, and the highest value of η simulated in a full indistinguishable system. Ideally one would compare highest combinations of η and x which are classically simulable. However, doing so is complicated by the fact that both methods handle combinations of noise differently: point truncation handles them as the parameter ηx^2, whereas state truncation handles them as ηx. With this in mind, we plot values for distinguishability and loss separately, and note that, for the same performance, a reduction in one implies an increase in the other.

6.7.1 Comparison at same level of truncation

We start by comparing the performance of the two algorithms when truncated at the same level k. This is of interest as in both approaches k is considered to be a parameter defining the

[1] Note that this computation could in the worst case take $O(n-i)^7 \log^4(n-i)$ time, depending on the matrix sparsity [238].

[2] Note that $m \in O(n^2)$ is only sufficient to ensure that the probability of seeing collisions from a Haar-random interferometer are small [186]. The classical hardness of Boson Sampling is also dependent on entries of U being drawn independently with high probability. To ensure this, m could be required to be as large as n^6 [71]. However, it is widely believed, and often referenced in Boson Sampling experiments, that $m = n^2$ should be sufficient [71, 72, 73, 74, 75, 76, 77, 191].

interference between photons. To do so, we consider the error bounds of classically simulating n-photon Boson Sampling for n ranging between 2 and 100. The values chosen for k depend on n: we consider $k = n - 1$ as the upper limit of what the two algorithms can achieve without simulating the full distribution, and also $k = n/2$ as a more feasible, though still exponential time, value.

The result is plotted in Figure 6.1, where in (6.1a) we show the highest value of x simulable assuming no loss ($\eta = 1$) and in (6.1b) we show the highest value of η simulable assuming the photons are fully indistinguishable ($x = 1$). For all cases, we are considering simulations up to 10% error.

There are a number of things we can note from Figure 6.1. First is that when $k = n - 1$, we can see that both algorithms tend to the same maximum values of distinguishability and loss. In the case of distinguishability, we can easily see why by considering the error bounds of both algorithms. One can see from Equation (6.84) that state truncation will have a simple error bound in this case of $\epsilon \leq x^n$, meaning that for constant error the largest value of x simulable is $x = \epsilon^{1/n}$. For point truncation, recall from Section 4.6.4 that the error is bounded as

$$\begin{aligned}\mathbb{E}[\Delta P] &\leq \frac{m^n}{n!}\sqrt{\mathrm{Var}[\Delta P]} && (6.99)\\ &\leq \frac{m^n}{n!}\sqrt{\sum_{j=k+1}^{n} x^{2j}\,\mathrm{Var}[c_j]} && (6.100)\\ &\leq \frac{1}{\sqrt{e}}\sqrt{\sum_{j=k+1}^{n} x^{2j}\sum_{l=0}^{n-j}\frac{1}{l!}}. && (6.101)\end{aligned}$$

Setting $k = n-1$ shows that the error is similarly bounded as $\epsilon \leq x^n/\sqrt{e}$, leading to a largest value of $x = (\epsilon\sqrt{e})^{1/n}$. Thus, although the highest value of x simulable via point truncation is higher than that via state truncation, the difference will decrease in the limit of large n. Curiously we see the same effect as well in the case of loss, but now the highest value of η simulable via state truncation is higher than that of point truncation. Again, this can be shown to hold theoretically: For state truncation the error scales as $\epsilon \leq \eta^n$ according to Equation (6.97), corresponding to $\eta = \epsilon^{1/n}$; whereas for point truncation we see from Equation (6.101) and substituting $x = \sqrt{\eta}$ that the error scales as $\epsilon \leq \eta^{n/2}/\sqrt{e}$, meaning a maximum value of η is $\eta = (e\epsilon^2)^{1/n}$. In the limit of large n these differences will also tail off.

For $k = n/2$, we see that for both distinguishability and loss point truncation is more powerful than state truncation. Although this is harder to formally prove, there is intuition to see why this is the case. For state truncation, we know that for a small error to be achievable we need $k \geq n\eta x$, as this is the mean of the binomial distribution. Thus for $k = n/2$, we have that $\eta x \leq 1/2$, and in both cases we see the highest value of x and η tending to a value below 1/2. For point truncation on the other hand, we know that the error tends to a constant value only dependent on k and ηx^2 in the limit of large n. As a result, it is unsurprising that for k increasing linearly with n the highest values of x and η will increase.

6.7.2 Comparison of runtimes

We next consider the runtime required to simulate n-photon Boson Sampling up to 10% error via either method. The motivation for this comparison is that the runtime of the two algorithms at the same value of k are significantly different. In particular, the runtime of state truncation is

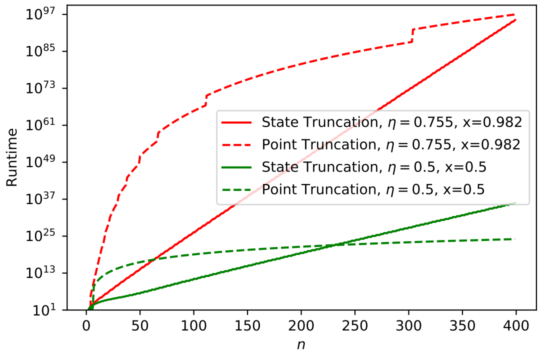


Figure 6.2: Approximate runtime (number of operations) to simulate n-photon Boson Sampling with chosen values of η and x up to 10% error ($\epsilon = 0.1$) via state (solid) or point (dashed) truncation. Note that a modern supercomputer running for an hour can perform roughly 10^{20} operations.

only dependent on k and not n, whereas the runtime for point truncation depends on a scaling of approximately $O(n^{2k})$.

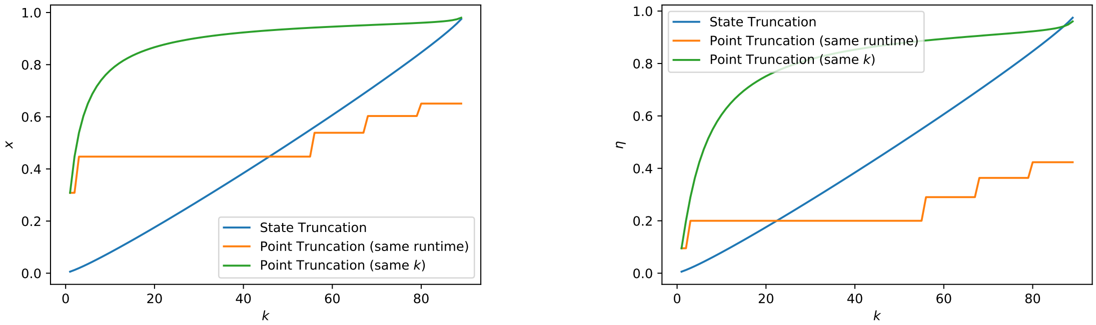
To understand how the runtimes scale, in Figure 6.2 we plot the runtimes of classically simulating n-photon Boson Sampling experiments via the two approaches for fixed values of η and x. The values of k chosen for each algorithm are the smallest values for an error of at most 10%. For choosing η and x, we give two example cases. The first (Figure 6.2, red), where $\eta = 0.755$ and $x = 0.982$, is an example of a hypothetical best experiment we could build with current technology, with the most lossless sources (82%) [285], interferometers (99%) [226] and detectors (93%) [286], and the highest level of photon indistinguishability (98%) [287]. The second case (Figure 6.2, green), where $\eta = x = 0.5$, is an example of how the two algorithms perform in what would be considered a poor experiment for both distinguishability and loss. Actual Boson Sampling experiments are likely to fall between these two extremes.

In both cases, state truncation appears to outperform point truncation for near-term photon experiments, with point truncation eventually being able to perform faster for larger values of n. When $\eta = x = 0.5$, point truncation performs better when n is approximately larger than 230. In the case of $\eta = 0.755, x = 0.982$, point truncation doesn't perform better for any $n < 400$ photons. This gives an idea of the regions in which the polynomial runtime of point truncation can be better or worse than the exponential runtime of state truncation.

It is also worth noting that just because point truncation is faster than state truncation for large enough n does not necessarily mean that either algorithm is efficient in these cases. When $\eta = x = 0.5$, point truncation only becomes more efficient at instances where both algorithms already require the order of 10^{22} operations. And in the case where $\eta = 0.755, x = 0.982$, both algorithms have runtimes of approximately 10^{96} operations while still not reaching a sufficiently large n for point truncation to outperform state truncation.

6.7.3 Comparison at same runtime

Now we consider both truncation level and runtime, and compare the algorithms when restricted to comparable runtimes. To do this, we shall consider the challenge of simulating a 90-photon Boson Sampling experiment, and the largest values of distinguishability and loss that can be



(a)

(b)

Figure 6.3: Highest value (6.3a) of x when $\eta = 1$ and (6.3b) of η when $x = 1$ simulable for 90-photon boson sampling at truncation level k up to 10% error ($\epsilon = 0.1$). Blue line indicates highest values simulable via state truncation at level k, green lines indicate highest values simulable via point truncation at level k, orange lines indicate highest values simulable by point truncation at level k' such that k' is the smallest level of truncation such that the approximate runtime of point truncation at level k' is longer than the runtime of state truncation at level k.

simulated at level k with 10% error. The motivation for this is that 90 photons has been suggested as strict upper bound for what is achievable using classical computation [162].

The results are shown in Figure 6.3, detailing for each algorithm the highest value of x simulable when $\eta = 1$ (6.3a) and the highest value of η simulable when $x = 1$ (6.3b). In both figures, the blue line indicates state truncation at level k, and the green line indicates point truncation at level k. However, the runtime of state truncation at level k and point truncation at level k are likely to be drastically different. To take runtime into consideration as well, we consider the orange line which indicates point truncation at level k', where k' is the smallest integer such that the approximated runtime of point truncation at level k' is longer than that of state truncation at level k. This allows us to compare the performance of the two algorithms when restricted to similar runtimes.

Considering distinguishability in Figure 6.3a, we can note that point truncation with comparable runtime performs better up to $k \leq 45$, after which the methods are roughly comparable with state truncation performing marginally better, before becoming more dominant for $k \geq 60$. It has been suggested that boson sampling with 50 indistinguishable photons is roughly the limit of what can be classically simulated on a supercomputer [235, 91, 288], so it appears that when considering distinguishability, the algorithms are roughly comparable in this case.

When considering loss in Figure 6.3b, we see a noticeable improvement for state truncation. Now point truncation under the same runtime only performs better up to $k \leq 22$, with state truncation performing considerably better for higher values of k. Boson Sampling with up to 30 indistinguishable photons is already known to be classically simulable on a standard laptop [235], so this appears to offer a noticeable improvement even for fast classical simulations.

6.8 Non-uniform loss

We finish by briefly considering non-uniform loss, where each photon survives a lossy optical component with probability τ. This model of loss has been considered before [83, 86], but without the incorporation of distinguishability. We can do this using the same methods as other non-uniform loss results, by extracting non-uniform losses into a layer of uniform losses followed by a lossy interferometer. The uniform loss layer means that each photon has probability $\eta = \tau^s$ of surviving, where τ is the loss of each optical component and s is the smallest number of lossy optical components a photon interacts with. If we take the total number of lossy components to be d, the remaining lossy circuit can be modelled as an $(m+d)$-mode interferometer, with lost photons ending up in the additional d modes. Thus we can achieve the same error as Equation (6.97) in $O(k2^k + \text{poly}(k, m, d))$ time. In typical schemes for linear interferometers, d is at most polynomial in m [176, 177], so the overhead from these additional modes is small. We can bound the error to a decreasing value in terms of n if $k = n\gamma$ for $\gamma > x\tau^s$. Taking the logarithm on both sides and rearranging for s, we find that this holds if

$$s > \frac{\log(1/\gamma) - \log 1/x}{\log 1/\tau}. \tag{6.102}$$

This suggests simulability of even constant depth Boson Sampling circuits, but requires our algorithm's runtime to increase linearly with n. If we are only interested in simulating up to a constant error, we can use Hoeffding's inequality again to find that we need

$$k \geq nx\tau^s + \sqrt{\frac{n\ln(1/\epsilon)}{2}} \tag{6.103}$$

for our simulator to be ϵ-close. Rearranging for s gives us that this holds if

$$s\log(1/\tau) \geq \frac{3\log n}{2} + \frac{\log(\ln(1/\epsilon))}{2} - \log k - \log(1/x) - \frac{\log 2}{2}. \tag{6.104}$$

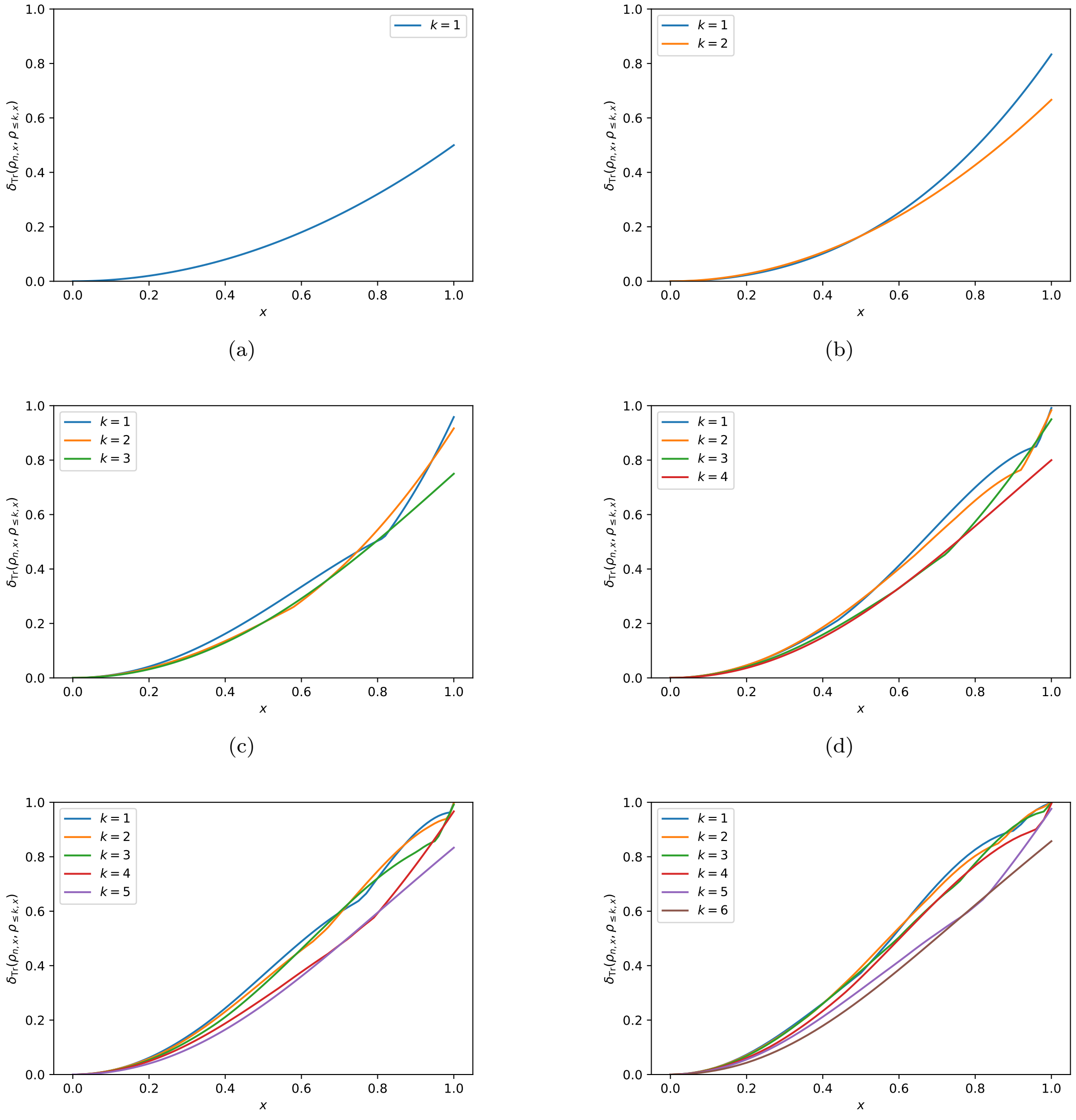
Thus we can simulate Boson Sampling up to accuracy ϵ under non-uniform loss in $O(k^2 2^k + mk^2)$ time if our circuit depth is

$$s = \Omega\left(\frac{\log n + \log\log(1/\epsilon) - \log(1/x) - \log k}{\log(1/\tau)}\right). \tag{6.105}$$

Note that even if $k = O(\log n)$, which would imply a polynomial runtime for our simulator, this bound holds if the circuit depth is at least logarithmic in terms of n. This matches results in [83, 86], showing that boson sampling can be classically simulated if each photon encounters at least a logarithmic number of lossy components. It also shows how distinguishability can affect the simulability of lossy components in Boson Sampling: if our photons are more distinguishable, corresponding to a smaller value of x, then we can simulate shallower (i.e. less total loss) optical circuits. Finally it is worth emphasising that our best-known universal interferometers, such as the ones mentioned in Section 4.3.5, require at least linear depth in terms of m, which translates to $O(n^2)$ depth overall.

6.9 Discussion of possible improvements

There are several possible ways one might seek to improve this research. In this section we shall give two potential improvements, and discuss their potential. Although we do not provide full analysis for general n, we can look at cases where there are a small number of photons and use this as a springboard for discussing how these improvements might work for general n.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 6.4: Exact trace distance betweem $\rho_{n,x}$ and $\rho_{\leq k,x}$ over x for all $k < n$ for (6.4a) $n = 2$, (6.4b) $n = 3$, (6.4c) $n = 4$, (6.4d) $n = 5$, (6.4e) $n = 6$ and (6.4f) $n = 7$.

	k		
n	1	2	3
2	$x^2/2$		
3	$x^2(3+2x)/6$	$2x^2/3$	
4	$x^2(30+16x-9x^2+9\lvert 2-3x^2\rvert)/48$	$x^2(18+4x-6x^2+3\lvert 1-3x^2\rvert+\lvert 1-4x+3x^2\rvert)/24$	$x^2(3+x-x^2)/4$

Table 6.1: Exact equations for $\delta_{\mathrm{tr}}(\rho_{n,x}, \rho_{\leq k,x})$ for $k < n$ and $n = 2, 3, 4$.

6.9.1 Computing the exact trace distance

In our analysis of the worst-case error bounds, we have simply used the property of the trace distance being convex. However, this is only an upper bound, and it could be that the actual trace distance is lower. How much can we improve this bound by more direct calculation?

Sadly there is not an easy way of calculating the exact trace distance, as we have managed to do for the fixed distinguishability cases in Section 6.2.2. In lieu of this, we use analytical code to compute the trace distance for small numbers of photons. Using symbolic programming in Mathematica, we are able to compute the exact trace distance in terms of x for varying k and up to four photons. Furthermore, we can use analytical programming in Matlab to produce figures plotting the trace distance over x for varying values of k and up to seven photons.

The formulae computed via Mathematica are given in Table 6.1, and the results from Matlab computation are given in Figure 6.4, with each subfigure showing the trace distance for varying n and k. It is very challenging to discern a general pattern here, unlike the fixed distinguishability cases discuseed in Section 6.2.2; the formulae produced by Mathematica start looking complicated even for $n = 4$. Larger values of n were attempted in both Matlab and Mathematica, but had to be abandoned due to computation time.

We can see that for high values of x all of these approximations start to fail. Part of the reason for this is how few photons we have; if on average we expect nx photons then for a good approximation with $k < n$ we would expect that $x < (n-1)/n$, which will be fairly low with these small values of n.

It is interesting to note from Figure 6.4 that for some values of x there are some values of k where $\delta_{\mathrm{tr}}(\rho_{n,x}, \rho_{\leq k,x}) < \delta_{\mathrm{tr}}(\rho_{n,x}, \rho_{\leq k+1,x})$. This is very surprising; we would expect that the trace distance would decrease for all x as k increases, as this would correspond to more indistinguishable photons. We do not know why this is and leave it as an open question for future research. It is however worth emphasising in these cases that the trace distance is an upper bound, and it might be that in practice interferometers which achieve this perfect maximisation of trace distance do not exist. We again leave this as a direction for future research, but note that this is similar to the question explored by Stanisic and Turner, who looked at this in the fixed distinguishability case [222].

6.9.2 Expanding in terms of representations

Another natural question is if we can instead expand in terms of representations rather than in terms of state. This would provide us with an alternative decomposition where rather than needing to sample which photons are indistinguishable, we would instead be sampling which immanants we need to compute to find the probability of an outcome. It is hard to analytically work out how the different immanants contribute for general n, but in this section we will use results noted by Stanisic and Turner [222] to show the decompositions for two and three particles, as well as offer some general comments on why this technique might not necessarily provide a better classical simulation but could still be of theoretical interest.

We shall start with the simpler case of two photons, which we shall assume to be in the first two spatial modes. As noted earlier, the state takes the form

$$\rho_x = x^2\rho_{\{1,2\}} + x(1-x)(\rho_{\{1\}} + \rho_{\{2\}}) + (1-x)^2\rho_\emptyset, \tag{6.106}$$

where ρ_I is the state corresponding to photons in modes $I \subseteq \{1,2\}$ being indistinguishable. The state corresponding to the fully indistinguishable case is fully symmetric, giving us

$$\rho_{\{1,2\}} = \left|\begin{array}{|c|c|}\hline 1 & 2 \\ \hline\end{array}\right\rangle \left\langle\begin{array}{|c|c|}\hline 1 & 2 \\ \hline\end{array}\right|, \tag{6.107}$$

where we have suppressed notation for the irrep as it is implied by the Young tableaux, and we have suppressed the symmetric irrep basis as it is one dimensional in the two particle cases. Note that states where only one photon is in the indistinguishable set are effectively fully distinguishable as well. As a result, the states $\rho_{\{1\}}, \rho_{\{2\}}$ are equal to $\rho_\emptyset$, the fully distinguishable state, which as described in the irrep basis in Equation 5.36 gives us

$$\rho_\emptyset = \frac{1}{2}\left(\left|\begin{array}{|c|c|}\hline 1 & 2 \\ \hline\end{array}\right\rangle \left\langle\begin{array}{|c|c|}\hline 1 & 2 \\ \hline\end{array}\right| + \left|\begin{array}{|c|}\hline 1 \\ \hline 2 \\ \hline\end{array}\right\rangle \left\langle\begin{array}{|c|}\hline 1 \\ \hline 2 \\ \hline\end{array}\right|\right). \tag{6.108}$$

The overall state can therefore be written as

$$\rho_x = x^2 \left|\begin{array}{|c|c|}\hline 1 & 2 \\ \hline\end{array}\right\rangle \left\langle\begin{array}{|c|c|}\hline 1 & 2 \\ \hline\end{array}\right| + \frac{(2x(1-x) + (1-x)^2)}{2}\left(\left|\begin{array}{|c|c|}\hline 1 & 2 \\ \hline\end{array}\right\rangle \left\langle\begin{array}{|c|c|}\hline 1 & 2 \\ \hline\end{array}\right| + \left|\begin{array}{|c|}\hline 1 \\ \hline 2 \\ \hline\end{array}\right\rangle \left\langle\begin{array}{|c|}\hline 1 \\ \hline 2 \\ \hline\end{array}\right|\right) \tag{6.109}$$

$$= \frac{1}{2}\left((2x^2 + 2x(1-x) + (1-x)^2)\left|\begin{array}{|c|c|}\hline 1 & 2 \\ \hline\end{array}\right\rangle \left\langle\begin{array}{|c|c|}\hline 1 & 2 \\ \hline\end{array}\right| + (2x(1-x) + (1-x)^2)\left|\begin{array}{|c|}\hline 1 \\ \hline 2 \\ \hline\end{array}\right\rangle \left\langle\begin{array}{|c|}\hline 1 \\ \hline 2 \\ \hline\end{array}\right|\right) \tag{6.110}$$

$$= \frac{1+x^2}{2}\left|\begin{array}{|c|c|}\hline 1 & 2 \\ \hline\end{array}\right\rangle \left\langle\begin{array}{|c|c|}\hline 1 & 2 \\ \hline\end{array}\right| + \frac{1-x^2}{2}\left|\begin{array}{|c|}\hline 1 \\ \hline 2 \\ \hline\end{array}\right\rangle \left\langle\begin{array}{|c|}\hline 1 \\ \hline 2 \\ \hline\end{array}\right|. \tag{6.111}$$

As $x \to 1$, this tends towards only the fully symmetric irrep contributing, as expected. However, even as $x \to 0$, the fully symmetric irrep still contributes to half of the outcome probability.

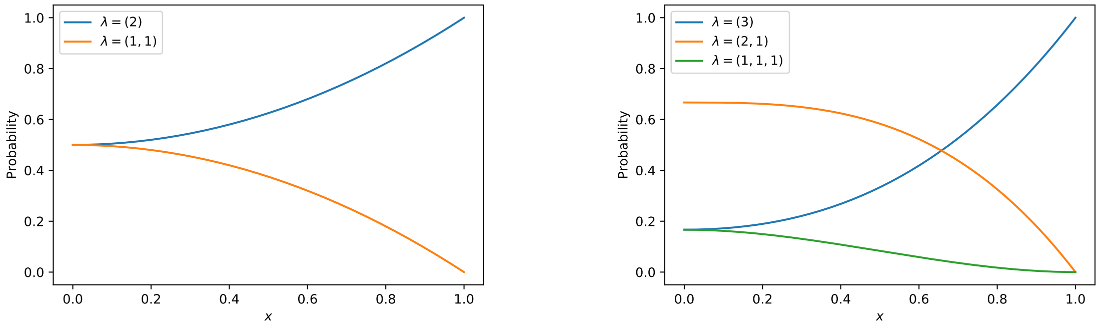
We can also work out the explicit decomposition for three particles, again assuming our photons start in the first three modes. The state is now

$$\rho_x = x^3\rho_{\{1,2,3\}} + x^2(1-x)(\rho_{\{1,2\}} + \rho_{\{1,3\}} + \rho_{\{2,3\}}) + x(1-x)^2(\rho_{\{1\}} + \rho_{\{2\}} + \rho_{\{3\}}) + (1-x)^3\rho_\emptyset. \tag{6.112}$$

The fully indistinguishable state is similar to before:

$$\rho_{\{1,2,3\}} = \left|\begin{array}{|c|c|c|}\hline 1 & 2 & 3 \\ \hline\end{array}\right\rangle \left\langle\begin{array}{|c|c|c|}\hline 1 & 2 & 3 \\ \hline\end{array}\right|. \tag{6.113}$$

Rather than writing out the full Young-Yagamouchi basis, we shall simply use a 1 or 2 subscript to denote the basis state of the $(1,2)$ irrep of the symmetric group, which, as noted in Theorem 4.1, induces a multiplicity on the corresponding irrep of the unitary group. This matches the notation used in Stanisic and Turner [222]. With this in mind, the uniform mixture of all three singly distinguishable states is



(a)

(b)

Figure 6.5: Probability of different irreps when (6.5a) $n = 2$ and (6.5b) $n = 3$. Note that if for the $\lambda = (2, 1)$ irrep in Figure 6.5b, we have shown the probability of sampling any of the multiplicities, when in reality this is a uniform distribution of the four multiplicities.

$$\begin{aligned}\rho_{\{1,2\}} + \rho_{\{1,3\}} + \rho_{\{2,3\}} = {}& \left|\begin{array}{|c|c|c|}\hline 1&2&3\\\hline\end{array}\right\rangle\left\langle\begin{array}{|c|c|c|}\hline 1&2&3\\\hline\end{array}\right| + \frac{1}{2}\left(\left|\begin{array}{|c|c|}\hline 1&2\\\hline 3\\\cline{1-1}\end{array}_1\right\rangle\left\langle\begin{array}{|c|c|}\hline 1&2\\\hline 3\\\cline{1-1}\end{array}_1\right| + \left|\begin{array}{|c|c|}\hline 1&2\\\hline 3\\\cline{1-1}\end{array}_2\right\rangle\left\langle\begin{array}{|c|c|}\hline 1&2\\\hline 3\\\cline{1-1}\end{array}_2\right|\right)\\ &+ \frac{1}{2}\left(\left|\begin{array}{|c|c|}\hline 1&3\\\hline 2\\\cline{1-1}\end{array}_1\right\rangle\left\langle\begin{array}{|c|c|}\hline 1&3\\\hline 2\\\cline{1-1}\end{array}_1\right| + \left|\begin{array}{|c|c|}\hline 1&3\\\hline 2\\\cline{1-1}\end{array}_2\right\rangle\left\langle\begin{array}{|c|c|}\hline 1&3\\\hline 2\\\cline{1-1}\end{array}_2\right|\right).\end{aligned} \tag{6.114}$$

Finally, we note that the fully distinguishable states $\rho_\emptyset, \rho_{\{1\}}, \rho_{\{2\}}, \rho_{\{3\}}$ can all be written as

$$\begin{aligned}\rho_\emptyset = {}& \frac{1}{6}\left(\left|\begin{array}{|c|c|c|}\hline 1&2&3\\\hline\end{array}\right\rangle\left\langle\begin{array}{|c|c|c|}\hline 1&2&3\\\hline\end{array}\right| + \left|\begin{array}{|c|c|}\hline 1&2\\\hline 3\\\cline{1-1}\end{array}_1\right\rangle\left\langle\begin{array}{|c|c|}\hline 1&2\\\hline 3\\\cline{1-1}\end{array}_1\right| + \left|\begin{array}{|c|c|}\hline 1&2\\\hline 3\\\cline{1-1}\end{array}_2\right\rangle\left\langle\begin{array}{|c|c|}\hline 1&2\\\hline 3\\\cline{1-1}\end{array}_2\right|\right)\\ &+ \frac{1}{6}\left(\left|\begin{array}{|c|c|}\hline 1&3\\\hline 2\\\cline{1-1}\end{array}_1\right\rangle\left\langle\begin{array}{|c|c|}\hline 1&3\\\hline 2\\\cline{1-1}\end{array}_1\right| + \left|\begin{array}{|c|c|}\hline 1&3\\\hline 2\\\cline{1-1}\end{array}_2\right\rangle\left\langle\begin{array}{|c|c|}\hline 1&3\\\hline 2\\\cline{1-1}\end{array}_2\right| + \left|\begin{array}{|c|}\hline 1\\\hline 2\\\hline 3\\\hline\end{array}\right\rangle\left\langle\begin{array}{|c|}\hline 1\\\hline 2\\\hline 3\\\hline\end{array}\right|\right).\end{aligned} \tag{6.115}$$

The overall state can be expanded as

$$\begin{aligned}\rho_x = {}& \frac{1}{6}\left((1+3x^2+2x^3)\left|\begin{array}{|c|c|c|}\hline 1&2&3\\\hline\end{array}\right\rangle\left\langle\begin{array}{|c|c|c|}\hline 1&2&3\\\hline\end{array}\right|\right)\\ &+ \frac{1}{6}\left((1-x^3)\left(\left|\begin{array}{|c|c|}\hline 1&2\\\hline 3\\\cline{1-1}\end{array}_1\right\rangle\left\langle\begin{array}{|c|c|}\hline 1&2\\\hline 3\\\cline{1-1}\end{array}_1\right| + \left|\begin{array}{|c|c|}\hline 1&2\\\hline 3\\\cline{1-1}\end{array}_2\right\rangle\left\langle\begin{array}{|c|c|}\hline 1&2\\\hline 3\\\cline{1-1}\end{array}_2\right|\right)\right)\\ &+ \frac{1}{6}\left((1-x^3)\left(\left|\begin{array}{|c|c|}\hline 1&3\\\hline 2\\\cline{1-1}\end{array}_1\right\rangle\left\langle\begin{array}{|c|c|}\hline 1&3\\\hline 2\\\cline{1-1}\end{array}_1\right| + \left|\begin{array}{|c|c|}\hline 1&3\\\hline 2\\\cline{1-1}\end{array}_2\right\rangle\left\langle\begin{array}{|c|c|}\hline 1&3\\\hline 2\\\cline{1-1}\end{array}_2\right|\right) + (1-3x^2+2x^3)\left|\begin{array}{|c|}\hline 1\\\hline 2\\\hline 3\\\hline\end{array}\right\rangle\left\langle\begin{array}{|c|}\hline 1\\\hline 2\\\hline 3\\\hline\end{array}\right|\right).\end{aligned} \tag{6.116}$$

It is again possible to verify that as $x \to 1$ then only the symmetric irrep contributes, and as $x \to 0$ then all irreps contribute uniformly.

In Figure 6.5 we have plotted the probability of sampling from different irreps over $x \in [0, 1]$ for (6.5a) $n = 2$ and (6.5b) $n = 3$. As already mentioned, this makes clear how at $x = 0$ all terms have equal probability, and as $x \to 1$ the fully symmetric irrep begins to dominate. It is interesting to note that for $n = 3$, $\lambda = (2, 1)$ is the most likely irrep; this is because of the multiplicities within this irrep for both S_3 and $\mathrm{U}(3)$. By solving the cubic equation

$$1 + 3x^2 + 2x^3 = 4 - 4x^3, \tag{6.117}$$

we find that the two overlap when $x \approx 0.657$, when both irreps have probability approximately 0.4773 of being sampled[3].

We now go back to the question at the start of this section, as to whether or not this could lead to another simulation. A potential method of simulation, akin to the point and state truncation methods, would be to truncate which irrep we sample from. In other words, we sample an irrep with at most $n - k$ nonzero rows for some truncation level k, and then perform sampling conditioned on that irrep.

There are a few limitations with this approach. The first is that this expansion is nontrivial, and we do not currently have a convenient general form for it, unlike with state truncation where we simply follow the binomial distribution. The second is that, even in the fully distinguishable case, the matrix permanent still plays a role in the distribution, so for any value of x including $x = 0$ we are going to have some error if we truncate before the symmetric irrep. And the third is that it is unclear how to classically simulate these irrep states under unitary action; it seems reasonable for them to be related to matrix immanants in a way similar to Tichy and Mølmer [289], but even if so classical algorithms for computing more general immanants are less well-known, though some do exist [260, 290, 261, 263]. Obviously classical algorithms for the special cases of the permanent and determinant are well-known with exponential [291] and polynomial [259] runtimes, respectively.

So if this decomposition does not provide us with faster classical algorithms, what might it be useful for? Well, there are some instances of λ for which matrix imminants are known to be hard. The best-known of these is the permanent [183, 71], but other forms of immanant such as for hook or rectangular diagrams are also know to be #P-hard [264]. If one is able to construct a form of distinguishable-photon Boson Sampling such that the probability ends up being largely concentrated in these hard instances, and show that computing the sum of these immanants is at least as hard as computing a single immanant, then this could potentially lead to new proofs of hardness for Boson Sampling under distinguishability. We leave this question as future work.

6.10 Conclusion

In recent years significant improvements have been made in the ability of classical computers to simulate Boson Sampling under various imperfections. However, while it is of theoretical interest to demonstrate asymptotic improvements in classical simulation, the whole reason for proposals such as Boson Sampling is to offer speedups for near-term devices. Although our algorithm will not scale polynomially as the number of photons increases, we find that a substantial improvement over current classical algorithms can be achieved for the numbers of photons that experimentalists are currently aiming for. In doing so, we have effectively set a benchmark for what is required of a 50–90 photon Boson Sampling device.

[3] Note that there are other solutions to this cubic equation, but $x \approx 0.657$ is the only real-valued solution between 0 and 1 and is therefore the only solution relevant to us.

6.10.1 Open questions

There are a number of ways one could improve this classical simulation. In particular, the approach of Ref. [84] for truncation when looking at near-term devices is dependent on Metropolised Independence Sampling. A direct adaptation of the Clifford & Clifford algorithm to this approach would almost certainly offer an improvement over our algorithm. However, such an adaptation is non-trivial, due to the fact that the terms in the expansion do not correspond to states, something that motivated our work here.

There are other open questions we would like to consider as well. The first would be to reduce the average-case error bounds to be less than our worst-case error bounds. This would most likely involve an alternative to using the triangle inequality. The second would be to find a way of explaining the difference in dependence on η and x between point and state truncation, and ideally improving either algorithm in the process.

As mentioned in Section 4.6.4, Renema et al. argue informally at the end of [84] that the expansion can also apply to more complex models of distinguishability with multiple parameters. We leave extending this classical simlation to more complex models of distinguishability as a question for future research, but note that the expansion considered in Equation 6.36 can also be made more generalised to the cases considered by Renema et al., as well as other imperfections such as those mentioned in Section 4.5.3.

Finally it would be interesting to extend these results to other photonics-based quantum advantage schemes, such as Gaussian Boson Sampling [82]. The probability distribution of Gaussian Boson Sampling is that of n indistinguishable squeezed states at the output of an m-mode linear optical interferometer, and depends on the Hafnian of a matrix. Unlike Boson Sampling, there is no known polynomial time classical algorithm for computing the probability of a single outcome from n fully distinguishable squeezed states. An attempt to remedy this was recently presented by Renema [292], by looking at the probability of an outcome from a Gaussian Boson Sampling as a linear combination of matrix permanents and extending the point truncation method accordingly, but this result hits a stumbling block in the form of an open problem related to the hardness of matrix permanents. On the other hand, it is classically efficient to sample n fully distinguishable squeezed states, via a similar approach to that used for classically sampling distinguishable single photons [188]: For each squeezed state, one first samples the number of photons in that squeezed state via the inverse binomial distribution [293], then each of those photons is sampled through the interferometer individually, as photons that start in the same spatial mode do not interfere with one another and their outcomes can therefore be sampled individually. It might seem odd to think that there is a probability distribution for which it is easy to compute a sample but hard to approximate the probability of a given outcome; after all, surely someone can approximate the probability distribution by simply taking a polynomial number of samples. However, while this will give an approximation up to additive error, it will not give an approximation up to multiplicative error. This is because the distribution space is exponentially large, and most outcomes have an exponentially small probability due to the anti-concentration conjecture. Indeed, simply stating that every outcome has zero probability of happening is a polynomial time approximation up to additive error.

As a result, adapting the Clifford & Clifford algorithm to non-ideal Boson Sampling models provides an important step towards being able to classically simulate imperfections in Gaussian Boson Sampling. This, combined with recent classical simulation algorithms for ideal Gaussian Boson Sampling [294, 293], show strong potential for these results to extent to Gaussian Boson Sampling.

6.10.2 Note added

During this work we were made aware of independent work by V. Shchesnovich, which also shows that the model of distinguishability considered by Renema et al. corresponds to that of selecting indistinguishable photons via the binomial distribution [295]. This is derived using significantly different methods from those used in this manuscript, and does not consider classical simulation of distinguishability via the above method (though this has been anticipated [296]).

Part IV

Conclusions and open questions

Chapter 7

Conclusion

It truly is astounding to see what progress has been made in quantum computing. From the 1980s, when quantum computers were a vague concept with no theoretical benefits beyond possibly simulation [7], to the 1990s when the first theoretical speedups were discovered [13, 15], to today where demonstration devices are now readily available and showcasing the potential that quantum computers hold [4, 3]. All of this culminating in Google's astonishing result: a quantum computer solving a problem significantly faster than what even the best supercomputers can accomplish [5]. This is a tremendous way to mark the start of a new decade in quantum computing.

And with this new decade, new goals must be sets. Now that we have quantum computers solving a problem faster than what is classically possible, we need to show that quantum computers can do so for a problem that is beneficial to us. And just as importantly, we need to show that quantum computers can solve this problem well.

In this thesis, we have strived to close the gap between what problems quantum computers can solve and the limitations of their architecture. In Part II, we have given a new example problem which quantum computers can offer a speedup for over the best classical algorithms: The Travelling Salesman Problem. As an NP-Hard problem with many applications, this demonstrates how quantum computers can offer speedups for even some of the most challenging problems we encounter in Computer Science and Mathematics. In Part III, we assess how photon distinguishability and loss affect the near-term quantum advantage architecture of Boson Sampling. In doing so, we devise new methods for mathematically modelling these imperfections via the first quantisation, and from there adapt the classical Clifford & Clifford algorithm [91] to take advantage of these issues. This emphasises how much these imperfections reduce any benefit we are likely to see, and how much effort we must do to overcome these experimental challenges in the near future.

7.1 Open questions

But alas, the gap between these directions still exists, and there is yet more work to be done. We shall conclude this thesis with some open questions, such that an inspired reader might choose to pursue these routes even deeper. As we have done so throughout this thesis, we shall consider the two approaches separately.

7.1.1 Application-focused directions

Several open directions have already been mentioned in Section 3.5. Of particular note is the question of what other classical algorithms for the Travelling Salesman Problem can be sped up by a quantum computer. Promising directions include cut-and-count [148, 146, 147], which use a combination of Monte Carlo algorithms and dynamic programming, and branch-and-bound & branch-and-cut [109, 110, 111], for which quantum speedups already exist but most complexity understanding is empirical rather than analytical.

The fact that such speedups exist for the Travelling Salesman Problem suggests that polynomial speedups also exist for other NP-Hard problems as well. Already several other problems have seen speedups: Campbell, Khurana and Montanaro [154] showed how the backtracking framework can also be applied to Boolean Satisfiability and Graph Colouring, and Montanaro [152] used the quantum speedup for branch-and-bound algorithms to find the exact ground state of Sherrington-Kirkpatrick Hamiltonians. Due to the same families of algorithms, such as dynamic programming or backtracking, being used to solve many different NP-Hard problems, it is likely that in the future we shall see polynomial speedups for many more.

But of course, this still does not address the main stumbling block when moving from these eventual speedups to a near-term speedup: That the quantum algorithm needs to run on near-term quantum computers. As noted in Campbell, Khurana and Montanaro [154] and mentioned in Section 4.1, this already rules out many of these algorithms, for which the significant dependence on error correction and fault tolerance, and the classical computational overhead the comes with it, reduces any quantum speedup in practice to nought. To overcome this, we need to see these algorithms adapted to architectures which might be realisable in the near future.

We already hint at this open direction in Section 4.1, when discussing the recent work on algorithms which use classical processing to break the problem down until it is of a size that a quantum algorithm with a small number of qubits can be used [68, 69]. It is likely that other quantum algorithms for these problems can also be adapted to a hybrid quantum-classical framework of this form. One that we think is particularly promising is the approach of Ambainis et al. [116], where a polynomial speedup for dynamic programming is obtained by using a classical computation to preprocess simple spaces of the problem, followed by using Quantum Minimum Finding to search over the larger spaces. It might be possible to adapt this architecture to quantum computers with constrained amounts of memory, by more careful analysis of the quantum algorithm's memory usage and new consideration of how to partition the quantum and classical aspects of the algorithm. It is worth noting that this approach still depends on universal fault-tolerant quantum computation, but the hope is that the smaller number of logical qubits required would lead to a more feasible speedup.

There are also other near-term architectures that can be considered. Of particular note is the Quantum Adiabatic Optimisation Algorithm. When originally proposed by Farhi et al. [58], it was used as a quantum algorithm for finding an approximation to the Max-Cut problem, which is known to be NP-Hard [99]. As a model of universal quantum computation [59, 60], it is likely that other applications for QAOA to NP-Hard problems will also be found in the future, though it is less clear how promising a speedup QAOA will offer. Other intermediate-scale, or NISQ, devices also show potential in different applications, such as the Variational Quantum Eigensolver for simulating physical systems [57, 56].

7.1.2 Architecture-focused directions

There are many directions we can go when looking at near-term quantum architectures. As mentioned at the end of Chapters 5 and 6, the most natural direction for continuing this work is to better understand the computational complexity of Boson Sampling under these imperfections.

We have already discussed several examples of classical simulation approaches, such as [84, 85, 83, 86, 87], and have hinted at how one might prove hardness, via the computational complexity of immanants [260, 290, 261, 264, 263]. Better understanding of how the Quantum Schur Transform acts on certain states would also assist in this goal.

But Boson Sampling is only one of many photonic quantum computing architectures. Another interesting question is whether or not these same imperfections can be applied to variants of Boson Sampling. Many of these simulation algorithms, including our own described in Chapter 6, can be naturally extended to Scattershot Boson Sampling, by simply choosing which modes our photons start in from the $\binom{m}{n}$ uniform distribution at the start. But what about Gaussian Boson Sampling? Already there have been some promising results by extending some of the classical algorithms for both ideal and imperfect Boson Sampling [297, 294, 293, 292], and it would be exciting to see the same for other algorithms. It would also be interesting to see if these simulators can be applied to universal linear optical quantum computing architectures, through simulating postselected, adaptive or measurement-based schemes [298, 299].

More broadly, we still need to work hard to find applications for Boson Sampling and its variants. As has already been mentioned, a number of promising directions exist, particularly for more general linear optics [92] and Gaussian Boson Sampling [93, 94, 95]. It will be interesting to see what other applications exist, and especially if these applications extend to other schemes.

And beyond linear optics, we need to look at the many quantum advantage proposals that have been proposed over the last several years, to see what extent we can classically simulate them as well as what applications they might have. Boson Sampling has received a lot of attention on this end, as already shown. Less is currently known about the extent to which IQP circuits and Random Circuit Sampling can be classically simulated. But there are some promising results in these directions. For IQP circuits, Bremner, Montanaro and Shepherd showed how polarisation error can lead to an efficient classical simulation, as well as how simple forms of error correction can mitigate these issues [169]. As for Random Circuit Sampling, Pednault et al. [53] give an estimate for what is likely to be the largest classical simulation we can achieve, and Morimae, Takeuchi and Tani [54] suggest a classical simulator which takes advantage of the poor fidelity. Another relevant direction, considered by Bravyi et al. [300], looks at how much we can classically simulate random quantum circuits when there are only a small number of non-Clifford gates[1], showing that circuits with 40–50 qubits and over 60 non-Clifford gates can be simulated on a standard computer.

7.2 The future

So what lies ahead for the world of quantum computing then? It is always hard to predict what will happen, but the current state of quantum computers offer a lot of promise and potential. Google's paper has proven that potentially relevant and significant quantum devices could be realisable within our lifetime, if not already here. But in order to truly demonstrate that these devices are worth the hype and effort behind them, now is more important than ever that we push for improving the architecture to as good a quality as possible, as well as refining the problems we are proposing for them to make them more physically realisable.

Eventually, we hope that these two paths will intersect, and we will have a useful application for a quantum computer. Then, dear reader, we will have found our quantum speedup.

[1] Note that quantum circuits composed entirely of Clifford gates are efficiently classically simulable, and the inclusion of non-Clifford gates such as the T or Toffoli gates is sufficient for universal quantum computation.

Bibliography

[1] Dominic W. Berry, Craig Gidney, Mario Motta, Jarrod R. McClean, and Ryan Babbush. Qubitization of Arbitrary Basis Quantum Chemistry Leveraging Sparsity and Low Rank Factorization. *Quantum*, 3:208, 2019. arXiv:1902.02134, doi:10.22331/q-2019-12-02-208.

[2] Iordanis Kerenidis and Anupam Prakash. Quantum Recommendation Systems. In Christos H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, volume 67 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 49:1–49:21, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: http://drops.dagstuhl.de/opus/volltexte/2017/8154, arXiv:1603.08675, doi:10.4230/LIPIcs.ITCS.2017.49.

[3] IBM. IBM Q Experience. Retrieved 07 January 2020. URL: https://quantum-computing.ibm.com/.

[4] Rigetti. Quantum Computing Systems. Retrieved 07 January 2020. URL: https://www.rigetti.com/systems/.

[5] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V. Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R. McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019. doi:10.1038/s41586-019-1666-5.

[6] Ashley Montanaro. Quantum algorithms: an overview. *npj Quantum Information*, 2, 2016. arXiv:1511.04206, doi:10.1038/npjqi.2015.23.

[7] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6):467–488, 1982. doi:10.1007/BF02650179.

[8] Seth Lloyd. Universal Quantum Simulators. *Science*, 273(5278):1073–1078, 1996. URL: `https://science.sciencemag.org/content/273/5278/1073`, `doi:10.1126/science.273.5278.1073`.

[9] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Simulating Hamiltonian Dynamics with a Truncated Taylor Series. *Physical Review Letters*, 114:090502, 2015. URL: `https://link.aps.org/doi/10.1103/PhysRevLett.114.090502`, `arXiv:1412.4687`, `doi:10.1103/PhysRevLett.114.090502`.

[10] D. W. Berry, A. M. Childs, and R. Kothari. Hamiltonian Simulation with Nearly Optimal Dependence on all Parameters. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS 2015)*, pages 792–809, 2015. `arXiv:1501.01715`, `doi:10.1109/FOCS.2015.54`.

[11] Guang Hao Low and Isaac L. Chuang. Optimal Hamiltonian Simulation by Quantum Signal Processing. *Physical Review Letters*, 118:010501, 2017. URL: `https://link.aps.org/doi/10.1103/PhysRevLett.118.010501`, `arXiv:1606.02685`, `doi:10.1103/PhysRevLett.118.010501`.

[12] David Deutsch and Richard Jozsa. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 439(1907):553–558, 1992. URL: `https://royalsocietypublishing.org/doi/abs/10.1098/rspa.1992.0167`, `doi:10.1098/rspa.1992.0167`.

[13] P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science (FOCS 1994)*, pages 124–134, 1994. `arXiv:quant-ph/9508027`, `doi:10.1109/SFCS.1994.365700`.

[14] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography.* Cryptography and Network Security. Chapman & Hall/CRC, Boca Raton, Florida, first edition, 2007.

[15] Lov K. Grover. A Fast Quantum Mechanical Algorithm for Database Search. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC 1996)*, pages 212–219, 1996. URL: `http://doi.acm.org/10.1145/237814.237866`, `arXiv:quant-ph/9605043`, `doi:10.1145/237814.237866`.

[16] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum Algorithm for Linear Systems of Equations. *Physical Review Letters*, 103:150502, 2009. URL: `https://link.aps.org/doi/10.1103/PhysRevLett.103.150502`, `arXiv:0811.3171`, `doi:10.1103/PhysRevLett.103.150502`.

[17] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum Support Vector Machine for Big Data Classification. *Physical Review Letters*, 113:130503, 2014. URL: `https://link.aps.org/doi/10.1103/PhysRevLett.113.130503`, `arXiv:1307.0471`, `doi:10.1103/PhysRevLett.113.130503`.

[18] Peter Wittek. *Quantum Machine Learning.* Elsevier, Amsterdam, Netherlands, 2014.

[19] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. An introduction to quantum machine learning. *Contemporary Physics*, 56(2):172–185, 2015. `arXiv:1409.3097`, `doi:10.1080/00107514.2014.964942`.

[20] Jeremy Adcock, Euan Allen, Matthew Day, Stefan Frick, Janna Hinchliff, Mack Johnson, Sam Morley-Short, Sam Pallister, Alasdair Price, and Stasja Stanisic. Advances in quantum machine learning, 2015. arXiv:1512.02900. `arXiv:1512.02900`.

[21] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017. `arXiv:1611.09347`, `doi:10.1038/nature23474`.

[22] Ewin Tang. A Quantum-Inspired Classical Algorithm for Recommendation Systems. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC 2019)*, pages 217—-228, New York, NY, USA, 2019. Association for Computing Machinery. `arXiv:1807.04271`, `doi:10.1145/3313276.3316310`.

[23] Ewin Tang. Quantum-inspired classical algorithms for principal component analysis and supervised clustering, 2018. arXiv:1811.00414. `arXiv:1811.00414`.

[24] András Gilyén, Seth Lloyd, and Ewin Tang. Quantum-inspired low-rank stochastic regression with logarithmic dependence on the dimension, 2018. arXiv:1811.04909. `arXiv:1811.04909`.

[25] Juan Miguel Arrazola, Alain Delgado, Bhaskar Roy Bardhan, and Seth Lloyd. Quantum-inspired algorithms in practice, 2019. arXiv:1905.10415. `arXiv:1905.10415`.

[26] Nai-Hui Chia, András Gilyén, Tongyang Li, Han-Hsuan Lin, Ewin Tang, and Chunhao Wang. Sampling-based sublinear low-rank matrix arithmetic framework for dequantizing quantum machine learning, 2019. arXiv:1910.06151. `arXiv:1910.06151`.

[27] Dhawal Jethwani, François Le Gall, and Sanjay K. Singh. Quantum-Inspired Classical Algorithms for Singular Value Transformation, 2019. arXiv:1910.05699. `arXiv:1910.05699`.

[28] Lieven M. K. Vandersypen, Matthias Steffen, Gregory Breyta, Costantino S. Yannoni, Mark H. Sherwood, and Isaac L. Chuang. Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature*, 414(6866):883–887, 2001. `arXiv:quant-ph/0112176`, `doi:10.1038/414883a`.

[29] B. P. Lanyon, T. J. Weinhold, N. K. Langford, M. Barbieri, D. F. V. James, A. Gilchrist, and A. G. White. Experimental Demonstration of a Compiled Version of Shor's Algorithm with Quantum Entanglement. *Physical Review Letters*, 99:250505, 2007. URL: `https://link.aps.org/doi/10.1103/PhysRevLett.99.250505`, `arXiv:0705.1398`, `doi:10.1103/PhysRevLett.99.250505`.

[30] Chao-Yang Lu, Daniel E. Browne, Tao Yang, and Jian-Wei Pan. Demonstration of a Compiled Version of Shor's Quantum Factoring Algorithm Using Photonic Qubits. *Physical Review Letters*, 99:250504, 2007. URL: `https://link.aps.org/doi/10.1103/PhysRevLett.99.250504`, `arXiv:0705.1684`, `doi:10.1103/PhysRevLett.99.250504`.

[31] Alberto Politi, Jonathan C. F. Matthews, and Jeremy L. O'Brien. Shor's Quantum Factoring Algorithm on a Photonic Chip. *Science*, 325(5945):1221–1221, 2009. URL: `https://science.sciencemag.org/content/325/5945/1221`, `arXiv:0911.1242`, `doi:10.1126/science.1173731`.

[32] Erik Lucero, R. Barends, Y. Chen, J. Kelly, M. Mariantoni, A. Megrant, P. O'Malley, D. Sank, A. Vainsencher, J. Wenner, T. White, Y. Yin, A. N. Cleland, and John M. Martinis. Computing prime factors with a Josephson phase qubit quantum processor. *Nature Physics*, 8(10):719–723, 2012. `arXiv:1202.5707`, `doi:10.1038/nphys2385`.

[33] Enrique Martín-López, Anthony Laing, Thomas Lawson, Roberto Alvarez, Xiao-Qi Zhou, and Jeremy L. O'Brien. Experimental realization of Shor's quantum factoring algorithm using qubit recycling. *Nature Photonics*, 6(11):773–776, 2012. `arXiv:1111.4147`, `doi:10.1038/nphoton.2012.259`.

[34] John A. Smolin, Graeme Smith, and Alexander Vargo. Oversimplifying quantum factoring. *Nature*, 499(7457):163–165, 2013. `doi:10.1038/nature12290`.

[35] Nanyang Xu, Jing Zhu, Dawei Lu, Xianyi Zhou, Xinhua Peng, and Jiangfeng Du. Quantum Factorization of 143 on a Dipolar-Coupling Nuclear Magnetic Resonance System. *Physical Review Letters*, 108:130501, 2012. URL: `https://link.aps.org/doi/10.1103/PhysRevLett.108.130501`, `doi:10.1103/PhysRevLett.108.130501`.

[36] Nikesh S. Dattani and Nathaniel Bryans. Quantum factorization of 56153 with only 4 qubits, 2014. arXiv:1411.6758v3. `arXiv:1411.6758`.

[37] Raouf Dridi and Hedayat Alghassi. Prime factorization using quantum annealing and computational algebraic geometry. *Scientific Reports*, 7(1):43048, 2017. `arXiv:1604.05796`, `doi:10.1038/srep43048`.

[38] Zhaokai Li, Nikesh S. Dattani, Xi Chen, Xiaomei Liu, Hengyan Wang, Richard Tanburn, Hongwei Chen, Xinhua Peng, and Jiangfeng Du. High-fidelity adiabatic quantum computation using the intrinsic Hamiltonian of a spin system: Application to the experimental factorization of 291311, 2017. arXiv:1706.08061v1. `arXiv:1706.08061`.

[39] Eric R. Anschuetz, Jonathan P. Olson, Alán Aspuru-Guzik, and Yudong Cao. Variational Quantum Factoring, 2018. arXiv:1808.08927v1. `arXiv:1808.08927`.

[40] Leah Crane. Quantum computer sets new record for finding prime number factors, 2019. Retrieved 24 February 2019. URL: `https://www.newscientist.com/article/2227387-quantum-computer-sets-new-record-for-finding-prime-number-factors/`.

[41] Scott Aaronson. Quantum computing motte-and-baileys, 2019. Retrieved 27 February 2020. URL: `https://www.scottaaronson.com/blog/?p=4447`.

[42] Fabrice Boudot, Pierrick Gaudry, Aurore Guillevic, Nadia Heninger, Emmanuel Thomé, and Paul Zimmermann. [cado-nfs-discuss] 795-bit factoring and discrete logarithms, 2019. Retrieved 31 January 2020. URL: `https://lists.gforge.inria.fr/pipermail/cado-nfs-discuss/2019-December/001139.html`.

[43] Fabrice Boudot, Pierrick Gaudry, Aurore Guillevic, Nadia Heninger, Emmanuel Thome, and Paul Zimmermann. Factorization of RSA-250, 2020. Retrieved 29 February 2020. URL: `https://listserv.nodak.edu/cgi-bin/wa.exe?A2=NMBRTHRY;dc42ccd1.2002`.

[44] Thorsten Kleinjung, Joppe W. Bos, and Arjen K. Lenstra. Mersenne Factorization Factory. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014*, pages 358–377, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg. Cryptology ePrint Archive, Report 2014/653, `https://eprint.iacr.org/2014/653`. URL: `https://link.springer.com/chapter/10.1007/978-3-662-45611-8_19`.

[45] X.-D. Cai, C. Weedbrook, Z.-E. Su, M.-C. Chen, Mile Gu, M.-J. Zhu, Li Li, Nai-Le Liu, Chao-Yang Lu, and Jian-Wei Pan. Experimental Quantum Computing to Solve Systems of Linear Equations. *Physical Review Letters*, 110:230501, 2013. URL: `https://link.aps.org/doi/10.1103/PhysRevLett.110.230501`, `arXiv:1302.4310`, `doi:10.1103/PhysRevLett.110.230501`.

[46] Jian Pan, Yudong Cao, Xiwei Yao, Zhaokai Li, Chenyong Ju, Hongwei Chen, Xinhua Peng, Sabre Kais, and Jiangfeng Du. Experimental realization of quantum algorithm for solving linear systems of equations. *Physical Review A*, 89:022313, 2014. URL: `https://link.aps.org/doi/10.1103/PhysRevA.89.022313`, `arXiv:1302.1946`, `doi:10.1103/PhysRevA.89.022313`.

[47] Stefanie Barz, Ivan Kassal, Martin Ringbauer, Yannick Ole Lipp, Borivoje Dakic, Alán Aspuru-Guzik, and Philip Walther. A two-qubit photonic quantum processor and its application to solving systems of linear equations. *Scientific Reports*, 4(1):6115, 2014. `doi:10.1038/srep06115`.

[48] Zhaokai Li, Xiaomei Liu, Nanyang Xu, and Jiangfeng Du. Experimental Realization of a Quantum Support Vector Machine. *Physical Review Letters*, 114:140504, 2015. URL: `https://link.aps.org/doi/10.1103/PhysRevLett.114.140504`, `doi:10.1103/PhysRevLett.114.140504`.

[49] Google Research. Datasets. Retrieved 28 February 2020. URL: `https://research.google/tools/datasets/`.

[50] Sergio Boixo, Sergei V. Isakov, Vadim N. Smelyanskiy, Ryan Babbush, Nan Ding, Zhang Jiang, Michael J. Bremner, John M. Martinis, and Hartmut Neven. Characterizing quantum supremacy in near-term devices. *Nature Physics*, 14(6):595–600, 2018. `arXiv:1608.00263`, `doi:10.1038/s41567-018-0124-x`.

[51] Dominik Hangleiter, Juan Bermejo-Vega, Martin Schwarz, and Jens Eisert. Anticoncentration theorems for schemes showing a quantum speedup. *Quantum*, 2:65, 2018. `arXiv:1706.03786`, `doi:10.22331/q-2018-05-22-65`.

[52] Adam Bouland, Bill Fefferman, Chinmay Nirkhe, and Umesh Vazirani. Quantum Supremacy and the Complexity of Random Circuit Sampling, 2018. arXiv:1803.04402. `arXiv:1803.04402`.

[53] Edwin Pednault, John A. Gunnels, Giacomo Nannicini, Lior Horesh, and Robert Wisnieff. Leveraging Secondary Storage to Simulate Deep 54-qubit Sycamore Circuits, 2019. arXiv:1910.09534. `arXiv:1910.09534`.

[54] Tomoyuki Morimae, Yuki Takeuchi, and Seiichiro Tani. Sampling of globally depolarized random quantum circuit. Technical Report YITP-19-90, Yukawa Institute for Theoretical Physics, Kyoto University, 2019. `arXiv:1911.02220`.

[55] Yiqing Zhou, E. Miles Stoudenmire, and Xavier Waintal. What limits the simulation of quantum computers?, 2020. arXiv:2002.07730v1. `arXiv:2002.07730`.

[56] John Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, 2:79, 2018. `arXiv:1801.00862`, `doi:10.22331/q-2018-08-06-79`.

[57] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5(1):4213, 2014. `arXiv:1304.3061`, `doi:10.1038/ncomms5213`.

[58] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A Quantum Approximate Optimization Algorithm. Technical Report MIT-CTP/4610, Centre for Theoretical Physics, Massachusetts Institute of Technology, 2014. `arXiv:1411.4028`.

[59] Seth Lloyd. Quantum approximate optimization is computationally universal, 2018. arXiv:1812.11075. `arXiv:1812.11075`.

[60] Mauro E. S. Morales, Jacob Biamonte, and Zoltán Zimborás. On the Universality of the Quantum Approximate Optimization Algorithm, 2019. arXiv:1909.03123. `arXiv:1909.03123`.

[61] Alexander Schrijver. On the History of Combinatorial Optimization (Till 1960). In K. Aardal, G.L. Nemhauser, and R. Weismantel, editors, *Discrete Optimization*, volume 12 of *Handbooks in Operations Research and Management Science*, pages 1–68. Elsevier, 2005. URL: `http://www.sciencedirect.com/science/article/pii/S0927050705120015`, `doi:https://doi.org/10.1016/S0927-0507(05)12001-5`.

[62] Clay Mathematics Institute. P vs NP problem. Retrieved 17 September 2019. URL: `https://www.claymath.org/millennium-problems/p-vs-np-problem`.

[63] Scott Aaronson. Reasons to believe, 2006. Retrieved 16 September 2019. URL: `https://www.scottaaronson.com/blog/?p=122`.

[64] David Eppstein. The Traveling Salesman Problem for Cubic Graphs. *Journal of Graph Algorithms and Applications*, 11(1):61–81, 2007. `arXiv:cs/0302030`, `doi:10.7155/jgaa.00137`.

[65] Mingyu Xiao and Hiroshi Nagamochi. An Exact Algorithm for TSP in Degree-3 Graphs Via Circuit Procedure and Amortization on Connectivity Structure. *Algorithmica*, 74(2):713–741, 2016. URL: `http://dx.doi.org/10.1007/s00453-015-9970-4`, `arXiv:1212.6831`, `doi:10.1007/s00453-015-9970-4`.

[66] Mingyu Xiao and Hiroshi Nagamochi. An Improved Exact Algorithm for TSP in Graphs of Maximum Degree 4. *Theory of Computing Systems*, 58(2):241–272, 2016. URL: `http://dx.doi.org/10.1007/s00224-015-9612-x`, `doi:10.1007/s00224-015-9612-x`.

[67] Ashley Montanaro. Quantum walk speedup of backtracking algorithms, 2015. arXiv:1509.02374v2. `arXiv:1509.02374`.

[68] Vedran Dunjko, Yimin Ge, and J. Ignacio Cirac. Computational Speedups Using Small Quantum Devices. *Physical Review Letters*, 121:250501, 2018. URL: `https://link.aps.org/doi/10.1103/PhysRevLett.121.250501`, `arXiv:1807.08970`, `doi:10.1103/PhysRevLett.121.250501`.

[69] Yimin Ge and Vedran Dunjko. A hybrid algorithm framework for small quantum computers with application to finding Hamiltonian cycles, 2019. arXiv:1907.01258v1. `arXiv:1907.01258`.

[70] Scott Aaronson and Alex Arkhipov. The Computational Complexity of Linear Optics. Technical Report TR10-170, Electronic Colloquium on Computational Complexity, Weizmann Institute of Science, 2010. URL: `https://eccc.weizmann.ac.il/report/2010/170/`, `arXiv:1011.3245`.

[71] Scott Aaronson and Alex Arkhipov. The Computational Complexity of Linear Optics. In *Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing (STOC 2011)*, pages 333–342, New York, NY, USA, 2011. ACM. URL: `http://doi.acm.org/10.1145/1993636.1993682`, `doi:10.1145/1993636.1993682`.

[72] Matthew A. Broome, Alessandro Fedrizzi, Saleh Rahimi-Keshari, Justin Dove, Scott Aaronson, Timothy C. Ralph, and Andrew G. White. Photonic Boson Sampling in a Tunable Circuit. *Science*, 339(6121):794–798, 2013. URL: `http://science.sciencemag.org/content/339/6121/794`, `arXiv:1212.2234`, `doi:10.1126/science.1231440`.

[73] Justin B. Spring, Benjamin J. Metcalf, Peter C. Humphreys, W. Steven Kolthammer, Xian-Min Jin, Marco Barbieri, Animesh Datta, Nicholas Thomas-Peter, Nathan K. Langford, Dmytro Kundys, James C. Gates, Brian J. Smith, Peter G. R. Smith, and Ian A. Walmsley. Boson Sampling on a Photonic Chip. *Science*, 339(6121):798–801, 2013. URL: `http://science.sciencemag.org/content/early/2012/12/19/science.1231692`, `arXiv:1212.2622`, `doi:10.1126/science.1231692`.

[74] Max Tillmann, Borivoje Dakić, René Heilmann, Stefan Nolte, Alexander Szameit, and Philip Walther. Experimental boson sampling. *Nature Photonics*, 7:540, 2013. URL: `http://dx.doi.org/10.1038/nphoton.2013.102`, `arXiv:1212.2240`, `doi:10.1038/nphoton.2013.102`.

[75] Andrea Crespi, Roberto Osellame, Roberta Ramponi, Daniel J. Brod, Ernesto F. Galvão, Nicolò Spagnolo, Chiara Vitelli, Enrico Maiorino, Paolo Mataloni, and Fabio Sciarrino. Integrated multimode interferometers with arbitrary designs for photonic boson sampling. *Nature Photonics*, 7:545, 2013. URL: `http://dx.doi.org/10.1038/nphoton.2013.112`, `arXiv:1212.2783`, `doi:10.1038/nphoton.2013.112`.

[76] Jacques Carolan, Christopher Harrold, Chris Sparrow, Enrique Martín-López, Nicholas J. Russell, Joshua W. Silverstone, Peter J. Shadbolt, Nobuyuki Matsuda, Manabu Oguma, Mikitaka Itoh, Graham D. Marshall, Mark G. Thompson, Jonathan C. F. Matthews, Toshikazu Hashimoto, Jeremy L. O'Brien, and Anthony Laing. Universal linear optics. *Science*, 349(6249):711–716, 2015. URL: `http://science.sciencemag.org/content/349/6249/711`, `arXiv:1505.01182`, `doi:10.1126/science.aab3642`.

[77] Hui Wang, Yu He, Yu-Huai Li, Zu-En Su, Bo Li, He-Liang Huang, Xing Ding, Ming-Cheng Chen, Chang Liu, Jian Qin, Jin-Peng Li, Yu-Ming He, Christian Schneider, Martin Kamp, Cheng-Zhi Peng, Sven Höfling, Chao-Yang Lu, and Jian-Wei Pan. High-efficiency multiphoton boson sampling. *Nature Photonics*, 11:361–365, 2017. URL: `http://dx.doi.org/10.1038/nphoton.2017.63`.

[78] Stefano Paesani, Yunhong Ding, Raffaele Santagati, Levon Chakhmakhchyan, Caterina Vigliar, Karsten Rottwitt, Leif K. Oxenløwe, Jianwei Wang, Mark G. Thompson, and Anthony Laing. Generation and sampling of quantum states of light in a silicon chip. *Nature Physics*, 15(9):925–929, 2019. `arXiv:1812.03158`, `doi:10.1038/s41567-019-0567-8`.

[79] Hui Wang, Jian Qin, Xing Ding, Ming-Cheng Chen, Si Chen, Xiang You, Yu-Ming He, Xiao Jiang, L. You, Z. Wang, C. Schneider, Jelmer J. Renema, Sven Höfling, Chao-Yang Lu, and Jian-Wei Pan. Boson Sampling with 20 Input Photons and a 60-Mode Interferometer in a 10^{14}-Dimensional Hilbert Space. *Physical Review Letters*, 123:250503, 2019. URL: https://link.aps.org/doi/10.1103/PhysRevLett.123.250503, arXiv:1910.09930, doi:10.1103/PhysRevLett.123.250503.

[80] Scott Aaronson. Scattershot BosonSampling: A new approach to scalable BosonSampling experiments, 2013. Retrieved 08 December 2019. URL: https://www.scottaaronson.com/blog/?p=1579.

[81] A. P. Lund, A. Laing, S. Rahimi-Keshari, T. Rudolph, J. L. O'Brien, and T. C. Ralph. Boson Sampling from a Gaussian State. *Physical Review Letters*, 113:100502, 2014. URL: https://link.aps.org/doi/10.1103/PhysRevLett.113.100502, arXiv:1305.4346, doi:10.1103/PhysRevLett.113.100502.

[82] Craig S. Hamilton, Regina Kruse, Linda Sansoni, Sonja Barkhofen, Christine Silberhorn, and Igor Jex. Gaussian Boson Sampling. *Physical Review Letters*, 119:170501, 2017. URL: https://link.aps.org/doi/10.1103/PhysRevLett.119.170501, arXiv:1612.01199, doi:10.1103/PhysRevLett.119.170501.

[83] Raúl García-Patrón, Jelmer J. Renema, and Valery Shchesnovich. Simulating boson sampling in lossy architectures. *Quantum*, 3:169, 2019. arXiv:1712.10037, doi:10.22331/q-2019-08-05-169.

[84] J. J. Renema, A. Menssen, W. R. Clements, G. Triginer, W. S. Kolthammer, and I. A. Walmsley. Efficient Classical Algorithm for Boson Sampling with Partially Distinguishable Photons. *Physical Review Letters*, 120:220502, 2018. URL: https://link.aps.org/doi/10.1103/PhysRevLett.120.220502, arXiv:1707.02793, doi:10.1103/PhysRevLett.120.220502.

[85] Jelmer Renema, Valery Shchesnovich, and Raul Garcia-Patron. Classical simulability of noisy boson sampling, 2018. arXiv:1809.01953. arXiv:1809.01953.

[86] Michał Oszmaniec and Daniel J Brod. Classical simulation of photonic linear optics with lost particles. *New Journal of Physics*, 20(9):092002, 2018. URL: https://doi.org/10.1088%2F1367-2630%2Faadfa8, arXiv:1801.06166, doi:10.1088/1367-2630/aadfa8.

[87] Daniel Jost Brod and Michał Oszmaniec. Classical simulation of linear optics subject to nonuniform losses, 2019. arXiv:1906.06696v1. arXiv:1906.06696.

[88] Dave Bacon, Isaac L. Chuang, and Aram W. Harrow. Efficient Quantum Circuits for Schur and Clebsch-Gordan Transforms. *Physical Review Letters*, 97:170502, 2006. URL: https://link.aps.org/doi/10.1103/PhysRevLett.97.170502, doi:10.1103/PhysRevLett.97.170502.

[89] Aram W. Harrow. *Applications of coherent classical communication and the Schur transform to quantum information theory.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2005. URL: https://dspace.mit.edu/handle/1721.1/34973, arXiv:quant-ph/0512255.

[90] Dave Bacon, Isaac L. Chuang, and Aram W. Harrow. The Quantum Schur and Clebsch-Gordan Transforms: I. Efficient Qudit Circuits. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2007)*, pages 1235–1244, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics. URL: `http://dl.acm.org/citation.cfm?id=1283383.1283516`, `arXiv:quant-ph/0601001`.

[91] Peter Clifford and Raphaël Clifford. The Classical Complexity of Boson Sampling. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2018)*, pages 146–155, 2018. URL: `http://epubs.siam.org/doi/abs/10.1137/1.9781611975031.10`, `arXiv:1706.01260`, `doi:10.1137/1.9781611975031.10`.

[92] Chris Sparrow, Enrique Martín-López, Nicola Maraviglia, Alex Neville, Christopher Harrold, Jacques Carolan, Yogesh N. Joglekar, Toshikazu Hashimoto, Nobuyuki Matsuda, Jeremy L. O'Brien, David P. Tew, and Anthony Laing. Simulating the vibrational quantum dynamics of molecules using photonics. *Nature*, 557(7707):660–667, 2018. `doi:10.1038/s41586-018-0152-9`.

[93] Joonsuk Huh, Gian Giacomo Guerreschi, Borja Peropadre, Jarrod R. McClean, and Alán Aspuru-Guzik. Boson sampling for molecular vibronic spectra. *Nature Photonics*, 9(9):615–620, 2015. `arXiv:1412.8427`, `doi:10.1038/nphoton.2015.153`.

[94] Kamil Brádler, Shmuel Friedland, Josh Izaac, Nathan Killoran, and Daiqin Su. Graph isomorphism and Gaussian boson sampling, 2018. arXiv:1810.10644. `arXiv:1810.10644`.

[95] Maria Schuld, Kamil Brádler, Robert Israel, Daiqin Su, and Brajesh Gupt. A quantum hardware-induced graph kernel based on Gaussian Boson Sampling, 2019. arXiv:1905.12646. `arXiv:1905.12646`.

[96] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Growth of Functions. In *Introduction to Algorithms*, chapter 3, pages 43–64. MIT Press, third edition, 2009.

[97] Scott Aaronson, Greg Kuperberg, Christopher Granade, and Vincent Russo. Complexity Zoo. Retrieved 06 January 2020. URL: `https://complexityzoo.uwaterloo.ca/Complexity_Zoo`.

[98] Charles H. Bennett, Ethan. Bernstein, Gilles. Brassard, and Umesh. Vazirani. Strengths and Weaknesses of Quantum Computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997. `arXiv:quant-ph/9701001`, `doi:10.1137/S0097539796300933`.

[99] Richard M. Karp. Reducibility among Combinatorial Problems. In Raymond E. Miller, James W. Thatcher, and Jean D. Bohlinger, editors, *Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations, held March 20–22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, and sponsored by the Office of Naval Research, Mathematics Program, IBM World Trade Corporation, and the IBM Research Mathematical Sciences Department*, pages 85–103. Springer US, Boston, MA, 1972. `doi:10.1007/978-1-4684-2001-2_9`.

[100] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

[101] Stephen A. Cook. The Complexity of Theorem-proving Procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing (STOC 1971)*, pages 151–158,

New York, NY, USA, 1971. ACM. URL: http://doi.acm.org/10.1145/800157.805047, doi:10.1145/800157.805047.

[102] N. Christofides. Worst-Case Analysis of a New Heuristic for the Travelling Salesman Problem. Technical Report 388, Management Sciences Research Group, Graduate School of Industrial Administration, Carnegie Mellon University, 1976. URL: https://apps.dtic.mil/dtic/tr/fulltext/u2/a025602.pdf.

[103] Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. The Complexity of Satisfiability of Small Depth Circuits. In Jianer Chen and Fedor V. Fomin, editors, *Parameterized and Exact Computation*, pages 75–85, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[104] Ali Asgar Sohanghpurwala, Mohamed W. Hassan, and Peter Athanas. Hardware accelerated SAT solvers—A survey. *Journal of Parallel and Distributed Computing*, 106:170–184, 2017. URL: http://www.sciencedirect.com/science/article/pii/S0743731516301903, doi:https://doi.org/10.1016/j.jpdc.2016.12.014.

[105] Marijn J. H. Heule, Matti Juhani Järvisalo, and Martin Suda. SAT Competition 2018. *Journal on Satisfiability, Boolean Modeling and Computation*, 11(1):133–154, 2019. URL: https://content.iospress.com/articles/journal-on-satisfiability-boolean-modeling-and-computation/sat190120, doi:10.3233/SAT190120.

[106] Michael Held and Richard M. Karp. A Dynamic Programming Approach to Sequencing Problems. *Journal of the Society for Industrial and Applied Mathematics*, 10(1):196–210, 1962. URL: http://www.jstor.org/stable/2098806.

[107] Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. The Travelling Salesman Problem in Bounded Degree Graphs. In *Proceedings of the 35th International Conference on Automata, Languages and Programming (ICALP'08)*, pages 198–209, 2008. URL: http://dx.doi.org/10.1007/978-3-540-70575-8_17, doi:10.1007/978-3-540-70575-8_17.

[108] Michael B. Cohen, Yin Tat Lee, and Zhao Song. Solving Linear Programs in the Current Matrix Multiplication Time. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC 2019)*, pages 938–942, New York, NY, USA, 2019. ACM. URL: http://doi.acm.org/10.1145/3313276.3316303, arXiv:1810.07896, doi:10.1145/3313276.3316303.

[109] John D. C. Little, Katta G. Murty, Dura W. Sweeney, and Caroline Karel. An Algorithm for the Traveling Salesman Problem. *Operations Research*, 11(6):972–989, 1963. URL: http://dx.doi.org/10.1287/opre.11.6.972, doi:10.1287/opre.11.6.972.

[110] Manfred Padberg and Giovanni Rinaldi. A Branch-and-Cut Algorithm for the Resolution of Large-Scale Symmetric Traveling Salesman Problems. *SIAM Review*, 33(1):60–100, 1991. URL: http://dx.doi.org/10.1137/1033004, doi:10.1137/1033004.

[111] David L. Applegate, Robert E. Bixby, Vasek Chvátal, and William J. Cook. *The Traveling Salesman Problem: A Computational Study.* Princeton University Press, Princeton, New Jersey, USA, 2006.

[112] University of Waterloo. UK49687: Shortest possible tour to nearly every pub in the United Kingdom. Retrieved 20 September 2019. URL: http://www.math.uwaterloo.ca/tsp/uk/index.html.

[113] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum Amplitude Amplification and Estimation. In Jr. Samuel J. Lomonaco, editor, *Quantum Computation and Quantum Information*, volume 305 of *AMS Contemporary Mathematics*, pages 53–74. American Mathematical Society, 2002. arXiv:quant-ph/0005055.

[114] Christoph Dürr and Peter Høyer. A Quantum Algorithm for Finding the Minimum, 1996. arXiv:quant-ph/9607014v2. arXiv:quant-ph/9607014.

[115] Andrew M. Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A. Spielman. Exponential Algorithmic Speedup by a Quantum Walk. In *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing (STOC 2003)*, pages 59–68, New York, NY, USA, 2003. ACM. URL: http://doi.acm.org/10.1145/780542.780552, arXiv:quant-ph/0209131, doi:10.1145/780542.780552.

[116] Andris Ambainis, Kaspars Balodis, Jānis Iraids, Martins Kokainis, Krišjānis Prūsis, and Jevgēnijs Vihrovs. Quantum Speedups for Exponential-Time Dynamic Programming Algorithms, 2018. arXiv:1807.05209v1. arXiv:1807.05209.

[117] Pooya Ronagh. Quantum Algorithms for Solving Dynamic Programming Problems, 2019. arXiv:1906.02229v2. arXiv:1906.02229.

[118] Nicolas J. Cerf, Lov K. Grover, and Colin P. Williams. Nested quantum search and structured problems. *Physical Review A*, 61:032303, 2000. URL: https://link.aps.org/doi/10.1103/PhysRevA.61.032303, arXiv:quant-ph/9806078, doi:10.1103/PhysRevA.61.032303.

[119] Edward Farhi and Sam Gutmann. Quantum computation and decision trees. *Physical Review A*, 58:915–928, 1998. URL: https://link.aps.org/doi/10.1103/PhysRevA.58.915, arXiv:quant-ph/9706062, doi:10.1103/PhysRevA.58.915.

[120] Martin Fürer. Solving NP-Complete Problems with Quantum Search. In Eduardo Sany Laber, Claudson Bornstein, Loana Tito Nogueira, and Luerbio Faria, editors, *LATIN 2008: Theoretical Informatics*, pages 784–792, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[121] Aleksandrs Belovs. Quantum Walks and Electric Networks, 2013. arXiv:1302.3143v1. arXiv:1302.3143.

[122] A. Belovs, A. Childs, S. Jeffery, R. Kothari, and F. Magniez. Time-Efficient Quantum Walks for 3-Distinctness. In *Proceedings of the 40th International Conference on Automata, Languages and Programming (ICALP'13)*, pages 105–122, 2013. URL: http://dx.doi.org/10.1007/978-3-642-39206-1_10, doi:10.1007/978-3-642-39206-1_10.

[123] Simon Apers, András Gilyén, and Stacey Jeffery. A Unified Framework of Quantum Walk Search, 2019. arXiv:1912.04233v1. arXiv:1912.04233.

[124] Stephen Piddock. Quantum walk search algorithms and effective resistance, 2019. arXiv:1912.04196v1. arXiv:1912.04196.

[125] Rina Dechter. Enhancement schemes for constraint processing: Backjumping, learning, and cutset decomposition. *Artificial Intelligence*, 41(3):273–312, 1990. URL: http://www.sciencedirect.com/science/article/pii/0004370290900463, doi:http://dx.doi.org/10.1016/0004-3702(90)90046-3.

[126] Andris Ambainis and Martins Kokainis. Quantum Algorithm for Tree Size Estimation, with Applications to Backtracking and 2-player Games. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC 2017)*, pages 989–1002, New York, NY, USA, 2017. ACM. URL: http://doi.acm.org/10.1145/3055399.3055444, arXiv:1704.06774, doi:10.1145/3055399.3055444.

[127] Michael Jarret and Kianna Wan. Improved quantum backtracking algorithms using effective resistance estimates. *Physical Review A*, 97:022337, 2018. URL: https://link.aps.org/doi/10.1103/PhysRevA.97.022337, arXiv:1711.05295, doi:10.1103/PhysRevA.97.022337.

[128] Arnab Das and Bikas K. Chakrabarti. Colloquium: Quantum annealing and analog quantum computation. *Reviews of Modern Physics*, 80:1061–1081, 2008. URL: https://link.aps.org/doi/10.1103/RevModPhys.80.1061, arXiv:0801.2193, doi:10.1103/RevModPhys.80.1061.

[129] C. C. McGeoch, R. Harris, S. P. Reinhardt, and P. I. Bunyk. Practical Annealing-Based Quantum Computing. *Computer*, 52(6):38–46, 2019. doi:10.1109/MC.2019.2908836.

[130] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum Computation by Adiabatic Evolution. Technical Report MIT-CTP-2936, Massachusetts Institute of Technology, 2000. arXiv:quant-ph/0001106.

[131] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, Joshua Lapan, Andrew Lundgren, and Daniel Preda. A Quantum Adiabatic Evolution Algorithm Applied to Random Instances of an NP-Complete Problem. *Science*, 292(5516):472–475, 2001. URL: https://science.sciencemag.org/content/292/5516/472, arXiv:quant-ph/0104129, doi:10.1126/science.1057726.

[132] Roberto Desimone, Paul Warburton, Yan-Long Fang, Ashley Montanaro, Stephen Piddock, Sheir Yarkoni, Andy Mason, Cathy White, and Tudor Popa. Quantum computing algorithms for optimised planning and scheduling applications (QCAPS). Technical Report QCAPS/133087/Final Report/D008/v1.0, Plantagenet Systems, University College London, University of Bristol, D-Wave Systems and BT Research, 2018. URL: https://everythingquantum.files.wordpress.com/2018/11/d008-final-report-qcaps-project-v1-0.pdf.

[133] Roman Martoňák, Giuseppe E. Santoro, and Erio Tosatti. Quantum annealing of the traveling-salesman problem. *Physical Review E*, 70:057701, 2004. URL: http://link.aps.org/doi/10.1103/PhysRevE.70.057701, arXiv:cond-mat/0402330, doi:10.1103/PhysRevE.70.057701.

[134] J. A. Barker. A quantum-statistical Monte Carlo method; path integrals with boundary conditions. *The Journal of Chemical Physics*, 70(6):2914–2918, 1979. URL: http://scitation.aip.org/content/aip/journal/jcp/70/6/10.1063/1.437829, doi:http://dx.doi.org/10.1063/1.437829.

[135] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953. URL: http://scitation.aip.org/content/aip/journal/jcp/21/6/10.1063/1.1699114, doi:http://dx.doi.org/10.1063/1.1699114.

[136] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal*, 49(2):291–307, 1970. doi:10.1002/j.1538-7305.1970.tb01770.x.

[137] Olivier C. Martin and Steve W. Otto. Combining simulated annealing with local search heuristics. *Annals of Operations Research*, 63(1):57–75, 1996. URL: http://dx.doi.org/10.1007/BF02601639, doi:10.1007/BF02601639.

[138] Hongwei Chen, Xi Kong, Bo Chong, Gan Qin, Xianyi Zhou, Xinhua Peng, and Jiangfeng Du. Experimental demonstration of a quantum annealing algorithm for the traveling salesman problem in a nuclear-magnetic-resonance quantum simulator. *Physical Review A*, 83:032314, 2011. URL: http://link.aps.org/doi/10.1103/PhysRevA.83.032314, doi:10.1103/PhysRevA.83.032314.

[139] Bettina Heim, Ethan W. Brown, Dave Wecker, and Matthias Troyer. Designing Adiabatic Quantum Optimization: A Case Study for the Traveling Salesman Problem, 2017. arXiv:1702.06248v1. arXiv:1702.06248.

[140] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A Quantum Approximate Optimization Algorithm Applied to a Bounded Occurrence Constraint Problem. Technical Report MIT-CTP/4628, Centre for Theoretical Physics, Massachusetts Institute of Technology, 2014. arXiv:1412.6062.

[141] Boaz Barak, Ankur Moitra, Ryan O'Donnell, Prasad Raghavendra, Oded Regev, David Steurer, Luca Trevisan, Aravindan Vijayaraghavan, David Witmer, and John Wright. Beating the Random Assignment on Constraint Satisfaction Problems of Bounded Degree. In Naveen Garg, Klaus Jansen, Anup Rao, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2015)*, volume 40 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 110–123, Dagstuhl, Germany, 2015. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: http://drops.dagstuhl.de/opus/volltexte/2015/5298, arXiv:1505.03424, doi:10.4230/LIPIcs.APPROX-RANDOM.2015.110.

[142] Takanori Akiyama and Takao Nishizeki. NP-completeness of the Hamiltonian cycle problem for bipartite graphs. *Journal of Information processing*, 3(2):73–76, 1980.

[143] M. R. Garey, D. S. Johnson, and R. Endre Tarjan. The planar hamiltonian circuit problem is np-complete. *SIAM Journal on Computing*, 5(4):704–714, 1976. doi:10.1137/0205049.

[144] Kazuo Iwama and Takuya Nakashima. An Improved Exact Algorithm for Cubic Graph TSP. In Guohui Lin, editor, *Proceedings of the 13th Annual International Computing and Combinatorics Conference (COCOON 2007)*, pages 108–117, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. URL: http://dx.doi.org/10.1007/978-3-540-73545-8_13, doi:10.1007/978-3-540-73545-8_13.

[145] Maciej Liśkiewicz and Martin R. Schuster. A new upper bound for the traveling salesman problem in cubic graphs. *Journal of Discrete Algorithms*, 27:1–20, 2014. URL: http://www.sciencedirect.com/science/article/pii/S1570866714000057, arXiv:1207.4694, doi:http://dx.doi.org/10.1016/j.jda.2014.02.001.

[146] Hans L. Bodlaender, Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Information and Computation*, 243:86–111, 2015. URL: http://www.sciencedirect.

com/science/article/pii/S0890540114001606, doi:http://dx.doi.org/10.1016/j.ic.2014.12.008.

[147] M. Cygan, J. Nederlof, M. Pilipczuk, M. Pilipczuk, J. van Rooij, and J. Wojtaszczyk. Solving Connectivity Problems Parameterized by Treewidth in Single Exponential Time. In *Proceedings of the IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS 2011)*, pages 150–159, 2011. URL: http://dx.doi.org/10.1109/FOCS.2011.23, arXiv:1103.0534, doi:10.1109/FOCS.2011.23.

[148] A. Björklund. Determinant sums for undirected Hamiltonicity. *SIAM Journal on Computing*, 43:280–299, 2014. URL: http://dx.doi.org/10.1137/110839229, arXiv:1008.0541, doi:10.1137/110839229.

[149] Sebastian Dörn. Quantum algorithms for graph traversals and related problems. In *Computation and Logic in the Real World: Proceedings of the Third Conference on Computability in Europe (CiE 2007)*, pages 123–131, 2007.

[150] Gilles Brassard and Peter Høyer. An Exact Quantum Polynomial-Time Algorithm for Simon's problem. In *Proceedings of the Fifth Israeli Symposium on Theory of Computing and Systems (ISTCS 1997)*, pages 12–23, 1997. arXiv:quant-ph/9704027, doi:10.1109/ISTCS.1997.595153.

[151] Salvatore Mandrà, Gian Giacomo Guerreschi, and Alán Aspuru-Guzik. Faster than classical quantum algorithm for dense formulas of exact satisfiability and occupation problems. *New Journal of Physics*, 18(7):073003, 2016. URL: http://stacks.iop.org/1367-2630/18/i=7/a=073003, arXiv:1512.00859.

[152] Ashley Montanaro. Quantum speedup of branch-and-bound algorithms, 2019. arXiv:1906.10375v1. arXiv:1906.10375.

[153] Ashley Montanaro. Quantum speedup of Monte Carlo methods. *Proceedings of the Royal Society A*, 471:20150301, 2015. URL: https://royalsocietypublishing.org/doi/10.1098/rspa.2015.0301, arXiv:1504.06987, doi:10.1098/rspa.2015.0301.

[154] Earl Campbell, Ankur Khurana, and Ashley Montanaro. Applying quantum algorithms to constraint satisfaction problems. *Quantum*, 3:167, 2019. arXiv:1810.05582, doi:10.22331/q-2019-07-18-167.

[155] K. Wiesner. The careless use of language in quantum information, 2017. arXiv:1705.06768. arXiv:1705.06768.

[156] Carmen Palacios-Berraquero, Leonie Mueck, and Divya M. Persaud. Instead of 'supremacy' use 'quantum advantage'. *Nature*, 576:213, 2019. doi:10.1038/d41586-019-03781-0.

[157] Larry Stockmeyer. The Complexity of Approximate Counting. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing (STOC 1983)*, pages 118–126, New York, NY, USA, 1983. ACM. URL: http://doi.acm.org/10.1145/800061.808740, doi:10.1145/800061.808740.

[158] Seinosuke Toda. PP is as Hard as the Polynomial-Time Hierarchy. *SIAM Journal on Computing*, 20(5):865–877, 1991. doi:10.1137/0220053.

[159] Madhur Tulsiani. Information and coding theory, university of chicago, 2017. URL: https://ttic.uchicago.edu/~madhurt/courses/infotheory2017/.

[160] Scott Aaronson and Lijie Chen. Complexity-theoretic Foundations of Quantum Supremacy Experiments. In *Proceedings of the 32nd Computational Complexity Conference (CCC 2017)*, pages 22:1–22:67, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. `arXiv:1612.05903`, `doi:10.4230/LIPIcs.CCC.2017.22`.

[161] Alexander M. Dalzell. Lower bounds on the classical simulation of quantum circuits for quantum supremacy. S.b. thesis, Massachusetts Institute of Technology, 2017. URL: `https://dspace.mit.edu/handle/1721.1/111859`.

[162] Alexander M. Dalzell, Aram W. Harrow, Dax Enshan Koh, and Rolando L. La Placa. How many qubits are needed for quantum computational supremacy? Technical Report MIT-CTP/5019, Centre for Theoretical Physics, Massachusetts Institute of Technology, 2018. `arXiv:1805.05224`.

[163] Tomoyuki Morimae and Suguru Tamaki. Additive-error fine-grained quantum supremacy. Technical Report YITP-19-120, Yukawa Institute for Theoretical Physics, Kyoto University, 2019. `arXiv:1912.06336`.

[164] Dominik Hangleiter, Martin Kliesch, Jens Eisert, and Christian Gogolin. Sample Complexity of Device-Independently Certified "Quantum Supremacy". *Physical Review Letters*, 122:210502, 2019. URL: `https://link.aps.org/doi/10.1103/PhysRevLett.122.210502`, `arXiv:1812.01023`, `doi:10.1103/PhysRevLett.122.210502`.

[165] Daniel James Shepherd. *Quantum Complexity: restrictions on algorithms and architectures.* PhD thesis, University of Bristol, 2009. `arXiv:1005.1425`.

[166] Dan Shepherd and Michael J. Bremner. Temporally unstructured quantum computation. *Proceedings of the Royal Society A*, 465:1413—1439, 2009. `arXiv:0809.0847`, `doi:10.1098/rspa.2008.0443`.

[167] Michael J. Bremner, Richard Jozsa, and Dan Shepherd. Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. *Proceedings of the Royal Society A*, 467:459—472, 2011. `arXiv:1005.1407`, `doi:10.1098/rspa.2010.0301`.

[168] Michael J. Bremner, Ashley Montanaro, and Dan J. Shepherd. Average-Case Complexity Versus Approximate Simulation of Commuting Quantum Computations. *Physical Review Letters*, 117:080501, 2016. URL: `https://link.aps.org/doi/10.1103/PhysRevLett.117.080501`, `arXiv:1504.07999`, `doi:10.1103/PhysRevLett.117.080501`.

[169] Michael J. Bremner, Ashley Montanaro, and Dan J. Shepherd. Achieving quantum supremacy with sparse and noisy commuting quantum computations. *Quantum*, 1:8, 2017. `arXiv:1610.01808`, `doi:10.22331/q-2017-04-25-8`.

[170] Scott Aaronson and Sam Gunn. On the Classical Hardness of Spoofing Linear Cross-Entropy Benchmarking, 2019. arXiv:1910.12085. `arXiv:1910.12085`.

[171] Mark Fox. *Quantum Optics: An Introduction.* Oxford University Press, Oxford, United Kingdom, 2006.

[172] Christopher Gerry and Peter Knight. *Introductory Quantum Optics.* Cambridge University Press, Cambridge, United Kingdom, 2012.

[173] M. D. Eisaman, J. Fan, A. Migdall, and S. V. Polyakov. Invited review article: Single-photon sources and detectors. *Review of Scientific Instruments*, 82(7):071101, 2011. `doi:10.1063/1.3610677`.

[174] Chandra M Natarajan, Michael G Tanner, and Robert H Hadfield. Superconducting nanowire single-photon detectors: physics and applications. *Superconductor Science and Technology*, 25(6):063001, 2012. URL: https://doi.org/10.1088%2F0953-2048%2F25%2F6%2F063001, arXiv:1204.5560, doi:10.1088/0953-2048/25/6/063001.

[175] A. Hurwitz. Über die Erzeugung der Invarianten durch Integration. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, 1897:71–90, 1897. URL: http://eudml.org/doc/58378.

[176] Michael Reck, Anton Zeilinger, Herbert J. Bernstein, and Philip Bertani. Experimental realization of any discrete unitary operator. *Physical Review Letters*, 73:58–61, 1994. URL: https://link.aps.org/doi/10.1103/PhysRevLett.73.58, doi:10.1103/PhysRevLett.73.58.

[177] William R. Clements, Peter C. Humphreys, Benjamin J. Metcalf, W. Steven Kolthammer, and Ian A. Walmsley. Optimal design for universal multiport interferometers. *Optica*, 3(12):1460–1465, 2016. URL: http://www.osapublishing.org/optica/abstract.cfm?URI=optica-3-12-1460, arXiv:1603.08788, doi:10.1364/OPTICA.3.001460.

[178] Ludwig Zehnder. Ein neuer Interferenzrefraktor. *Zeitschrift für Instrumentenkunde*, 11:275—-285, 1891.

[179] Ludwig Mach. Ueber einen Interferenzrefraktor. *Zeitschrift für Instrumentenkunde*, 11:89—-93, 1892.

[180] C. K. Hong, Z. Y. Ou, and L. Mandel. Measurement of subpicosecond time intervals between two photons by interference. *Physical Review Letters*, 59:2044–2046, 1987. URL: https://link.aps.org/doi/10.1103/PhysRevLett.59.2044, doi:10.1103/PhysRevLett.59.2044.

[181] Nicolás Quesada, Juan Miguel Arrazola, and Nathan Killoran. Gaussian boson sampling using threshold detectors. *Physical Review A*, 98:062322, 2018. URL: https://link.aps.org/doi/10.1103/PhysRevA.98.062322, arXiv:1807.01639, doi:10.1103/PhysRevA.98.062322.

[182] Stefan Scheel and Stefan Yoshi Buhmann. Macroscopic Quantum Electrodynamics - concepts and applications. *Acta Physica Slovaca*, 58(5):675–809, 2008. arXiv:0902.3586.

[183] L.G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979. URL: http://www.sciencedirect.com/science/article/pii/0304397579900446, doi:https://doi.org/10.1016/0304-3975(79)90044-6.

[184] Scott Aaronson. A linear-optical proof that the permanent is #P-hard. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 467(2136):3393–3405, 2011. URL: https://royalsocietypublishing.org/doi/abs/10.1098/rspa.2011.0232, arXiv:1109.1674, doi:10.1098/rspa.2011.0232.

[185] Mark A. Itzler, Xudong Jiang, Mark Entwistle, Krystyna Slomkowski, Alberto Tosi, Fabio Acerbi, Franco Zappa, and Sergio Cova. Advances in ingaasp-based avalanche diode single photon detectors. *Journal of Modern Optics*, 58(3-4):174–200, 2011. doi:10.1080/09500340.2010.547262.

[186] Alex Arkhipov and Greg Kuperberg. The bosonic birthday paradox. *Geometry & Topology Monographs*, 18:1–7, 2012. arXiv:1106.0849, doi:10.2140/gtm.2012.18.1.

[187] C. Gogolin, M. Kliesch, L. Aolita, and J. Eisert. Boson-Sampling in the light of sample complexity, 2013. arXiv:1306.3995. arXiv:1306.3995.

[188] Scott Aaronson and Alex Arkhipov. Bosonsampling is Far from Uniform. *Quantum Information & Computation*, 14(15-16):1383–1423, 2014. URL: http://dl.acm.org/citation.cfm?id=2685179.2685186, arXiv:1309.7460.

[189] Nicolò Spagnolo, Chiara Vitelli, Marco Bentivegna, Daniel J. Brod, Andrea Crespi, Fulvio Flamini, Sandro Giacomini, Giorgio Milani, Roberta Ramponi, Paolo Mataloni, Roberto Osellame, Ernesto F. Galvão, and Fabio Sciarrino. Experimental validation of photonic boson sampling. *Nature Photonics*, 8:615, 2014. URL: http://dx.doi.org/10.1038/nphoton.2014.135, arXiv:1311.1622, doi:10.1038/nphoton.2014.135.

[190] Iris Agresti, Niko Viggianiello, Fulvio Flamini, Nicolò Spagnolo, Andrea Crespi, Roberto Osellame, Nathan Wiebe, and Fabio Sciarrino. Pattern Recognition Techniques for Boson Sampling Validation. *Physical Review X*, 9:011013, 2019. URL: https://link.aps.org/doi/10.1103/PhysRevX.9.011013, arXiv:1712.06863, doi:10.1103/PhysRevX.9.011013.

[191] Han-Sen Zhong, Yuan Li, Wei Li, Li-Chao Peng, Zu-En Su, Yi Hu, Yu-Ming He, Xing Ding, Weijun Zhang, Hao Li, Lu Zhang, Zhen Wang, Lixing You, Xi-Lin Wang, Xiao Jiang, Li Li, Yu-Ao Chen, Nai-Le Liu, Chao-Yang Lu, and Jian-Wei Pan. 12-Photon Entanglement and Scalable Scattershot Boson Sampling with Optimal Entangled-Photon Pairs from Parametric Down-Conversion. *Physical Review Letters*, 121:250505, 2018. URL: https://link.aps.org/doi/10.1103/PhysRevLett.121.250505, arXiv:1810.04823, doi:10.1103/PhysRevLett.121.250505.

[192] Jacques Carolan, Jasmin D. A. Meinecke, Peter J. Shadbolt, Nicholas J. Russell, Nur Ismail, Kerstin Wörhoff, Terry Rudolph, Mark G. Thompson, Jeremy L. O'Brien, Jonathan C. F. Matthews, and Anthony Laing. On the experimental verification of quantum complexity in linear optics. *Nature Photonics*, 8(8):621–626, 2014. arXiv:1311.2913, doi:10.1038/nphoton.2014.152.

[193] Malte C. Tichy, Klaus Mayer, Andreas Buchleitner, and Klaus Mølmer. Stringent and Efficient Assessment of Boson-Sampling Devices. *Physical Review Letters*, 113:020502, 2014. URL: https://link.aps.org/doi/10.1103/PhysRevLett.113.020502, arXiv:1312.3080, doi:10.1103/PhysRevLett.113.020502.

[194] Mattia Walschaers, Jack Kuipers, Juan-Diego Urbina, Klaus Mayer, Malte Christopher Tichy, Klaus Richter, and Andreas Buchleitner. Statistical benchmark for BosonSampling. *New Journal of Physics*, 18(3):032001, 2016. URL: https://doi.org/10.1088%2F1367-2630%2F18%2F3%2F032001, arXiv:1410.8547, doi:10.1088/1367-2630/18/3/032001.

[195] Mattia Walschaers. *Efficient Quantum Transport*. PhD thesis, Albert-Ludwigs-Universität Freiburg, 2016. URL: https://freidok.uni-freiburg.de/data/11065.

[196] Taira Giordani, Fulvio Flamini, Matteo Pompili, Niko Viggianiello, Nicolò Spagnolo, Andrea Crespi, Roberto Osellame, Nathan Wiebe, Mattia Walschaers, Andreas Buchleitner, and Fabio Sciarrino. Experimental statistical signature of many-body quantum interference. *Nature Photonics*, 12:173, 2018. doi:10.1038/s41566-018-0097-4.

[197] Rodney Loudon. *The Quantum Theory of Light*. Oxford University Press, Oxford, United Kingdom, 2000.

[198] Andreas Christ and Christine Silberhorn. Limits on the deterministic creation of pure single-photon states using parametric down-conversion. *Physical Review A*, 85:023829, 2012. URL: `https://link.aps.org/doi/10.1103/PhysRevA.85.023829`, `arXiv:1111.4095`, `doi:10.1103/PhysRevA.85.023829`.

[199] Regina Kruse, Craig S. Hamilton, Linda Sansoni, Sonja Barkhofen, Christine Silberhorn, and Igor Jex. Detailed study of gaussian boson sampling. *Physical Review A*, 100:032326, 2019. URL: `https://link.aps.org/doi/10.1103/PhysRevA.100.032326`, `arXiv:1801.07488`, `doi:10.1103/PhysRevA.100.032326`.

[200] Francesco Guerra. *Quantum Field Theory and Renormalization Theory in the Early Scientific Activity of Eduardo R. Caianiello*, pages 93–108. Springer Milan, Milano, 2006. `doi:10.1007/88-470-0472-1_8`.

[201] Juan Miguel Arrazola and Thomas R. Bromley. Using gaussian boson sampling to find dense subgraphs. *Physical Review Letters*, 121:030503, 2018. URL: `https://link.aps.org/doi/10.1103/PhysRevLett.121.030503`, `arXiv:1803.10730`, `doi:10.1103/PhysRevLett.121.030503`.

[202] Stephen D. Bartlett and Barry C. Sanders. Requirement for quantum computation. *Journal of Modern Optics*, 50(15-17):2331–2340, 2003. URL: `https://www.tandfonline.com/doi/abs/10.1080/09500340308233564`, `arXiv:quant-ph/0302125`, `doi:10.1080/09500340308233564`.

[203] Han-Sen Zhong, Li-Chao Peng, Yuan Li, Yi Hu, Wei Li, Jian Qin, Dian Wu, Weijun Zhang, Hao Li, Lu Zhang, Zhen Wang, Lixing You, Xiao Jiang, Li Li, Nai-Le Liu, Jonathan P. Dowling, Chao-Yang Lu, and Jian-Wei Pan. Experimental Gaussian Boson sampling. *Science Bulletin*, 64(8):511–515, 2019. URL: `http://www.sciencedirect.com/science/article/pii/S2095927319301938`, `arXiv:1905.00170`, `doi:https://doi.org/10.1016/j.scib.2019.04.007`.

[204] Peter P. Rohde. Boson sampling with photons of arbitrary spectral structure. *Physical Review A*, 91:012307, 2015. URL: `https://link.aps.org/doi/10.1103/PhysRevA.91.012307`, `arXiv:1410.3979`, `doi:10.1103/PhysRevA.91.012307`.

[205] V. S. Shchesnovich. Partial indistinguishability theory for multiphoton experiments in multiport devices. *Physical Review A*, 91:013844, 2015. URL: `https://link.aps.org/doi/10.1103/PhysRevA.91.013844`, `arXiv:1410.1506`, `doi:10.1103/PhysRevA.91.013844`.

[206] Malte C. Tichy. Sampling of partially distinguishable bosons and the relation to the multidimensional permanent. *Physical Review A*, 91:022316, 2015. URL: `https://link.aps.org/doi/10.1103/PhysRevA.91.022316`, `arXiv:1410.7687`, `doi:10.1103/PhysRevA.91.022316`.

[207] V. Tamma and S. Laibacher. Boson sampling with non-identical single photons. *Journal of Modern Optics*, 63(1):41–45, 2016. `arXiv:1512.05579`, `doi:10.1080/09500340.2015.1088096`.

[208] Adrian J. Menssen, Alex E. Jones, Benjamin J. Metcalf, Malte C. Tichy, Stefanie Barz, W. Steven Kolthammer, and Ian A. Walmsley. Distinguishability and Many-Particle Interference. *Physical Review Letters*, 118:153603, 2017. URL: `https://link.aps.org/doi/10.1103/PhysRevLett.118.153603`, `arXiv:1609.09804`, `doi:10.1103/PhysRevLett.118.153603`.

[209] V. S. Shchesnovich. Sufficient condition for the mode mismatch of single photons for scalability of the boson-sampling computer. *Physical Review A*, 89:022333, 2014. URL: `https://link.aps.org/doi/10.1103/PhysRevA.89.022333`, `arXiv:1311.6796`, `doi:10.1103/PhysRevA.89.022333`.

[210] Peter P. Rohde and Timothy C. Ralph. Error tolerance of the boson-sampling model for linear optics quantum computing. *Physical Review A*, 85:022332, 2012. URL: `https://link.aps.org/doi/10.1103/PhysRevA.85.022332`, `arXiv:1111.2426`, `doi:10.1103/PhysRevA.85.022332`.

[211] Scott Aaronson and Daniel J. Brod. BosonSampling with lost photons. *Physical Review A*, 93:012335, 2016. URL: `https://link.aps.org/doi/10.1103/PhysRevA.93.012335`, `arXiv:1510.05245`, `doi:10.1103/PhysRevA.93.012335`.

[212] Vincenzo Tamma. Sampling of bosonic qubits. *International Journal of Quantum Information*, 12(07n08):1560017, 2014. `arXiv:1506.04948`, `doi:10.1142/S0219749915600175`.

[213] Vincenzo Tamma and Simon Laibacher. Multiboson Correlation Interferometry with Arbitrary Single-Photon Pure States. *Physical Review Letters*, 114:243601, 2015. URL: `https://link.aps.org/doi/10.1103/PhysRevLett.114.243601`, `arXiv:1410.8121`, `doi:10.1103/PhysRevLett.114.243601`.

[214] Simon Laibacher and Vincenzo Tamma. From the Physics to the Computational Complexity of Multiboson Correlation Interference. *Physical Review Letters*, 115:243605, 2015. URL: `https://link.aps.org/doi/10.1103/PhysRevLett.115.243605`, `arXiv:1507.01541`, `doi:10.1103/PhysRevLett.115.243605`.

[215] Vincenzo Tamma and Simon Laibacher. Multi-boson correlation sampling. *Quantum Information Processing*, 15(3):1241–1262, 2016. `arXiv:1512.05605`, `doi:10.1007/s11128-015-1177-8`.

[216] Simon Laibacher and Vincenzo Tamma. Symmetries and entanglement features of inner-mode-resolved correlations of interfering nonidentical photons. *Physical Review A*, 98:053829, 2018. URL: `https://link.aps.org/doi/10.1103/PhysRevA.98.053829`, `arXiv:1706.05578`, `doi:10.1103/PhysRevA.98.053829`.

[217] Xu-Jie Wang, Bo Jing, Peng-Fei Sun, Chao-Wei Yang, Yong Yu, Vincenzo Tamma, Xiao-Hui Bao, and Jian-Wei Pan. Experimental Time-Resolved Interference with Multiple Photons of Different Colors. *Physical Review Letters*, 121:080501, 2018. URL: `https://link.aps.org/doi/10.1103/PhysRevLett.121.080501`, `arXiv:1803.04696`, `doi:10.1103/PhysRevLett.121.080501`.

[218] Venkata Vikram Orre, Elizabeth A. Goldschmidt, Abhinav Deshpande, Alexey V. Gorshkov, Vincenzo Tamma, Mohammad Hafezi, and Sunil Mittal. Interference of Temporally Distinguishable Photons Using Frequency-Resolved Detection. *Physical Review Letters*, 123:123603, 2019. URL: `https://link.aps.org/doi/10.1103/PhysRevLett.123.123603`, `arXiv:1904.03222`, `doi:10.1103/PhysRevLett.123.123603`.

[219] R. B. A. Adamson, P. S. Turner, M. W. Mitchell, and A. M. Steinberg. Detecting hidden differences via permutation symmetries. *Physical Review A*, 78:033832, 2008. URL: https://link.aps.org/doi/10.1103/PhysRevA.78.033832, arXiv:quant-ph/0612081, doi:10.1103/PhysRevA.78.033832.

[220] Hubert de Guise, Si-Hui Tan, Isaac P. Poulin, and Barry C. Sanders. Coincidence landscapes for three-channel linear optical networks. *Physical Review A*, 89:063819, 2014. URL: https://link.aps.org/doi/10.1103/PhysRevA.89.063819, arXiv:1402.2391, doi:10.1103/PhysRevA.89.063819.

[221] Peter S. Turner. Postselective quantum interference of distinguishable particles, 2016. arXiv:1608.05720. arXiv:1608.05720.

[222] Stasja Stanisic and Peter S. Turner. Discriminating distinguishability. *Physical Review A*, 98:043839, 2018. URL: https://link.aps.org/doi/10.1103/PhysRevA.98.043839, arXiv:1806.01236, doi:10.1103/PhysRevA.98.043839.

[223] J. W. Silverstone, D. Bonneau, J. L. O'Brien, and M. G. Thompson. Silicon quantum photonics. *IEEE Journal of Selected Topics in Quantum Electronics*, 22(6):390–402, 2016. arXiv:1707.02334, doi:10.1109/JSTQE.2016.2573218.

[224] Lawrence M. Rosenfeld, Dominic A. Sulway, Gary F. Sinclair, Vikas Anant, Mark G. Thompson, John G. Rarity, and Joshua W. Silverstone. Mid-infrared quantum optics in silicon, 2019. arXiv:1906.10158v1. arXiv:1906.10158.

[225] Ludovico Latmiral, Nicolò Spagnolo, and Fabio Sciarrino. Towards quantum supremacy with lossy scattershot boson sampling. *New Journal of Physics*, 18(11):113008, 2016. URL: https://doi.org/10.1088%2F1367-2630%2F18%2F11%2F113008, arXiv:1610.02279, doi:10.1088/1367-2630/18/11/113008.

[226] Hui Wang, Wei Li, Xiao Jiang, Y.-M. He, Y.-H. Li, X. Ding, M.-C. Chen, J. Qin, C.-Z. Peng, C. Schneider, M. Kamp, W.-J. Zhang, H. Li, L.-X. You, Z. Wang, J. P. Dowling, S. Höfling, Chao-Yang Lu, and Jian-Wei Pan. Toward Scalable Boson Sampling with Photon Loss. *Physical Review Letters*, 120:230502, 2018. URL: https://link.aps.org/doi/10.1103/PhysRevLett.120.230502, arXiv:1801.08282, doi:10.1103/PhysRevLett.120.230502.

[227] Anthony Leverrier and Raúl García-Patrón. Analysis of circuit imperfections in BosonSampling. *Quantum Information and Computation*, 15:0489–0512, 2015. arXiv:1309.4687, doi:10.26421/QIC15.5-6.

[228] Alex Arkhipov. Bosonsampling is robust against small errors in the network matrix. *Physical Review A*, 92:062326, 2015. URL: https://link.aps.org/doi/10.1103/PhysRevA.92.062326, arXiv:1412.2516, doi:10.1103/PhysRevA.92.062326.

[229] Daniel J. Brod. Complexity of simulating constant-depth BosonSampling. *Physical Review A*, 91:042316, 2015. URL: https://link.aps.org/doi/10.1103/PhysRevA.91.042316, arXiv:1412.6788, doi:10.1103/PhysRevA.91.042316.

[230] Abhinav Deshpande, Bill Fefferman, Minh C. Tran, Michael Foss-Feig, and Alexey V. Gorshkov. Dynamical Phase Transitions in Sampling Complexity. *Physical Review Letters*, 121:030501, 2018. URL: https://link.aps.org/doi/10.1103/PhysRevLett.121.030501, arXiv:1703.05332, doi:10.1103/PhysRevLett.121.030501.

[231] Nishad Maskara, Abhinav Deshpande, Minh C. Tran, Adam Ehrenberg, Bill Fefferman, and Alexey V. Gorshkov. Complexity phase diagram for interacting and long-range bosonic hamiltonians, 2019. arXiv:1906.04178. arXiv:1906.04178.

[232] J. Kitaygorsky, J. Zhang, A. Verevkin, A. Sergeev, A. Korneev, V. Matvienko, P. Kouminov, K. Smirnov, B. Voronov, G. Gol'tsman, and R. Sobolewski. Origin of dark counts in nanostructured NbN single-photon detectors. *IEEE Transactions on Applied Superconductivity*, 15(2):545–548, 2005. doi:10.1109/TASC.2005.849914.

[233] V. S. Shchesnovich. Conditions for an experimental Boson-sampling computer to disprove the Extended Church-Turing thesis, 2014. arXiv:1403.4459v6. arXiv:1403.4459.

[234] Gil Kalai and Guy Kindler. Gaussian Noise Sensitivity and BosonSampling, 2014. arXiv:1409.3093v2. arXiv:1409.3093.

[235] Alex Neville, Chris Sparrow, Raphael Clifford, Eric Johnston, Patrick M. Birchall, Ashley Montanaro, and Anthony Laing. Classical boson sampling algorithms with superior performance to near-term experiments. *Nature Physics*, 13:1153–1157, 2017. URL: http://dx.doi.org/10.1038/nphys4270, arXiv:1705.00686, doi:10.1038/nphys4270.

[236] Sonia Buckley, Kelley Rivoire, and Jelena Vučković. Engineered quantum dot single-photon sources. *Reports on Progress in Physics*, 75(12):126503, nov 2012. URL: https://doi.org/10.1088%2F0034-4885%2F75%2F12%2F126503, doi:10.1088/0034-4885/75/12/126503.

[237] Mark Jerrum, Alistair Sinclair, and Eric Vigoda. A polynomial-time Approximation Algorithm for the Permanent of a Matrix with Nonnegative Entries. *J. ACM*, 51(4):671–697, 2004. URL: http://doi.acm.org/10.1145/1008731.1008738, doi:10.1145/1008731.1008738.

[238] Mark Huber and Jenny Law. Fast Approximation of the Permanent for Very Dense Problems. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2008)*, pages 681–689, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics. URL: http://dl.acm.org/citation.cfm?id=1347082.1347157.

[239] William Fulton and Joe Harris. *Representation Theory: A First Course*, volume 129 of *Graduate Texts in Mathematics*. Springer, 2004.

[240] D. J. Rowe, M. J. Carvalho, and J. Repka. Dual pairing of symmetry and dynamical groups in physics. *Reviews of Modern Physics*, 84(2):711–757, 2012. URL: http://link.aps.org/doi/10.1103/RevModPhys.84.711, arXiv:1207.0148, doi:10.1103/RevModPhys.84.711.

[241] Arne Alex, Matthias Kalus, Alan Huckleberry, and Jan von Delft. A numerical algorithm for the explicit calculation of SU(N) and SL(N,C) Clebsch–Gordan coefficients. *Journal of Mathematical Physics*, 52(2):023507, 2011. URL: http://dx.doi.org/10.1063/1.3521562, arXiv:1009.0437, doi:10.1063/1.3521562.

[242] Ish Dhand, Barry C. Sanders, and Hubert de Guise. Algorithms for SU(n) boson realizations and D-functions. *Journal of Mathematical Physics*, 56(11):111705, 2015. URL: http://dx.doi.org/10.1063/1.4935433, arXiv:1507.06274, doi:10.1063/1.4935433.

[243] Christopher M. Dawson and Michael A. Nielsen. The Solovay-Kitaev Algorithm. *Quantum Information & Computation*, 6(1):81–95, 2006. URL: http://dl.acm.org/citation.cfm?id=2011679.2011685, arXiv:quant-ph/0505030.

[244] Michael Nielsen and Isaac Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, Cambridge, United Kingdom, 2010.

[245] William M. Kirby and Frederick W. Strauch. A practical quantum algorithm for the Schur transform. *Quantum Information and Computation*, 18:0721–0742, 2018. arXiv:1709.07119, doi:10.26421/QIC18.9-10.

[246] Hari Krovi. An efficient high dimensional quantum Schur transform. *Quantum*, 3:122, 2019. arXiv:1804.00055, doi:10.22331/q-2019-02-14-122.

[247] G. Vidal, J. I. Latorre, P. Pascual, and R. Tarrach. Optimal minimal measurements of mixed states. *Physical Review A*, 60:126–135, 1999. URL: https://link.aps.org/doi/10.1103/PhysRevA.60.126, arXiv:quant-ph/9812068, doi:10.1103/PhysRevA.60.126.

[248] M. Keyl and R. F. Werner. Estimating the spectrum of a density operator. *Physical Review A*, 64:052311, 2001. URL: https://link.aps.org/doi/10.1103/PhysRevA.64.052311, arXiv:quant-ph/0102027, doi:10.1103/PhysRevA.64.052311.

[249] Masahito Hayashi and Keiji Matsumoto. Quantum universal variable-length source coding. *Physical Review A*, 66:022311, 2002. URL: https://link.aps.org/doi/10.1103/PhysRevA.66.022311, arXiv:quant-ph/0202001, doi:10.1103/PhysRevA.66.022311.

[250] Masahito Hayashi and Keiji Matsumoto. Simple construction of quantum universal variable-length source coding. *Quantum Information and Computation*, 2:519–529, 2002. arXiv:quant-ph/0209124, doi:10.26421/QIC2.s.

[251] Stephen D. Bartlett, Terry Rudolph, and Robert W. Spekkens. Classical and Quantum Communication without a Shared Reference Frame. *Physical Review Letters*, 91:027901, 2003. URL: https://link.aps.org/doi/10.1103/PhysRevLett.91.027901, arXiv:quant-ph/0302111, doi:10.1103/PhysRevLett.91.027901.

[252] Vojtěch Havlíček and Sergii Strelchuk. Quantum Schur Sampling Circuits can be Strongly Simulated. *Physical Review Letters*, 121:060505, 2018. URL: https://link.aps.org/doi/10.1103/PhysRevLett.121.060505, arXiv:1809.05171, doi:10.1103/PhysRevLett.121.060505.

[253] Stephen P. Jordan. Permutational Quantum Computing. *Quantum Information and Computation*, 10:0470–0497, 2010. arXiv:0906.2508, doi:10.26421/QIC10.5-6.

[254] Vojtěch Havlíček, Sergii Strelchuk, and Kristan Temme. Classical algorithm for quantum SU(2) Schur sampling. *Physical Review A*, 99:062336, 2019. URL: https://link.aps.org/doi/10.1103/PhysRevA.99.062336, doi:10.1103/PhysRevA.99.062336.

[255] Bill Fefferman and Christopher Umans. On the Power of Quantum Fourier Sampling. In *11th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2016)*, pages 1:1–1:19, 2016. arXiv:1507.05592, doi:10.4230/LIPIcs.TQC.2016.1.

[256] Maarten Van den Nest. Simulating quantum computers with probabilistic methods. *Quantum Information and Computation*, 11:0784–0812, 2011. arXiv:0911.1624, doi:10.26421/QIC11.9-10.

[257] Martin Schwarz and Maarten Van den Nest. Simulating Quantum Circuits with Sparse Output Distributions. Technical Report TR13-154, Electronic Colloquium on Computational Complexity, Weizmann Institute of Science, 2013. URL: `https://eccc.weizmann.ac.il/report/2013/154/`, `arXiv:1310.6749`.

[258] Charles Thomas Hepler. On the complexity of computing characters of finite groups. Master's thesis, University of Calgary, 1994. URL: `https://prism.ucalgary.ca/handle/1880/45530`.

[259] Vissarion Fisikopoulos and Luis Peñaranda. Faster geometric algorithms via dynamic determinant computation. *Computational Geometry*, 54:1–16, 2016. URL: `http://www.sciencedirect.com/science/article/pii/S0925772115001261`, `arXiv:1206.7067`, `doi:https://doi.org/10.1016/j.comgeo.2015.12.001`.

[260] W. Hartmann. On the complexity of immanants. *Linear and Multilinear Algebra*, 18(2):127–140, 1985. `doi:10.1080/03081088508817680`.

[261] Peter Bürgisser. The Computational Complexity to Evaluate Representations of General Linear Groups. *SIAM Journal on Computing*, 30(3):1010–1022, 2000. `doi:10.1137/S0097539798367892`.

[262] Jean-Luc Brylinski and Ranee Brylinski. Complexity and Completeness of Immanants, 2003. arXiv:cs/0301024. `arXiv:arXiv:cs/0301024`.

[263] Stephan Mertens and Cristopher Moore. The Complexity of the Fermionant and Immanants of Constant Width. *Theory of Computing*, 9(6):273–282, 2013. URL: `http://www.theoryofcomputing.org/articles/v009a006`, `arXiv:1110.1821`, `doi:10.4086/toc.2013.v009a006`.

[264] Peter Bürgisser. The Computational Complexity of Immanants. *SIAM Journal on Computing*, 30(3):1023–1040, 2000. `doi:10.1137/S0097539798367880`.

[265] A. Barenco. A universal two-bit gate for quantum computation. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 449(1937):679–683, 1995. URL: `http://rspa.royalsocietypublishing.org/content/449/1937/679`, `arXiv:quant-ph/9505016`, `doi:10.1098/rspa.1995.0066`.

[266] D. J. Rowe, B. C. Sanders, and H. de Guise. Representations of the Weyl group and Wigner functions for SU(3). *Journal of Mathematical Physics*, 40(7):3604–3615, 1999. URL: `http://dx.doi.org/10.1063/1.532911`, `arXiv:math-ph/9811012`, `doi:10.1063/1.532911`.

[267] Rajendra Bhatia. *Matrix Analysis*. Graduate Texts in Mathematics. Springer-Verlag New York, Heidelberg, Germany, 1997. `doi:10.1007/978-1-4612-0653-8`.

[268] Roe Goodman and Nolan R. Wallach. *Symmetry, Representations, and Invariants*. Graduate Texts in Mathematics. Springer, Heidelberg, Germany, 2009. `doi:10.1007/978-0-387-79852-3`.

[269] M. Hamermesh. *Group Theory and Its Application to Physical Problems*. Addison-Wesley, 1962.

[270] Keisuke Fujii, Hirotada Kobayashi, Tomoyuki Morimae, Harumichi Nishimura, Shuhei Tamate, and Seiichiro Tani. Impossibility of Classically Simulating One-Clean-Qubit Computation, 2014. arXiv:1409.6777. `arXiv:1409.6777`.

[271] Tomoyuki Morimae and Takeshi Koshiba. Classical simulatability of the one clean qubit model, 2014. arXiv:1405.6840v2. `arXiv:1405.6840`.

[272] Tomoyuki Morimae, Keisuke Fujii, and Joseph F. Fitzsimons. Hardness of Classically Simulating the One-Clean-Qubit Model. *Physical Review Letters*, 112:130502, 2014. URL: `https://link.aps.org/doi/10.1103/PhysRevLett.112.130502`, `arXiv:1312.2496`, `doi:10.1103/PhysRevLett.112.130502`.

[273] Aram W. Harrow and Michael A. Nielsen. Robustness of quantum gates in the presence of noise. *Physical Review A*, 68:012308, 2003. URL: `https://link.aps.org/doi/10.1103/PhysRevA.68.012308`, `arXiv:quant-ph/0301108`, `doi:10.1103/PhysRevA.68.012308`.

[274] S. Virmani, Susana F. Huelga, and Martin B. Plenio. Classical simulability, entanglement breaking, and quantum computation thresholds. *Physical Review A*, 71:042328, 2005. URL: `https://link.aps.org/doi/10.1103/PhysRevA.71.042328`, `arXiv:quant-ph/0408076`, `doi:10.1103/PhysRevA.71.042328`.

[275] H. Buhrman, R. Cleve, M. Laurent, N. Linden, A. Schrijver, and F. Unger. New Limits on Fault-Tolerant Quantum Computation. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006)*, pages 411–419, 2006. `arXiv:quant-ph/0604141`, `doi:10.1109/FOCS.2006.50`.

[276] Guifré Vidal. Efficient Classical Simulation of Slightly Entangled Quantum Computations. *Physical Review Letters*, 91:147902, 2003. URL: `https://link.aps.org/doi/10.1103/PhysRevLett.91.147902`, `arXiv:quant-ph/0301063`, `doi:10.1103/PhysRevLett.91.147902`.

[277] Animesh Datta and Guifre Vidal. Role of entanglement and correlations in mixed-state quantum computation. *Physical Review A*, 75:042310, 2007. URL: `https://link.aps.org/doi/10.1103/PhysRevA.75.042310`, `arXiv:quant-ph/0611157`, `doi:10.1103/PhysRevA.75.042310`.

[278] N. Anari, L. Gurvits, S. O. Gharan, and A. Saberi. Simply Exponential Approximation of the Permanent of Positive Semidefinite Matrices. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS 2017)*, pages 914–925, 2017. `arXiv:1704.03486`, `doi:10.1109/FOCS.2017.89`.

[279] S. L. Braunstein, C. M. Caves, R. Jozsa, N. Linden, S. Popescu, and R. Schack. Separability of Very Noisy Mixed States and Implications for NMR Quantum Computing. *Physical Review Letters*, 83:1054–1057, 1999. URL: `https://link.aps.org/doi/10.1103/PhysRevLett.83.1054`, `arXiv:quant-ph/9811018`, `doi:10.1103/PhysRevLett.83.1054`.

[280] P. Rungta, W.J. Munro, K. Nemoto, P. Deuar, G. J. Milburn, and C. M. Caves. Qudit Entanglement. In Howard J. Carmichael, Roy J. Glauber, and Marlan O. Scully, editors, *Directions in Quantum Optics: A Collection of Papers Dedicated to the Memory of Dan Walls Including Papers Presented at the TAMU-ONR Workshop Held at Jackson, Wyoming, USA, 26–30 July 1999*, pages 149–164, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg. `arXiv:quant-ph/0001075`, `doi:10.1007/3-540-40894-0_14`.

[281] Kai Chen and Ling-An Wu. The generalized partial transposition criterion for separability of multipartite quantum states. *Physics Letters A*, 306(1):14–20, 2002. URL: `http://www.sciencedirect.com/science/article/pii/S0375960102015384`, `arXiv:quant-ph/0208058`, `doi:https://doi.org/10.1016/S0375-9601(02)01538-4`.

[282] Heinz-Peter Breuer. Optimal Entanglement Criterion for Mixed Quantum States. *Physical Review Letters*, 97:080501, 2006. URL: https://link.aps.org/doi/10.1103/PhysRevLett.97.080501, arXiv:quant-ph/0605036, doi:10.1103/PhysRevLett.97.080501.

[283] Max Tillmann, Si-Hui Tan, Sarah E. Stoeckl, Barry C. Sanders, Hubert de Guise, René Heilmann, Stefan Nolte, Alexander Szameit, and Philip Walther. Generalized Multiphoton Quantum Interference. *Physical Review X*, 5:041015, 2015. URL: https://link.aps.org/doi/10.1103/PhysRevX.5.041015, arXiv:1403.3433, doi:10.1103/PhysRevX.5.041015.

[284] R. Arratia and L. Gordon. Tutorial on large deviations for the binomial distribution. *Bulletin of Mathematical Biology*, 51(1):125–131, 1989. doi:10.1007/BF02458840.

[285] Sergei Slussarenko, Morgan M. Weston, Helen M. Chrzanowski, Lynden K. Shalm, Varun B. Verma, Sae Woo Nam, and Geoff J. Pryde. Unconditional violation of the shot-noise limit in photonic quantum metrology. *Nature Photonics*, 11:700–703, 2017. arXiv:1707.08977, doi:10.1038/s41566-017-0011-5.

[286] F. Marsili, V. B. Verma, J. A. Stern, S. Harrington, A. E. Lita, T. Gerrits, I. Vayshenker, B. Baek, M. D. Shaw, R. P. Mirin, and S. W. Nam. Detecting single infrared photons with 93*Nature Photonics*, 7:210–214, 2013. arXiv:1209.5774, doi:10.1038/nphoton.2013.13.

[287] Dalia P. Ornelas-Huerta, Alexander N. Craddock, Elizabeth A. Goldschmidt, Andrew J. Hachtel, Yidan Wang, P. Bienias, Alexey V. Gorshkov, Steve L. Rolston, and James V. Porto. On-demand indistinguishable single photons from an efficient and pure source based on a Rydberg ensemble, 2020. arXiv:2003.02202v1. arXiv:2003.02202.

[288] Baida Zhang, Huiquan Wang, Yang Wang, Yong Liu, Junjie Wu, Xuejun Yang, and Xianmin Jin. A benchmark test of boson sampling on Tianhe-2 supercomputer. *National Science Review*, 5(5):715–720, 2018. URL: https://dx.doi.org/10.1093/nsr/nwy079, arXiv:1606.05836, doi:10.1093/nsr/nwy079.

[289] Malte C. Tichy and Klaus Mølmer. Extending exchange symmetry beyond bosons and fermions. *Physical Review A*, 96:022119, 2017. URL: https://link.aps.org/doi/10.1103/PhysRevA.96.022119, arXiv:1702.03528, doi:10.1103/PhysRevA.96.022119.

[290] A. I. Barvinok. Computational complexity of immanents and representations of the full linear group. *Functional Analysis and Its Applications*, 24(2):144–145, 1990. doi:10.1007/BF01077707.

[291] David G. Glynn. The permanent of a square matrix. *European Journal of Combinatorics*, 31(7):1887–1891, 2010. URL: http://www.sciencedirect.com/science/article/pii/S0195669810000211, doi:https://doi.org/10.1016/j.ejc.2010.01.010.

[292] Jelmer J. Renema. Simulability of Imperfect Gaussian and Superposition Boson Sampling, 2019. arXiv:1911.10112. arXiv:1911.10112.

[293] Bujiao Wu, Bin Cheng, Fei Jia, Jialin Zhang, Man-Hong Yung, and Xiaoming Sun. Speedup in classical simulation of gaussian boson sampling. *Science Bulletin*, 65(10):832 – 841, 2020. URL: http://www.sciencedirect.com/science/article/pii/S2095927320300748, arXiv:1908.10070, doi:https://doi.org/10.1016/j.scib.2020.02.012.

[294] Nicolás Quesada and Juan Miguel Arrazola. The Classical Complexity of Gaussian Boson Sampling, 2019. arXiv:1908.08068. arXiv:1908.08068.

[295] V. S. Shchesnovich. Noise in boson sampling and the threshold of efficient classical simulatability. *Physical Review A*, 100:012340, 2019. URL: https://link.aps.org/doi/10.1103/PhysRevA.100.012340, arXiv:1902.02258, doi:10.1103/PhysRevA.100.012340.

[296] V. S. Shchesnovich. personal communication, 2019.

[297] Haoyu Qi, Daniel J. Brod, Nicolás Quesada, and Raúl García-Patrón. Regimes of classical simulability for noisy gaussian boson sampling. *Physical Review Letters*, 124:100502, Mar 2020. URL: https://link.aps.org/doi/10.1103/PhysRevLett.124.100502, arXiv:1905.12075, doi:10.1103/PhysRevLett.124.100502.

[298] E. Knill, R. Laflamme, and G. J. Milburn. A scheme for efficient quantum computation with linear optics. *Nature*, 409(6816):46–52, 2001. doi:10.1038/35051009.

[299] Mercedes Gimeno-Segovia, Pete Shadbolt, Dan E. Browne, and Terry Rudolph. From Three-Photon Greenberger-Horne-Zeilinger States to Ballistic Universal Quantum Computation. *Physical Review Letters*, 115:020502, 2015. URL: https://link.aps.org/doi/10.1103/PhysRevLett.115.020502, arXiv:1410.3720, doi:10.1103/PhysRevLett.115.020502.

[300] Sergey Bravyi, Dan Browne, Padraic Calpin, Earl Campbell, David Gosset, and Mark Howard. Simulation of quantum circuits by low-rank stabilizer decompositions. *Quantum*, 3:181, 2019. arXiv:1808.00128, doi:10.22331/q-2019-09-02-181.