mathematics

Special Issue Reprint

Quantum Computing Algorithms and Computational Complexity

Edited by
Fernando L. Pelayo and Mauro Mezzini

www.mdpi.com/journal/mathematics

MDPI

Quantum Computing Algorithms and Computational Complexity

Quantum Computing Algorithms and Computational Complexity

Editors

Fernando L. Pelayo
Mauro Mezzini

MDPI • Basel • Beijing • Wuhan • Barcelona • Belgrade • Manchester • Tokyo • Cluj • Tianjin

Editors
Fernando L. Pelayo
University of
Castilla-La Mancha,
Albacete, Spain

Mauro Mezzini
Roma Tre University,
Rome, Italy

Editorial Office
MDPI
St. Alban-Anlage 66
4052 Basel, Switzerland

This is a reprint of articles from the Special Issue published online in the open access journal *Mathematics* (ISSN 2227-7390) (available at: https://www.mdpi.com/si/mathematics/Quantum_Comput_Algorithm_Comput_Complex).

For citation purposes, cite each article independently as indicated on the article page online and as indicated below:

LastName, A.A.; LastName, B.B.; LastName, C.C. Article Title. *Journal Name* **Year**, *Volume Number*, Page Range.

ISBN 978-3-0365-7746-3 (Hbk)
ISBN 978-3-0365-7747-0 (PDF)

Contents

About the Editors

Fernando L. Pelayo

Fernando L. Pelayo has been an Associate Professor at Universidad de Castilla- La Mancha and Tutor Professor at UNED since 2000. He has published about sixty research papers, with one third of them being in JCR journals. He obtained the best university paper award in the IFIP TC10: International Embedded Systems Symposium (IESS'2005). He has been a Guest Editor for the Special Issues "Quantum Computing Algorithms and Computational Complexity" and "Quantum Computing Algorithms and Quantum Computing Simulators", both belonging to the Mathematics and Computer Science sections of the Mathematics Journal. He chaired the First European Performance Engineering Workshop (EPEW'2004) and the Third and Fourth Workshops on Formal Methods in the Development of Software (WFMDS'2013–2014). His research interests include quantum computing, algorithm complexity, formal models of concurrency and discrete dynamical systems.

Mauro Mezzini

Mauro Mezzini has been an Assistant Professor of Computer Science at the Roma Tre University since 2012. Previously, he was Adjunct Professor at La Sapienza University (2005–2011) and Project Manager at Telecom Italia (1990–2012). Some of his works have been published in SIAM Journals on computing, aCm transaction on information and system security, discrete applied mathematics and theoretical computer science. He has been a Guest Editor for the Special Issues "Quantum Computing Algorithms and Computational Complexity" and "Quantum Computing Algorithms and Quantum Computing Simulators", both belonging to the Mathematics and Computer Science sections of the Mathematics journal. His research interests include quantum computing, machine learning, graph theory and algorithm, databases and discrete mathematics.

mathematics

MDPI

Editorial

Preface to the Special Issue on "Quantum Computing Algorithms and Computational Complexity"

Fernando L. Pelayo [1,*] and Mauro Mezzini [2,*]

[1] Computing Systems Department, Faculty of Computer Science Engineering, University of Castilla-La Mancha, 02071 Albacete, Spain
[2] Department of Education, Roma Tre University, 00185 Rome, Italy
* Correspondence: fernandol.pelayo@uclm.es (F.L.P.); mauro.mezzini@uniroma3.it (M.M.)

1. Call for Papers

In 1982, Richard Feynman stated that in order to simulate quantum systems, we would rather go for a sort of brand-new powered quantum processor instead of a classical one. Since then, Quantum Computation has been growing in terms of both architectural issues associated with such quantum computers and the algorithms that can be run with them. All this has attracted much interest from the computer science community.

Just to mention some facts, it is obvious that the intrinsic parallelism that comes with the superposition of quantum states together with interference features provides us with a very good perspective to deal with heavy computational problems, such as encrypting/decrypting tasks or studying quantum issues of matter.

Quantum computing is a hot field of research at the intersection of mathematics, computer science, and physics that promises to significantly revolutionise many aspects of the technology industry such as medicine, machine learning, artificial intelligence, cryptography, and operations research to name a few. Investors and governments from all over the world promote its development considering that it is crucial and of strategic importance for countries, companies and, therefore, society as a whole. The huge investments in resources to develop quantum computing by countries such as China, India, the United States, Russia, and so on only confirms this reality.

This Special Issue was mainly concerned with quantum algorithms, the mathematics underlying them, and those complexity issues arising from them.

2. Published Papers

This is a Special Issue of *Mathematics* belonging to the section "Mathematics and Computer Science", which was closed on 30 June 2022.

A total of 13 papers were submitted to it, of which 7 have been accepted. This represents an acceptance ratio of 53.8%. The average time for accepted papers to be published is 43.4 days.

The main contributions of these seven papers are the following:

Two of these seven papers are focused on improving the performance by means of quantum algorithms over the best instances of classical ones:

- Yan Li, Dapeng Hao, Yang Xu, and Kinkeung Lai, in their paper "A Fast Quantum Image Component Labeling Algorithm" [1], improve the performance of one of the most time-consuming tasks within digital image processing. They propose a fast quantum image component labelling algorithm that improves the efficiency of its classical computing counterpart. The time and spatial complexities are $O(n^2)$ and $O(n)$, respectively.
- Kamil Khadiev, Artem Ilikaev, and Jevgenijs Vihrovs, in their paper "Quantum Algorithms for Some Strings Problems Based on Quantum String Comparator" [2], improve the performance of three classical problems over strings: "sorting of n strings of length

Citation: Pelayo, F.L.; Mezzini, M. Preface to the Special Issue on "Quantum Computing Algorithms and Computational Complexity". *Mathematics* **2022**, *10*, 4032. https://doi.org/10.3390/math10214032

Received: 19 October 2022
Accepted: 21 October 2022
Published: 30 October 2022

k", "the most frequent string search problem", and "searching intersection of two sequences of strings". Based on the quantum procedure for comparing two strings of length k in $O(\sqrt{k})$ queries, they are able to reduce time complexities, thus moving the factor k to $\sqrt{k}$ in all its instances as parameters.

Another four papers deal with former quantum algorithms (or part of them) for which some improvements or different perspectives have been addressed:

- Daniil Rabinovich, Richik Sengupta, Ernesto Campos, Vishwanathan Akshay, and Jacob Biamonte, in their paper "Progress towards Analytically Optimal Angles in Quantum Approximate Optimisation": [3], present proof that the optimal quantum approximate optimisation algorithm's (QAOA) parameters for a single layer reduce to one free variable and that optimal angles can be recovered in the thermodynamic limit. They also demonstrate that conditions for vanishing gradients of the overlap function are so similar that reveals a linear relationship between both parameters regardless the number of qubits.
- Tieyu Zhao, Tianyu Yang, and Yingying Chi, in their paper "Quantum Weighted Fractional Fourier Transform" [4], present a reformulation of the weighted fractional Fourier transform (WFRFT) and prove its unitarity, thereby proposing a quantum weighted fractional Fourier transform (QWFRFT) which seems to be very usable for signal processing.
- Mauro Mezzini, Jose J. Paulet, Fernando Cuartero, Hernan I. Cruz, and Fernando L. Pelayo, in their paper "On the Amplitude Amplification of Quantum States Corresponding to the Solutions of the Partition Problem" [5], present a quantum computing piece of code that increases the amplitude of the states corresponding to the solutions of the partition problem by a factor of almost two. Unfortunately, this algorithm cannot be iterated in contrast to the amplitude amplification part of Grover's algorithm.
- Serena Di Giorgio and Paulo Mateus, in their paper "On the Complexity of Finding the Maximum Entropy Compatible Quantum State" [6], follow Jaynes' principle in order to characterize a compatible density operator with maximum entropy. They first stated that comparing the entropy of compatible density operators is complete for the quantum computational complexity class QSZK, even for the simplest case of three chains. They show that for the case of quantum Markov chains and trees, there exists a procedure which is polynomial in the number of subsystems that constructs the maximum entropy compatible density operator. An extension of the Chow–Liu algorithm to the same subclass of quantum states is also provided.

Finally, there is a paper that researches a classical Operational Research problem by means of quantum annealing:

- Saul Gonzalez-Bermejo, Guillermo Alonso-Linaje, and Parfait Atchade-Adelomou, in their paper "GPS: A New TSP Formulation for Its Generalizations Type QUBO" [7], propose a new Quadratic Unconstrained Binary Optimization (QUBO) formulation of the Travelling Salesman Problem (TSP) with a smaller number of necessary variables, together with a thorough study of the constraints and their management. This study includes a practical test over D-wave quantum annealers platform.

As Guest Editors of this Special Issue, we would like to thank all the authors who make contributions on these quite conceptually similar fields of research.

We also would like to thank all the reviewers for their big effort in developing so constructive reports that contribute to improve the quality and quantity of the results provided within this Special Issue on "Quantum Computing Algorithms and Computational Complexity".

We hope that the research papers published in this Special Issue promote more extensive research and lend further support to quantum computing. We are believers of the wide and crucial effect that quantum computing can have in our society, from the domain of energy-efficient computing, through to high-performance computing, up to the management of many of the most challenging problems which still remain open.

Author Contributions: All authors have contributed equally. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, Y.; Hao, D.; Xu, Y.; Lai, K. A Fast Quantum Image Component Labeling Algorithm. *Mathematics* **2022**, *10*, 2718. [CrossRef]
2. Khadiev, K.; Ilikaev, A.; Vihrovs, J. Quantum Algorithms for Some Strings Problems Based on Quantum String Comparator. *Mathematics* **2022**, *10*, 377. [CrossRef]
3. Rabinovich, D.; Sengupta, R.; Campos, E.; Akshay, V.; Biamonte, J. Progress towards Analytically Optimal Angles in Quantum Approximate Optimisation. *Mathematics* **2022**, *10*, 2601. [CrossRef]
4. Zhao, T.; Yang, T.; Chi, Y. Quantum Weighted Fractional Fourier Transform. *Mathematics* **2022**, *10*, 1896. [CrossRef]
5. Mezzini, M.; Paulet, J.J.; Cuartero, F.; Cruz, H.I.; Pelayo, F.L. On the Amplitude Amplification of Quantum States Corresponding to the Solutions of the Partition Problem. *Mathematics* **2021**, *9*, 2027. [CrossRef]
6. Di Giorgio, S.; Mateus, P. On the Complexity of Finding the Maximum Entropy Compatible Quantum State. *Mathematics* **2021**, *9*, 193. [CrossRef]
7. Gonzalez-Bermejo, S.; Alonso-Linaje, G.; Atchade-Adelomou, P. GPS: A New TSP Formulation for Its Generalizations Type QUBO. *Mathematics* **2022**, *10*, 416. [CrossRef]

mathematics

MDPI

Article

On the Complexity of Finding the Maximum Entropy Compatible Quantum State

Serena Di Giorgio [1,2,*,†] and Paulo Mateus [1,2,†]

[1] Departamento de Matemática, Instituto Superior Técnico, Universidade de Lisboa, Av. Rovisco Pais, 1049-001 Lisboa, Portugal; pmat@math.ist.utl.pt
[2] Instituto de Telecomunicações, Universidade de Lisboa, Av. Rovisco Pais, 1049-001 Lisboa, Portugal
* Correspondence: serena.giorgio@tecnico.ulisboa.pt
† These authors contributed equally to this work.

Abstract: Herein we study the problem of recovering a density operator from a set of compatible marginals, motivated by limitations of physical observations. Given that the set of compatible density operators is not singular, we adopt Jaynes' principle and wish to characterize a compatible density operator with maximum entropy. We first show that comparing the entropy of compatible density operators is complete for the quantum computational complexity class QSZK, even for the simplest case of 3-chains. Then, we focus on the particular case of quantum Markov chains and trees and establish that for these cases, there exists a procedure polynomial in the number of subsystems that constructs the maximum entropy compatible density operator. Moreover, we extend the Chow–Liu algorithm to the same subclass of quantum states.

Keywords: quantum Markov chains; maximum von Neumann entropy; QSZK-completeness

Citation: Di Giorgio, S.; Mateus, P. On the Complexity of Finding the Maximum Entropy Compatible Quantum State. *Mathematics* **2021**, *9*, 193. https://doi.org/10.3390/math9020193

Received: 15 December 2020
Accepted: 15 January 2021
Published: 19 January 2021

1. Introduction

Quantum tomography [1] allows us to associate a unique quantum state over a finite-dimensional Hilbert space provided that multiple copies of the quantum system are available, together with a complete set of measurements. Observe that when the degrees of freedom increase, the amount of resources for performing the latter grows exponentially. However, physically relevant phenomena are entirely determined by few-body correlations—their Hamiltonians are in general highly local [2]—and when we restrict ourselves to k-order dependencies, the data collection results in an exponential speed-up in the number of subsystems, leading to efficient tomography techniques [3]. Clearly, a partial dataset admits many possible compatible density operators. The overlap between (quantum) statistical mechanics and quantum information theory provides a well-established tool, entropy maximization, to dealing with the remaining degrees of freedom. By using von Neumann entropy within Jaynes' principle [4], we define a criterion to estimate density operators, maximally unbiased with regards to the provided partial information.

Problem statement.

A question that naturally arises is the following: is there an efficient and effective procedure for inferring the aforementioned quantum state? More concretely, is it possible to find a density operator describing a finite-dimensional multipartite quantum system that maximizes the von Neumann entropy under the constraints given by its few-body marginals? In this work, we focus on this problem for the case of direct correlations, that is, 2-body marginals.

The problem we address is strictly related to the (quantum) Hamiltonian learning problem [5–7]—every density operator is thermal for a determined Hamiltonian. In general, the Hamiltonian is given, and one tries to find out its properties, so the problem of its characterization is not well explored. Recent developments in (quantum) machine learning techniques [8] renewed the interest in the Hamiltonian learning problem. In [9], an effective

neural-networks approach to the problem has been proposed, and an upper bound, which is polynomial in the number of qudits, has been established for its sample complexity [10].

One of the main reasons for the little background on the problem at hand relies on the computational hardness of well-known problems that reduce to it. First, the *quantum marginal problem* [11,12], that consists of determining whether a set of marginal quantum states has a global density operator compatible with them, and for which a solution is known just in some particular cases [13–15]. Then, the classical inference problem of a probability distribution via graphical models [16] also leads to a maximum entropy estimation. Density operators naturally encompass classical probability distributions on the finite-dimensional setup; therefore, when considering direct correlations between the subsystems, the hardness results for classical graph-inference should be considered. In particular, the classical problem is well known for being computationally hard [17,18]. The only cases for which a polynomial procedure is known is when the direct correlations have the structure of a tree (undirected acyclic graph), and moreover, for this case, there exists an efficient procedure for determining the most likely tree from a general graph—the Chow–Liu algorithm [19]. The speed-up is due to the Markov condition, which can be directly inferred from the graphical structure, resulting in the factorization of the maximum entropy joint probability distribution. Many attempts have been made for developing appropriate operatorial graphical models [20,21], but none of them naturally encodes the desired generalization of Markovianity. For obtaining a compression of the learning procedure, further conditions [22] need to be verified.

In this article, we study the aforementioned problem restricted to a tree-structured set of marginals density operators and the abstraction of the Chow–Liu algorithm. Namely, we focus on two questions. First, is the inference efficiency limited to mutually commuting (and acyclic connected) density operators, which encode classical probability distributions? Second, can we determine a broader set of density operators for which an extended efficient procedure is similarly achieved?

Contributions of the paper.

We start by showing that comparing the entropies of 3-chains—quantum states compatible with two given 2-body marginals—is a complete problem for the class QSZK [23–25]—Quantum Statistical Zero Knowledge. This result hints that finding the maximum entropy compatible state given two marginals should be not feasible, even for a quantum computer [26], at least by performing an entropy-monotonic step-by-step optimization into the compatibility space of the provided marginals. Indeed, the complexity class QSZK, originally defined by J. Watrous in 2002 [25], collects promise problems whose true instances can be verified by a zero knowledge quantum proof between two quantum entities, generalizing the class Statistical Zero Knowledge (SZK) to quantum computers. Natural complete problems for the class represent its hardness, including distinguishing two quantum states (Problem 4) and determining their quantum entropy difference [27].

Next, we restrict the class of quantum states to make the problem feasible. We consider quantum Markov trees, states for which each 3-subchains form a quantum Markov chain [28]. In this case, we show that the maximum entropy compatible problem is in P, and also that there exists a polynomial-time quantum circuit that constructs the maximal entropy compatible state. Finally, we use this result to extend the Chow–Liu algorithm [19] for quantum states whose all 3-subchains are quantum Markov chains. The results obtained in this paper provide a natural extension of prior work [29] to the many-body scenario.

Organization of the paper.

In Section 2, we give some background and state clearly the problems we are addressing. In Section 3, we attain the hardness of comparing the entropy of a compatible chain. In Section 4, we consider the restriction of the maximum entropy problem to quantum Markov trees. There, we provide the polynomial-time solution for this case, how to construct the solution with a polynomial-quantum circuit, and the generalization of Chow–Liu algorithm. Some of the proofs are left to the appendices. Finally, we draw some conclusions and leave some open problems in Section 5.

2. Background and Problem Statement

Throughout this work, we assume all quantum states and operators to be defined over a finite dimensional Hilbert space $\mathcal{H}$ that is composed of n parts, such that $\mathcal{H} = \otimes_{i=1}^{n} \mathcal{H}_i$. We denote by $\mathcal{I}$ a collection of subsets of $\{1, \ldots n\}$ and throughout the text we call $\mathcal{I}$ the *set of marginals indexes*. Elements of $\mathcal{I}$ are denoted by J, and its complement is represented by $\overline{J}$. Given $\mathcal{I}$, we are interested in density operators that are compatible with a $\mathcal{I}$-indexed family of marginal density operators $\mathcal{C}$ where $\mathcal{C} = \{\rho_J \in \mathcal{B}(\mathcal{H}_J)\}_{J \in \mathcal{I}}$ such that

$$\mathrm{Tr}_{\overline{J \cap J'}}[\rho_J] = \mathrm{Tr}_{\overline{J \cap J'}}[\rho_{J'}] \text{ for all } J, J' \in \mathcal{I}, \tag{1}$$

where $\mathcal{H}_J = \bigotimes_{i \in J} \mathcal{H}_i$. We call each element ρ_J a *marginal density operator*. We also denote by $\mathcal{Q}(\mathcal{C}) = \{Q_J\}_{J \in \mathcal{I}}$ a family of quantum circuits such that Q_J constructs the density operator ρ_J.

The *compatibility set* $\mathrm{Comp}(\mathcal{C})$ associated to a given family of compatible marginals $\mathcal{C}$ is the set of density operators over $\mathcal{H}$ that admits as partial traces all the elements of $\mathcal{C}$, that is:

$$\mathrm{Comp}(\mathcal{C}) := \left\{\rho \in \mathcal{B}(\mathcal{H}) : \mathrm{Tr}_{\overline{J}}[\rho] = \rho_J \text{ for all } J \in \mathcal{I}\right\}. \tag{2}$$

The family $\mathcal{C}$ is said to be admissible when $\mathrm{Comp}(\mathcal{C}) \neq 0$, that is, if it admits at least one density operator whose marginals coincide with those in $\mathcal{C}$.

We start by noticing that, the problem of the admissibility for a compatible set where all marginal density operators are diagonal for the same basis—that is, density operators encoding discrete probability distributions—collapses in the classical compatible marginal problem [30]. This classical problem has been shown to be NP-complete for the three-dimensional case [31]. There are many cases for which it is solvable [32], and there is always a solution if we consider only two-body marginals (bipartite marginals) that form an acyclic graph.

The relevant case where the marginals are not diagonal for the same basis has been the target of several research works and is called the quantum compatible marginal problem. Liu showed that this problem is Quantum Merlin Arthur (QMA)-complete, that is, it is one of the hardest problem in the computational complexity class QMA [12]. The class Quantum Merlin Arthur (QMA) [33] collects promise problems whose "yes" answer can be verified by a 1-message quantum interactive proof, generalizing to the quantum realm the class NP of problems classically verifiable in poly-time.

Problem 1. *Quantum Compatible Marginal Problem (QCMP)*

- `Input:` *A family of circuits $\mathcal{Q}(\mathcal{C})$ that construct the family of marginal density operators $\mathcal{C}$.*
- `Accept:` *if $\mathcal{C}$ is admissible.*
- `Reject:` *if $\mathcal{C}$ is not admissible.*

In some cases, we know that $\mathcal{C}$ is admissible, for instance when we are promised that the marginals ρ_J are indeed partial traces of a global state. In Physics, it is reasonable to assume that we can prepare many copies of a global system, but in general, we can only partially observe it. In this case, given that we have many copies of the global system, we would be able to characterize in full detail the partial traces and know that they form an admissible set. The question now is to infer the global state with maximum entropy among those in the compatibility set. This leads to the following problem.

Problem 2. *Maximum Entropy Compatible Marginal Problem (MECMP)*

- `Input:` *A family of circuits $\mathcal{Q}(\mathcal{C})$ promised to construct an admissible $\mathcal{C}$, and a real value k.*
- `Accept:` *if there exists a $\rho \in Comp(\mathcal{C})$ such that $S(\rho) \geq k$*
- `Reject:` *otherwise.*

Given the general complexity of this problem, we focus on the more straightforward case where all sets J in $\mathcal{I}$ have two indexes. Thus, we consider that we are given a set of

compatible two-body marginals, and we want to reconstruct the maximum entropy state compatible with those marginals. For this two-body case, it is possible to construct an associated graph, where each two-body marginal denotes an edge.

Definition 1. *Let $\mathcal{C}$ be a $\mathcal{I}$-indexed family of two-body compatible marginal density operators. The associated graph $\mathcal{G}_{\mathcal{C}}$ is $(\{1,\ldots,n\}, E)$, where $(i,j) \in E$ if $\{i,j\} \in \mathcal{I}$.*

In the simplest non-trivial case, we have that $n = 3$ and $\mathcal{I} = \{\{1,2\},\{2,3\}\}$. We call this case a 3-chain. In the next section, we show that given two density operators ρ_0 and ρ_1 in the compatible set of a 3-chain, comparing who has higher entropy is QSZK-complete. We denote the subspaces $\mathcal{H}_1$, $\mathcal{H}_2$ and $\mathcal{H}_3$ by $\mathcal{H}_A$, $\mathcal{H}_B$ and $\mathcal{H}_C$, respectively.

3. Hardness of Comparing Entropy of a Compatible Chain

Ben-Aroya et al. [27] showed that, given two quantum circuits Q_0 and Q_1 that generate two mixed states ρ_0 and ρ_1, respectively, such that $|S(\rho_0) - S(\rho_1)| > \frac{1}{2}$, determining whether $S(\rho_0) > S(\rho_1)$ is QSZK-complete. Thus, they conclude that it is quite improbable that computing the von Neumann entropy of a mixed state can be done in BQP [34]. We further look into this problem by restricting to the case when ρ_0 and ρ_1 live in the same Hilbert space and have the same marginals. We state our problem as follows:

Problem 3. *3-Chain Compatible Quantum Entropy Difference (3cQED)*

- `Input:` *Two quantum circuits Q_0 and Q_1 that generate tripartite density operators ρ_0 and ρ_1, respectively, over the same Hilbert space of the form $\mathcal{H}_A \otimes \mathcal{H}_B \otimes \mathcal{H}_C$, promised that:*
  - $\mathrm{Tr}_A(\rho_0) = \mathrm{Tr}_A(\rho_1)$;
  - $\mathrm{Tr}_C(\rho_0) = \mathrm{Tr}_C(\rho_1)$;
  - $|S(\rho_0) - S(\rho_1)| \geq 1/2$;

  then,
- `Accept:` *if $S(\rho_0) - S(\rho_1) \geq 1/2$;*
- `Reject:` *if $S(\rho_1) - S(\rho_0) \geq 1/2$.*

Clearly, 3cQED is a particular case of QED, wherein the latter the Hilbert space of ρ_0 and ρ_1 does not have to be the same, nor do the densities need to be tripartite.

Obviously, 3cQED is reducible to QED, and therefore it lies in QSZK. It remains to show that it is QSZK hard. To do so, we adapt the proof of Ben-Aroya et al., and reduce $\mathrm{QSD}_{\alpha,\beta}$, natural complete problem for the class QSZK [25], to 3cQED, for $0 \leq \alpha < \beta^2 \leq 1$.

Problem 4. *Quantum state distance ($QSD_{\alpha,\beta}$) with $0 \leq \alpha < \beta^2 \leq 1$:*

- `Input:` *Two quantum circuits Q_0 and Q_1, acting on m qubits, that prepare the states ρ_0 or ρ_1 promised that*
  - *either $||\rho_0 - \rho_1||_{tr} \geq \beta$;*
  - *or $||\rho_0 - \rho_1||_{tr} \leq \alpha$;*

  then,
- `Accept:` $||\rho_0 - \rho_1||_{tr} \geq \beta$,
- `Reject:` $||\rho_0 - \rho_1||_{tr} \leq \alpha$.

In Problem 4, $||\rho_0 - \rho_1||_{tr}$ denotes the trace distance between the operators ρ_0 and ρ_1.

Theorem 1. *For any $0 \leq \alpha < \beta^2 \leq 1$, $QSD_{\alpha,\beta}$ is reducible to 3cQED.*

Proof of Theorem 1. The idea of the proof is the following. From quantum circuits Q_0 and Q_1 acting on m bits that generate, respectively, ρ_0 and ρ_1 fulfilling the promise of $\mathrm{QSD}_{\alpha,\beta}$, we are going to construct, in polynomial-time, two quantum circuits Q_0' and

Q_1' that generate tripartite density operators ρ_0' and ρ_1', fulfilling the promise of 3cQED, such that $\mathrm{QSD}_{\alpha,\beta}(Q_0, Q_1)_{NO}$ iff $\mathrm{3cQED}(Q_0', Q_1')_{NO}$.

Concretely, given circuits Q_0, Q_1, that construct ρ_0 and ρ_1, we first apply the polarization lemma (Lemma A1 in Appendix A) with $n = m$ and obtain circuits R_0 and R_1 that output density operators μ_0, μ_1, respectively. We then construct two circuits Z_0 and Z_1 as follows. Z_1 is implemented by a circuit which first applies a Hadamard gate on a single qubit b, measures b and then conditioned on the result it applies either R_0 or R_1. The output of Z_1 is $\xi_1 = \frac{1}{2}|0\rangle\langle 0| \otimes \mu_0 + \frac{1}{2}|1\rangle\langle 1| \otimes \mu_1$. Since we need to construct a tripartite system, we introduce a non-orthodox, but useful, notation ξ_1^{AC} to denote a copy of ξ_1 where the qubit part of ξ_1 belongs to the system of A and the remaining part belongs to system of C. Similarly, we denote by ξ_1^{CA} to indicate a copy of ξ_1 where the qubit part belongs to C and remaining part to A. Circuit Z_0 is the same as Z_1 except that the qubit b is traced out. The output of Z_0 is $\xi_0 = \frac{1}{2}\mu_0 + \frac{1}{2}\mu_1$. We shall denote by ξ_0^A and ξ_0^C a copy of ξ_0 belonging to the subsystem of A or C, respectively.

Finally, we denote by $|\phi^{\pm}\rangle^{AC}$ two maximally entangled states between A and C. Moreover, take $\zeta = \frac{1}{2}|\phi^+\rangle\langle\phi^+| + \frac{1}{2}|\phi^-\rangle\langle\phi^-|$ and note that $S(\zeta) = 1$. We denote by Q the circuit that prepares ζ. Consider:

- $\rho' = \xi_0^A \otimes \zeta^{AC} \otimes \xi_0^C \otimes |0\rangle\langle 0|_B$;
- $\rho'' = \xi_1^{AC} \otimes \xi_1^{CA} \otimes |0\rangle\langle 0|_B$.

Note that in ρ' the subsystem of A contains ξ_0^A and a qubit of ζ^{AC}; the subsystem of C contains ξ_0^C and the other qubit of ζ^{AC}. Moreover, in ρ'', the subsystem of A has a qubit entangled with μ_0 and μ_1 in the subsystem C (ξ_1^{AC}); and has another μ_0 and μ_1 entangled with a qubit of C (ξ_1^{CA}).

The reduction outputs the following pair of density operators (ρ', ρ'') together with the circuits that construct them, namely $Q_0' = Z_0 \otimes Z_0 \otimes Q$ and $Q_1' = Z_1 \otimes Z_1$. We ignore the construction of the state $|0\rangle\langle 0|_B$, which is trivial.

Start by observing that by tracing C from both ρ' and ρ'' we obtain $(\frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1|) \otimes (\frac{1}{2}\mu_0 + \frac{1}{2}\mu_1) \otimes |0\rangle\langle 0|$. The same state will be obtained by tracing subsystem A from both ρ' and ρ''. So, ρ' and ρ'' have compatible marginals.

Part 1

If $(Q_0, Q_1) \in (\mathrm{QSD}_{\alpha,\beta})_{NO}$ then $(Z_0 \otimes Z_0 \otimes Q, Z_1 \otimes Z_1) \in \mathrm{3cQED}_{NO}$.

We know that $\| \rho_0 - \rho_1 \|_{\mathrm{tr}} \leq \alpha$. By the Polarization lemma (Lemma A1 in Appendix A) we get $\| \mu_0 - \mu_1 \|_{\mathrm{tr}} \leq 2^{-m}$. By the joint-entropy theorem (Lemma A2),

$$S(\xi_1) = \frac{1}{2}(S(\mu_0) + S(\mu_1)) + 1. \tag{3}$$

On the other hand, ξ_0 is very close both to μ_0 and to μ_1. Specifically, $\| \xi_0 - \mu_1 \|_{\mathrm{tr}} = \left\| \frac{1}{2}\mu_0 - \frac{1}{2}\mu_1 \right\|_{\mathrm{tr}} \leq 2^{-m}$. Thus, by Fannes' inequality (Lemma A3 in Appendix A) $|S(\xi_0) - S(\mu_1)| \leq 2^{-m} \cdot \mathrm{poly}(m) \leq 0.1$, for large enough m_0. Similarly, $|S(\xi_0) - S(\mu_0)| \leq 0.1$. It follows that

$$|S(\xi_0) - \frac{1}{2}(S(\mu_0) + S(\mu_1))| \leq 0.1. \tag{4}$$

Combining the two equations we get $S(\xi_1) - S(\xi_0) \geq 0.9$. Thus, $S(\rho'') - S(\rho') \geq 2 \times 0.9 - 1 = 0.8$. Therefore, $(Z_0 \otimes Z_0 \otimes Q, Z_1 \otimes Z_1) \in \mathrm{3cQED}_{NO}$.

Part 2

If $(Q_0, Q_1) \in (\mathrm{QSD}_{\alpha,\beta})_{YES}$ then $(Z_0 \otimes Z_0 \otimes Q, Z_1 \otimes Z_1) \in \mathrm{3cQED}_{YES}$.

By the Polarization lemma (Lemma A1 in Appendix A) $\| \mu_0 - \mu_1 \|_{\mathrm{tr}} \geq 1 - 2^{-m}$. Using Lemma A5 (in Appendix A), we get that $S(\xi_0) \geq \frac{1}{2}[S(\mu_0) + S(\mu_1)] + 1 - H(\frac{1}{2} + \frac{\| \mu_0 - \mu_1 \|_{\mathrm{tr}}}{2}) \geq \frac{1}{2}[S(\mu_0) + S(\mu_1)] + 1 - H(2^{-m_0})$. By Lemma A2 (in Appendix A) we know that $S(\xi_1) = \frac{1}{2}(S(\mu_0) + S(\mu_1)) + 1$. Therefore, $S(\xi_1) - S(\xi_0) = H(2^{-m}) < 0.1$ for sufficiently large m.

In particular, $S(\rho'') - S(\rho') \leq 2 * 0.1 - 1 = -0.8$ and $(Z_0 \otimes Z_0 \otimes Q, Z_1 \otimes Z_1) \in$ 3cQED$_{YES}$. □

It follows that comparing the entropy of a set of compatible marginals is QSZK-complete, as this problem is also an instance of QED. As a consequence, we expect that finding the maximum entropy state is also generally hard, at least by performing a step by step entropy-increasing procedure. We now focus our attention on a particular sub-case in which this problem can be addressed.

4. Quantum Markov Chains and Trees

Given that the general problem of finding the maximum entropy state is hard, we focus on a well-behaved subset of density operators, namely *quantum Markov trees* (QMT)—Definition 4—which extends the notion of quantum Markov chains (QMC) [35] to the multi-partite scenario. By defining QMTs, we were able to extend the learning techniques provided by classical graphical models—Bayes composition [16] and Chow–Liu algorithm—to an enlarged set of density operators with respect to the mutually-commuting ones.

We refer to a set of two-body density operators as *tree-structured* when its associated graph—Definition 1—is a tree. In particular, we showed that given a tree-structured set of two-body marginal density operators:

- it admits a QMT in the compatibility space iff every sub-3-chain is compatible with a QMC—Theorem 3;
- the QMT coincides with the density operator that maximizes the von Neumann entropy, constrained by the provided set of two-body marginals—Corollary 1;
- defining a proper order in the graph—constructive ordering—we can construct the unique compatible QMT directly from the marginals. The Lagrange multipliers in the optimization problem are then obtained through Theorem 2.

We were then able to show that for QMTs, the MECMP is in P—Theorem 3. Moreover, given a general set of two-body marginals, we found that if all the sub-3-chains are compatible with a QMC, the *optimal-sub-tree*, which is a QMT, can be efficiently determined by generalizing the Chow–Liu learning algorithm—Theorem 5.

The main achievement consists in the exponential speed-up of the general Markov condition. For the case at hand, QMC-compatibility of every tree chain—polynomial in the number of 1-body subsystem n—implies the QMC-compatibility of every further sub-chains formed by sub-groups of nodes—exponential in n.

In order of proving the mentioned results, first we give the essential background on QMC—Section 4.1, then we formally define QMT—Section 4.2. In Section 4.3 we provide the entropic characterization of QMTs, then in Section 4.4 we derive the compatibility condition for a given set of tree-structured marginals with a QMT, and in Section 4.5 we study the MECM problem restricted to QMTs. Finally, in Section 4.6, we extend the Chow–Liu algorithm for determining the *optimal tree* when the provided set is not tree-structured.

4.1. Background on Quantum Markov Chains

We consider QMCs that rely on the Hilbert space $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B \otimes \mathcal{H}_C$ and take $\mathcal{C} = \{\rho_{\{A,B\}} \in \mathcal{B}\Big(\mathcal{H}_{\{AB\}}\Big)\}, \rho_{\{B,C\}} \in \mathcal{B}\Big(\mathcal{H}_{\{BC\}}\Big)\}$. To simplify notation, we drop the brackets and commas in the indexes and so, for instance, the partial trace $\rho_{\{A,B\}}$ is just denoted by ρ_{AB} (the same simplification is applied for the Hilbert subspaces $\mathcal{H}_{\{A,B\}}$, which are denoted just by $\mathcal{H}_{AB}$).

Recall the definition of quantum Markov chain:

Definition 2 ([36]). *A quantum Markov chain (QMC) is a 3-chain $A - B - C$ for which there exists a recovery map $\mathcal{R}_{B\to BC} : \mathcal{B}(\mathcal{H}_B) \to \mathcal{B}(\mathcal{H}_{BC})$, i.e., an arbitrary trace-preserving completely positive (CPTP) map (see, for instance, [37,38]), s.t. $\rho_{ABC} = (\mathcal{I}_A \otimes \mathcal{R}_{B\to BC})(\rho_{AB})$, where $\mathcal{I}_A$ denotes the identity map on $\mathcal{B}(\mathcal{H}_A)$.*

By definition, the recovery map must fulfill that $\mathcal{R}_{B\to BC}(\rho_B) = \rho_{BC}$.

Definition 3. *A family of QMC's $\{\rho^{(n)}_{ABC}\}_{n\in\mathbb{N}}$ is said to be constructed in polynomial time if all elements $\rho^{(n)}_{ABC}$ rely in the same (finite) Hilbert space $\mathcal{H}_A \otimes \mathcal{H}_B \otimes \mathcal{H}_C$ (that does not depend on n) and there is polynomial-time family of quantum circuits that generate both $\rho^{(n)}_{AB}$ and $\mathcal{R}^{(n)}_{B\to BC}$.*

Given that the dimension of (a polynomial-time) quantum Markov chain does not grow with n, it can be represented in matrix form in polynomial-time by multiplying all the gates involved in the circuits that generate $\rho^{(n)}_{AB}$ and $\mathcal{R}^{(n)}_{B\to BC}$. We stress that to design circuits for density operators and CPTP maps we require only an ancilla space of the same dimension of the support of these operators/maps [39]. Therefore, the number of gates is polynomial in n, but the full dimension of the space (including ancillae) does not grow with n.

From this point on, we assume that ρ_{ABC} is invertible (on its support), as invertible density operators are dense. To derive the main result of the paper, we need to establish a central lemma listing some known characterizations of QMCs. We give the proof in Appendix B.

Lemma 1. *Let ρ_{ABC} be an invertible density operator. The following four assertions are equivalent:*

1. *ρ_{ABC} is a QMC over the chain $A - B - C$.*
2. *$I_\rho(A : C|B) = 0$, where $I_\rho(A : C|B) := S(\rho_{AB}) + S(\rho_{BC}) - S(\rho_B) - S(\rho_{ABC})$.*
3. *$\mathcal{P}_{B\to BC}(X) := \rho_{BC}^{\frac{1}{2}}((\rho_B^{-\frac{1}{2}} X \rho_B^{-\frac{1}{2}}) \otimes id_C)\rho_{BC}^{\frac{1}{2}}$, is a CPTP map for any $X \in \mathcal{B}(\mathcal{H}_B)$ and preserves the partial trace ρ_{AB}.*
4. *$\log \rho_{ABC} - (\log \rho_{AB}) \otimes id_C = id_A \otimes (\log \rho_{BC}) - id_A \otimes (\log \rho_B) \otimes id_C$.*

The map $\mathcal{P}_{B\to BC}(X)$ is known as *Petz recovery map* or *transpose map*. Again, to ease notation, we drop the identities whenever they are obvious, for instance, we drop them in the expressions $\rho_{BC}^{\frac{1}{2}}((\rho_B^{-\frac{1}{2}} X \rho_B^{-\frac{1}{2}}) \otimes \mathrm{id}_C)\rho_{BC}^{\frac{1}{2}}$ to just $\rho_{BC}^{\frac{1}{2}}\rho_B^{-\frac{1}{2}} X \rho_B^{-\frac{1}{2}}\rho_{BC}^{\frac{1}{2}}$, and the same for $\log \rho_{ABC} - (\log \rho_{AB}) \otimes \mathrm{id}_C$, which we write just $\log \rho_{ABC} - \log \rho_{AB}$.

Observe that we can also recover a tripartite density operator from ρ_{BC} through $\mathcal{P}_{B\to AB}(\cdot)$:

$$\rho_{AB}^{\frac{1}{2}}\rho_B^{-\frac{1}{2}}\rho_{BC}\rho_B^{-\frac{1}{2}}\rho_{AB}^{\frac{1}{2}}, \tag{5}$$

and by uniqueness, since the von Neumann entropy is operator-concave [40,41], we have $\mathcal{P}_{B\to BC}(\rho_{AB}) = \mathcal{P}_{B\to AB}(\rho_{BC})$. However, it is not known whether, given a family of QMC that can be constructed in polynomial time via $\mathcal{P}^{(n)}_{B\to BC}(X)$, it is possible to build $\mathcal{P}^{(n)}_{B\to AB}(X)$ in polynomial-time. The next result states that solution to MECMP (Problem 2) and also QCMP (Problem 1), for 3-chains can be fully determined when a QMC belongs to the compatibility set, the proofs can be found in [29].

Lemma 2. *Given a 3-chain $\{\rho_{AB}, \rho_{BC}\}$ compatible with a QMC, say ρ_{ABC}, then the solution of the maximum entropy estimator $\widetilde{\rho}_{ABC}$ is precisely ρ_{ABC}. Moreover, the 3-chain $\{\rho_{AB}, \rho_{BC}\}$ is compatible with a QMC in $\mathcal{B}(\mathcal{H}_{ABC})$ iff $\mathrm{Tr}_A(\rho_{AB}) = \mathrm{Tr}_C(\rho_{BC})$ and the operator $\Theta_{ABC} = \rho_{BC}^{\frac{1}{2}}\rho_B^{-\frac{1}{2}}\rho_{AB}^{\frac{1}{2}}$ is normal. Moreover, if two marginals $\{\rho_{AB}, \rho_{BC}\}$ are compatible with a QMC on $\mathcal{B}(\mathcal{H}_{ABC})$, say ρ_{ABC}, then the operator Θ_{ABC} is its square root.*

4.2. Definition of Quantum Markov Trees

We are now able to extend the above result from 3-chains to a more general setting, namely to trees. From this point on, we make the following assumption.

Assumption 1. *Assume the graph $\mathcal{G}_\mathcal{C}$ associated to a Maximum Entropy Compatible Marginal Problem $\mathcal{C}$ over $\mathbb{X} = \{X_1, \ldots X_n\}$ is a tree, that is, $\mathcal{G}_\mathcal{C}$ is an acyclic connected graph over $\mathbb{X}$.*

By taking any node as a root of $\mathcal{G}_{\mathcal{C}}$, we construct an arborescence (or a directed tree). For the sake of readability, we introduce the following notation. We call a *constructive ordering of* $\mathcal{C}$ any total order compatible with the topological order of an arborescence of $\mathcal{G}_{\mathcal{C}}$. Without loss of generality, we consider a constructive order of the form $X_1 < \cdots < X_n$ and denote by $\mathcal{G}_k$ the induced subgraph of $\mathcal{G}_{\mathcal{C}}$ containing all the nodes $V_k = \{X_1, \dots X_k\}$ for $k \in \{1 \dots n\}$. We also denote by $\mathcal{C}_k$ the marginals in $\mathcal{C}$ containing nodes in $\{X_1, \dots X_k\}$ and by Y_k, for $k \geq 2$, the node in V_{k-1} connected to X_k in $\mathcal{G}_k$ (the adjacent node of X_k in $\mathcal{G}_k$). Finally, we denote by $\overline{Y_k}$ the set $V_{k-1} \setminus \{Y_k\}$, which is non-empty for $k \geq 3$.

The next result follows easily:

Proposition 1. *If $\mathcal{G}_{\mathcal{C}}$ is a tree, than all the subgraphs $\mathcal{G}_k$ are trees, and moreover, X_k is a leaf of $\mathcal{G}_k$.*

We now define a quantum Markov tree, which, as we shall see later on, generalizes the notion of Markov random field, when the underlying graph is a tree.

Definition 4. *Let $\rho \in \mathcal{B}(\mathcal{H}_{\mathbb{X}})$ with $\mathbb{X} := \{X_1, \dots X_n\}$ be an invertible density operator (over its support) and $\mathcal{C}$ is a (non-trivial) set of two-body marginals of ρ. We say that ρ is* quantum Markov tree (QMT) *or is factorizable via Petz according to $\mathcal{C}$ if its square root is such that $\rho = \Theta\Theta^\dagger = \Theta^\dagger\Theta$ where Θ admits a decomposition, for some constructive order $X_1 < \cdots < X_n$, of the form*

$$\Theta = \Delta_n \dots \Delta_3 (\rho^{\frac{1}{2}}_{X_1X_2} \otimes id_{\overline{\{X_1X_2\}}}) \tag{6}$$

with $\Delta_k = \left(\rho^{\frac{1}{2}}_{X_kY_k}\left(id_{X_k} \otimes \rho^{-\frac{1}{2}}_{Y_k}\right)\right) \otimes id_{\overline{\{X_kY_k\}}}$, for all $k = 3 \dots n$.

We note that for Equation (6) to be well defined, it must be the case that $\mathcal{G}_{\mathcal{C}}$ is a tree, that is, that we are working under Assumption 1. It is relatively simple to extend the notion to acyclic graphs (which may not be connected).

4.3. QMT as Max-Entropy Density Operator

The following result will shed some light on the relationship between Markov random fields and QMTs.

Theorem 2. *Let $\rho \in \mathcal{B}(\mathcal{H}_{\mathbb{X}})$ be an invertible density operator over (its support) and $\mathcal{C}$ is a (non-trivial) set of two-body marginals s.t. $\mathcal{G}_{\mathcal{C}}$ is a spanning tree over $\mathbb{X}$, then there exists $\rho \in \mathrm{Comp}(\mathcal{C})$ factorizable via Petz according to $\mathcal{C}$ iff there exists $\rho \in \mathcal{B}(\mathcal{H}_{\mathbb{X}})$ such that, equivalently, one of the following two hold:*

(i) $\log \rho = \sum_{\mathcal{C}} \log \rho_{X_iX_j} - \sum_{i=1}^{n} (deg(X_i) - 1) \log \rho_{X_i}$;

(ii) we have

$$\forall k = 2, \dots, n: \quad \rho_k = Tr_{\overline{Y_k}}[\rho] \text{ is s.t. } I_{\rho_k}(X_k : \overline{Y_k}|Y_k) = 0 \tag{7}$$

for some constructive ordering $X_1 < \cdots < X_n$.

Proof of Theorem 2. The proof follows by induction on k, that is, by adding one edge per node following a constructive ordering in $\mathcal{C}$. Therefore, we have that

$$\mathcal{C} = \left\{\rho_{X_kY_k} \in \mathcal{B}(\mathcal{H}_{X_kY_k}) : \ Y_k \in \{X_1, \dots, X_{k-1}\};\ k = 2, \dots, n\right\}, \tag{8}$$

The proof follows by complete induction on k.

(Basis $k = 3$): The first chain occurs when the third node is added, that is, when $k = 3$. Assume there exists $\rho_3 \in \mathrm{Comp}(\mathcal{C}_3)$ that is factorizable via Petz, i.e.,

$$\Theta_3 = \rho^{\frac{1}{2}}_3 = \rho^{\frac{1}{2}}_{X_3Y_3}\rho^{-\frac{1}{2}}_{Y_3}\rho^{\frac{1}{2}}_{\overline{Y_3}Y_3} = \rho^{\frac{1}{2}}_{X_3Y_3}\rho^{-\frac{1}{2}}_{Y_3}\rho^{\frac{1}{2}}_{X_1X_2}. \tag{9}$$

Observe that we can use Lemma 2 and so, Θ_3 is exactly the operator described in the lemma, and since it is a square root, it is normal. Then, by Lemma 1, we have the following equivalences: ρ_3 is a QMC iff $I_{\rho_3}\left(X_3 : \overline{Y_3}|Y_3\right) = 0$ iff $\log \rho_3 = \log \rho_{X_3Y_3} + \log \rho_{X_1X_2} - \log \rho_{Y_3}$. The other direction follows immediately.

(Induction step $k \longrightarrow k+1$):

Complete induction hypothesis: $\forall j = 3, \ldots, k \; \exists \rho_j \in \mathrm{Comp}\left(\mathcal{C}_j\right)$ factorizable via Petz according with $\mathcal{C}_j$ iff there exists $\rho_j \in \mathcal{B}\left(\mathcal{H}_{V_j}\right)$ such that, equivalently, one of the following two hold:

- $\log \rho_j = \sum_{\mathcal{C}_j} \log \rho_{X_iX_t} - \sum_{i=1}^{j} (\deg_{\mathcal{G}_j}(X_i) - 1) \log \rho_{X_i}$;
- $I_{\rho_j}\left(X_j : \overline{Y_j}|Y_j\right) = 0$.

Induction step: Assume there $\exists \rho_{k+1} \in \mathrm{Comp}(\mathcal{C}_{k+1})$ factorizable via Petz according with $\mathcal{C}_{k+1}$, then, our goal is to show that the following holds for ρ_{k+1}:

- $\log \rho_{k+1} = \sum_{\mathcal{C}_{k+1}} \log \rho_{X_iX_t} - \sum_{i=1}^{k+1} (\deg_{\mathcal{G}_k}(X_i) - 1) \log \rho_{X_i}$ and ,
- $I_{\rho_{k+1}}\left(X_{k+1} : \overline{Y_{k+1}}|Y_{k+1}\right) = 0$.

Therefore, assume $\exists \rho_{k+1} \in \mathrm{Comp}(\mathcal{C}_{k+1})$ factorizable via Petz, i.e.,

$$\Theta_{k+1} = \rho_{k+1}^{\frac{1}{2}} = \Delta_{k+1}\Delta_k \ldots \Delta_3 \rho_{X_1X_2}^{\frac{1}{2}} \quad \text{where} \quad \Delta_i := \rho_{X_iY_i}^{\frac{1}{2}} \rho_{Y_i}^{-\frac{1}{2}}. \tag{10}$$

Then:

$$\begin{aligned}
\rho_{k+1} &= \Theta_{k+1}\Theta_{k+1}^{\dagger} \\
&= \Delta_{k+1}\Delta_k \ldots \Delta_2 \rho_{X_1Y_1} \Delta_2^{\dagger} \ldots \Delta_k^{\dagger}\Delta_{k+1} \\
&= \rho_{X_{k+1}Y_{k+1}}^{\frac{1}{2}} \rho_{Y_{k+1}}^{-\frac{1}{2}} \rho_k \, \rho_{Y_{k+1}}^{-\frac{1}{2}} \rho_{X_{k+1}Y_{k+1}}^{\frac{1}{2}} \\
&= \Theta_{k+1}^{\dagger}\Theta_{k+1} \\
&= \rho_{X_1X_2}^{\frac{1}{2}} \Delta_3^{\dagger} \ldots \Delta_k^{\dagger}\Delta_{k+1}\Delta_{k+1}\Delta_k \ldots \Delta_3 \rho_{X_1X_2}^{\frac{1}{2}} \\
&= \rho_k^{\frac{1}{2}} \rho_{Y_{k+1}}^{-\frac{1}{2}} \rho_{X_{k+1}Y_{k+1}} \rho_{Y_{k+1}}^{-\frac{1}{2}} \rho_k^{\frac{1}{2}}.
\end{aligned} \tag{11}$$

We can use Lemma 2 on the set $\{\, \rho_{X_{k+1}Y_{k+1}}, \rho_k \,\}$ and conclude that ρ_{k+1} is a QMC in the order $X_{k+1} - \overline{Y_{k+1}} - Y_{k+1}$. Therefore, using Lemma 1, we have ρ_{k+1} is a QMC iff $I_{\rho_{k+1}}\left(X_{k+1} : \overline{Y_{k+1}}|Y_{k+1}\right) = 0$ iff

$$\begin{aligned}
\log \rho_{k+1} &= \log \rho_{X_{k+1}Y_{k+1}} + \log \rho_{\overline{Y_{k+1}}Y_{k+1}} - \log(\rho_{Y_{k+1}}) \\
&\overset{I.H.}{=} \sum_{\mathcal{C}_{k+1}} \log \rho_{X_iX_t} - \sum_{i=1}^{k+1} (\deg_{\mathcal{C}_i}(X_i) - 1) \log \rho_{X_i}.
\end{aligned} \tag{12}$$

The other direction is straightforward. Just notice that $\mathrm{Tr}_{X_{k+1}}(\rho_{k+1}) = \rho_k$, and by induction hypothesis ρ_k is compatible with $\mathcal{C}_k$, and so is ρ_{k+1}. Moreover, by construction of ρ_{k+1} it is also compatible with $\mathcal{C}_{k+1}$. □

Note that the proof of the previous theorem does not depend on which constructive ordering one chooses. This follows from the fact that condition (*i*) is equivalent to condition (*ii*), and condition (*i*) does not assume any ordering.

The reader conversant in Markov random fields will identify condition (ii) as the quantum analogue of the *Local Markov Property* of a Markov random field—any variable X_i is conditionally independent of the remaining nodes given its adjacent nodes:

$$X_i \perp\!\!\!\perp \overline{\{X_i\} \cup \mathrm{Ad}X_i} \,|\, \mathrm{Ad}X_i, \tag{13}$$

where $\mathrm{Ad}X_i$ is the set of adjacent nodes to X_i. The notion of conditional independence is equivalently replaced by the conditional mutual information being null, that is

$$I(X_i : \overline{\{X_i\} \cup \mathrm{Ad}X_i} \,|\, \mathrm{Ad}X_i) = 0, \tag{14}$$

which, for the case of the tree $\mathcal{G}_k$ and for the node X_k, we have

$$I(X_k : \overline{Y_k} \,|\, Y_k) = 0. \tag{15}$$

The following results state how to compute the solution Maximum Entropy Compatible Marginal Problem when $\mathcal{G}_\mathcal{C}$ is a tree and there exists $\rho \in \mathrm{Comp}(\mathcal{C})$ that factorizes via Petz according to $\mathcal{C}$.

Corollary 1. *Let $\rho \in \mathcal{B}(\mathcal{H})$ factorize via Petz according to $\mathcal{C}$ and $\mathcal{G}_\mathcal{C}$ a spanning tree. Then,*

$$\rho = \underset{\rho' \in Comp(\mathcal{C})}{arg\,max}\ S(\rho'). \tag{16}$$

Proof of Corollary 1. It follows that in the case $\rho \in \mathcal{B}(\mathcal{H})$ factorizes via Petz according to $\mathcal{C}$, we have $\log \rho = \sum_\mathcal{C} \log \rho_{X_i X_j} - \sum_{i=1}^n (\deg(X_i) - 1) \log \rho_{X_i}$, which saturates the subadditivity of the von Neumann entropy for every 3-chain $\overline{Y_k} - Y_k - X_k$, $k = 3, \ldots, n$ in the spanning tree. □

4.4. Compatibility with a QMT

We are now ready to state our main theorem, which gives a stronger characterization for the existence of a compatible density operator that is a QMT. Previously, we needed multivariate measurements to establish whether there exists a QMT in the given compatibility set. Herein, we show that it is enough to consider two-body measurements, which makes the procedure feasible in practice. The proof requires some technical lemmas that we placed in Appendix C.

Theorem 3. *Let $\mathcal{C} := \{\rho_{X_i X_j} \in \mathcal{B}\left(\mathcal{H}_{X_i X_j}\right), i \neq j \in \{1, \ldots, n\}\}$ be a set of admissible two-body marginals and such that the associate graph $\mathcal{G}_\mathcal{C} = (V, E)$ is a spanning tree. Then, there exists $\tilde{\rho} \in \mathcal{B}(\mathcal{H})$ such that $\tilde{\rho} \in Comp(\mathcal{C})$ factorizable via Petz according to $\mathcal{C}$ iff*

$$I_\rho(X_i : ad\, X_j |\, X_j) = 0,\ \forall \rho_{X_i X_j} \in \mathcal{C} \text{ and } \quad \forall ad\, X_j,\ ad\, X_j \neq X_i, \tag{17}$$

where ad X_j indicates an adjacent node of X_j in $\mathcal{G}_\mathcal{C}$, that is $adX_i \in AdX_i$. Moreover,

$$\tilde{\rho} := \underset{\rho' \in Comp(\mathcal{C})}{arg\,max}\ S(\rho). \tag{18}$$

Proof of Theorem 3. As in the previous theorem, we assume a constructive ordering $X_1 < \cdots < X_n$ for $\mathcal{C}$ which will be used in the induction proof. Moreover, we can rewrite $\mathcal{C}$ using such order as in Equation (8). Thus, the set of conditions in Equation (17) are:

$$I_\rho(X_k : \mathrm{ad}\, Y_k |\, Y_k) = 0,\ \forall\, \mathrm{ad}\, Y_k \in V_{k-1},\ k = 3, \ldots, n. \tag{19}$$

($\Rightarrow$) Using the previous theorem we have that

$$I_{\rho_k}(X_k : \overline{Y_k} | Y_k) = I_\rho(X_k : \overline{Y_k} | Y_k) = 0. \tag{20}$$

Moreover, by Proposition 1, X_k is leaf in $\mathcal{G}_k$ and it is only connected to Y_k. Finally, by applying the chain rule of the quantum conditional mutual information (c.f. in Appendix C Equation (A13)) and choosing the chain to start in a node adjacent to X_k, say $\mathrm{ad}X_k$, it follows that $I_\rho(X_k : \mathrm{ad}\, Y_k |\, Y_k) = 0$.

($\Leftarrow$) The proof follows again by complete induction in the number of nodes k, following the assumed constructive ordering of $\mathcal{C}$. Again, the simplest tree where the equation has any meaning requires three nodes.

(Basis $k = 3$): for this case the statement of this theorem coincides with (ii) of Theorem 2, since $\mathrm{ad}\, Y_3 = \overline{Y_3}$.

(Induction step $k \longrightarrow k+1$):

Induction hypothesis: We assume

$$I_\rho(X_k : \mathrm{ad}\, Y_k | Y_k) = 0,\ \forall\, \mathrm{ad}\, Y_k \in V_{k-1},\ k = 3, \ldots, n, \tag{21}$$

and so, by hypothesis, ρ_ℓ is factorizable via Petz according to $\mathcal{C}_\ell$, and so, by Theorem 2, we have

$$I_{\rho_\ell}\left(X_\ell : \overline{Y_\ell} | Y_\ell\right) = 0\ \forall \ell = 3, \ldots k. \tag{22}$$

Induction step: We assume $I_\rho(X_{k+1} : \mathrm{ad} Y_{k+1} | Y_{k+1}) = 0\ \forall\, \mathrm{ad}\, Y_{k+1} \in V_k$ and our goal is to show that there exists ρ_{k+1} factorizable via Petz according to $\mathcal{C}_{k+1}$ such that its partial traces hold

$$I_{\rho_{k+1}}\left(X_{k+1} : \overline{Y_{k+1}} | Y_{k+1}\right) = 0. \tag{23}$$

Observe that, by definition, $Y_{k+1} \in V_k$, let m_{k+1} be some step in which Y_{k+1} was connected to some node (note that it might connect to some node in many steps). Clearly, we have $3 \leq m_{k+1} \leq k$. We consider two cases, depending on the degree of Y_{k+1} in $\mathcal{G}_k$.

Case (1) $\deg Y_{k+1} = 1$, then by construction, it must be that $Y_{k+1} = X_{m_k}$ and by Equation (22) we have that for ρ_{m_k} its partial traces hold

$$I_{\rho_{m_k}}\left(X_{m_k} : \overline{Y_{m_k}} | Y_{m_k}\right) = 0. \tag{24}$$

By Lemma A7 (in Appendix C) since

$$V_k \setminus \{X_{m_k}, Y_{m_k}\} \supseteq \overline{Y_{m_k}} = V_{m_k} \setminus \{X_{m_k}, Y_{m_k}\}, \tag{25}$$

we also have for ρ_k that

$$I_\rho\left(X_{m_k} : V_k \setminus \{X_{m_k}, Y_{m_k}\} | Y_{m_k}\right) = I_\rho(Y_{k+1} : V_k \setminus \{Y_{k+1}, \mathrm{ad} Y_{k+1}\} | \mathrm{ad} Y_{k+1}) = 0, \tag{26}$$

where the last equality is obtained by noticing that $X_{m_k} = Y_{k+1}$ and $Y_{m_k} = \mathrm{ad} Y_{k+1}$. Recall that we have,

$$I_\rho(X_{k+1} : \mathrm{ad}\, Y_{k+1} | Y_{k+1}) = 0. \tag{27}$$

Moreover, the set $\{V_k \setminus \{Y_{k+1}, \mathrm{ad} Y_{k+1}\}, \mathrm{ad}\, Y_{k+1}, Y_{k+1}, X_{k+1}\}$, forms the chain

$$V_k \setminus \{Y_{k+1}, \mathrm{ad} Y_{k+1}\} - \mathrm{ad}\, Y_{k+1} - Y_{k+1} - X_{k+1}. \tag{28}$$

Then, by using Lemma A6 (a) (in Appendix C), there exists a density operator $\rho_{k+1} \in \mathcal{B}\left(\mathcal{H}_{V_{k+1}}\right)$ such that its partial traces fulfill

$$I_\rho(X_{k+1} : V_k \setminus \{Y_{k+1}\} | Y_{k+1}) = I_{\rho_{k+1}}(X_{k+1} : \overline{Y_{k+1}} | Y_{k+1}) = 0. \tag{29}$$

Furthermore, by construction of this ρ_{k+1} in Lemma A6 (a) (in Appendix C) we have $\mathrm{Tr}_{X_{k+1}}[\rho_{k+1}] = \rho_k$, and so ρ_{k+1} is s.t.:

$$I_\rho(X_i : V_i \setminus \{X_i, Y_i\} | Y_i) = 0 \quad \forall i :\ 2 \leq i \leq k+1. \tag{30}$$

(Case 2) $\deg Y_{k+1} > 1$, then $\mathcal{G}_{k+1}$ can be seen as a star centered in Y_{k+1}, with as many branches, as many as adjacent nodes $(\mathrm{ad} Y_{k+1})_i$ in $\mathcal{G}_{k+1}$, whose number is precisely the degree r_k of Y_{k+1} in $\mathcal{G}_k$, plus the new added node X_{k+1} (c.f. Figure 1).

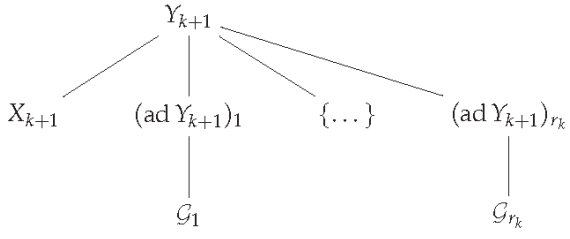


Figure 1. The associate graph $\mathcal{G}_{k+1}$: can be seen as a star centered in Y_{k+1}, where every branch is an adjacent of Y_{k+1} in V_k, plus the link to X_{k+1}. $\mathcal{G}_i$ indicates the rest of the graph (a tree) that is connected to the i-th adjacent $(\mathrm{ad}Y_{k+1})_i$. The number of adjacent nodes to Y_{k+1} in $\mathcal{G}_{k+1}$ is $r_k + 1$ by adding X_{k+1} to other r_k nodes in $\mathcal{G}_k$.

To prove the thesis we must find ρ_{k+1} such that, if

$$I_\rho(X_{k+1} : (\mathrm{ad}Y_{k+1})_i | Y_{k+1}) = 0 \; \forall i = 1 \dots r_k \tag{31}$$

then, accordingly to Theorem 2, it is enough to show:

$$I_{\rho_{k+1}}\left(X_{k+1} : \overline{Y_{k+1}} | Y_{k+1}\right) = 0. \tag{32}$$

Moreover, by induction hypothesis, we know that

$$I_\rho(X_\ell : \mathrm{ad}Y_\ell | Y_\ell) = 0 \; \forall \mathrm{ad}Y_\ell \in V_k, \; \ell = 3, \dots k. \tag{33}$$

and again, by Theorem 2, we must have:

$$I_{\rho_\ell}\left(X_\ell : \overline{Y_\ell} | Y_\ell\right) = 0 \; \forall \ell = 3, \dots k. \tag{34}$$

We proceed to show Equation (32) by using Corollary A2 (in Appendix C). Indeed, this results guarantees that the star

$$\{X_{k+1}, Y_{k+1}, (\mathrm{ad}Y_{k+1})_1 \cup \mathcal{G}_1, \dots, (\mathrm{ad}Y_{k+1})_{r_k} \cup \mathcal{G}_{r_k}\} \tag{35}$$

factorizes via Petz according to

$$\{X_{k+1}Y_{k+1}, Y_{k+1}(\mathrm{ad}Y_{k+1})_1 \cup \mathcal{G}_1, \dots, Y_{k+1}(\mathrm{ad}Y_{k+1})_{r_k} \cup \mathcal{G}_{r_k}\} \tag{36}$$

iff

$$I_\rho(X_{k+1} : (\mathrm{ad}Y_{k+1})_i \cup \mathcal{G}_i \mid Y_{k+1}) = 0, \quad \forall i \in 1, \dots, r_k; \tag{37}$$

$$I_\rho\Big((\mathrm{ad}Y_{k+1})_i \cup \mathcal{G}_i : (\mathrm{ad}Y_{k+1})_j \cup \mathcal{G}_j \mid Y_{k+1}\Big) = 0, \quad \forall i \neq j \in 1 \dots r_k. \tag{38}$$

Using Theorem 2 in Equation (37), we get the goal, stated in Equation (32). The conditions in Equation (38) come from the complete induction hypothesis Equation (34). On the other hand, the conditions stated in Equation (37), come from observing that, for every $(\mathrm{ad}Y_{k+1})_i$, there is a chain

$$X_{k+1} - Y_{k+1} - (\mathrm{ad}Y_{k+1})_i - \mathcal{G}_i, \tag{39}$$

for which we already have the conditions:

$$I_\rho(X_{k+1} : (\mathrm{ad}Y_{k+1})_i | Y_{k+1}) = 0, \tag{40}$$

$$I_\rho(Y_{k+1} : \mathcal{G}_i | (\mathrm{ad}Y_{k+1})_i) = 0. \tag{41}$$

Equation (40) follows from induction hypothesis Equation (33). Moreover, Equation (41) follows from the fact that, by hypothesis, ρ_k is a QMT, and so

$$Y_{k+1} - (\mathrm{ad}Y_{k+1})_i - \mathcal{G}_i \tag{42}$$

is a quantum Markov chain. Therefore, by using Lemma A6 (a) (in Appendix C), we get the desired condition

$$I_\rho(X_{k+1} : (\mathrm{ad}Y_{k+1})_i \cup \mathcal{G}_i \mid Y_{k+1}) = 0. \tag{43}$$

Since the argument holds for all the adjacent nodes $(\mathrm{ad}Y_{k+1})_i$, we derive the whole set of conditions (37), which ends the proof for case (2).

Finally, the fact that the obtained state maximizes the von Neumann entropy with the provided marginals comes for free from Corollary 1. □

4.5. QMT and the MECM Problem

We are now able to show that for QMTs, the MECM problem is in P and that there is a polynomial quantum circuit that constructs the Maximum entropy compatible density operator. Moreover, we also show that it is possible to extend the Chow–Liu algorithm efficiently for quantum Markov networks. To derive these results, we need first to compute the number of 3-chains in a graph with n nodes—proof in Appendix D.

Lemma 3. *The number of 3-chains #c in a tree with $n \geq 2$ vertices satisfies $n-2 \leq \#c \leq \frac{1}{2}(n-1)(n-2)$. Moreover, the number of 3-chains for any graph is upper-bounded by $\frac{1}{2}n(n-1)(n-2)$, and it reaches the bound for a complete graph of n nodes.*

We are now able to establish a sufficient condition for the MECMP problem to be in P.

Theorem 4. *The Maximum Entropy Compatible Marginal Problem for $\mathcal{C}$ is in P when*

1. *$\mathcal{G}_C$ is a spanning tree*
2. *ρ_{ijk} is a QMC constructed in polynomial-time (with respect with the number of nodes n) where $\rho_{i,j}, \rho_{j,k} \in \mathcal{C}$ and $i < j < k$ for some given constructive order of $\mathcal{G}_C$.*

Moreover, there exists a quantum polynomial circuit that constructs the maximum entropy compatible tree.

Proof of Theorem 4. From Theorem 3, the density operator that maximizes the Entropy is a QMT. Moreover, we can compute its entropy in polynomial time, by considering the constructive ordering of point 2. Indeed, from Theorem 2 (i), when ρ is a QMT we have that

$$S(\rho) = \sum_{\mathcal{C}} S(\rho_{X_i X_j}) - \sum_{i=1}^{n} (\deg(X_i) - 1) S(\rho_{X_i}). \tag{44}$$

Moreover, since each $\rho_{X_i X_j}$ belongs to a QMC constructed in polynomial time, we can compute a matrix representation of the density operator of the QMC in polynomial-time as well. Recall in Definition 3, that the Hilbert space of a polynomial-time QMC is fixed, and does not depend on the complexity parameter, that is, as usual, the dimension of the Hilbert space associated with each node is fixed (regarding) the complexity parameter n (the number of nodes).

Moreover, given the constructive order, we are also able to make a quantum circuit (c.f. Figure 2) to construct the maximum entropy compatible tree by constructing the first Markov chain ρ_{X_1,X_2,X_3} and then applying the circuits for the recovery maps $\mathcal{R}$ of the remaining nodes. □

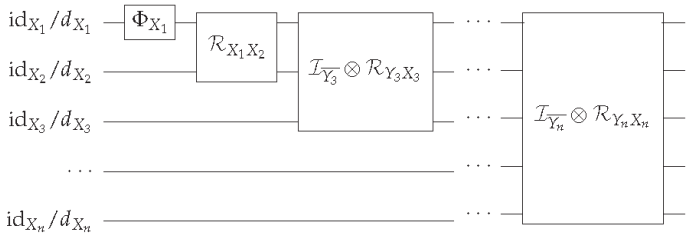


Figure 2. Quantum circuit that outputs the optimal quantum Markov trees (QMT). Note that the k-th block $\mathcal{I}_{\overline{Y_k}} \otimes \mathcal{R}_{Y_kX_k}$ operates only over two components X_k and Y_k, for all $k = 3 \dots n$.

4.6. QMT and Chow–Liu Algorithm

Two-body marginals for which all 3-chains form a QMC have another interesting property. It is possible to find the QMT closest, with regards to the quantum relative entropy (the generalization of the Kullback–Leibler divergence [42]), to the unknown density operator. Note that the number of spanning trees over a complete graph is given by Cayley's formula [43], n^{n-2} which is exponential on n. To extract the closest QMT, we need to construct a weighted graph (where the nodes are each component of the density operator), and the edges are weighted with the von Neumann mutual information between every two components. The optimal spanning tree, which can be found using the polynomial-time algorithm by Chow–Liu Algorithm 1, gives the support to a QMT. Moreover, this QMT will be the one closest to the unknown state. When the density operators are diagonal, that is, describe a probability distribution, this algorithm coincides with the well-established Chow–Liu algorithm.

Algorithm 1 Chow–Liu Algorithm

`Input`: $\{ \mathcal{C}', I_{\mathcal{C}'} \}$ from a set of RVs $X = \{X_1, \dots X_n\}$, where

- $\mathcal{C}' = \{ p(X_i, X_j); X_i \not\equiv X_j \in X \}$ is a set of two-body marginal probability distributions;
- $I_{\mathcal{C}'} = \{ I(X_i : X_j) : p(X_i, X_j) \in \mathcal{C}' \}$ is the associated set of conditional mutual information.

`Output`: $\{ \mathcal{C}'_T \subseteq \mathcal{C}'$ s.t. $H(p(X) \mid\mid p_T(X))$ is minimal $\}$, where

- $\mathcal{C}'_T$ subset of 2-body marginals which associate graph is a tree;
- $p_T(X) \in \mathrm{Comp}\, \mathcal{C}'_T$.

1. Sort $\mathcal{C}' = \{p(X_i, X_j) = p_\alpha\}_{\alpha=1}^{M \leq \frac{1}{2}n(n-1)}$ s.t. $I_1 \geq I_2 \geq .... \geq I_N$;
2. Initialize: $\mathcal{C}'_T = \emptyset$ $\alpha = 0$.
3. Iterate: `while` $a \leq M$ `do`

```
if C'_T ∪ p_α s.t. G_T is a tree
    then C'_T = C'_T ∪ { p_α };
α = α + 1 ;
return C'_T
```

Theorem 5. *If the set of two body marginals $\mathcal{C}$ is s.t. every 3-chain is compatible with a QMC then every subtree is a QMT. A QMT that minimizes the quantum relative entropy with respect to the (unknown) given quantum state, is the maximum weighted tree $\mathcal{G}_{T_C}$ where the weight of each edge is given by the quantum mutual information. Such tree can be obtained efficiently using the (generalized) Chow–Liu learning algorithm [19].*

Proof of Theorem 5. The proof consists in applying Theorem 3 to the main result of Section 6 in the paper [29], that we are going to briefly recall.

Let $\rho_{\mathbb{X}}$ be the unknown quantum state that describes best the quantum system for which the bipartite marginals are known (for instance, they have been measured and collected in $\mathcal{C}$). Moreover, let $\widetilde{\rho}_{\mathcal{C}_T}$ be the maximum von Neumann entropy d.o. compatible with a subset $\mathcal{C}_T \subseteq \mathcal{C}$ s.t. $\mathcal{G}_{\mathcal{C}_T}$ is a tree. We refer to $\widetilde{\rho}_{\mathcal{C}_T}$ as quantum tree. Their relative entropy can be written as

$$S(\rho_{\mathbb{X}}||\widetilde{\rho}_{\mathcal{C}_T}) = -S(\rho_{\mathbb{X}}) - \mathrm{Tr}(\rho_{\mathbb{X}} \log \widetilde{\rho}_{\mathcal{C}_T}) = S(\widetilde{\rho}_{\mathcal{C}_T}) - S(\rho_{\mathbb{X}}), \tag{45}$$

where we have used condition (i) in Theorem 2 on $\log \widetilde{\rho}_{\mathcal{C}_T}$.

Therefore, the optimal maximum entropy estimator $\widetilde{\rho}$ is computed over the subtree with minimal von Neumann entropy:

$$\widetilde{\rho} = \underset{\mathcal{C}_T \subseteq \mathcal{C}}{\operatorname{argmin}} \max_{\rho \in \mathrm{Comp}(\mathcal{C}_T)} S(\rho). \tag{46}$$

Since the number of possible spanning trees is n^{n-2} [43], we can not choose the best fitting tree efficiently, in general. However, in the case at hand, we can manipulate Equation (45) and derive subcases for which the computation can be performed efficiently.

Observe that

$$\sum_{\mathcal{C}_T} S\left(\rho_{X_i X_j}\right) - \sum_{i=1}^{n} (\deg X_i - 1) S(\rho_{X_i}) = -\sum_{\mathcal{C}_T} I_\rho(X_i, X_j) + \sum_{i=1}^{n} S(\rho_{X_i}), \tag{47}$$

and set

$$\Delta S(\widetilde{\rho}_{\mathcal{C}_T}) := \sum_{\mathcal{C}_T} S\left(\rho_{X_i X_j}\right) - \sum_{i=1}^{n} (\deg X_i - 1) S(\rho_{X_i}) - S(\widetilde{\rho}_{\mathcal{C}_T}), \tag{48}$$

which is always non-negative. By adding and subtracting the term

$$\sum_{\mathcal{C}_T} S\left(\rho_{X_i X_j}\right) + \sum_{i=1}^{n} (\deg X_i - 1) S(\rho_{X_i}) \tag{49}$$

to Equation (45), it assumes the form

$$S(\rho_{\mathbb{X}}||\widetilde{\rho}_{\mathcal{C}_T}) = -\sum_{\mathcal{C}_T} I_\rho(X_i, X_j) - \Delta S(\widetilde{\rho}_{\mathcal{C}_T}) + \sum_{i=1}^{n} S(\rho_{X_i}) - S(\rho_{\mathbb{X}}). \tag{50}$$

By using condition (i) of Theorem 2, we can replace the log term of $S(\widetilde{\rho}_{\mathcal{C}_T})$ of Equation (48) and thus, for a QMT, $\Delta S(\widetilde{\rho}_{\mathcal{C}}) = 0$. Moreover, we also have the converse, that is, $\Delta S(\widetilde{\rho}_{\mathcal{C}}) = 0$ holds only for QMTs. The latter result can be derived by observing that

$$\Delta S(\widetilde{\rho}_{\mathcal{C}_T}) = \sum_{i=1}^{n-2} I_\rho(X_{l_i} : V_i \backslash \{X_{l_i}, \mathrm{ad}X_{l_i}\} | \mathrm{ad}X_{l_i}), \tag{51}$$

which, by positivity of quantum conditional mutual information, is 0 iff all the terms in the sum are 0. Then, by Theorem 3, we have $\Delta S(\widetilde{\rho}_{\mathcal{C}}) = 0$ iff all the 3-chains in $\mathcal{C}_T$ are QMC.

Therefore, when the provided set of marginals $\mathcal{C}$ is s.t. every 3-chain is compatible with a QMC, $\Delta S(\widetilde{\rho}_{\mathcal{C}}) = 0$ in Equation (50). Therefore, the best tree is the one that maximizes the term

$$\sum_{\mathcal{C}_T} I_\rho(X_i, X_j), \tag{52}$$

i.e., the maximum weighted spanning sub-tree, where the weights are given by the mutual information between every couple of linked nodes.

This problem is efficiently solved for classical graphs by the Chow–Liu algorithm, which we have here generalized to quantum states, be replacing the Shannon entropy with the von Neumann entropy. □

The general case of efficiently finding the optimal spanning tree which gives the support to a quantum tree remains open. Minimizing the general form of Equation (50) would require the maximization of the quantity $\sum_{\mathcal{C}_\mathcal{T}} I_\rho(X_i, X_j) + \Delta S(\widetilde{\rho}_{\mathcal{C}_\mathcal{T}})$ with $S(\widetilde{\rho}_{\mathcal{C}_\mathcal{T}}) > 0$. Already in the tripartite scenario, it is evident that the maximum weighted tree is not a necessarily solution to the problem.

For the sake of completeness, we present the Chow–Liu algorithm in pseudo-code. In its quantum version, the Shannon entropy is replaced by the von Neumann entropy, so as the relative entropy with the quantum relative entropy.

5. Conclusions

In this paper, we addressed the problem of learning the maximum entropy density operator, describing an unknown quantum system on a finite-dimensional Hilbert space, from a set of two-body marginals.

First, we have shown that comparing the entropies of 3-chains—the simplest non-trivial scenario, where two marginals are known in a tripartite quantum system—is QSZK-complete. The result hints that finding the maximum entropy compatible state should be in general not feasible, with a step by step entropy-monotonic procedure.

Then, we determined a subclass of density operators where the addressed problem is in P. Concretely, by observing that the problem at hand naturally abstracts the inference problem for classical probability distribution within graphical models, we ask whether an exact efficient max-entropy learning procedure is limited to classical Markovian systems—the set of constraints is a tree-structured set of mutually commuting density operators. We generalize and extend the classical procedure to a larger subset of density operators, namely two-body marginals compatible with a quantum Markov tree (QMT), whose 3-chains are polynomial-time quantum Markov chains. In addition, for a general set of quantum states whose 3-subchains are quantum Markov chains, we were able to generalize the Chow–Liu algorithm for extracting the optimal QMT. Moreover, we showed that, in the case at hand, the maximum entropy quantum state could be constructed by a polynomial-time quantum circuit.

We stress that the obtained procedures overcome the quantum marginal problem, for which a solution is known in the case of compatibility of the provided set of marginals with a QMT.

Understanding other classes of quantum states for which this problem is tractable (at least in quantum polynomial time) would be a relevant problem. In particular, a further study on the robustness of the procedure can shed some light on the power of quantum machine learning techniques on solving the same problem beyond the Markovian assumption. Indeed, differently from the classical scenario, quantum Markov chains have been proven to be in general distant in trace distance from approximately-Markovian chains—that is, tripartite density operators ρ_{ABC} s.t. $I_\rho(A:B|C) \ll \epsilon$, $\epsilon > 0$—and the result naturally extends to QMT and many body density operators.

Author Contributions: Conceptualization, S.D.G. and P.M.; Formal analysis, S.D.G.; Funding acquisition, P.M.; Investigation, S.D.G. and P.M.; Methodology, S.D.G. and P.M.; Supervision, P.M.; Writing—original draft, P.M.; Writing—review & editing, S.D.G. and P.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by Security and Quantum Information Group of Instituto de Telecomunicações, by Programme (COMPETE 2020) of the Portugal 2020 framework [Project Q.DOT with Nr. 039728 (POCI-01-0247-FEDER-039728)] and the Fundação para a Ciência e a Tecnologia (FCT) through national funds, by FEDER, COMPETE 2020, and by Regional Operational Program of Lisbon, under UIDB/50008/2020 (actions QuRUNNER, QUESTS), Project QuantumMining POCI-01-0145-FEDER-031826 and Project PREDICT PTDC/CCI-CIF/29877/2017.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MDPI	Multidisciplinary Digital Publishing Institute
DOAJ	Directory of open access journals
TLA	Three letter acronym
LD	linear dichroism
QSZK	Quantum Statistical Zero Knowledge
CPTP	Completely Positive Trace Preserving
QMA	Quantum Merlin Arthur
QCMP	Quantum Compatible Marginal Problem
MECMP	Maximum Entropy Compatible Marginal Problem
BQP	Bounded-Error Quantum Polynomial Time
(3c)QED	(3-chain) Quantum Entropy Difference
QSD	Quantum Statistical Difference
QMC	Quantum Markov Chain
QMT	Quantum Markov Tree

Appendix A. Lemmas for Theorem 1

To perform the proof of Theorem 1, we need the following lemmas.

Lemma A1 (Polarization lemma, Theorem 5 in [25]). *Let α and β satisfy $0 \leq \alpha < \beta^2 \leq 1$. Then, there is a deterministic polynomial-time procedure that, on input $(Q_0, Q_1, 1^n)$ where Q_0 and Q_1 are quantum circuits, outputs descriptions of quantum circuits (R_0, R_1) (each having size polynomial in n and in the size of Q_0 and Q_1) such that*

$$\begin{aligned} \| \rho_0 - \rho_1 \|_{\mathrm{tr}} \leq \alpha \quad &\Rightarrow \quad \| \mu_0 - \mu_1 \|_{\mathrm{tr}} \leq 2^{-n}, \\ \| \rho_0 - \rho_1 \|_{\mathrm{tr}} \geq \beta \quad &\Rightarrow \quad \| \mu_0 - \mu_1 \|_{\mathrm{tr}} \geq 1 - 2^{-n}. \end{aligned}$$

The proof of the following lemmas can be found in [38].

Lemma A2 (Joint entropy theorem [38]). *Suppose p_i are probabilities, $|i\rangle$ are orthogonal states for a system A and ρ_i is any set of density operators for another system B. Then,*

$$S\left(\sum_i p_i |i\rangle\langle i| \otimes \rho_i\right) = H(p_i) + \sum_i p_i S(\rho_i). \tag{A1}$$

Lemma A3 (Fannes' inequality [38]). *Suppose ρ and σ are density matrices over a Hilbert space of dimension d. Suppose further that the trace distance between them satisfies $t = \| \rho - \sigma \|_{\mathrm{tr}} \leq 1/e$. Then,*

$$|S(\rho) - S(\sigma)| \leq t(\ln d - \ln t). \tag{A2}$$

Lemma A4 (Lemma 3.2 in [44]). *Let ρ_0 and ρ_1 be two density matrices, and let $\rho = \frac{1}{2}(\rho_0 + \rho_1)$. If there exists a measurement with outcome 0 or 1 such that making the measurement on ρ_b yields the bit b with probability at least p, then*

$$S(\rho) \geq \frac{1}{2}[S(\rho_0) + S(\rho_1)] + (1 - H(p)). \tag{A3}$$

In particular, by choosing the right observable we have

Lemma A5 (Theorem 1 in [25]). *Let ρ_0 and ρ_1 be two density matrices, and let $\rho = \frac{1}{2}(\rho_0 + \rho_1)$. Then,*

$$S(\rho) \geq \frac{1}{2}[S(\rho_0) + S(\rho_1)] + (1 - H(\frac{1}{2} + \frac{\| \rho_0 - \rho_1 \|_{\mathrm{tr}}}{2})). \tag{A4}$$

Appendix B. Proof of the Central Lemma 1

The proof of the lemma comes straightforwardly from the following definitions and previously established theorems.

Definition A1. *Let $\mathcal{H}$ be a finite dimensional Hilbert space and $\rho_i \in \mathcal{B}(\mathcal{H})$, i = 1,2, density operators. Their relative entropy is defined as:*

$$S(\rho_1 \| \rho_2) := \begin{cases} Tr\rho_1(\log \rho_1 - \log \rho_2) & \text{if } supp(\rho_1) \subseteq supp(\rho_2) \\ +\infty & \text{otherwise.} \end{cases} \tag{A5}$$

Originally defined by Umegaki [45]. A relevant property of the quantum relative entropy is its monotonicity under CPTP maps, also known as Uhlmann's theorem [46].

Theorem A1. *Let $\mathcal{H}$ and $\mathcal{K}$ be finite dimensional Hilbert spaces, $\rho_i \in \mathcal{B}(\mathcal{H})$, i = 1,2, density operators with $supp(\rho_1) \subseteq supp(\rho_2)$. For a CPTP map $\Phi : \mathcal{B}(\mathcal{H}) \to \mathcal{B}(\mathcal{K})$ the following inequality holds:*

$$S(\rho_1 \| \rho_2) \geq S(\Phi(\rho_1) \| \Phi(\rho_2)). \tag{A6}$$

Corollary A1. *The von Neumann entropy is strong sub-additivite:*

$$S\left(\rho_{ABC} \| \rho_{AB} \otimes \frac{id_c}{d_C}\right) \geq S\left(\rho_{BC} \| \rho_B \otimes \frac{id_c}{d_C}\right). \tag{A7}$$

Proof of Corollary A1. Observe that setting in (A6) $\rho_1 \to \rho_{ABC}$, $\rho_2 \to \rho_{AB} \otimes \mathrm{id}_C / d_C$ and $\phi(\cdot) \to \mathrm{Tr}_A[\cdot]$, we obtain which is equivalent to the non-negativity of the quantum conditional mutual information $I_\rho(A : C|B) \geq 0$. □

The following two theorems characterize the case of the equality and they will be the core of the proof of Lemma 1.

Theorem A2 (Theorem 2 in [47]). *Let $\Phi : \mathcal{B}(\mathcal{H}) \to \mathcal{B}(\mathcal{K})$ be a CPTP map and let $\rho_i \in \mathcal{B}(\mathcal{H})$, i = 1,2, and $\phi(\rho_i) \in \mathcal{B}(\mathcal{K})$ be all invertible density operators. Then, the equality holds in the Uhlmann theorem iff the following equivalent conditions hold:*

(i) $\phi^\dagger\left(\phi(\rho_2)^{it}\phi(\rho_1)^{-it}\right) = \rho_2^{it}\rho_1^{-it}$ $t \in \mathbb{R}$;

(ii) $\phi^\dagger(\log \phi(\rho_1) - \log \phi(\rho_2)) = \log \rho_1 - \log \rho_2$;

where (ii) is obtained differentiating (i) in t = 0.

The adjoint map $\phi^\dagger(\cdot)$ is understood with respect to the *Hilbert–Schmidt inner product*.

Theorem A3 (Theorem 5.2 in [35]). *A tripartite state $\rho_{ABC} \in \mathcal{B}(\mathcal{H}_{ABC})$ is a QMC in the order A-B-C iff $I_\rho(A:C|B) = 0$. Furthermore, one can always choose as recovery map the rotated Petz map:*

$$\mathcal{P}^t_{B\to BC}(X) := \rho_{BC}^{\frac{1+it}{2}} \left(\rho_B^{-\frac{1+it}{2}} X \rho_B^{-\frac{1-it}{2}} \otimes id_C \right) \rho_{BC}^{\frac{1-it}{2}}, \text{ for any } X \in \mathcal{B}(\mathcal{H}_B),\ t \in \mathbb{R}. \tag{A8}$$

Proof of Theorem A3 of Lemma 1.

(3 ⇒ 1) This implication comes for free from the definition of QMC. Moreover, the map $\mathcal{P}_{B\to BC}(\cdot)$ is clearly CPTP. The complete positivity indeed comes for free from the Hermitianicity of $\rho_B \otimes \mathrm{id}_C/d_C$ and ρ_{BC}, then of their square-roots.
(1 ⇒ 3) Equation (A8) for t = 0 gives exactly the Petz map in (3), so the implication comes as corollary of Theorem A3.
(1 ⇔ 2) This follows from the statement of Theorem A3.
(2 ⇔ 4) It comes as corollary of Theorem A2, using the settings in Corollary A1.
□

Appendix C. Lemmas for Theorem 2 and 3

We need the following Lemmas to derive the proof.

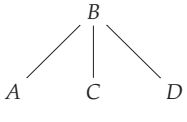
Lemma A6. *Let $\mathbb{X} = \{A, B, C, D\}$ be the labeling of parts of a finite dimensional Hilbert space $\mathcal{H}$ and $\mathcal{C} = \{\rho_{XY} \in \mathcal{B}(\mathcal{H}_{XY}),\ X, Y \in \mathbb{X}\}$ an admissible set of two-body marginals. Assume $\rho_{AB}, \rho_{BC} \in \mathcal{C}$ and $\exists\, \tilde{\rho}_{ABC} \in \mathcal{B}(\mathcal{H}_{ABC}) : \tilde{\rho}_{ABC} \in Comp(\rho_{AB}, \rho_{BC})$ such that*

$$I_\rho(A:C|B) = 0. \tag{A9}$$

(a) If the associate graph $\mathcal{G}_\mathcal{C}$ is a chain A-B-C-D (i.e., $\mathcal{C} = \{\rho_{AB}, \rho_{BC}, \rho_{CD}\}$) then

$$\begin{aligned} &\exists\, \tilde{\rho} \in \mathcal{B}(\mathcal{H}) :\ \tilde{\rho} \in Comp(\mathcal{C}) \text{ s.t } I_\rho(A:CD|B) = 0 \quad \text{iff} \\ &\exists\, \tilde{\rho}_{BCD} \in \mathcal{B}(\mathcal{H}_{BCD}) :\ \tilde{\rho}_{BCD} \in Comp(\rho_{BC}, \rho_{CD}) \text{ s.t } I_\rho(B:D|C) = 0. \end{aligned} \tag{A10}$$

(b) If the associate graph $\mathcal{G}_\mathcal{C}$ is a star centered in B (i.e., $\mathcal{C} = \{\rho_{AB}, \rho_{BC}, \rho_{BD}\}$)

$\mathcal{G}_\mathcal{C}$:  (A11)

then

$$\exists\, \tilde{\rho} \in \mathcal{B}(\mathcal{H}) :\ \tilde{\rho} \in Comp(\mathcal{C}) \text{ s.t } I_\rho(A:CD|B) = 0 \quad \text{iff} \tag{A12}$$

(i) $\exists\, \tilde{\rho}_{CBD} \in Comp(\rho_{BC}, \rho_{BD}),\ \tilde{\rho}_{BCD} \in \mathcal{B}(\mathcal{H}_{BCD})$ *s.t.* $I_\rho(C:D|B) = 0$ *and*
(ii) $\exists\, \tilde{\rho}_{ABD} \in Comp(\rho_{AB}, \rho_{BD}),\ \tilde{\rho}_{ABD} \in \mathcal{B}(\mathcal{H}_{ABD})$ *s.t.* $I_\rho(A:D|B) = 0$.

In both the cases, $\tilde{\rho} = \arg\max_{\rho \in Comp(\mathcal{C})} S(\rho)$ and factorizes over the elements of $\mathcal{C}$ via Petz following a constructive ordering for $\mathcal{C}$.

Proof of Lemma A6. We prove cases (a) and (b) together, but each direction of the equivalence at a time. We notice than one direction follows easily from the chain rule, we start with that direction (⇐) Recall the chain rule for quantum conditional mutual information:

$$\begin{aligned} I_\rho(A:X_1,\ldots,X_n|B) = I_\rho(A:X_1|B) + I_\rho(A:X_2|BX_1) + \\ + \cdots + I_\rho(A:X_n|BX_1,\ldots,X_{n-1}) \end{aligned} \tag{A13}$$

and recall that the conditional mutual information is non negative.

Case (a)

$$I_\rho(AB:D|C) = I_\rho(B:D|C) + I_\rho(B:D|AC) = 0 \quad \Rightarrow \quad I_\rho(B:D|C) = 0 \tag{A14}$$

The case (b) is analogous:

$$\begin{aligned} I_\rho(AC:D|B) = I_\rho(A:D|B) + I_\rho(A:D|BC) = 0 \quad &\Rightarrow \quad I_\rho(A:D|B) = 0 \\ I_\rho(AC:D|B) = I_\rho(C:D|B) + I_\rho(C:D|AB) = 0 \quad &\Rightarrow \quad I_\rho(C:D|B) = 0 \end{aligned} \tag{A15}$$

($\Rightarrow$) (a) To prove the other direction of the statement, we show that there exists a $\tilde{\rho} \in \mathcal{B}(\mathcal{H})$: $\tilde{\rho} \in \mathrm{Comp}(\tilde{\rho}_{ABC}, \tilde{\rho}_{BCD})$ and QMC on the order AB-C-D.

By hypothesis and using Lemma 1, the tripartite states can be recovered from two of its two-body marginals using the Petz recovery map:

$$I_\rho(A:C|B) = 0 \quad \text{iff} \quad \tilde{\rho}_{ABC} = \rho_{BC}^{\frac{1}{2}} \rho_B^{-\frac{1}{2}} \rho_{AB} \rho_B^{-\frac{1}{2}} \rho_{BC}^{\frac{1}{2}} = \rho_{AB}^{\frac{1}{2}} \rho_B^{-\frac{1}{2}} \rho_{BC} \rho_B^{-\frac{1}{2}} \rho_{AB}^{\frac{1}{2}}, \tag{A16}$$

$$I_\rho(B:D|C) = 0 \quad \text{iff} \quad \tilde{\rho}_{BCD} = \rho_{BC}^{\frac{1}{2}} \rho_C^{-\frac{1}{2}} \rho_{CD} \rho_C^{-\frac{1}{2}} \rho_{BC}^{\frac{1}{2}} = \rho_{CD}^{\frac{1}{2}} \rho_C^{-\frac{1}{2}} \rho_{BC} \rho_C^{-\frac{1}{2}} \rho_{CD}^{\frac{1}{2}}. \tag{A17}$$

Using Lemma 2, we check the compatibility of the two marginals with the desired QMC showing that the operator $\Theta_{ABCD} := \tilde{\rho}_{ABC}^{\frac{1}{2}} \rho_C^{-\frac{1}{2}} \rho_{CD}^{\frac{1}{2}}$ is normal:

$$\Theta_{ABCD}\Theta_{ABCD}^\dagger = \tilde{\rho}_{ABC}^{\frac{1}{2}} \rho_C^{-\frac{1}{2}} \rho_{CD} \rho_C^{-\frac{1}{2}} \tilde{\rho}_{ABC}^{\frac{1}{2}} \tag{A18}$$

$$= \rho_{AB}^{\frac{1}{2}} \rho_B^{-\frac{1}{2}} \underbrace{\rho_{BC}^{\frac{1}{2}} \rho_C^{-\frac{1}{2}} \rho_{CD} \rho_C^{-\frac{1}{2}} \rho_{BC}^{\frac{1}{2}}}_{\tilde{\rho}_{BCD}} \rho_B^{-\frac{1}{2}} \rho_{AB}^{\frac{1}{2}} \tag{A19}$$

$$= \underbrace{\rho_{AB}^{\frac{1}{2}} \rho_B^{-\frac{1}{2}} \rho_{CD}^{\frac{1}{2}} \rho_C^{-\frac{1}{2}}}\, \rho_{BC} \underbrace{\rho_C^{-\frac{1}{2}} \rho_{CD}^{\frac{1}{2}} \rho_B^{-\frac{1}{2}} \rho_{AB}^{\frac{1}{2}}} \tag{A20}$$

$$= \rho_{CD}^{\frac{1}{2}} \rho_C^{-\frac{1}{2}} \underbrace{\rho_{AB}^{\frac{1}{2}} \rho_B^{-\frac{1}{2}} \rho_{BC} \rho_B^{-\frac{1}{2}} \rho_{AB}^{\frac{1}{2}}}_{\tilde{\rho}_{ABC}} \rho_C^{-\frac{1}{2}} \rho_{CD}^{\frac{1}{2}} \tag{A21}$$

$$= \Theta_{ABCD}^\dagger \Theta_{ABCD}. \tag{A22}$$

Equality in Equation (A19) follows for Equation (A16) and Lemma 2. Equality in Equation (A20) follows from permuting density operators in different Hilbert spaces.

(b) Similarly to (a), we show that there exists a $\tilde{\rho} \in \mathcal{B}(\mathcal{H})$: $\tilde{\rho} \in \mathrm{Comp}(\tilde{\rho}_{ABC}, \tilde{\rho}_{CBD}, \tilde{\rho}_{ABD})$ and QMC on the order AC-B-D. Again, using Lemma 1, the tripartite states can be recovered from two of its two-body marginals using the Petz recovery map:

$$I_\rho(A:C|B) = 0 \quad \text{iff} \quad \tilde{\rho}_{ABC} = \rho_{BC}^{\frac{1}{2}} \rho_B^{-\frac{1}{2}} \rho_{AB} \rho_B^{-\frac{1}{2}} \rho_{BC}^{\frac{1}{2}} = \rho_{AB}^{\frac{1}{2}} \rho_B^{-\frac{1}{2}} \rho_{BC} \rho_B^{-\frac{1}{2}} \rho_{AB}^{\frac{1}{2}}, \tag{A23}$$

$$I_\rho(C:D|B) = 0 \quad \text{iff} \quad \tilde{\rho}_{CBD} = \rho_{BC}^{\frac{1}{2}} \rho_B^{-\frac{1}{2}} \rho_{BD} \rho_B^{-\frac{1}{2}} \rho_{BC}^{\frac{1}{2}} = \rho_{BC}^{\frac{1}{2}} \rho_B^{-\frac{1}{2}} \rho_{BD} \rho_B^{-\frac{1}{2}} \rho_{BC}^{\frac{1}{2}}, \tag{A24}$$

$$I_\rho(A:D|B) = 0 \quad \text{iff} \quad \tilde{\rho}_{ABD} = \rho_{AB}^{\frac{1}{2}} \rho_B^{-\frac{1}{2}} \rho_{BD} \rho_B^{-\frac{1}{2}} \rho_{AB}^{\frac{1}{2}} = \rho_{AB}^{\frac{1}{2}} \rho_B^{-\frac{1}{2}} \rho_{BD} \rho_B^{-\frac{1}{2}} \rho_{AB}^{\frac{1}{2}}. \tag{A25}$$

First, by using Lemma 2, we check the compatibility of the two marginals ρ_{BD} and ρ_{ABC} with the desired QMC showing that the operator $\Theta_{ABCD} := \tilde{\rho}_{ABC}^{\frac{1}{2}} \rho_B^{-\frac{1}{2}} \rho_{BD}^{\frac{1}{2}}$ is normal:

$$\Theta_{ABCD}\Theta_{ABCD}^{\dagger} = \tilde{\rho}_{ABC}^{\frac{1}{2}} \rho_B^{-\frac{1}{2}} \rho_{BD} \rho_B^{-\frac{1}{2}} \tilde{\rho}_{ABC}^{\frac{1}{2}} \tag{A26}$$

$$= \rho_{AB}^{\frac{1}{2}} \rho_B^{-\frac{1}{2}} \underbrace{\rho_{BC}^{\frac{1}{2}} \rho_B^{-\frac{1}{2}} \rho_{BD} \rho_B^{-\frac{1}{2}} \rho_{BC}^{\frac{1}{2}}}_{\tilde{\rho}_{BCD}} \rho_B^{-\frac{1}{2}} \rho_{AB}^{\frac{1}{2}} \tag{A27}$$

$$= \underbrace{\rho_{AB}^{\frac{1}{2}} \rho_B^{-\frac{1}{2}} \rho_{BD}^{\frac{1}{2}}}_{\tilde{\rho}_{ABD}} \rho_B^{-\frac{1}{2}} \rho_{BC} \rho_B^{-\frac{1}{2}} \underbrace{\rho_{BD}^{\frac{1}{2}} \rho_B^{-\frac{1}{2}} \rho_{AB}^{\frac{1}{2}}}_{\tilde{\rho}_{ABD}} \tag{A28}$$

$$= \rho_{BD}^{\frac{1}{2}} \rho_B^{-\frac{1}{2}} \underbrace{\rho_{AB}^{\frac{1}{2}} \rho_B^{-\frac{1}{2}} \rho_{BC} \rho_B^{-\frac{1}{2}} \rho_{AB}^{\frac{1}{2}}}_{\tilde{\rho}_{ABC}} \rho_B^{-\frac{1}{2}} \rho_{BD}^{\frac{1}{2}} \tag{A29}$$

$$= \Theta_{ABCD}^{\dagger}\Theta_{ABCD}. \tag{A30}$$

Moreover, the QMC $\tilde{\rho} = \Theta_{ABCD}\Theta_{ABCD}^{\dagger}$ is in $\mathrm{Comp}(\tilde{\rho}_{ABC}, \tilde{\rho}_{CBD}, \tilde{\rho}_{ABD})$ by using Equations (A23)–(A25). □

The previous lemma can be trivially extended to the n-partite scenario, i.e., to an arbitrary chain and a star:

Corollary A2. *Let $\mathbb{X} = \{X_1, \ldots X_n\}$ be the labeling set of the parts of a finite dimensional Hilbert space $\mathcal{H}$ and $\mathcal{C} = \{\rho_{XY} \in \mathcal{B}(\mathcal{H}_{XY}), X, Y \in \mathbb{X}\}$ a set of two-body marginals on it classically compatible. Assume $\mathcal{G}_{\mathcal{C}}$ is a star centered in some $Y \in \mathbb{X}$, i.e., $\mathcal{C} = \{\rho_{X_iY}, i = 1, \ldots, n-1\}$ then there exists $\tilde{\rho} \in \mathcal{B}(\mathcal{H}) : \tilde{\rho} \in Comp(\mathcal{C})$ such that $\tilde{\rho}$ is a quantum Markov network iff*

$$I_{\rho}(X_i : X_j|Y) = 0 \; \forall i \neq j \in 1, \ldots, n-1. \tag{A31}$$

Moreover,

$$\tilde{\rho} = arg \max_{\rho \in Comp(\mathcal{C})} S(\rho) \tag{A32}$$

and factorizes over the elements of $\mathcal{C}$ via Petz following a constructive ordering for $\mathcal{C}$.

Proof of Corollary A2. The proof follows by adding at each step a node to the setting of Lemma A6 (case b). Shortly, consider the constructive ordering for the graph $\mathbb{X} = \{Y, X_1, \ldots, X_n\}$. Start from graph $\mathcal{G}_3$, where $V_3 \equiv Y, X_1, X_2, X_3$, clearly in this case we are in the situation of Lemma A6 (b), then:

$$I_{\rho}(X_2 : X_1|Y) = 0 \; I_{\rho}(X_3 : X_1|Y) = 0 \; \Leftrightarrow \; I_{\rho}(X_3 : X_1X_2|Y) = 0. \tag{A33}$$

Observe that the two conditions $I_{\rho}(X_3 : X_1X_1|Y) = 0$ and $I_{\rho}(X_2 : X_1|Y) = 0$ are those required by Theorem 2 s.t. there exists a Petz-factorizable d.o. ρ_3 over $\mathcal{G}_3$. Next, we add the link $X_4 - Y$ to the graph and verify that $I_{\rho}(X_4 : X_1X_2X_3|Y) = 0$ also holds. We need to use again Theorem 2 to construct a Petz-factorizable ρ_4. This condition follows by applying Lemma A6 (b):

$$\begin{aligned} I_{\rho}(X_4 : X_1X_2X_3|Y) = 0 \text{ iff} \\ I_{\rho}(X_3 : X_1X_2|Y) = 0 \text{ and } I_{\rho}(X_4 : X_1X_2|Y) = 0. \end{aligned} \tag{A34}$$

where the first condition is the one we got in the previous step, the second comes from Lemma A6 (b):

$$\begin{aligned} I_{\rho}(X_4 : X_1X_2|Y) = 0 \text{ iff} \\ I_{\rho}(X_4 : X_1|Y) = 0 \text{ and } I_{\rho}(X_4 : X_2|Y) = 0. \end{aligned} \tag{A35}$$

Then, we keep adding nodes and decomposing the next required condition by Theorem 2. We notice that at each step, i.e., every time we add a node, in order to have a Petz decomposable d.o. on the new graph we have to add to the previous set of 3-chains, all the new 3-chains, i.e., the ones that involve the last added node. □

Lemma A7. *Let $\rho \in \mathcal{B}(\mathcal{H})$, where $\mathbb{X} = \{X_1, \ldots, X_n\}$ and $\mathcal{H} = \bigotimes_{i=1}^{n} \mathcal{H}_{X_i}$, such that $\rho \in Comp(\mathcal{C})$ with $\mathcal{G}_{\mathcal{C}}$ a tree (i.e., we work under Assumption 1, and take $X_1 < \cdots < X_n$ the constructive order). If for some $\ell \leq n$ the following conditions hold*

$$I_{\rho_j}(X_i : \overline{Y_j}|\, Y_j) = 0,\ \forall j = 3, \ldots, \ell, \tag{A36}$$

then, by taking any i and $m_i \geq i$ such that

$$deg X_i|_{\mathcal{G}_i} = deg X_i|_{\mathcal{G}_{m_i}} \quad \text{and} \quad deg Y_i|_{\mathcal{G}_i} = deg Y_i|_{\mathcal{G}_{m_i}}, \tag{A37}$$

the following conditions also hold

$$I_{\rho}(X_i : V_{r_i} \setminus \{X_i, Y_i\}|\, Y_i) = 0,\ \forall r_i :\ i \leq r_i \leq m_i \leq \ell, \tag{A38}$$

Proof of Lemma A7. Take that Equation (A36) with $j = \rho_{r_i}$. By Theorem 2, we know that ρ_{r_i} factorizes via Petz over its two-body marginals according to $\mathcal{C}_{r_i}$. Then, set

$$\Delta_k := \rho_{X_k Y_k}^{\frac{1}{2}} \rho_{Y_k}^{-\frac{1}{2}}, \tag{A39}$$

it follows that the factorization via Petz can be written as follows:

$$\rho_{r_i} = \Delta_{r_i} \Delta_{r_i - 1} \ldots \Delta_i \ldots\ldots \rho_{X_1 X_2} \ldots \Delta_i \ldots \Delta_{r_i - 1} \Delta_{r_i}, \tag{A40}$$

where, in general, $[\Delta_i, \Delta_j] \neq 0$. Note that, from the definition of m_i it must be the case that $[\Delta_{r_i}, \Delta_s] = 0\ \forall s : i \leq s \leq r_i$. This follows since Equation (A37) imposes that no more nodes are connected to X_i and Y_i when adding nodes from step i to m_i; and therefore, the additional Δ_k's operate on different Hilbert spaces. Then, Equation (A40) is to be written as:

$$\rho_k = \Delta_i \ldots \Delta_{r_i} \ldots \rho_{X_1 X_2} \ldots \Delta_{r_i} \ldots \Delta_i. \tag{A41}$$

Now consider a new, but equivalent, constructive ordering $<'$

$$X_1 <' \cdots <' X_{i-1} <' X_{r_i} <' X_{i+1} <' \cdots <' X_{r_i - 1} <' X_i, \tag{A42}$$

obtained from the order $<$ by exchanging r_i with i. By using Theorem 2 (ii) with the order $<'$, we get in $\mathcal{C}'_{r_i}$ the condition

$$I_{\rho_{r'_i}}\left(X_{r'_i} : \overline{Y_{r'_i}}|Y_{r'_i}\right) = 0. \tag{A43}$$

Which for the usual order $<$ can be stated as:

$$I_{\rho}(X_i : V_{r_i} \setminus \{X_i, Y_i\}|Y_i) = 0. \tag{A44}$$

The latter equality is valid for all $r_i :\ i \leq r_i \leq m_i \leq \ell$, since the only property used was the fact that $\deg X_i|_{\mathcal{G}_i} = \deg X_i|_{\mathcal{G}_{r_i}}$. □

Appendix D. Number of 3-Chains

Proof of Lemma 3. We make the proof by counting, for each node X_i, how many 3-chains $X_j - X_i - X_k$ can be formed, and summing all of them afterwards.

For a spanning tree, the lower bound is the number of 3-chains in a n-chain (all nodes have degree 2, with exception of the root and the leaf). In this case, every node is the

central node of only one 3-chain, aside for the root and the leaf; thus, $\#c = n-2$. The upper bound is derived by counting the number of 3-chains in a n-star (there is a root and all the remaining nodes are leaves). The root, say Y, has $\deg Y = n-1$, and the remaining nodes (enumerate them as $X_1, \dots X_{n-1}$), have degree one. In this case, consider the first edge X_1Y, it can be linked through Y to more n-2 nodes, which also gives the number of 3-chains it can be part of. The next edge X_2Y, it can be connected through Y to $n-3$ nodes to form $n-3$ different chains (the chain $X_2 - Y - X_1$ is the same as $X_1 - Y - X_2$, which has been already counted for). It is now clear that the number of 3-chains in an n-star is

$$\#c_i = \sum_{k=2}^{n}(n-k) = \sum_{k=1}^{n-2} k = \frac{1}{2}(n-1)(n-2). \tag{A45}$$

The number of chains in a n-star is also the number of 3-chains that a node contributes in a complete graph. Then, to obtain the number of 3-chains in a complete graph it is enough to multiply Equation (A45) by the number of nodes, and so $\#c = n\#c_i = \frac{1}{2}n(n-1)(n-2)$.

Another way of obtaining this value consists in using well-known formulas from combinatorial calculus, and observing that the number of 3-chains in a complete graph of n vertices is the number of *simple dispositions*, i.e., the number of ordered sequences of length 3 without repetitions in a set of n elements, divided by two. The factor 2 comes from the symmetry of the 3-chains; that is, $A - B - C$ is the same 3-chain as $C - B - A$. Then, once again,

$$\#c = \frac{1}{2}\frac{n!}{(n-3)!} = \frac{1}{2}n(n-1)(n-2) \tag{A46}$$

□

References

1. D'Ariano, M.; G Paris, M.G.; Sacchi, M.F. Quantum tomography. *Adv. Imaging Electron. Phys.* **2003**, *128*, 206–309.
2. Huber, F.M.; Gühne, O. Characterizing ground and thermal states of few-body Hamiltonians. *Phys. Rev. Lett.* **2016**, *117*, 010403. [CrossRef] [PubMed]
3. Cotler, J.; Wilczek, F. Quantum overlapping tomography. *Phys. Rev. Lett.* **2020**, *124*, 100401. [CrossRef] [PubMed]
4. Jaynes, E.T. Information theory and statistical mechanics. *Phys. Rev.* **1957**, *106*, 620. [CrossRef]
5. Bairey, E.; Arad, I.; Lindner, N.H. Learning a local Hamiltonian from local measurements. *Phys. Rev. Lett.* **2019**, *122*, 020504. [CrossRef] [PubMed]
6. Bairey, E.; Guo, C.; Poletti, D.; Lindner, N.H.; Arad, I. Learning the dynamics of open quantum systems from their steady states. *New J. Phys.* **2020**, *22*, 032001. [CrossRef]
7. Wiebe, N.; Granade, C.; Ferrie, C.; Cory, D.G. Hamiltonian Learning and Certification Using Quantum Resources. *Phys. Rev. Lett.* **2014**, *112*, 190501. [CrossRef]
8. Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N.; Lloyd, S. Quantum machine learning. *Nature* **2017**, *549*, 195. [CrossRef]
9. Anshu, A.; Arunachalam, S.; Kuwahara, T.; Soleimanifar, M. Sample-efficient learning of quantum many-body systems. *arXiv* **2020**, arXiv:2004.07266.
10. Cao, N.; Xie, J.; Zhang, A.; Hou, S.Y.; Zhang, L.; Zeng, B. Supervised learning for quantum maximum entropy estimation. *arXiv* **2020**, arXiv:2005.01540.
11. Klyachko, A.A. Quantum marginal problem and N-representability. In *Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, 2006; Volume 36, p. 72.
12. Liu, Y.K. Consistency of local density matrices is QMA-complete. In *Approximation, Randomization, and Combinatorial Optimization Algorithms and Techniques*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 438–449.
13. Higuchi, A.; Sudbery, A.; Szulc, J. One-qubit reduced states of a pure many-qubit state: Polygon inequalities. *Phys. Rev. Lett.* **2003**, *90*, 107902. [CrossRef] [PubMed]
14. Bravyi, S. Requirements for Compatibility between Local and Multipartite Quantum States. *Quantum Inf. Comput.* **2004**, *4*, 12–26.
15. Huber, F.M. Quantum States and Their Marginals: From Multipartite Entanglement to Quantum Error-Correcting Codes. Ph.D. Thesis, University of Siegen (DE), Siegen, Germany, 2018.
16. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2006.
17. Dagum, P.; Luby, M. Approximating probabilistic inference in Bayesian belief networks is NP-hard. *Artif. Intell.* **1993**, *60*, 141–153. [CrossRef]
18. Valiant, L.G. The complexity of computing the permanent. *Theor. Comput. Sci.* **1979**, *8*, 189–201. [CrossRef]

19. Chow, C.K.; Liu, C.N. Approximating discrete probability distributions with dependence trees. *IEEE Trans. Inf. Theory* **1968**, *14*, 462–467. [CrossRef]
20. Leifer, M.S.; Spekkens, R.W. Towards a formulation of quantum theory as a causally neutral theory of Bayesian inference. *Phys. Rev. A* **2013**, *88*, 052130. [CrossRef]
21. Fitzsimons, J.F.; Jones, J.A.; Vedral, V. Quantum correlations which imply causation. *Sci. Rep.* **2015**, *5*, 18281. [CrossRef]
22. Horsman, D.; Heunen, C.; Pusey, M.F.; Barrett, J.; Spekkens, R.W. Can a quantum state over time resemble a quantum state at a single time? *Proc. R. Soc. A Math. Phys. Eng. Sci.* **2017**, *473*, 20170395. [CrossRef]
23. Watrous, J. Quantum computational complexity. *arXiv* **2008**, arXiv:0804.3401.
24. Bernstein, E.; Vazirani, U. Quantum complexity theory. *Siam J. Comput.* **1997**, *26*, 1411–1473. [CrossRef]
25. Watrous, J. Limits on the power of quantum statistical zero-knowledge. In Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science, Vancouver, BC, Canada 16–19 November 2002; IEEE: New York, NY, USA, 2002; pp. 459–468.
26. Arora, S.; Barak, B. *Computational Complexity: A Modern Approach*; Cambridge University Press: Cambridge, UK, 2009.
27. Ben-Aroya, A.; Schwartz, O.; Ta-Shma, A. Quantum Expanders: Motivation and Construction. *Theory Comput.* **2020**, *6*, 47–79. [CrossRef]
28. Fawzi, O.; Renner, R. Quantum conditional mutual information and approximate Markov chains. *Commun. Math. Phys.* **2015**, *340*, 575–611. [CrossRef]
29. Giorgio, S.D.; Mateus, P.; Mera, B. Recoverability from direct quantum correlations. *J. Phys. A Math. Theor.* **2020**, *53*, 185301. [CrossRef]
30. Wang, Y.J. Compatibility among Marginal Densities. *Biometrika* **2004**, *91*, 234–239. [CrossRef]
31. De Loera, J.; Onn, S. The Complexity of Three-Way Statistical Tables. *Siam J. Comput.* **2004**, *33*, 819–836. [CrossRef]
32. Fritz, T.; Chaves, R. Entropic Inequalities and Marginal Problems. *IEEE Trans. Inf. Theory* **2013**, *59*, 803–817. [CrossRef]
33. Watrous, J. Succinct quantum proofs for properties of finite groups. In Proceedings of the 41st Annual Symposium on Foundations of Computer Science, Redondo Beach, CA, USA, 12–14 November 2000; IEEE: New York, NY, USA, 2000; pp. 537–546.
34. Aaronson, S. Quantum computing, postselection, and probabilistic polynomial-time. *Proc. R. Soc. A Math. Phys. Eng. Sci.* **2005**, *461*, 3473–3482. [CrossRef]
35. Sutter, D. Approximate Quantum Markov Chains. In *Approximate Quantum Markov Chains*; Springer International Publishing: Cham, Switzerland, 2018; pp. 75–100.
36. Sutter, D.; Fawzi, O.; Renner, R. Universal recovery map for approximate Markov chains. *Proc. R. Soc. A* **2016**, *472*, 20150623. [CrossRef]
37. Choi, M.D. Completely positive linear maps on complex matrices. *Linear Algebra Appl.* **1975**, *10*, 285–290. [CrossRef]
38. Nielsen, M.A.; Chuang, I. *Quantum Computation and Quantum Information*; Cambridge University Press: Cambridge, UK, 2010.
39. Aharonov, D.; Kitaev, A.; Nisan, N. Quantum circuits with mixed states. In Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, Berkeley, CA, USA, 28–30 May 1998; pp. 20–30.
40. Löwner, K. Über monotone matrixfunktionen. *Math. Z.* **1934**, *38*, 177–216. [CrossRef]
41. Bengtsson, I.; Życzkowski, K. *Geometry of Quantum States: An Introduction to Quantum Entanglement*; Cambridge University Press: Cambridge, UK, 2006.
42. Cover, T.M.; Thomas, J.A. *Elements of Information Theory*; John Wiley & Sons: New York, NY, USA, 2012.
43. Cayley, A. A theorem on trees. *Quart. J. Math.* **1889**, *23*, 376–378.
44. Ambainis, A.; Nayak, A.; Ta-Shma, A.; Vazirani, U. Dense quantum coding and quantum finite automata. *J. Assoc. Comput. Mach. (JACM)* **2002**, *49*, 496–511. [CrossRef]
45. Umegaki, H. Conditional expectation in an operator algebra, IV (entropy and information). In *Kodai Mathematical Seminar Reports*; Department of Mathematics, Tokyo Institute of Technology: Tokyo, Japan, 1962; Volume 14, pp. 59–85.
46. Uhlmann, A. Relative entropy and the Wigner-Yanase-Dyson-Lieb concavity in an interpolation theory. *Commun. Math. Phys.* **1977**, *54*, 21–32. [CrossRef]
47. Petz, D. Monotonicity of quantum relative entropy revisited. *Rev. Math. Phys.* **2003**, *15*, 79–91. [CrossRef]

mathematics

MDPI

Article

On the Amplitude Amplification of Quantum States Corresponding to the Solutions of the Partition Problem

Mauro Mezzini [1,*], Jose J. Paulet [2], Fernando Cuartero [3], Hernan I. Cruz [3] and Fernando L. Pelayo [3]

[1] Department of Education, Roma Tre University, 00154 Roma, Italy
[2] Instituto De Inv. EN Informática AB, University of Castilla-La Mancha, 13001 Ciudad Real, Spain; JoseJavier.Paulet@uclm.es
[3] Departamento de Sistemas Informáticos, University of Castilla-La Mancha, 13001 Ciudad Real, Spain; Fernando.Cuartero@uclm.es (F.C.); HernanIndibil.LaCruz@alu.uclm.es (H.I.C.); FernandoL.Pelayo@uclm.es (F.L.P.)
* Correspondence: mauro.mezzini@uniroma3.it

Abstract: In this paper we investigate the effects of a quantum algorithm which increases the amplitude of the states corresponding to the solutions of the partition problem by a factor of almost two. The study is limited to one iteration.

Keywords: amplitude amplification; subset sum problem; quantum algorithms; computational efficiency

Citation: Mezzini, M.; Paulet, J.J.; Cuartero, F.; Cruz, H.I.; Pelayo, F.L. On the Amplitude Amplification of Quantum States Corresponding to the Solutions of the Partition Problem. *Mathematics* **2021**, *9*, 2027. https://doi.org/10.3390/math9172027

Academic Editor: Jan Sładkowski

Received: 27 July 2021
Accepted: 19 August 2021
Published: 24 August 2021

1. Introduction and Preliminaries

Quantum computing promises to offer a different paradigm for solving long standing complex problems. Shor's algorithm [1] is a clear example of the success of the application of quantum computation to a difficult problem, the factoring of integers. Other algorithms like Deutsch-Josza's [2] or Grover's [3] give strong hints that the intrinsic parallelism of quantum computation could be used in order to efficiently solve computationally complex problems. In particular, Grover's algorithm uses the concept of an oracle $f(x)$ which, given a configuration x of the problem at hand, computes whether or not x is a solution of the problem. In other words, if x is a configuration that solves an instance of a combinatorial problem then the oracle returns $f(x) = 1$ otherwise the oracle outputs $f(x) = 0$. By using this oracle, Grover's algorithm is capable of speeding up the time to search for a solution of many, very complex, combinatorial problems. In Grover's algorithm the Grover iteration amplifies the amplitude of the configuration states corresponding to solutions of the problem (i.e., those states x for which $f(x) = 1$) by a factor less than $O(1/\sqrt{N})$ (in the worst case in which there is only one configuration which is the solution of the problem), where $N = 2^n$ is the number of possible configurations. In this way, after approximately $O(\sqrt{N})$ iterations, the amplitudes of the states corresponding to the solutions of the problem approach 1.

We exploit the parallelism of quantum computation, therefore devising a quantum algorithm that is capable of doubling the amplitude of the states corresponding to the solution of a problem. In this paper, we will focus on the subset sum problem, a well known NP-complete problem. The subset sum problem is very important both at a theoretical level in complexity theory, and at a practical level in applications such as cryptography. The problem can be stated in the following way in which we denote by $\mathbb{N}^+$, the set of non-zero natural numbers. Let E be a finite set of elements, $s : E \to \mathbb{N}^+$ a function and $W \in \mathbb{N}^+$ a target number. The subset sum problem wonders whether there exists a subset $E' \subset E$ such that $\sum_{e \in E'} s(e) = W$. Usually this problem can also be reformulated as follows. There exists a subset $E' \subset E$ such that $\sum_{e \in E'} s(e) = \sum_{e \in E \setminus E'} s(e)$? In this later formulation it is called the partition problem (PP) [4]. From now on we do not lose any generality in

considering the set E equals the set of the first $|E|$ natural numbers; that is, we always consider $E = \{0, 1, \ldots, n-1\}$. Furthermore, we note that if PP has a solution E' then $E - E'$ is also a solution of the PP.

The literature is abundant in the study of this problem, for which the dynamic programming solving strategy is the most commonly applied. More recently, it has also been approached from the point of view of quantum computing, among these references we can mention [5] where an instance of the subset sum problem can be implemented by a quantum algorithm using the nuclear magnetic resonance (NMR) technique. In that paper, even at a very early stage and with a low number of qubits, they limit themselves to showing that the problem can be approached from this new perspective.

In [6], the authors studied the problem through both an asymptotic heuristic and a new data structure for using quantum gates. There, the possibility of obtaining an improvement over classical algorithms is shown, specifically the Howgrave–Graham–Joux [7], which of course is fully coherent with the NP $\subseteq$ BQP conjecture, obtaining a bound time $\tilde{O}(2^{2/3n})$.

More recently, in 2018, Helm and May [8] proposed a quantum algorithm that reduces the heuristic time to $\tilde{O}(2^{0.226n})$. A couple of years later, Li and Li [9] reduced this time beyond that, i.e., to $\tilde{O}(2^{0.209n})$.

In this work we will get rid of the heuristics of these studies to better go for the modest approach of [5]. In this sense, we propose a piece of quantum code using a quantum circuit to model the problem, consequently we devise a transformation that will allow us to increase the amplitudes of those corresponding to the solution by 50%, leaving the line of how and how many times this process could be iterated, as the key issue to be dealt with. In fact, we think that we could find the same limitations as those from Grover's algorithm. Anyway, we believe that this line of research deserves to be addressed.

Let $x \in \mathbb{N}$, $0 \leq x < 2^n$ then we say that $|x\rangle$ is a *computational state*, the binary representation of which is $|x_{n-1}x_{n-2}\ldots x_0\rangle$ with x_{n-1} being the most significant bit of the binary representation of $|x\rangle$.

The S gate for a single qubit is represented by the following matrix:

$$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$$

Other usual gates are Pauli X gate, also known as NOT gates, and $\sqrt{X}$ gates.

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad \sqrt{X} = \frac{1}{2}\begin{bmatrix} 1+i & 1-i \\ 1-i & 1+i \end{bmatrix}$$

The Hadamard gate for a single qubit is

$$H = \frac{\sqrt{2}}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

While the same gate for n qubits is represented by the following matrix

$$H^{\otimes^n} = \bigotimes_{i=1}^{n} \frac{\sqrt{2}}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

As stated before, we are interested in amplifying amplitudes for the search problem. In order to do this we make use of sequences of quantum gates in the way of Hadamard-S-Hadamard. Let us start by describing graphically the effect of such quantum gates and so resorting to the positions of the state vector represented in the Bloch sphere.

The effect of applying an H q-gate over a qubit represented on the Bloch Sphere can be seen as "first rotating an angle of π radians around the Z axis and then rotating an angle of $\frac{\pi}{2}$ radians around the Y axis" (fully equivalent to "first rotating an angle of $\frac{\pi}{2}$ radians around the Y axis and then rotating an angle of π radians around the X axis", which are the movements made by H in Figure 1). Hadamard q-gate application changes a qubit from a computational basis to a Hadamard basis. To better understand the effects of the sequence

H-S-H we start, as usual, from a qubit initialized to $|0\rangle$, i.e., located at the north pole of the Bloch s. which, moved through a Hadamard gate, gets located at the equator of the Bloch s., specifically on the X axis, in the position usually known as a $|+\rangle$, which corresponds to the qubit

$$|+\rangle = \frac{\sqrt{2}}{2}|0\rangle + \frac{\sqrt{2}}{2}|1\rangle$$

After this gate, we apply an S gate, the effect of which is a rotation of the state vector through an angle of $\frac{\pi}{2}$ radians around the Z axis. This places the qubit at position $|i\rangle$.

$$|i\rangle = \frac{\sqrt{2}}{2}|0\rangle + i \cdot \frac{\sqrt{2}}{2}|1\rangle$$

Finally, the effect of the second application of gate H is to reset the computational base, and then the vector from position $|i\rangle$ moves to the opposite point on the Y axis, that is, to the position

$$|-i\rangle = \frac{\sqrt{2}}{2}|0\rangle - i \cdot \frac{\sqrt{2}}{2}|1\rangle$$

In general, we have the following result

$$HSH \quad = \quad \sqrt{X}$$

That is, an HSH gate is equivalent to a $\frac{\pi}{2}$ radians clockwise rotation around the X axis as Figure 1 shows over a single qubit initialized to $|0\rangle$.

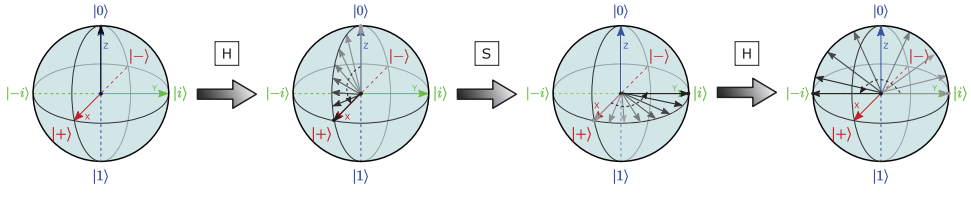


Figure 1. H-S-H over $|0\rangle$.

Let us go for the general case of a qubit register initialized, as usual, to $|0\rangle^{\otimes^n}$.

2. The Effect of Applying Hadamard-S-Hadamard Gates to $|0\rangle^{\otimes^n}$

First of all we will introduce a notation which will be heavily used in the rest of the paper. Let $x \in \mathbb{N}$, $0 \leq x < 2^n$ and $x_{n-1} \ldots, x_1 x_0$ be its binary representation. The Hamming weight of x will be denoted as $w(x) = \sum_{j=0}^{n-1} x_j$. Let $z \in \mathbb{N}$, $0 \leq z < 2^n$, $|z\rangle = |z_{n-1} z_{n-2} \ldots z_0\rangle$. The term $x \cdot z$ denotes $\sum_{j=0}^{n-1} x_j z_j$. Furthermore, $\forall k \in \mathbb{N}$, $0 < k < n$, we denote by $z_{\text{-k}}$ a natural number obtained from z by considering only the least $n-k$ significant bits, that is, the binary representation of $z_{\text{-k}}$ is $z_{n-k-1} z_{n-k-2} \ldots z_0$. In other words $z_{\text{-k}}$ is the bitwise AND between z and $2^{n-k} - 1$.

In this section we want to determine the effect of applying Hadamard first, then S and finally Hadamard gates on quantum state $|0\rangle^{\otimes^n}$, that is, we want to determine a formula for $|\alpha\rangle$ such that

$$|\alpha\rangle = H^{\otimes^n} S^{\otimes^n} H^{\otimes^n} |0\rangle^{\otimes^n} \tag{1}$$

and we show that in the out-coming computational state $|\alpha\rangle = \sum_{z=0}^{2^n-1} a_z |z\rangle$ the amplitude a_z of a single state $|z\rangle$ depends just on $w(z)$.

It is well known, see [3], that given any computational state $|x\rangle$ where $0 \leq x < 2^n$

$$|\psi\rangle = H^{\otimes^n} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{z=0}^{2^n-1} (-1)^{x \cdot z} |z\rangle \tag{2}$$

Now we start with the following Lemma

Lemma 1. *Let* $0 \leq x < 2^n$

$$S^{\otimes^n}|x\rangle = i^{w(x)}|x_{n-1}x_{n-2}\dots x_0\rangle$$

Proof. Induction on n is the base case with $n = 1$ straightforward. So suppose that the statement holds for $n-1$. Then

$$\begin{aligned} S^{\otimes^n}|x\rangle &= S|x_{n-1}\rangle \otimes S|x_{n-2}\rangle \cdots \otimes S|x_0\rangle = S^{\otimes^{n-1}}|x_{n-1}\dots x_1\rangle \otimes S|x_0\rangle = \\ &= i^{\sum_{j=1}^{n-1} x_j}|x_{n-1}\dots x_1\rangle \otimes S|x_0\rangle = \text{ (by induction hypothesis)} \\ &= i^{\sum_{j=1}^{n-1} x_j}|x_{n-1}\dots x_1\rangle \otimes i^{x_0}|x_0\rangle = \\ &= i^{w(x)}|x_{n-1}x_{n-2}\dots x_0\rangle \end{aligned}$$

□

Now by Lemma 1 and Equation (2), we have that

$$|\psi_1\rangle = S^{\otimes^n}H^{\otimes^n}|0\rangle^{\otimes^n} = S^{\otimes^n}\frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle = \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1} i^{w(x)}|x\rangle$$

and applying the Hadamard to $|\psi_1\rangle$, by (2), we have that

$$|\psi_2\rangle = H^{\otimes^n}|\psi_1\rangle = \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1} i^{w(x)}\left[\frac{1}{\sqrt{2^n}}\sum_{z=0}^{2^n-1}(-1)^{x\cdot z}|z\rangle\right]$$

and reordering the term of the sum we have that

$$|\psi_2\rangle = \frac{1}{2^n}\sum_{z=0}^{2^n-1}\sum_{x=0}^{2^n-1}(-1)^{x\cdot z}i^{w(x)}|z\rangle = \frac{1}{2^n}\sum_{z=0}^{2^n-1}\left(\sum_{x=0}^{2^n-1} i^{w(x)}(-1)^{x\cdot z}\right)|z\rangle$$

So in order to compute the amplitudes of $|\psi_2\rangle$ we need to compute the sum

$$\sum_{x=0}^{2^n-1} i^{w(x)}(-1)^{x\cdot z} \tag{3}$$

for every $0 \leq z < 2^n$. We will do this in the following two theorems. First of all we need the next Lemma which will be heavily used in the rest of the paper.

Lemma 2. *Let* $0 \leq z < 2^{2m+1}$ *and* $0 \leq x < 2^{2m+1}$ *and let* $z_{2m}z_{2m-1}\dots z_0$ *and* $x_{2m}x_{2m-1}\dots x_0$ *be the binary representation, respectively, of* z *and* x*. We have that*

$$\sum_{x=2^{2m}}^{2^{2m+1}-1} i^{w(x)}(-1)^{\sum_{j=0}^{2m} z_j\cdot x_j} = i(-1)^{z_{2m}}\sum_{x=0}^{2^{2m}-1} i^{w(x)}(-1)^{\sum_{j=0}^{2m-1} z_j\cdot x_j} \tag{4}$$

Proof. We note that, on the left hand of Equation (4), for every element of the sum, we have that $x_{2m} = 1$. Therefore $\sum_{j=0}^{2m} z_j \cdot x_j = \sum_{j=0}^{2m-1} z_j \cdot x_j + z_{2m}$. Based on this we have that

$$\sum_{x=2^{2m}}^{2^{2m+1}-1} i^{w(x)}(-1)^{\sum_{j=0}^{2m} z_j\cdot x_j} = (-1)^{z_{2m}}\sum_{x=2^{2m}}^{2^{2m+1}-1} i^{w(x)}(-1)^{\sum_{j=0}^{2m-1} z_j\cdot x_j}$$

Furthermore for the same reason above, if $2^{2m} \leq x < 2^{2m+1}$ and if $0 \leq \bar{x} < 2^{2m}$ then we have that $w(x) = w(\bar{x}) + 1$ and this proves Equation (4). □

Theorem 1. *Let $0 \leq z < 2^n$, $z \in \mathbb{N}$. If $n = 2m$ is even we have that*

$$\sum_{x=0}^{2^n-1} i^{w(x)}(-1)^{z\cdot x} = (-1)^{w(z)} i^{m+w(z)} 2^m \tag{5}$$

Proof. We prove Equation (5) on induction on m being the base case with $m = 1$ being easily verifiable for all $z \in \{0, 1, 2, 3\}$. So suppose the statement holds for all $h \leq m$ and for all $0 \leq z < 2^{2m}$. Then, for any $0 \leq z < 2^{2m+2}$ we have

$$\sum_{x=0}^{2^{2m+2}-1} i^{w(x)}(-1)^{z\cdot x} = \sum_{x=0}^{2^{2m}-1} i^{w(x)}(-1)^{\sum_{j=0}^{2m-1} z_j\cdot x_j} \tag{6}$$

$$+ \sum_{x=2^{2m}}^{2^{2m+1}-1} i^{w(x)}(-1)^{\sum_{j=0}^{2m} z_j\cdot x_j} + \tag{7}$$

$$+ \sum_{x=2^{2m+1}}^{2^{2m+2}-1} i^{w(x)}(-1)^{\sum_{j=0}^{2m+1} z_j\cdot x_j} \tag{8}$$

Now, by Equation (4) and by induction hypothesis, we have that (7) is equal to

$$\sum_{x=2^{2m}}^{2^{2m+1}-1} i^{w(x)}(-1)^{\sum_{j=0}^{2m} z_j\cdot x_j} = i(-1)^{z_{2m}} \sum_{x=0}^{2^{2m}-1} i^{w(x)}(-1)^{\sum_{j=0}^{2m-1} z_j\cdot x_j} =$$

$$= i(-1)^{z_{2m}}(-1)^{w(z_{\text{-}2})} i^{m+w(z_{\text{-}2})} 2^m \tag{9}$$

Likewise, in the term (8), for each, x is the sum, the bit x_{2m+1} is always set to 1, so we have that (8) is, by Equation (4), equal to

$$\sum_{x=2^{2m+1}}^{2^{2m+2}-1} i^{w(x)}(-1)^{\sum_{j=0}^{2m+1} z_j\cdot x_j} = i(-1)^{z_{2m+1}} \sum_{x=0}^{2^{2m+1}-1} i^{w(x)}(-1)^{\sum_{j=0}^{2m} z_j\cdot x_j} \tag{10}$$

Now by repeatedly applying Equation (4) and the induction hypothesis we have that the sum in the right hand of Equation (10) is

$$\sum_{x=0}^{2^{2m+1}-1} i^{w(x)}(-1)^{\sum_{j=0}^{2m} z_j\cdot x_j} = \tag{11}$$

$$= \sum_{x=0}^{2^{2m}-1} i^{w(x)}(-1)^{\sum_{j=0}^{2m-1} z_j\cdot x_j} + \sum_{x=2^{2m}}^{2^{2m+1}-1} i^{w(x)}(-1)^{\sum_{j=0}^{2m} z_j\cdot x_j} =$$

$$= (-1)^{w(z_{\text{-}2})} i^{m+w(z_{\text{-}2})} 2^m + i(-1)^{z_{2m}} \sum_{x=0}^{2^{2m}-1} i^{w(x)}(-1)^{\sum_{j=0}^{2m-1} z_j\cdot x_j} =$$

$$= (-1)^{w(z_{\text{-}2})} i^{m+w(z_{\text{-}2})} 2^m [1 + i(-1)^{z_{2m}}] \tag{12}$$

So if we replace (12) in (10) and if we sum together (6), (9) and (10) we obtain

$$\begin{aligned} a_z =& (-1)^{w(z_{\text{-}2})} i^{m+w(z_{\text{-}2})} 2^m \left[1 + i(-1)^{z_{2m}} + i(-1)^{z_{2m+1}} + i^2(-1)^{z_{2m}+z_{2m+1}}\right] = \\ =& (-1)^{w(z_{\text{-}2})} i^{m+1+w(z_{\text{-}2})} 2^m \left[-i + (-1)^{z_{2m}} + (-1)^{z_{2m+1}} + i(-1)^{z_{2m}+z_{2m+1}}\right] \end{aligned} \tag{13}$$

Now if we denote by $P = (-1)^{w(z_{\text{-}2})} i^{m+1+w(z_{\text{-}2})} 2^m$ we have that (13) is

$$a_z = \begin{cases} 2P & \text{if } z_{2m} = z_{2m+1} = 0 \\ -2iP & \text{if } z_{2m} \neq z_{2m+1} \\ -2P & \text{if } z_{2m} = z_{2m+1} = 1 \end{cases}$$

and it is now easy to verify that

$$a_z = (-1)^{w(z)} i^{m+1+w(z)} 2^{m+1}$$

for every $0 \leq z < 2^{2m+2}$, and this proves the induction step. □

Theorem 2. *Let $n = 2m + 1$ be an odd natural, $m \in \mathbb{N}$ and let $0 \leq z < 2^n$, $z \in \mathbb{N}$. Then*

$$\sum_{x=0}^{2^n-1} i^{w(x)} (-1)^{z \cdot x} = (-1)^{w(z)} i^{m+w(z)} 2^m (1+i) \tag{14}$$

Proof. First of all we note that Equation (14) holds if $m = 0$ and $z \in \{0, 1\}$. So in the following we suppose that $m > 1$. We have that

$$\begin{aligned} a_z &= \sum_{x=0}^{2^{2m+1}-1} i^{w(x)} (-1)^{z \cdot x} = \\ &= \sum_{x=0}^{2^{2m}-1} i^{w(x)} (-1)^{\sum_{j=0}^{2m-1} z_j \cdot x_j} + \sum_{x=2^{2m}}^{2^{2m+1}-1} i^{w(x)} (-1)^{\sum_{j=0}^{2m} z_j \cdot x_j} \end{aligned}$$

and, by Theorem 1, and by Equation (4), we have

$$\begin{aligned} a_z &= (-1)^{w(z_{-1})} i^{m+w(z_{-1})} 2^m + i(-1)^{z_{2m}} \sum_{x=0}^{2^{2m}-1} i^{w(x)} (-1)^{\sum_{j=0}^{2m-1} z_j \cdot x_j} = \\ &= (-1)^{w(z_{-1})} i^{m+w(z_{-1})} 2^m + i(-1)^{z_{2m}} (-1)^{w(z_{-1})} i^{m+w(z_{-1})} 2^m = \\ &= (-1)^{w(z_{-1})} i^{m+w(z_{-1})} 2^m [1 + i(-1)^{z_{2m}}] \end{aligned} \tag{15}$$

Let $z_{2m} z_{2m-1} \ldots z_0$ be the binary representation of z. Suppose first that $z_{2m} = 0$. Then Equation (15) become

$$(-1)^{w(z)} i^{m+w(z)} 2^m + (-1)^{w(z)} i^{m+w(z)+1} 2^m \tag{16}$$

and the Theorem is therefore proved. So suppose that $z_{2m} = 1$. Then Equation (15) become

$$(-1)^{w(z)-1} i^{m+w(z)-1} 2^m + (-1)^{w(z)} i^{m+w(z)} 2^m \tag{17}$$

but observing that

$$(-1)^{w(z)} i^{m+w(z)+1} = (-1)^{w(z)-1} i^{m+w(z)-1} \tag{18}$$

we have that also in this case the theorem is satisfied. □

As an example we have computed the amplitudes a_z (disregarding the normalization factor) for $n \in \{3, 4\}$ and we report them on Figure 2.

$\|\mathbf{z}\rangle$	$\mathbf{a_z}$
$\|000\rangle$	$-2+2i$
$\|001\rangle$	$2+2i$
$\|010\rangle$	$2+2i$
$\|011\rangle$	$2-2i$
$\|100\rangle$	$2+2i$
$\|101\rangle$	$2-2i$
$\|110\rangle$	$2-2i$
$\|111\rangle$	$-2-2i$

$\|\mathbf{z}\rangle$	$\mathbf{a_z}$
$\|0000\rangle$	-4
$\|0001\rangle$	$4i$
$\|0010\rangle$	$4i$
$\|0011\rangle$	4
$\|0100\rangle$	$4i$
$\|0101\rangle$	4
$\|0110\rangle$	4
$\|0111\rangle$	$-4i$
$\|1000\rangle$	$4i$
$\|1001\rangle$	4
$\|1010\rangle$	4
$\|1011\rangle$	$-4i$
$\|1100\rangle$	4
$\|1101\rangle$	$-4i$
$\|1110\rangle$	$-4i$
$\|1111\rangle$	-4

Figure 2. (**Left**) The amplitudes of a_z for $n = 3$. (**Right**) The amplitudes of a_z for $n = 4$. In order to get the final amplitudes one should multiply them by a suitable normalization factor.

3. Doubling the Amplitudes of the Solution States of the PP

In this section we consider a quantum circuit for doubling the amplitude of the solution's states of the partition problem.

We describe an application of the gates described in the previous section in a quantum circuit to deal with PP (see Figure 3). While the following results apply specifically to the PP, they can be applied to any other search problem.

Let us denote $\sum_{e\in E} s(e)/2$ by $\mathcal{S}$. Notice that S belonging to $\mathbb{Z}$ is a requirement for PP to have a solution. We use the two's complement representation of $-\mathcal{S}$, so requiring for it $m = \lceil \log_2 \mathcal{S} \rceil + 1$ qubits. Then for each $e \in E$, we use $k_e = \lfloor \log_2 s(e) \rfloor + 1$ qubits to encode $s(e)$. These qubits will remain constant in every phase of the circuit and therefore we will not consider them in the rest of the discussion. As usual, we use n qubits to encode a subset E' of E, i.e., if $|x_{n-1}x_{n-2}\dots x_0\rangle$ is the state of those n qubits, then an element e, $0 \le e < n$, is included in the set E' if and only if $x_e = 1$. We will use m qubits, denoted in the following by $|\sigma\rangle$, to store the sum $\sigma = -\mathcal{S} + \sum_{e\in E'} s(e)$ for the elements selected in $|x\rangle$. In this way $|\sigma\rangle = |0\rangle^{\otimes m}$ for a solution $|x\rangle$ of the PP. Qubit $|c\rangle$ is used for control purposes.

For a top-down description we can distinguish four groups of bits: $|x\rangle$, $|\sigma\rangle$, $|c\rangle$ and the sets of qubits used to represent the constants $s(e)$ for each element of E. Note that the number of qubits of the circuit, $n + m + 1 + \sum_{e\in E} k_e$, is polynomial in the size of a concise specification of the PP.

At the beginning of the circuit we have the superposition:

$$|\varphi_0\rangle = |0\rangle^{\otimes n}|\sigma\rangle|c\rangle$$

where σ is set to the two's complement of $-\mathcal{S}$ and $|c\rangle$ is set to $|1\rangle$. Then, we apply the Hadamard q-gate to the first n qubits, so obtaining

$$|\varphi_1\rangle = \left(H^{\otimes^n} \otimes I^{m+1}\right)|\varphi_0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle|\sigma\rangle|c\rangle$$

Next, we use each qubit x_e to conditionally sum the element $s(e)$ to $|\sigma\rangle$. If there exists a solution to the PP then, in the final superposition of $|\sigma\rangle$, the amplitude of the state $|x\rangle|0\rangle^{\otimes^m}|c\rangle$ will not be 0. The states $|x\rangle$ for which $|\sigma\rangle$ is zero will be referred to as the solutions states of the PP. The control qubit $|c\rangle$ will be set to zero exactly for those states for which $|\sigma\rangle = |0\rangle^{\otimes^m}$. At this point we apply an uncomputational step in order to set $|\sigma\rangle = |-\mathcal{S}\rangle$. Now if we apply the S gate to the first n qubits we obtain, by Lemma 1

$$|\varphi_2\rangle = \left(S^{\otimes^n} \otimes I^{m+1}\right)|\varphi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} i^{w(x)}|x\rangle|\sigma\rangle|c\rangle$$

Afterwards Hadamard gate is applied to the first n qubits controlled by the control qubit $|c\rangle$ such that it only affects non-solution states (see Figure 3). For the sake of simplicity we suppose that PP has only two solutions whose numeric representation is y and its bitwise complement $\overline{y}$. By Equation (3), we obtain

$$\begin{aligned}
|\varphi_2\rangle \xrightarrow{\text{contr } H^{\otimes^n}} |\varphi_3\rangle &= \frac{1}{\sqrt{2^n}} \sum_{z\in\{y,\overline{y}\}} i^{w(z)}|z\rangle|\sigma\rangle|0\rangle + \\
&+ \frac{1}{2^n} \sum_{z=0}^{2^n-1} \sum_{x\notin\{y,\overline{y}\}} i^{w(x)}(-1)^{x\cdot z}|z\rangle|\sigma\rangle|1\rangle = \\
&= \frac{1}{2^n}\left[\sqrt{2^n} \sum_{z\in\{y,\overline{y}\}} i^{w(z)}|z\rangle|\sigma\rangle|0\rangle + \sum_{z=0}^{2^n-1} \sum_{x\notin\{y,\overline{y}\}} i^{w(x)}(-1)^{x\cdot z}|z\rangle|\sigma\rangle|1\rangle\right] \qquad (19)
\end{aligned}$$

Now we want to quantify the amplitude of the state $|y\rangle|\sigma\rangle|1\rangle$ and $|\overline{y}\rangle|\sigma\rangle|1\rangle$ of Equation (19). We consider only the state $|y\rangle|\sigma\rangle|1\rangle$ since the same arguments can be applied to state $|\overline{y}\rangle|\sigma\rangle|1\rangle$. The amplitude b_y of the state $|y\rangle|\sigma\rangle|1\rangle$ (in the following we disregard the normalization factor $1/2^n$) is given by the following formula

$$b_y = \sum_{x\notin\{y,\overline{y}\}} i^{w(x)}(-1)^{x\cdot y} \qquad (20)$$

We may write the above sum as

$$b_y = \sum_{x\notin\{y,\overline{y}\}} i^{w(x)}(-1)^{x\cdot y} = \sum_{x=0}^{2^n-1} i^{w(x)}(-1)^{x\cdot y} - \sum_{x\in\{y,\overline{y}\}} i^{w(x)}(-1)^{x\cdot y} \qquad (21)$$

We have that

$$\begin{aligned}
\sum_{x\in\{y,\overline{y}\}} i^{w(x)}(-1)^{x\cdot y} &= i^{w(y)}(-1)^{y\cdot y} + i^{n-w(y)}(-1)^{\overline{y}\cdot y} = \\
&= i^{w(y)}(-1)^{w(y)} + i^{n-w(y)} = \qquad (22)
\end{aligned}$$

Then, recalling that $i^x = i^{-x}$ when x is even and $i^{-x} = -i^x$ when x is odd, we have two cases:

- $w(y)$ is even

$$\begin{aligned}
i^{w(y)}(-1)^{w(y)} + i^{n-w(y)} &= i^{w(y)}(-1)^{w(y)} + i^{n+w(y)} = \\
&= i^{w(y)}(1+i^n) \qquad (23)
\end{aligned}$$

- $w(y)$ is odd

$$i^{w(y)}(-1)^{w(y)} + i^{n-w(y)} = i^{w(y)}(-1)^{w(y)} - i^{n+w(y)} = \\ = -i^{w(y)}(1+i^n) \tag{24}$$

For simplicity of notation in the following we denote $w(y)$ as simply $\overline{w}$. We have that if $n = 2m$ is even then, by Theorem 1, b_y is

$$b_y = \begin{cases} (-1)^{\overline{w}} i^{m+\overline{w}} 2^m - i^{\overline{w}}(1+i^{2m}) & \text{if } \overline{w} \text{ is even} \\ (-1)^{\overline{w}} i^{m+\overline{w}} 2^m + i^{\overline{w}}(1+i^{2m}) & \text{if } \overline{w} \text{ is odd} \end{cases} \tag{25}$$

while if $n = 2m+1$ is odd, by Theorem 2, b_y is

$$b_y = \begin{cases} (-1)^{\overline{w}} i^{m+\overline{w}} 2^m (1+i) - i^{\overline{w}}(1+i^{2m+1}) & \text{if } \overline{w} \text{ is even} \\ (-1)^{\overline{w}} i^{m+\overline{w}} 2^m (1+i) + i^{\overline{w}}(1+i^{2m+1}) & \text{if } \overline{w} \text{ is odd} \end{cases} \tag{26}$$

It is immediate to check that in Equations (25) and (26) the term $i^{\overline{w}}(1+i^n)$ becomes negligible, with respect to the other term in the equation, as m becomes bigger. We conclude that the amplitude of the state $|y\rangle|\sigma\rangle|1\rangle$ is almost the same as the amplitude of state $|y\rangle|\sigma\rangle|0\rangle$, thus effectively duplicating the chances of state $|y\rangle$ at the end of the circuit.

Figure 3 captures on the Quirk simulator the instance of the PP where elements in $\mathcal{E}$ are $s(0) = 2, s(1) = 1$ and $s(2) = 3$ depicted in rows 1–2, 3–4 and 5–6, bottom-up referring to the number of the rows as well as the significance of the qubits. The $7th$ row is used for the control qubit. Rows 8–11 encode $-\mathcal{S} = -3$ which in two's complement is represented with four bits $\sigma = 1101$ (bootom-up). The last three qubits are used to be set on the superposition as usual.

Since $n = 2m+1 = 3$ and $|y\rangle = |011\rangle$, we have that $b_y = 3 - 3i$, thus the probability of getting $|y\rangle$ is, by (19), $\frac{1}{64}\left[|2\sqrt{2}|^2 + |3-3i|^2\right] = \frac{26}{64} = 0.40625$ which is exactly the output of Quirk simulator as it can be checked in [10].

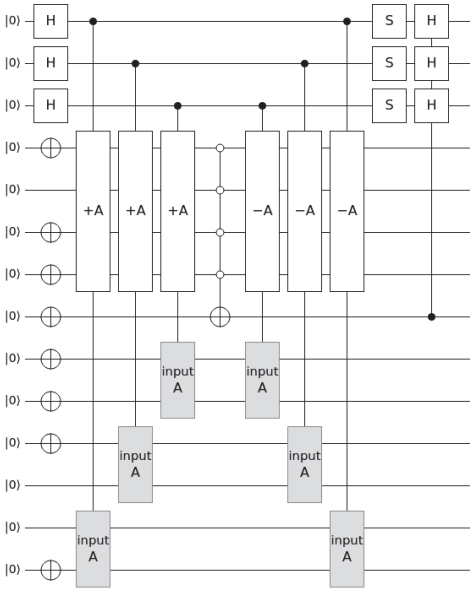


Figure 3. An instance of PP in Quirk.

As can be easily seen, this circuit resembles Grover's algorithm. Both have an initial superposition in the states where the solution will be encoded, afterwards the function to be solved (also called oracle) is computed in order to identify the solution state that should be marked in some way. Finally, a sort of transformation (Z axis turns, Hadamards) before measuring is required. However, they have important differences in some of these generic steps that can generate on the one hand why our proposal amplifies the probability of the solution state more than Grover's but on the other hand Grover's one can be iterated; meanwhile, ours cannot.

The main difference relies on how the algorithms *mark* the solution state. In our proposal the X axis is used for that; i.e., formerly the control qubit is set to $|1\rangle$ which means "no solution"; meanwhile, as soon as $\sigma = -\mathcal{S} + \sum_{e \in E'} s(e)$ equals 0 (solution condition) it is set to $|0\rangle$. In Grover's algorithm, the solution state is *marked* on the Z axis; that is, by means of a negative sign in the solution state. This last fact allows the calculations to be carried without any entanglement with the solution state.

Finally, the amplitude amplification part is also different but follows somehow the same fashion. In our proposal, we perform the amplification of the probability amplitude by means of turns in the Z axis and Hadamard transformations, which is very similar to the inversion on the mean of Grover's algorithm nevertheless, we apply Hadamard gates controlled by a qubit in which the solution has been marked. This fact generates an entanglement with the result of the function, which probably disables the interference required to iterate the corresponding piece of code.

4. Conclusions and Future Work

We have presented a quantum algorithm for doubling the amplitude of the state which corresponds to the solution of the partition problem. We further studied in detail the effects of applying first Hadamard, then S and finally Hadamard gates to the state $|0\rangle^{\otimes^n}$.

Since the proposed method doubles the amplitude of the states corresponding to a solution of the referred problem, one can infer that the reiteration of the method could lead us to a quantum polynomial algorithm that could solve the problem P = NP. Of course, we emphasize, that this is not our intention. Due to the way in which we *mark* the solution state pointed out at the end of the previous section, our piece of quantum code cannot directly be iterated as it can in the case of Grover's algorithm, therefore our proposal is simply to research to what extent this circuit could be either iterated/modified to be iterable, or, used as a shortcut to finish sooner some algorithms like Grover´s one.

The algorithm presented makes use of an oracle, and the approach is similar to the black box model as described in [11]; the idea of iterating this algorithm will suffer inevitably from the same limitation described there, and then the maximum speed-up should be limited to the polynomial, as it occurs with Grover's algorithm. It is our belief that together with any transformation aimed to make iterable the proposed code, it would come as drawback that its computational cost will compensate such an advantage.

Another idea related to our proposal is the possibility of application to the hidden subgroup problem. It is described in Section 5.4.3 of [3] and it has a known quantum solution by a variant of Shor's algorithm for the specific case of finite Abelian groups. However, the general problem remains open, with important consequences, for example for the graph isomorphism problem. This problem is discussed in Section 16.3 of [12] by using Boolean functions and their parity. This discussion is very similar to our Figure 2; thus, we think that it is worth investigating the possible relationship between them and, moreover, we also think that the problem of the hidden subgroup on Lie groups of general nature, such as SU(2) groups representing the quantum states of a single qubit, SU(4) for two qubits, and so on deserves to be studied. This could also offer an interesting research line for future works.

Author Contributions: Individual contributions are as follows: Conceptualization M.M. and J.J.P.; Formal analysis M.M., F.C. and F.L.P.; Investigation M.M., J.J.P., F.C., H.I.C. and F.L.P.; Writing—

original draft M.M., J.J.P., H.I.C. and F.C.; Writing—reviewing & editing M.M. and F.L.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received non-external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the anonymous reviewer who has pointed out the similarity between our treatment of the partition problem with the one discussed by Svozil for the hidden subgroup problem. This work was supported by project "FAME (Formal modeling and advanced testing methods. Applications to medicine and computing systems)", reference RTI2018-093608-B-C32, and by the Junta de Comunidades de Castilla-La Mancha project SBPLY/17/180501/000276/01 (cofunded with FEDER funds, EU).

Conflicts of Interest: The authors declare non-conflict of interest.

References

1. Shor, P. Algorithms for quantum computation: Discrete logarithms and factoring. In Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, USA, 20–22 November 1994; pp. 124–134. [CrossRef]
2. Deutsch, D.; Jozsa, R. Rapid solution of problems by quantum computation. *Proc. R. Soc. London Ser. Math. Phys. Sci.* **1992**, *439*, 553–558. [CrossRef]
3. Nielsen, M.A.; Chuang, I.L. *Quantum Computation and Quantum Information: 10th Anniversary Edition*, 10th ed.; Cambridge University Press: Cambridge, MA, USA, 2011.
4. Karp, R. Reducibility among combinatorial problems. In *Complexity of Computer Computations*; Miller, R., Thatcher, J., Eds.; Plenum Press: Berlin, Germany, 1972; pp. 85–103.
5. Chang, W.L.; Ren, T.T.; Feng, M.; Lu, L.C.; Lin, K.W.; Guo, M. Quantum Algorithms of the Subset-Sum Problem on a Quantum Computer. In Proceedings of the 2009 WASE International Conference on Information Engineering, Taiyuan, China, 10–11 July 2009; Volume 2, pp. 54–57. [CrossRef]
6. Bernstein, D.J.; Jeffery, S.; Lange, T.; Meurer, A. Quantum Algorithms for the Subset-Sum Problem. In *Post-Quantum Cryptography*; Gaborit, P., Ed.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 16–33.
7. Howgrave-Graham, N.; Joux, A. New Generic Algorithms for Hard Knapsacks. Cryptology ePrint Archive, Report 2010/189. 2010. Available online: https://eprint.iacr.org/2010/189 (accessed on 27 July 2021).
8. Helm, A.; May, A. Subset Sum Quantumly in 1.17^n. In Proceedings of the 13th Conference on the Theory of Quantum Computation, Communication and Cryptography, Sydney, Australia, 16 July 2018; pp. 5:1–5:16.
9. Li, Y.; Li, H. Improved quantum algorithm for the random subset sum problem. *arXiv* **2019**, arXiv:1912.09264.
10. Mezzini, M. Quirk's Url of an Instance of the Partition Problem. 2021. Available online: https://tinyurl.com/kp38rajd (accessed on 27 July 2021).
11. Beals, R.; Buhrman, H.; Cleve, R.; Mosca, M.; de Wolf, R. Quantum Lower Bounds by Polynomials. *J. ACM* **2001**, *48*, 778–797. [CrossRef]
12. Svocil, K. Quantum Queries Associated with Equi-partitioning of States and Multipartite Relational Encoding Across Space-Time. *Adv. Unconv. Comput. Emerg. Complex. Comput.* **2017**, *22*, 429–438. [CrossRef]

mathematics

MDPI

Article

Quantum Algorithms for Some Strings Problems Based on Quantum String Comparator †

Kamil Khadiev [1,*], Artem Ilikaev [1] and Jevgenijs Vihrovs [2]

[1] Institute of Computational Mathematics and Information Technologies, Kazan Federal University, Kremlevskaya 18, 420008 Kazan, Russia; artemka.tema1998@gmail.com
[2] Center for Quantum Computer Science, Faculty of Computing, University of Latvia, Raina 19, LV-1586 Riga, Latvia; jevgenijs.vihrovs@lu.lv
* Correspondence: kamil.hadiev@kpfu.ru
† This paper is an extended version of our paper published in Proceedings of TPNC2019 Conference.

Abstract: We study algorithms for solving three problems on strings. These are sorting of n strings of length k, "the Most Frequent String Search Problem", and "searching intersection of two sequences of strings". We construct quantum algorithms that are faster than classical (randomized or deterministic) counterparts for each of these problems. The quantum algorithms are based on the quantum procedure for comparing two strings of length k in $O(\sqrt{k})$ queries. The first problem is sorting n strings of length k. We show that classical complexity of the problem is $\Theta(nk)$ for constant size alphabet, but our quantum algorithm has $\tilde{O}(n\sqrt{k})$ complexity. The second one is searching the most frequent string among n strings of length k. We show that the classical complexity of the problem is $\Theta(nk)$, but our quantum algorithm has $\tilde{O}(n\sqrt{k})$ complexity. The third problem is searching for an intersection of two sequences of strings. All strings have the same length k. The size of the first set is n, and the size of the second set is m. We show that the classical complexity of the problem is $\Theta((n+m)k)$, but our quantum algorithm has $\tilde{O}((n+m)\sqrt{k})$ complexity.

Keywords: quantum computation; quantum algorithms; string processing; sorting

Citation: Khadiev, K.; Ilikaev, A.; Vihrovs, J. Quantum Algorithms for Some Strings Problems Based on Quantum String Comparator. *Mathematics* **2022**, *10*, 377. https://doi.org/10.3390/math10030377

Academic Editors: Fernando L. Pelayo, Mauro Mezzini and Jan Sładkowski

Received: 12 November 2021
Accepted: 22 January 2022
Published: 26 January 2022

1. Introduction

Quantum computing [1–3] is one of the hot topics in computer science in the last few decades. There are many problems where quantum algorithms outperform the best known classical algorithms [4–12].

One of these problems are problems for strings. Researchers show the power of quantum algorithms for such problems in [13–22].

In this paper, we consider three problems:

- Strings Sorting problem;
- the Most Frequent String Search problem;
- Intersection of Two String Sequences problem.

Our algorithms use some quantum algorithms as a subroutine, and the remaining part is classical. We investigate the problems in terms of query complexity. The query model is one of the most popular in the case of quantum algorithms. Such algorithms can do a query to a black box that has access to the sequence of strings. As a running time of an algorithm, we mean a number of queries to the black box.

In the paper, we suggested a quantum comparison procedure for two strings. We show that its quantum complexity is $\Theta(\sqrt{k})$, where k is the length of strings. The classical complexity is $\Theta(k)$. Thus, the quantum algorithm has a quadratic speed-up compared to classical algorithms. We propose a quantum algorithm that is based on "the first one search" (The minimal element satisfying a condition) problem algorithm from [23–26]. This algorithm is a modification of Grover's search algorithm [27,28]. Another important

algorithm for the search is described in [29]. Using this idea, we obtain quantum algorithms for several problems.

The first problem is the String Sorting problem. Assume that we have n strings of length k. It is known [30] that no quantum algorithm can sort arbitrary comparable objects faster than $O(n \log n)$. At the same time, several researchers tried to improve the hidden constant [31,32]. Other researchers investigated the space bounded case [33]. We focus on sorting strings. In a classical case, we can use an algorithm that is better than arbitrary comparable objects sorting algorithms. It is radix sort that has $O(nk)$ query complexity [34] for a finite size alphabet. It is also a lower bound for classical (randomized or deterministic) algorithms that is $\Omega(nk)$. Our quantum algorithm for the string sorting problem has query complexity $O(n(\log n) \cdot \sqrt{k}) = \tilde{O}(n\sqrt{k})$, where $\tilde{O}$ does not consider log factors. It is based on standard sorting algorithms [34] or Heapsort [34,35] and the quantum algorithm for comparing strings. Additionally, we use the idea of a noisy comparison procedure for sorting [36].

The second problem is the following. We have n strings of length k. We can assume that string symbols are letters from any constant size alphabet, for example, binary, Latin alphabet, or Unicode. The problem is finding the string that occurs in the sequence most often. The problem [37] is one of the most well-studied ones in the area of data streams [38–41]. Many applications in packet routing, telecommunication logging, and tracking keyword queries in search machines are critically based on such routines. The best-known classical (randomized or deterministic) algorithms require $\Omega(nk)$ queries because an algorithm should at least test all symbols of all strings. The deterministic solution can use the radix sort algorithm [34] or the Trie (prefix tree) [42–45] that allow achieving the required complexity.

We propose a quantum algorithm that is based on the sorting algorithm from the first problem. Our algorithm for the most frequent string search problem has query complexity $O(n(\log n) \cdot \sqrt{k}) = \tilde{O}(n\sqrt{k})$. If $\log_2 n = o(\sqrt{k})$, then our algorithm is better than classical counterparts. Note that this setup makes sense in practical cases.

The third problem is the Intersection of Two String Sequences problem. Assume that we have two sequences of strings of length k. The size of the first set is n, and the size of the second one is m. The first sequence is given, and the second one is given in an online fashion, one by one. After each requested string from the second sequence, we want to check whether this string belongs to the first sequence. We propose a quantum algorithm for the problem with quantum query complexity $O((n + m(\log m + \log\log n)) \cdot \log n \cdot \sqrt{k}) = \tilde{O}((n + m)\sqrt{k})$. The algorithm uses a quantum algorithm for sorting strings. At the same time, the best-known classical (randomized or deterministic) algorithm requires $\Omega((n + m)k)$ queries, and this bound is achieved using the radix sort algorithm or the Trie data structure.

The paper is an extended version of a conference paper [46].

The structure of the paper is the following. Discussion on the computation model is situated in Section 2. We present the quantum subroutine that compares two strings in Section 3. Then, we discuss three problems: Strings Sorting problem in Section 4, the Most Frequent String Search problem in Section 5, and Intersection of Two String Sequences problem in Section 6. Section 7 contains the conclusions.

2. Preliminaries

We use the standard form of the quantum query model. Let $f : D \to \{0,1\}, D \subseteq \{0,1\}^N$ be an N variable function. An input for the function is $x \in D$. We are given an oracle access to the input x, i.e., it is realized by a specific unitary transformation usually defined as $|i\rangle|z\rangle|w\rangle \to |i\rangle|z + x_i \pmod 2\rangle|w\rangle$, where the $|i\rangle$ register indicates the index of the variable we are querying, $|z\rangle$ is the output register, and $|w\rangle$ is some auxiliary workspace. Note that we use Dirac notation vectors. An algorithm in the query model consists of alternating applications of arbitrary unitaries independent of the input and the query unitary and a measurement in the end.

In the case of non-binary input, we present the input variables in binary form. Using alternating unitaries independent of the input, we can store bits in auxiliary work-space $|w\rangle$ and use the obtained variable in an algorithm. In the case of computing a complex function f and additionally non-binary input, we can consider a block of alternating unitaries independent of the input and the query unitary that stores required variables in the auxiliary work-space $|w\rangle$. Then, we compute the Boolean value of the function f on arguments and store them in the auxiliary work-space $|w\rangle$. After that, we can use the value of the function f in our algorithms.

The smallest number of queries for an algorithm that outputs $f(x)$ with a probability that is at least $\frac{2}{3}$ on all x is called the quantum query complexity of the function f and is denoted by $Q(f)$. We refer the readers to [1–3] for more details on quantum computing.

In the quantum algorithms in this article, we discuss quantum query complexity. We use modifications of Grover's search algorithm [27,28] as quantum subroutines. For these subroutines, time complexity (number of gates in a circuit) is more than query complexity for an additional log factor. Note that the query can be implemented using the CNOT gate.

3. The Quantum Algorithm for Comparing Two Strings

Firstly, we discuss a quantum subroutine that compares two strings of length k. Assume that this subroutine is COMPARE_STRINGS(s,t,k), and it compares s and t in the lexicographical order. It returns:

- -1 if $s < t$;
- 0 if $s = t$;
- 1 if $s > t$.

As a base for our algorithm, we use the algorithm of finding the minimal argument with 1-result of a Boolean-value function. Formally, we have:

Lemma 1 ([24,25], Theorem 10; [23], Section 2.2; [26], Proposition 4)**.** *Suppose we have a function $f : \{1,\ldots,N\} \to \{0,1\}$ for some integer N. There is a quantum algorithm for finding $j_0 = \min\{j \in \{1,\ldots,N\} : f(j) = 1\}$. The algorithm finds j_0 with the expected query complexity $O(\sqrt{j_0})$ and error probability that is, at most, $\frac{1}{2}$.*

Let us choose the function $f(j) = (s_j \neq t_j)$. Thus, we search for j_0 that is the index of the first unequal symbol of the strings. Then, we can claim that s precedes t in the lexicographical order if the symbol s_{j_0} precedes the symbol t_{j_0}. The claim is right by the definition of the lexicographical order. If there are no unequal symbols, then the strings are equal.

If we discuss the implementation of the f, then we can say that for computing the value $f(j)$, we store the binary representation of s_j and t_j in the auxiliary work-space, for example, $|\psi_s\rangle$ and $|\psi_t\rangle$. Then, compute the value of $f(j)$ and store it in a qubit $|\phi\rangle$. After that, we can use this value in the algorithm. The last step is clearing $|\phi\rangle$ using values of $|\psi_s\rangle$ and $|\psi_t\rangle$ and the CNOT gate; then, clearing $|\psi_s\rangle$ and $|\psi_t\rangle$ repeatedly using the same queries (that use CNOT gates). All these manipulations take a constant number of queries because of the constant size of the input alphabet.

We use THE_FIRST_ONE_SEARCH(f,k) as a subroutine from Lemma 1, where $f(j) = (s_j \neq t_j)$. Assume that this subroutine returns $k+1$ if it does not find any solution or the found argument j' is such that $f(j') = 0$.

We use the standard technique of boosting success probability. Thus, we repeat the subroutine $\lceil \log_2(\delta^{-1}) \rceil$ times and return the minimal answer.

Suppose the subroutine has an error. There are two cases. The first one is finding the index of unequal symbols that is not the minimal one. In the second case, the algorithm does not find unequal symbols. Then, we assume that it returns $k+1$. Thus, in a case of an error, the subroutine returns a value that is bigger than the correct answer.

Therefore, if at least one subroutine invocation has no error, then the whole algorithm succeeds. All error events are independent. The error probability of the whole algorithm is the probability of error for all invocations of the subroutine, that is $O\left(\frac{1}{2^{\log_2(\delta^{-1})}}\right) = O(\delta)$.

Let us present the Algorithm 1.

Algorithm 1 COMPARE_STRINGS(s, t, k). The Quantum Algorithm for Comparing Two Strings.

```
j_0 ← THE_FIRST_ONE_SEARCH(f, k)                          ▷ The initial value
for i ∈ {1, ..., ⌈log_2 δ^{-1}⌉} do
    j_0 ← min(j_0, THE_FIRST_ONE_SEARCH(f, k))
end for
if j_0 = k + 1 then
    result ← 0                                            ▷ The strings are equal.
else
    if (s_{j_0} < t_{j_0}) then
        result ← −1                                       ▷ s precedes t.
    else
        result ← 1                                        ▷ s succeeds t.
    end if
end if
return result
```

The next property follows from the previous discussion.

Lemma 2. *Algorithm 1 compares two strings of length k in the lexicographical order with query complexity $O(\sqrt{k}\log\delta^{-1})$ and error probability $O(\delta)$ for some integer k and $0 < \delta < 1$.*

The algorithm finds the minimal index of unequal symbols j_0. We can say that $j_0 - 1$ is the length of the longest common prefix for these strings.

We can show that the lower bound for the problem is $\Omega(\sqrt{k})$.

Lemma 3. *Any quantum algorithm for Comparing Two Strings problem has $\Omega(\sqrt{k})$ query complexity.*

Proof. Let us show that the problem is at least as hard as the unstructured search problem. Let $s_{\lfloor k/2\rfloor} = 1$ and $s_j = 0$ for all $j \in \{1, \ldots, \lfloor k/2\rfloor - 1, \lfloor k/2\rfloor + 1, \ldots k\}$. The string t is such that there is only one 1 in position z. In other words, there is $z \in \{1, \ldots, k\}$ such that $t_z = 1$ and $t_j = 0$ for all $j \in \{1, \ldots, z-1, z+1, \ldots k\}$.

If $z < \lfloor k/2\rfloor$, then $t > s$. If $z = \lfloor k/2\rfloor$, then $t = s$. If $z > \lfloor k/2\rfloor$, then $t < s$. Therefore, the problem is at least as hard as the search for 1 among the first $\lfloor k/2\rfloor$ variables in the string t.

It is known [14] that the quantum query complexity of the unstructured search among $\lfloor k/2\rfloor$ variables is $\Omega(\sqrt{k})$. □

At the same time, the classical complexity of the problem is $\Theta(k)$.

Lemma 4. *Randomized query complexity for Comparing Two Strings problem is $\Theta(k)$.*

Proof. Due to the proof of Lemma 3, the problem is at least as hard as the search for 1 among the first $k/2$ variables in the string t.

It is known [14] that the randomized query complexity of the unstructured search among $k/2$ variables is $\Omega(k)$.

At the same time, we can check all symbols sequentially to search the first unequal symbol. This algorithm has $O(k)$ query complexity. □

Additionally, we can compute the complexity of any algorithm based on the two strings comparison procedure.

Lemma 5. *Suppose we have some integer n, integer $A = A(n)$ and ε such that $\lim\limits_{n\to\infty} \varepsilon/A = 0$. Then, if a quantum algorithm does $A(n)$ comparisons of strings of length k and has $O(\varepsilon)$ error probability, then it does at most $O(A\sqrt{k}\log(A/\varepsilon))$ queries.*

Proof. As a strings comparison procedure, we use COMPARE_STRINGS subroutine for $\delta = \varepsilon/A$. Because of Lemma 2, the complexity of the subroutine is $O(\sqrt{k}\log(A/\varepsilon))$, and the error probability is $O(\varepsilon/A)$. Because of A comparison operations, the total complexity of the algorithm is $O(A\sqrt{k}\log(A/\varepsilon))$.

Let us discuss the error probability. Events of error in the algorithm are independent. Thus, all events should be correct. The error probability for one event is $1-(1-\varepsilon/A)$. Hence, the error probability for all A events is at least $1-(1-\varepsilon/A)^A = 1-(1-\varepsilon/A)^A$.

Note that

$$\lim_{n\to\infty} \frac{1-\left(1-\frac{\varepsilon}{A}\right)^A}{\varepsilon} = \lim_{n\to\infty} \frac{1-\left(1-\frac{\varepsilon}{A}\right)^{\frac{A}{\varepsilon}\cdot\varepsilon}}{\varepsilon} \leq 1;$$

Hence, the total error probability is at most $O(\varepsilon)$. □

4. Strings Sorting Problem

Let us consider the following problem.

Problem. For some positive integers n and k, we have the sequence of strings $s = (s^1, \ldots, s^n)$. Each $s^i = (s^i_1, \ldots, s^i_k) \in \Sigma^k$ for some finite size alphabet Σ. We search an order $ORDER = (i_1, \ldots, i_n)$ such that for any $j \in \{1, \ldots, n-1\}$, we have $s^{i_j} \leq s^{i_{j+1}}$ in the lexicographical order.

We use one of the existing sorting algorithms (for example, Heapsort algorithm [34,35] or the Merge sort algorithm [34]) as a base and the quantum algorithm for string comparison from Section 3. In fact, our comparison function can have errors. That is why we use the result for "noisy computation" from [36]. The result is presented in the following lemma.

Lemma 6 ([36], Theorem 3.5)**.** *Suppose we have a comparison procedure that works with error probability ε. Then there is a sorting algorithm with query complexity $O(n\log(n/\varepsilon))$ and error probability at most ε.*

The complexity of the algorithm is presented in the following theorem.

Theorem 1. *The algorithm sorts $s = (s^1, \ldots, s^n)$ with query complexity $O(n(\log n)\cdot\sqrt{k}) = \tilde{O}(n\sqrt{k})$ and constant error probability.*

Proof. The correctness of the algorithm follows from the description. Let $\varepsilon = 0.1$. Then, we apply the result from Lemma 6 and use the quantum comparison procedure that has ε error probability and $O(\sqrt{k}\log\varepsilon^{-1}) = O(\sqrt{k})$ query complexity. Therefore, the query complexity of the algorithm is $O(n(\log(n/\varepsilon))\cdot\sqrt{k}) = O(n(\log n)\cdot\sqrt{k}) = \tilde{O}(n\sqrt{k})$, and the error probability is ε. □

We can show the lower bound for the problem.

Theorem 2. *Any quantum algorithm for the Sorting problem has $\Omega(\sqrt{nk})$ query complexity.*

Proof. Let us show that the problem is at least as hard as the unstructured search problem. Assume that strings $s^1, \ldots, s^n$ are such that

- There is a pair (u, v) such that $s^u_v = 1$;
- For all pairs $(i, j) \neq (u, v)$, $s^i_j = 0$.

In that case, the answer is $ORDER = (i_1, \ldots, i_{n-1}, u)$, where $(i_1, \ldots, i_{n-1})$ is a permutation of integers from $\{1, \ldots, u-1, u+1, \ldots, n\}$. The searching for the required index u is at least as hard as the search for the 1-value variable s^u_v.

It is known [14] that the quantum complexity of the unstructured search among nk variables is $\Omega(\sqrt{nk})$. □

The lower bound for classical complexity can be proven by the same way as in Theorem 2.

Theorem 3. *The randomized query complexity of the Sorting problem is* $\Theta(nk)$.

Proof. Due to the proof of Theorem 2, the problem is at least as hard as the search for 1 among nk variables in the strings $s^1, \ldots, s^n$.

It is known [14] that the randomized query complexity of the unstructured search among nk variables is $\Omega(nk)$.

The Radix sort [34] algorithm reaches this bound and has $O(nk)$ complexity in a case of a finite alphabet. □

5. The Most Frequent String Search Problem

Let us formally present the problem.

Problem. For some positive integers n and k, we have a sequence of strings $s = (s^1, \ldots, s^n)$. Each $s^i = (s^i_1, \ldots, s^i_k) \in \Sigma^k$ for some finite size alphabet Σ. Let $\#(t) = |\{i \in \{1, \ldots, n\} : s^i = t\}|$ be the number of occurrences of a string t. We search for $i = argmax_{i \in \{1,\ldots n\}} \#(s^i)$. If several strings satisfy the condition, then the answer is the index of the string with minimal index in the set s. Formally, i is such that:

$$i = min\{j : \#(s^j) = \max_{z \in \{1,\ldots,n\}} \#(s^z)\}$$

Firstly, we present an idea of the algorithm.

The algorithm contains two steps. The first step is sorting the sequence of strings and obtaining $ORDER = (i_1, \ldots, i_n)$ such that for any $j \in \{1, \ldots, n-1\}$, we have $s^{i_j} \leq s^{i_{j+1}}$ in the lexicographical order. In that case, equal strings are situated sequentially. On the second step, we find each segment $[i_\ell, i_r]$ of indexes for equal strings, i.e., $s^j = s^{i_\ell}$ for $j \in \{i_\ell, \ldots, i_r\}$ and $s^{i_{\ell-1}} \neq s^{i_\ell}$ or $\ell = 1$, and $s^{i_{r+1}} \neq s^{i_r}$ or $r = n$. We check segments for different strings one by one. We store the longest segment's length as c_{max} and the minimal index of the string that corresponds to this segment in j_{max}. As in the sorting algorithm, in the second step of the algorithm, we apply the COMPARE_STRINGS subroutine for checking the equality of strings. Assume that we have the SORT_STRINGS(s) subroutine that implements the algorithm from Section 4.

Let us present the algorithm formally in Algorithm 2.

Let us discuss the complexity of the algorithm.

Theorem 4. *Algorithm 2 finds the most frequent string from* $s = (s^1, \ldots, s^n)$ *with query complexity* $O(n(\log n) \cdot \sqrt{k}) = \tilde{O}(n\sqrt{k})$ *and constant error probability.*

Proof. The correctness of the algorithm follows from the description. Let us discuss the query complexity. Because of Theorem 1, the sorting algorithm's complexity is $O(n(\log n)\sqrt{k})$, and the error probability is constant. The second step does $O(n)$ comparison operations. Let $\varepsilon' = 0.1$. Thus, because of Lemma 5, the second step of the algorithm algorithm does $O(n(\log n)\sqrt{k})$ queries, and the error probability is constant. The total complexity is $O(n(\log n)\sqrt{k} + n(\log n)\sqrt{k}) = O(n(\log n)\sqrt{k})$.

Error events of two steps are independent. Therefore, the error probability of the whole algorithm is also constant. We can achieve any required constant error probability by repetition. The technique is standard in both one-side and two-side errors. It can be seen, for example, in [16]. □

Algorithm 2 The Quantum Algorithm for the Most Frequent String Problem.

$(i_1, \ldots, i_n) = ORDER \leftarrow$ SORT_STRINGS(s) ▷ We sort $s = (s^1, \ldots, s^n)$.
$c_{max} \leftarrow 0, j_{max} \leftarrow -1$
$c \leftarrow 1, j \leftarrow i_1$
for $b \in (1, \ldots, n)$ **do**
  if $b = n$ or ($b \neq n$ and COMPARE_STRINGS$(s^{i_b}, s^{i_{b+1}}, k) \neq 0$) **then** ▷ We find the end of a segment
    if $c > c_{max}$ **then** ▷ If the current segment is longer than the current longest one
      $c_{max} \leftarrow c, j_{max} \leftarrow j$
    end if
    $c \leftarrow 1$
    if $b \neq n$ **then**
      $j = i_{b+1}$
    end if
  else
    $c \leftarrow c + 1$
    if $i_{b+1} < j$ **then** ▷ j is the minimal index of the current segment
      $j \leftarrow i_{b+1}$.
    end if
  end if
end for
return j_{max}

Theorem 5. *Suppose we have a constant ε such that $0 < \varepsilon < 3/4$. If the length of the strings $k \geq \log_2 n$, then any quantum algorithm for the Most Frequent String Search problem has $\Omega(\sqrt{nk} + n^{3/4-\varepsilon})$ query complexity. If $k < \log_2 n$, then any quantum algorithm for the Most Frequent String Search problem has $\Omega(\sqrt{nk})$ query complexity*

Proof. Let us show that the problem is at least as hard as the unstructured search problem. Assume that $n = 2t$ and $k > 1$ for some integer t. Then, let $s^{t+1}, \ldots, s^{2t} = 0^k$, where 0^k is a string of k zeros. Other strings can be $s^1, \ldots, s^t = 1^k$ or there are $z \in \{1, \ldots, t\}$ and $u \in \{1, \ldots, k\}$ such that $s^z_u = 0$ and $s^z_{u'} = 1$ for all $u' \in \{1, \ldots, u-1, u+1, \ldots, k\}$.

In the first case, the answer is 1^k. In the second case, the answer is 0^k. Therefore, solving the problem for this instance is equivalent to the search for 0 among the first $tk = nk/2$ variables.

According to [14], the quantum complexity of the unstructured search among $nk/2$ is $\Omega(\sqrt{nk})$.

In the case of odd n, we assign $s^n = 1^{k/2}0^{k/2}$, and it is not used in the search. Then, we can consider only $n-1$ strings. Thus, $n-1$ is even.

Let us consider the case of $k = 1$. If n is odd, then $s^n = 2$. Let $s^i = 0$ for $i \geq t+1$, and $t = \lfloor n/2 \rfloor$. Let us consider two cases. The first one is $s^i = 1$ for all $i \in \{1, \ldots, t\}$. The second case is $s^i = 1$ for all $i \in \{1, \ldots, t\} \backslash \{i_1\}$ and $s^{i_1} = 0$ for some $i_1 \in \{1, \ldots, t\}$. In the first case, the answer is 1. In the second case, the answer is 0. Therefore, solving the problem for this instance is equivalent to the search for 0 among the first $t = n/2 = nk/2$ variables.

Let us show that the problem is at least as hard as the d-distinctness problem [47]. Let d be such that $\frac{1}{4d} = \varepsilon/2$. Let b be the maximal integer that satisfies $n \geq b \cdot (d-1) + 1$. Let u^j be a binary representation of j for $j \in \{0, \ldots, b\}$.

Assume that $s^1 = u^1$ for other strings. We have two cases:

- **Case 1.** The sequence s contains $d-1$ copies of each u^j, where $j \geq 1$ and other strings are u^0. Formally:
  - $\#(u^j) = d-1$ for $j \in \{1, \ldots, b\}$;
  - $\#(u^0) = n - b \cdot (d-1)$.

- **Case 2.** The sequence s contains $d-1$ copies of each u^j, where $j \geq 1$ except some $j_m \in \{2,\ldots,b\}$; d copies of u^{j_m} and other strings are u^0. Formally:
  - $\#(u^{j_m}) = d$ for some $j_m \in \{2,\ldots,b\}$;
  - $\#(u^j) = d-1$ for $j \in \{1,\ldots,b\}\backslash\{j_m\}$;
  - $\#(u^0) = n - b\cdot(d-1)+1$.

In the first case, $\#(u^j) = d-1$ for $j \in \{1,\ldots,b\}$, $\#(u^0) \leq d-1$ and $s^1 = u^1$. Therefore, the answer is 1. In the second case, $\#(u^j) = d-1$ for $j \in \{1,\ldots,b\}\backslash\{j_m\}$, $\#(u^0) \leq d-1$ and $\#(u^{j_m}) = d$. Therefore, the answer is $i_m = \min\{i : s^i = u^{j_m}\}$. Note that $i_m \neq 1$ because $j_m \geq 2$ and $s^1 = u^1 \neq u^{j_m}$.

Hence, solving the problem for this instance is equivalent to checking whether there is a string that occurs in the input at least d times. It is the d-distinctness problem from [47]. It is known that the complexity of the problem is $\Omega\left(\frac{1}{4^d d^2 \cdot \log^{5/2} R}\cdot R^{3/4-1/(4d)}\right)$ for $R = \Theta(d^{d/2}n)$. In our case, the complexity is $\Omega(n^{3/4-\varepsilon})$. □

Secondly, let us discuss the classical complexity of the problem.

Theorem 6. *Any randomized algorithm for the Most Frequent String Search problem has* $\Theta(nk)$ *query complexity.*

Proof. The best-known classical algorithm uses the radix sort algorithm and does steps similar to the steps of the quantum algorithm.

The running time of this algorithm is $O(nk)$. At the same time, we can show that it is also a lower bound.

As it was shown in the proof of Theorem 5, the problem is at least as hard as the unstructured search problem among $nk/2$ variables. It is known [14] that the randomized complexity of the unstructured search among $nk/2$ variables is $\Omega(nk)$. □

6. Intersection of Two Sequences of Strings Problem

Let us consider the following problem.

Problem. For some positive integers n, m and k, we have the sequence of strings $s = (s^1,\ldots,s^n)$. Each $s^i = (s^i_1,\ldots,s^i_k) \in \Sigma^k$ for some finite size alphabet Σ. Then, we obtain m requests $t = (t^1 \ldots t^m)$, where $t^i = (t^i_1,\ldots,t^i_k) \in \Sigma^k$. The answer for a request t^i is 1 if there is $j \in \{1,\ldots,n\}$ such that $t^i = s^j$. We should answer 0 or 1 to each of m requests.

Let us present the algorithm that is based on the sorting algorithm from Section 4. We sort strings from s. Then, we answer each request using a binary search in the sorted sequence of strings [34] and COMPARE_STRINGS quantum subroutine for strings comparison during the binary search.

Let us present Algorithm 3. Assume that the sorting algorithm from Section 4 is the subroutine SORT_STRINGS(s), and it returns the order $ORDER = (i_1,\ldots,i_n)$. The subroutine BINARY_SEARCH_FOR_STRINGS$(t^i, s, ORDER)$ is the binary search algorithm with the COMPARE_STRINGS subroutine as a comparator, and it searches for t^i in the ordered sequence $(s^{i_1},\ldots,s^{i_n})$. Suppose that the subroutine BINARY_SEARCH_FOR_STRINGS returns 1 if it finds t and 0 otherwise.

Algorithm 3 The Quantum Algorithm for Intersection of Two Sequences of Strings Problem using sorting algorithm.

```
ORDER ← SORT_STRINGS(s)                          ▷ We sort s = (s^1,...,s^n).
for i ∈ {1,...,m} do
    ans ← BINARY_SEARCH_FOR_STRINGS(t^i, s, ORDER)  ▷ We search t^i in the ordered sequence.
    return ans
end for
```

The algorithms have the following query complexity.

Theorem 7. *Algorithm 3 solves Intersection of Two Sequences of Strings Problem with query complexity* $O((n+m)\sqrt{k}\cdot\log n\cdot\log(n+m)) = \tilde{O}((n+m)\sqrt{k})$ *and error probability* $O\left(\frac{1}{n+m}\right)$.

Proof. The correctness of the algorithm follows from the description.

Because of Theorem 1, the sorting algorithm's complexity is $O(n\log n\cdot\sqrt{k})$ and constant error probability.

Let us consider the second part of the algorithm. It does $O(m\log n)$ comparison operations for all invocations of the binary search algorithm. Let $\varepsilon = 0.1$. Thus, because of Lemma 5, the second part of the algorithm does

$$O(m\sqrt{k}\log n\log(m\log n)) = O(m\sqrt{k}\log n(\log m + \log\log n))$$

queries, and the error probability is constant.

Thus, the total complexity is $O((n+m(\log m+\log\log n))\sqrt{k}\log n)$. Error events of the two steps are independent. Therefore, the error probability of the whole algorithm is also constant. We can achieve any required constant error probability by repetition. □

The lower bound for the classical case can be proven using a result stated in [48] (Lemma 7, Section 5.1).

Theorem 8. *The randomized query complexity of Intersection of Two Sequences of Strings Problem is* $\Theta((n+m)k)$.

Proof. Assume that $n > m$. Let us consider $t^1 = 0^k$, and s^i contains only 0 s and 1 s, i.e., $s^i_j \in \{0,1\}$ for all $i \in \{1,\ldots,n\}, j \in \{1,\ldots,k\}$.

For checking $s^i = t^1$, it is enough to check $\neg \bigvee_{j=1}^{k}(s^i_j = 1)$ because this implies $s^i_j = 0$ for all $j \in \{1,\ldots,k\}$. In that case, checking for the existence of t^1 among s^i is the same as checking the following condition:

$$\neg \bigwedge_{i=1}^{n}\bigvee_{j=1}^{k}(s^i_j = 1)$$

This condition means that not all string s^i contains at least one 1.

The randomized complexity of computing $\neg \bigvee_{j=1}^{k}(s^i_j = 1)$ is the same as the complexity of the unstructured search for 1 among k variables, which is $\Omega(k)$. According to [48] (Lemma 7, Section 5.1), the total complexity of the function is $\Omega(nk)$.

Assume that $m > n$. Let us consider $s^i = 0^k$ for all $i \in \{1,\ldots,n\}$. The checking existence t^j among $s^1,\ldots,s^n$ is at least as hard as the search for 1 among $t^j_1,\ldots,t^j_k$ that requires $\Omega(k)$ queries. It is true for all $j \in \{1,\ldots,m\}$. Therefore, the total randomized complexity is $\Omega(mk)$.

Hence, if we join both cases, the randomized complexity of solving the problem is $\Omega(\max(n,m)\cdot k) = \Omega((n+m)\cdot k)$.

This complexity $O((n+m)k)$ can be reached if we use the radix sort algorithm and perform the same operations as in the quantum algorithm. □

Note that we can use the quantum algorithm for element distinctness [49,50] for this problem. The algorithm solves the problem of finding two identical elements in the sequence. The query complexity of the algorithm is $O(D^{2/3})$, where D is the number of elements in the sequence. The complexity is tight because of [51]. The algorithm can be the following. On j-th request, we can add the string t^j to the sequence $s^1,\ldots,s^n$ and invoke the element distinctness algorithm that finds a collision of t^j with other strings. Such approach

requires $\Omega(n^{2/3}\sqrt{k})$ queries for each request and $\Omega(mn^{2/3}\sqrt{k})$ for processing all requests. Note that the online nature of requests does not allow us to access all $t^1, \ldots, t^m$. Thus, each request should be processed separately.

In a case of $n \ll m$, we can use the Grover search algorithm for searching t^j among $(s_1, \ldots, s_n)$. The complexity is $\tilde{O}(m\sqrt{nk})$ in that case.

Because of the probabilistic behavior of the Oracle, we should use the approach similar to [52] that uses ideas of Amplitude Amplification [53].

7. Conclusions

In the paper, we propose a quantum algorithm for a comparison of strings and a general idea for any algorithm that does A string comparison operations. Then, using these results, we construct a quantum strings sorting algorithm that works faster than the radix sort algorithm, which is the best known deterministic algorithm for sorting a sequence of strings.

We propose quantum algorithms for two problems using the sorting algorithm: the Most Frequent String Search and Intersection of Two String Sequences. These quantum algorithms are more efficient than classical (deterministic or randomized) counterparts in a case of $\log_2(n) = o(\sqrt{k})$, where k is the length of strings and n is the number of strings. In a case of the Intersection of Two String Sequences problem, the condition is $\log_2(n)(\log_2 m + \log_2 \log_2 n) = o(\sqrt{k})$, where n and m are the number of strings in two sequences. Note that these assumptions are reasonable.

We discussed quantum and classical lower bounds for these problems. Classical lower bounds are tight, and at the same time, there is room to improve the quantum lower bounds.

Author Contributions: The main idea and algorithms, K.K. and A.I.; lower bounds, J.V. and K.K.; constructions and concepts, K.K., A.I. and J.V. All authors have read and agreed to the published version of the manuscript.

Funding: This paper has been supported by the Kazan Federal University Strategic Academic Leadership Program ("PRIORITY-2030"). J.V. Supported by the ERDF project 1.1.1.5/18/A/020 "Quantum algorithms: from complexity theory to experiment".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: We thank Aliya Khadieva, Farid Ablayev, Kazan Federal University quantum group and Krišjānis Prūsis from the University of Latvia for useful discussions.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Nielsen, M.A.; Chuang, I.L. *Quantum Computation and Quantum Information*; Cambridge University Press: Cambridge, UK, 2010.
2. Ambainis, A. Understanding Quantum Algorithms via Query Complexity. In Proceedings of the International Congress of Mathematicians, Rio de Janeiro, Brazil, 1–9 August 2018; Volume 4, pp. 3283–3304.
3. Ablayev, F.; Ablayev, M.; Huang, J.Z.; Khadiev, K.; Salikhova, N.; Wu, D. On quantum methods for machine learning problems part I: Quantum tools. *Big Data Min. Anal.* **2019**, *3*, 41–55. [CrossRef]
4. de Wolf, R. *Quantum Computing and Communication Complexity*; Institute for Logic, Language and Computation: Amsterdam, The Netherlands, 2001.
5. Jordan, S. Quantum Algorithms Zoo. 2021. Available online: http://quantumalgorithmzoo.org/ (accessed on 12 November 2021).
6. Khadiev, K.; Safina, L. Quantum Algorithm for Dynamic Programming Approach for DAGs. Applications for Zhegalkin Polynomial Evaluation and Some Problems on DAGs. In Proceedings of the UCNC, Tokyo, Japan, 3–7 June 2019; LNCS: Cham, Switzerland, 2019; Volume 4362, pp. 150–163.
7. Khadiev, K.; Kravchenko, D.; Serov, D. On the Quantum and Classical Complexity of Solving Subtraction Games. In Proceedings of the CSR 2019, Novosibirsk, Russia, 1–5 July 2019; LNCS: Cham, Switzerland, 2019; Volume 11532, pp. 228–236.

8. Khadiev, K.; Mannapov, I.; Safina, L. The Quantum Version Of Classification Decision Tree Constructing Algorithm C5. 0. In Proceedings of the CEUR Workshop Proceedings, Como, Italy, 9–11 September 2019; Volume 2500.
9. Kravchenko, D.; Khadiev, K.; Serov, D.; Kapralov, R. Quantum-over-Classical Advantage in Solving Multiplayer Games. *Lect. Notes Comput. Sci.* **2020**, *12448*, 83–98.
10. Khadiev, K.; Mannapov, I.; Safina, L. Classical and Quantum Improvements of Generic Decision Tree Constructing Algorithm for Classification Problem. *CEUR Workshop Proc.* **2021**, *2842*, 83–93.
11. Glos, A.; Nahimovs, N.; Balakirev, K.; Khadiev, K. Upper bounds on the probability of finding marked connected components using quantum walks. *Quantum Inf. Process.* **2021**, *20*, 6. [CrossRef]
12. Khadiev, K.; Safina, L. The quantum version of random forest model for binary classification problem. *CEUR Workshop Proc.* **2021**, *2842*, 30–35.
13. Montanaro, A. Quantum pattern matching fast on average. *Algorithmica* **2017**, *77*, 16–39. [CrossRef]
14. Bennett, C.H.; Bernstein, E.; Brassard, G.; Vazirani, U. Strengths and weaknesses of quantum computing. *SIAM J. Comput.* **1997**, *26*, 1510–1523. [CrossRef]
15. Ramesh, H.; Vinay, V. String matching in $O(\sqrt{n}+\sqrt{m})$ quantum time. *J. Discret. Algorithms* **2003**, *1*, 103–110. [CrossRef]
16. Ambainis, A.; Balodis, K.; Iraids, J.; Khadiev, K.; Kļevickis, V.; Prūsis, K.; Shen, Y.; Smotrovs, J.; Vihrovs, J. Quantum Lower and Upper Bounds for 2D-Grid and Dyck Language. In Proceedings of the 45th International Symposium on Mathematical Foundations of Computer Science (MFCS 2020), Prague, Czech Republic, 25–26 August 2020; Volume 170, pp. 8:1–8:14.
17. Khadiev, K.; Remidovskii, V. Classical and quantum algorithms for constructing text from dictionary problem. *Nat. Comput.* **2021**, *20*, 713–724. [CrossRef]
18. Khadiev, K.; Remidovskii, V. Classical and Quantum Algorithms for Assembling a Text from a Dictionary. *Nonlinear Phenom. Complex Syst.* **2021**, *24*, 207–221. [CrossRef]
19. Khadiev, K.; Kravchenko, D. Quantum Algorithm for Dyck Language with Multiple Types of Brackets. In Proceedings of Unconventional Computation and Natural Computation (UCNC 2021), Espoo, Finland, 18–22 October 2021; LNCS: Cham, Switzerland, 2021; Volume 12984, pp. 68–83.
20. Gall, F.L.; Seddighin, S. Quantum Meets Fine-grained Complexity: Sublinear Time Quantum Algorithms for String Problems. *arXiv* **2020**, arXiv:2010.12122.
21. Akmal, S.; Jin, C. Near-Optimal Quantum Algorithms for String Problems. *arXiv* **2021**, arXiv:2110.09696.
22. Ablayev, F.; Ablayev, M.; Khadiev, K.; Salihova, N.; Vasiliev, A. Quantum Algorithms for String Processing. *arXiv* **2020**, arXiv:2012.00372.
23. Kothari, R. An optimal quantum algorithm for the oracle identification problem. In Proceedings of the 31st International Symposium on Theoretical Aspects of Computer Science, Lyon, France, 5–8 March 2014; p. 482.
24. Lin, C.Y.Y.; Lin, H.H. Upper Bounds on Quantum Query Complexity Inspired by the Elitzur-Vaidman Bomb Tester. In Proceedings of the 30th Conference on Computational Complexity (CCC 2015), Portland, OR, USA, 17–19 June 2015; Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik: Dagstuhl, Germany, 2015.
25. Lin, C.Y.Y.; Lin, H.H. Upper Bounds on Quantum Query Complexity Inspired by the Elitzur–Vaidman Bomb Tester. *Theory Comput.* **2016**, *12*, 1–35. [CrossRef]
26. Kapralov, R.; Khadiev, K.; Mokut, J.; Shen, Y.; Yagafarov, M. Fast Classical and Quantum Algorithms for Online k-server Problem on Trees. *CEUR Workshop Proc.* **2022**, *3072*, 287–301.
27. Grover, L.K. A fast quantum mechanical algorithm for database search. In Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, Philadelphia, PA, USA, 22–24 May 1996; ACM: New York, NY, USA, 1996; pp. 212–219.
28. Boyer, M.; Brassard, G.; Høyer, P.; Tapp, A. Tight bounds on quantum searching. *Fortschritte Phys.* **1998**, *46*, 493–505. [CrossRef]
29. Long, G.L. Grover algorithm with zero theoretical failure rate. *Phys. Rev. A* **2001**, *64*, 022307. [CrossRef]
30. Høyer, P.; Neerbek, J.; Shi, Y. Quantum complexities of ordered searching, sorting, and element distinctness. *Algorithmica* **2002**, *34*, 429–448.
31. Odeh, A.; Elleithy, K.; Almasri, M.; Alajlan, A. Sorting N elements using quantum entanglement sets. In Proceedings of the Third International Conference on Innovative Computing Technology (INTECH 2013), London, UK, 29–31 August 2013; pp. 213–216.
32. Odeh, A.; Abdelfattah, E. Quantum sort algorithm based on entanglement qubits $\{00, 11\}$. In Proceedings of the 2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT), Farmingdale, NY, USA, 29–29 April 2016; pp. 1–5.
33. Klauck, H. Quantum time-space tradeoffs for sorting. In Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing, San Diego, CA, USA, 9–11 June 2003; ACM: New York, NY, USA, 2003; pp. 69–76.
34. Cormen, T.H.; Leiserson, C.E.; Rivest, R.L.; Stein, C. *Introduction to Algorithms*; MIT Press: Cambridge, UK, 2001.
35. Williams, J.W.J. Algorithm 232—Heapsort. *Commun. ACM* **1964**, *7*, 347–349.
36. Feige, U.; Raghavan, P.; Peleg, D.; Upfal, E. Computing with noisy information. *SIAM J. Comput.* **1994**, *23*, 1001–1018. [CrossRef]
37. Cormode, G.; Hadjieleftheriou, M. Finding frequent items in data streams. *Proc. Vldb Endow.* **2008**, *1*, 1530–1541. [CrossRef]
38. Muthukrishnan, S. Data streams: Algorithms and applications. *Found. Trends Theor. Comput. Sci.* **2005**, *1*, 117–236. [CrossRef]
39. Aggarwal, C.C. *Data Streams: Models and Algorithms*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2007; Volume 31.
40. Becchetti, L.; Chatzigiannakis, I.; Giannakopoulos, Y. Streaming techniques and data aggregation in networks of tiny artefacts. *Comput. Sci. Rev.* **2011**, *5*, 27–46. [CrossRef]

41. Boyar, J.; Larsen, K.S.; Maiti, A. The frequent items problem in online streaming under various performance measures. *Int. J. Found. Comput. Sci.* **2015**, *26*, 413–439. [CrossRef]
42. De La Briandais, R. File searching using variable length keys. In Proceedings of the Papers Presented at the the Western Joint Computer Conference, San Francisco, CA, USA, 3–5 March 1959; ACM: New York, NY, USA, 1959; pp. 295–298.
43. Black, P.E. Dictionary of Algorithms and Data Structures | NIST. 1998. Available online: http://www.nist.gov/dads (accessed on 12 November 2021).
44. Brass, P. *Advanced Data Structures*; Cambridge University Press: Cambridge, UK, 2008; Volume 193.
45. Knuth, D. *Searching and Sorting, the Art of Computer Programming*; Addison-Wesley: Reading, MA, USA, 1973; Volume 3.
46. Khadiev, K.; Ilikaev, A. Quantum Algorithms for the Most Frequently String Search, Intersection of Two String Sequences and Sorting of Strings Problems. In Proceedings of the International Conference on Theory and Practice of Natural Computing, Kingston, ON, Canada, 9–11 December 2019; pp. 234–245.
47. Mande, N.S.; Thaler, J.; Zhu, S. Improved Approximate Degree Bounds for k-Distinctness. In Proceedings of the 15th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2020), Riga, Latvia, 9–12 June 2020; Schloss Dagstuhl-Leibniz-Zentrum für Informatik: Dagstuhl, Germany, 2020.
48. Göös, M.; Jayram, T.; Pitassi, T.; Watson, T. Randomized communication vs. partition number. In Proceedings of the 44th International Colloquium on Automata, Languages, and Programming (ICALP 2017), Warsaw, Poland, 10–14 July 2017; Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik: Dagstuhl, Germany, 2017.
49. Ambainis, A. Quantum walk algorithm for element distinctness. *SIAM J. Comput.* **2007**, *37*, 210–239. [CrossRef]
50. Ambainis, A. Quantum Walk Algorithm for Element Distinctness. In Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, FOCS '04, Rome, Italy, 17–19 October 2004; pp. 22–31.
51. Aaronson, S.; Shi, Y. Quantum lower bounds for the collision and the element distinctness problems. *J. ACM* **2004**, *51*, 595–605. [CrossRef]
52. Høyer, P.; Mosca, M.; de Wolf, R. Quantum Search on Bounded-Error Inputs. In *Automata, Languages and Programming*; Baeten, J.C.M., Lenstra, J.K., Parrow, J., Woeginger, G.J., Eds.; Springer: Berlin/Heidelberg, Germany, 2003; pp. 291–299.
53. Brassard, G.; Høyer, P.; Mosca, M.; Tapp, A. Quantum amplitude amplification and estimation. *Contemp. Math.* **2002**, *305*, 53–74.

mathematics

MDPI

Article

GPS: A New TSP Formulation for Its Generalizations Type QUBO

Saul Gonzalez-Bermejo [1], Guillermo Alonso-Linaje [1] and Parfait Atchade-Adelomou [2,3,*]

[1] Facultad de Ciencias, Campus Miguel Delibes, Universidad de Valladolid, C/Plaza de Santa Cruz, 8, 47002 Valladolid, Spain; saul.gonzalez@alumnos.uva.es (S.G.-B.); guillermo.alonso.alonso-linaje@alumnos.uva.es or guillermo@alonso-linaje.com (G.A.-L.)
[2] Research Group on Data Science for the Digital Society, La Salle, Universitat Ramon Llull, Carrer de Sant Joan de La Salle, 42, 08022 Barcelona, Spain
[3] Lighthouse Disruptive Innovation Group, LLC, 7 Broadway Terrace, Apt. 1, Cambridge, MA 02139, USA
* Correspondence: parfait.atchade@salle.url.edu or parfait.atchade@lighthouse-dig.com

Abstract: We propose a new Quadratic Unconstrained Binary Optimization (QUBO) formulation of the Travelling Salesman Problem (TSP), with which we overcame the best formulation of the Vehicle Routing Problem (VRP) in terms of the minimum number of necessary variables. After, we will present a detailed study of the constraints subject to the new TSP model and benchmark it with MTZ and native formulations. Finally, we will test whether the correctness of the formulation by entering it into a QUBO problem solver. The solver chosen is a D-Wave_2000Q6 quantum computer simulator due to the connection between Quantum Annealing and QUBO formulations.

Keywords: quantum computing; quantum annealing; combinatorial optimization; QUBO; TSP; VRP

Citation: Gonzalez-Bermejo, S.; Alonso-Linaje, G.; Atchade-Adelomou, P. GPS: A New TSP Formulation for Its Generalizations Type QUBO. *Mathematics* **2022**, *10*, 416. https://doi.org/10.3390/math10030416

Academic Editors: Fernando L. Pelayo, Mauro Mezzini and Armin Fügenschuh

Received: 9 December 2021
Accepted: 24 January 2022
Published: 28 January 2022

1. Introduction

The Travelling Salesman Problem, known as TSP [1], is one of the most studied statements belonging to the combinatorial optimisation problems. In this, we are given a set of cities and the distances between them with which we must try to find the best route to travel all the towns, minimizing the total length.

Both the TSP and its more well-known derivative, the Vehicle Routing Problem (VRP), are routing problems with a great impact on most of the issues in our society. For this reason, and because both are NP-Hard [2], the scientific community has continued the search for a better formulation that makes their resolution efficient. However, unfortunately, we cannot use traditional search methods based on differentiability when defining the problem with discrete variables.

One of the models that allows us to write TSP like problems more generically is Quadratic Unconstrained Binary Optimization (QUBO) [3]. QUBO is a framework that enables us to model problems in a quadratic form subject to linear restrictions natively. However, with the help of penalty functions, it is possible to reformulate the tasks of order greater than two and inequality constraints to the QUBO model. Another characteristic that makes QUBO a very important modelling environment is its close connection with the Ising model [4]. The QUBO model constitutes a central problem for adiabatic quantum computing [5], which is solved through a physical process called quantum annealing [6,7].

It is known that the best current QUBO formulation of the TSP requires N^2 binary variables Appendix A.1. However, when we try to generalise this formulation to other set of problems such as VRP, we find that polynomial terms of order greater than two appear in these models. As QUBO modelling requires that the function to minimize must be quadratic, it is necessary to decrease the degree of these terms by introducing auxiliary variables, which greatly increases the number of required variables. This is crucial to

achieving good results through the solvers dedicated to it, especially if it is going to be implemented on a quantum computer.

Quantum annealing is the paradigm of using quantum processes to solve combinatorial optimization problems. This paradigm uses entropy as a target to force exploration, given that any function that smoothes the probability in the search space can have the same purpose according to the adiabatic theorem [8,9].

The D-Wave Quantum Processor Unit (QPU) is considered as a heuristic that minimizes the objective QUBO functions using a physically performed version of quantum annealing; this shows how the number of variables in the QUBO model is related to the number of qubits in a quantum computer [10].

The VRP encompasses two different problems: one in which the distance travelled by vehicles subject to capacity restrictions is minimized [11–14] and another, in which the time taken for cars to complete their routes is minimized. In this article, everything related to the VRP will optimize the time to complete these routes, equivalent to reducing the total of the distances travelled by all vehicles.

As we will explain later in the section dedicated to the VRP, to generalise a QUBO formulation from the TSP to the VRP, the objective function for calculating the distance travelled by the vehicles must be linear so that the formulation discussed above with N^2 variables cannot be used.

The most widely used QUBO model of the TSP that can represent distance linearly uses more than N^3 variables (native TSP formulation). However, there is indeed a formulation that uses $N^2 log_2(N)$ variables; this formulation is known as MTZ [15]. But generalising the MTZ formulation for the VRPs that minimises the maximum distances travelled by all the vehicles give us quadratic restrictions and, therefore, a penalty function of the order greater than two.

Our purpose in this work is to present a new QUBO model of the TSP in which the travelled distance calculation is linear, and uses only $3N^2$ variables, considerably improving the existing TSP models (both of N^3 and $N^2 log_2(N)$ variables). Furthermore, this new formulation of the TSP we refer to as GPS will be generalized to define an efficient formulation which consider the number of variables of a new VRP formulation. Unfortunately, after reviewing state of the art, we have not found a QUBO formulation of the VRP that minimises the maximum of the distances travelled by all the vehicles, thus we have not been able to make comparisons. When we talk about the number of variables in a model, we will describe it according to its dominant term, so for a model that, for example, requires $N^3 + 3N^2 + 2N$ variables, we will say that it is modelling with N^3 variables since it gives us enough information on its scalability.

The document is organised as follows. Section 2 presents our main motivation behind this work. Section 3 shows previous work on the TSP algorithm and its derivatives. Section 4 presents the QUBO framework and its connection to quantum annealing. In Section 5, we recall the native formulation of TSP and the MTZ QUBO model. Section 6 presents our TSP proposal with the improvements in the numbers of variables. A generalisation of our contribution is seen in Section 7 where we propose our VRP into the QUBO model. Section 8 presents the obtained results, and finally, Section 9 concludes the work carried out, and we open ourselves to some lines of the future of the proposed model.

2. Motivation

Our primary motivation is to find a suitable formulation that uses the minimum number of variables; and thus, the minimum number of qubits when implementing said models in quantum computers. This motivation is increased by solving the problem presented in our article [14] in which we desire that the mobile robots minimise the time, which is equivalent to reducing the maximum of the distances travelled by all the vehicles. What implies reducing the number of qubits necessary to implement this model in this era of very few qubits.

3. Related Work

In the mid-1920s, the following referenced articles [16,17], were the first articles to provide a solution to the minimal spanning tree (MST) problem. Based on these works, the mathematical researcher, Joseph B. Kruskal Jr, applied these solutions to the TSP [18], giving life to some of the first solutions to this problem that will arise during the next decades.

Towards the end of the sixties, the following article [19] offered a compilation and synthesis of the research on the travelling salesman problem. The authors began by defining the problem and presenting several relevant theorems. They also classified and detailed the solution techniques and computational results. Before that, in the mid-1960s, the TSP started to emerge in many different contexts. The following article [20] highlights some applications that began to occur in everyday life, such as vehicle routing or job shop scheduling problems. Other applications such as planning, logistics and the manufacture of electronic circuits became of particular interest.

By making a few small modifications to the original TSP, we could apply it in many fields such as SWP [21] and DNA sequencing [22,23] among others. In this last application, the concept of 'city' would come to be fragments of DNA and the idea of 'distance', a measure of similarity between the pieces of DNA. In many applications, additional restrictions such as resource limits or time windows make the problem considerably difficult.

Computationally, the TSP [24] is an NP-Hard problem within combinatorial optimization. As an NP-Hard problem, it is computationally complex, and heuristics are continually being developed to get as close as possible to the optimal solution. However, considering the computational complexity nature of these problems, the new approach that quantum computing takes is very promising.

Many works are related to the standard/native TSP or some related variant in a quantum environment within this new approach. For example, the referenced work [25], the authors introduced a different quantum annealing scheme based on a path-integral Monte Carlo process to address the symmetric version of the Travelling Salesman Problem (sTSP). In these other articles [26,27], the authors did a comparative study using the D-Wave platform to evaluate and compare the efficiency of quantum annealing with classical methods for solving standard TSP.

In this reference [28], several comparisons of heuristic techniques were made for some TSP Libraries (TSPLIB) [29] problems, both symmetric and asymmetric, and their results have been compared to other methods such as Self Organizing Maps and Simulated Annealing [30]. In both cases, the local search technique was applied to the results found with Wang's Recurrent Neural Network with "Winner Takes All" that improved the Self Organizing Maps [31]. Other techniques such as the co-adaptive neural network approach to the Euclidean Travelling Salesman Problem [32] equally important.

One of the generalizations of the TSP, known as the VRP, was studied on the D-Wave platform [33,34]. In tasks where routing and planning capacity (time) was required, the TSP with time windows (TSPTW) was generalized [35,36], and has high inherent complexity which presents enormous resolution difficulties. In the following references [21,37–39], the authors modelled combinatorial optimisation problems in which social workers visit their patients at their respective homes and attend to them at a specific time, called Social Workers' Problem (SWP). SWP is a significant problem because additional time constraints allow more realistic scenarios to be modelled than native TSP. The optimal or near-optimal solution for such issues is important in minimising distance and time and environmental problems such as reducing fuel consumption.

The generalization of the TSP that we will use in our work will be the VRP. However, there are other TSP derivatives, such as the Job Shop Scheduling Problem (JSSP) [40] that are not included in the study of this work.

During state of the art of these formulations carried out, we have found several articles [33,34,41] that solve the TSP and VRP (focusing on minimising distance and not time) for annealing computers [7,30,42]. However, the number of variables is still intractable for the current size of quantum computers. For this reason, we propose a new TSP formulation

with a representation of the linear distance that uses only $3N^2$ variables, which we will use to outperform the current best VRP modelling in terms of the number of required variables. For example, a possible formulation of the VRP uses N^3 variables where N is the number of cities, thus with only 10 towns, we would go to 1000 necessary variables. In quantum computing, each of these variables can be represented with a qubit, and that is why computers possessing 1000 qubits would be needed to carry out these tasks. However, the gate-based computers that mark this era of quantum computing [43] have around 100 qubits making this task intractable today. The number of qubits is higher for computers based on quantum annealing, reaching 2000 qubits like the D-Wave computer. However, the correspondence between variables and qubits will not be one-to-one due to the architecture of these computers, so that we will have a smaller number of useful qubits. The following reference [44] deals with the topology and graph problem mapping on the D-Wave 2000Q QPU computer in detail.

4. QUBO Model in Quantum Computing

Quantum computing as a new computational paradigm can help solve a set of complex problems (routing, scheduling, banking problems, etc.) or solve tasks that respond to the law of quantum mechanics. However, before solving a problem, we first need to express it in a mathematical formulation that is largely compatible with the underlying physical hardware. This methodology is also useful for quantum computation. One of the frameworks that allows us to define said mathematical formulation to be solved in a quantum computer is the QUBO.

Adiabatic computation was born from the use of the adiabatic theorem [8,9] to perform the calculations using the tunnel effect to go from the global minimum of a simple Hamiltonian (A Hamiltonian system is a dynamic system governed by Hamilton equations. In physics, these active systems describe the evolution of a physical system, such as an electron in an electromagnetic field.) [45,46] to the global minimum of the problem of interest.

One of the market leaders for this type of computing is D-Wave, which roughly solves the quadratic unconstrained binary optimisation problem (QUBO). The QUBO formulation (1) is suitable for running a D-Wave architecture [47]; however, QUBO can be mapped to the Ising [6] model and thus be used in computers based on quantum gates, for example, IBMQ, Rigetti, Xanadu (strawberryfields), etc.

The problems that D-Wave quantum computers are prepared to solve are those that consist of finding the minimum of a function of the following form:

$$\sum_{i=1}^{n} b_i x_i + \sum_{i=1}^{n} \sum_{j=1}^{n} q_{i,j} x_i x_j, \tag{1}$$

where the variables $x_i \in \{0,1\}$ and the coefficients $b_i, q_{i,j} \in \mathbb{R}$.

We are then, given a problem, we need to model it with the above structure where the variables that form the solution will only take the values 0 or 1. Let us observe that, by taking the variables x_i the values 0 or 1, it is true that $x_i^2 = x_i$. Therefore, we can group the linear terms with the quadratic terms and express the above equation in matrix format:

$$x^t Q x, \tag{2}$$

with $x \in \{0,1\}^n$ and $Q \in \mathfrak{M}_{n\times n}$ which is compactly representing the QUBO formulation. QUBO can be mapped into the Ising model with the change variable: $z = 1 - 2x$. Thus, we pass a binary variable $(0,1)$ to a spin variable $(-1,1)$. Therefore, given a formulation of a problem to the QUBO, we can implement it and solve it in computers based on quantum gates, only applying the change of variable mentioned.

5. TSP Formulation

As discussed in the introduction, before presenting our GPS model in section four, we will analyze the native TSP model and the MTZ that we aim to improve in terms of the number of variables.

5.1. Native Formulation

In this section, we will recall the formulation of the native TSP [41]. This modelling, which has been defined in [48], despite appearing in a very natural way which facilitates its understanding, requires N^3 variables to be implemented.

The variables that appear in this model are the variables $x_{i,j,t}$ such that $i, j \in \{0, \ldots, N+1\}$ and $t \in \{0, \ldots, N\}$. Let us consider that the variables $x_{i,i,t}$ do not exist in this model. The interpretation of the variables $x_{i,j,t}$ is simple, since $x_{i,j,t} = 1$ if at instant t we traverse the edge that connects the cities i and j, and $x_{i,j,t} = 0$ for all other cases.

We can define the objective function of the native (Native in the sense of general, the most used) TSP [41] as:

$$\sum_{u=0}^{N+1} \sum_{v=0}^{N+1} \sum_{t=0}^{N} x_{u,v,t} d_{u,v}. \tag{3}$$

where $d_{u,v}$ represents the distance between nodes u and v. This objective function is subject to a series of restrictions:

- Constraint 1. The salesman must leave each city once.

$$\text{For each } u \in \{0, \ldots, N\}: \sum_{v=1}^{N+1} \sum_{t=0}^{N} x_{u,v,t} = 1. \tag{4}$$

- Constraint 2. Each city must be reached once.

$$\text{For each } v \in \{1, .., N+1\}: \sum_{u=0}^{N} \sum_{t=0}^{N} x_{u,v,t} = 1. \tag{5}$$

- Constraint 3. If the salesman leaves a city, he cannot return to it later. This constraint ensures that no unconnected cycles are formed as a solution. There are two ways of posing this constraint.
  - Imposing that once he leaves a city he cannot return to it.
    For each $u \in \{1, \ldots, N+1\}$:

$$\sum_{v=0}^{N+1} \sum_{t=0}^{N} \sum_{w=0}^{N+1} \sum_{j=t+1}^{N} x_{u,v,t} x_{w,u,j} = 0. \tag{6}$$

  - Imposing that once he arrives in a city, he must leave it.
    For each $t \in \{0, \ldots, N-1\}$, $u, v \in \{0, \ldots, N\}$:

$$x_{u,v,t}(1 - \sum_{w=1}^{N+1} x_{v,w,t+1}) = 0. \tag{7}$$

This formulation requires N^3 variables. Next, we will analyse another model used to define the TSP which is less commonly used in quantum *annealing* articles.

5.2. MTZ Formulation

Recalling the idea of this formulation is to consider the variables $x_{i,j} = 1$ if the edge that connects the cities i and j appears in the solution path, where $x_{i,j} = 0$ for all other cases. Once we have these variables, we can establish order on the route by employing a set of variables that will represent the moment the salesman arrives at that city (the variable u_i

expressed in binary format, will take the integer value t if the city i is reached in the tth position.). This model requires $N^2 log_2(N)$, greatly improving the number of variables in the general formulation. However, when implemented using *annealing* it presents surprisingly inaccurate results. This is because the *annealing* algorithm gets stuck trying to minimise the part of the objective function generated by the sub-tour's constraint [15], since the representation of integers in their binary format has the disadvantage that close numbers such as $2^n - 1$ and 2^n differ by a large number of qubits, so from the *annealing* they are perceived as very different solutions.

Once the two most common QUBO models of the TSP have been presented, let us analyze the formulation with which we improve the number of variables of the previous two.

6. GPS Formulation

Now we are starting to present our work. To develop this model, we take the variables $x_{i,j,r}$ with $i, j \in \{0, \ldots, N+1\}$ and $r \in \{0, 1, 2\}$. In all the modelling, the variables $x_{i,j,r}$ such that $i = j$ are not considered. We work with directional edges, that is, if in the model the edge (i, j) appears, we understand that first we must go through node i and immediately after that we go to j. Let us analyse what each variable represents:

- $x_{i,j,0} = 1$ means that the edge (i, j) does not appear in the path and the node i is reached earlier than the j.
- $x_{i,j,1} = 1$ means that the edge (i, j) appears in the path, so the node i is reached earlier than the j.
- $x_{i,j,2} = 1$ means that the edge (i, j) does not appear in the path, and the node j is reached earlier than the i.

Let us, therefore, see some examples (Figure 1) in which these variables do not take the value zero:

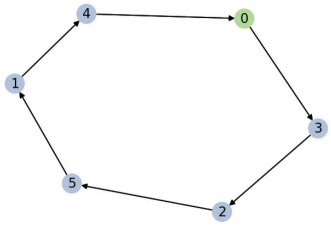


Figure 1. Example of a TSP solution with six different cities. It begins at node 0, and the arrows indicate the order in which the towns will be visited.

In this particular case, $x_{4,5,0} = 0$ and $x_{4,5,2} = 0$ since the edge $(4, 5)$ does appear in the solution. On the other hand $x_{4,5,1}$ will also be 0 because although edge $(4, 5)$ does appear in the graph, node 5 will be visited before node 4.

Let us, therefore, see examples in which these variables do not take the value zero:

- $x_{5,1,1} = 1$: in this case it will take the value 1 since edge (5, 1) appears in the solution and node 5 is visited first.
- $x_{4,1,2} = 1$: because in the solution we don't have the connection (4, 1), we have the connection (1, 4) and the node 4 is visited later node 1.
- $x_{5,3,2} = 1$: since the edge $(3, 5)$ does not appear and node 3 is visited first.

From the definition of our variables, we can define the distance travelled through the following objective function as:

$$\sum_{i=0}^{N+1} \sum_{j=0}^{N+1} d_{i,j} x_{i,j,1}. \tag{8}$$

The constraints that must be met are:

- Constraint 1: For each i, j one and only one of the 3 cases of r must be given, so

$$\text{For all } i, j\text{: } \sum_{r=0}^{2} x_{i,j,r} = 1. \tag{9}$$

- Constraint 2: Each node must be exited once.

$$\text{For each } i \in \{0, \ldots, N\}\text{: } \sum_{j=0}^{N+1} x_{i,j,1} = 1. \tag{10}$$

- Constraint 3: Each node must be reached once.

$$\text{For each } j \in \{1, \ldots, N+1\}\text{: } \sum_{i=0}^{N} x_{i,j,1} = 1. \tag{11}$$

- Constraint 4: If node i is reached before j, then node j is reached after i, so, for all $i, j \in \{0, \ldots, N+1\}$ such that $i \neq j$:

$$x_{i,j,2} = 1 - x_{j,i,2}. \tag{12}$$

  It would also have to be specified for $r = 0$ and $r = 1$, however this restriction is sufficient since by (9) it is implicit.
- Constraint 5: If node i is reached before node j and node j is reached before node k, then node i must be reached before k. This condition will prevent the route from returning to a node from which it had already exited, thus preventing cycles from forming. We then arrive at the penalty function Equation (13).

$$\sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{N} (x_{j,i,2} x_{k,j,2} - x_{j,i,2} x_{k,i,2} - x_{k,j,2} x_{k,i,2} + x_{k,i,2}). \tag{13}$$

With only the cases in which $i \neq j$, $i \neq k$ and $j \neq k$ will be taken in the summation and in the annex (Appendix B) we will provide the approach followed to arrive at it.

The following is deduced from the Equation (13). We have $x_{i,j,2} = 0$ if i is reached before j and $x_{i,j,2} = 1$ in the case where j is reached before i. Thus, with the previous equation we penalise these following cases in which $x_{i,j,2} = 0, x_{j,k,2} = 0$ and $x_{i,k,2} = 1$ and $x_{i,j,2} = 1, x_{j,k,2} = 1$ and $x_{i,k,2} = 0$ which lead to cases in which it would be forming cycles (for these two situations the value of the parentheses is 1 and for the rest 0). In (27) we have a similar situation for our VRP formulation, where we offer more details. For this condition, we must have directly constructed a penalty function that avoids erroneous cases without first posing linear conditions through which to generate its corresponding penalty function.

Formulated in this way we have managed to reduce the number of variables required from $N^2 \log_2 N$ to $3N^2$, achieving very noticeable reductions when working with large problems. Once we have this formulation, let us see how we can generalise it to the new VRP formulation.

7. New VRP Formulation

This section develops our VRP into the QUBO model using the GPS formulation.

As discussed in the introduction, this model is optimal concerning the number of binary variables used. However, this generalisation does not appear as naturally as expected because it requires a delicate step to get the constraints of the Equation (27). To do this, we will detail each step and explain each of the constraints step by step.

Original Formulation $5N^2Q$

For this new VRP, we will consider that N is the number of cities and Q is the number of available vehicles. We first present the variables that will form the problem. We then take the following set of variables.

$$x_{i,j,r,q} \text{ with } i,j \in \{0,\ldots,N+1\}, r \in \{0,1,2,3,4\} \text{ and } q \in \{1,\ldots,Q\} \tag{14}$$

In all the modelling, the variables $x_{i,j,r}$ such that $i = j$ are not considered. The variables i, j refer to the cities must travel to, and the variable q refers to the vehicle. The nodes 0 and $N+1$ correspond to the starting and ending points. Note that they may be the same node but we will separate them for convenience in the formulation. The values $d_{i,j}$ with $i,j \in \{0,\ldots,N+1\}$ correspond to the distance between node i and j. Let us dive into the interpretation of each variable:

- $x_{i,j,0,q} = 1$ means that the vehicle q travels to the cities i and j, does not travel across the edge (i,j) and arrives at the city i before the j.
- $x_{i,j,1,q} = 1$ means that the vehicle q travels to the cities i and j travels across the edge (i,j) (that is, once it passes through the city i the next city it reaches is the j) and therefore the city i is reached earlier than the city j.
- $x_{i,j,2,q} = 1$ means that the vehicle q travels through the cities i and j and arrives at the city j earlier than at the city i.
- $x_{i,j,3,q} = 1$ means that the vehicle q does not go through the cities i and j, and the city i is reached earlier than the city j. Note that $x_{i,j,3,q}$ can take the value 1 whether the vehicle q passes through one of both cities or neither of them.
- $x_{i,j,4,q} = 1$ means that the vehicle q does not travel to the cities i and j, and the city j is reached earlier than the city i.

Even if no vehicle passes through the objects i and j, the formulation must establish an order between them. However, this restriction does not make the modelling meaningless, since we can assume that if the vehicles are ordered in the order of $\{1,\ldots,Q\}$, then i will be reached before j if the vehicle that passes through node i has a lower number than the one that passes through node j. Once the interpretation of each variable is explained, let us analyse the constraints that must be met.

- Constraint 1: For each i,j,q, one and only one of the possibilities must be met for r, so:

$$\text{For all } i,j,q\text{: } \sum_{r=0}^{4} x_{i,j,r,q} = 1, \tag{15}$$

- Constraint 2: Each vehicle has to fulfill that it leaves the starting position. For this situation, we are going to impose that:

$$\text{For all } q\text{: } \sum_{j=1}^{N+1} x_{0,j,1,q} = 1, \tag{16}$$

No vehicle can return to the starting position from a city, so:

$$\text{For all } q\text{: } \sum_{i=0}^{N+1} x_{i,0,1,q} = 0, \tag{17}$$

- Constraint 3: Every vehicle must finish in the final position. For this, it must be fulfilled that:

$$\text{For all } q\text{: } \sum_{i=0}^{N} x_{i,N+1,1,q} = 1, \tag{18}$$

No vehicle can leave the final position. We then have that:

$$\text{For all } q\text{: } \sum_{j=0}^{N+1} x_{N+1,j,1,q} = 0. \tag{19}$$

Vehicles that do not travel on any road will meet all constraints when taking the following condition:

$$x_{0,N+1,1,q} = 1.$$

- Constraint 4: The vehicle must leave once and only once from each city, then:

$$\text{For each } i \in \{1,\ldots,N\}\text{: } \sum_{q=1}^{Q}\sum_{j=1}^{N+1} x_{i,j,1,q} = 1. \tag{20}$$

- Constraint 5: The vehicle must arrive once and only once to each city, then:

$$\text{For each } j \in \{1,\ldots,N\}\text{: } \sum_{q=1}^{Q}\sum_{i=0}^{N} x_{i,j,1,q} = 1. \tag{21}$$

- Constraint 6: The city i is reached before the city j does not depend on each vehicle. Therefore, for all the vehicles that either arrive at city i earlier than j, or arrive at city j earlier than i. Introducing the auxiliary variables $a_{i,j}$, we have the following constraint. For all $i, j \in \{1,\ldots,N\}$:

$$\sum_{q=1}^{Q} x_{i,j,0,q} + x_{i,j,1,q} + x_{i,j,3,q} = a_{i,j}Q. \tag{22}$$

It will then be true that for each i, j or $a_{i,j} = 1$, which means that the city i is reached earlier than the city j and therefore for each q we will have $x_{i,j,r,q} = 1$ for any value of the r in which i is reached before j, or $a_{i,j} = 0$, and we will have $x_{i,j,r,q} = 0$ for all the vehicles and for values r where i is reached before j.

- Constraint 7: If the vehicle q arrives in the city j, then the vehicle q must leave the city j. For this we impose the constraint that for $i \in \{0,\ldots,N\}$, $j \in \{1,\ldots,N\}$ and $q \in \{1,\ldots,Q\}$:

$$x_{i,j,1,q}(1 - \sum_{k=1}^{N+1} x_{j,k,1,q}) = 0. \tag{23}$$

Let us now impose the conditions that make vehicles run on a tour.

- Constraint 8: It must be fulfilled that either the vehicle pass through the city i before the j or arrive before to the city j rather than the city i. Therefore, it must be verified that, for $i \in \{0,\ldots,N\}$, $j \in \{1,\ldots,N\}$ and $q \in \{1,\ldots,Q\}$:

$$x_{i,j,0,q} + x_{i,j,1,q} + x_{i,j,3,q} = 1 - (x_{j,i,0,q} + x_{j,i,1,q} + x_{j,i,3,q}). \tag{24}$$

- Constraint 9: If city i is reached before j and city j is reached before city k, then city i must be reached before city k. This condition will prevent the vehicle from returning to a city it has already passed through and therefore prevents a cycle from forming. To introduce this constraint, we will directly calculate a penalty function worth 0 in the correct cases and 1 in those that are not. To facilitate the understanding of the penalty function, we are going to take, for i, j, k, q, the following variables:

$$\begin{aligned} a_{i,j} &= x_{i,j,0,1} + x_{i,j,1,1} + x_{i,j,3,1} \\ a_{j,k} &= x_{j,k,0,1} + x_{j,k,1,1} + x_{j,k,3,1} \\ a_{i,k} &= x_{i,k,0,1} + x_{i,k,1,1} + x_{i,k,3,1}. \end{aligned} \tag{25}$$

Remember that it is not necessary to introduce these conditions because the constraint (22) establishes the correct values of the variables $a_{i,j}$. Therefore, $a_{i,j} = 1$ means that the city i is reached before the city j and the same with j and k. Also, it is very important to remember that due to the same constraint (22), we can take any of the vehicles as a reference. In this case, we have taken the first vehicle as a reference.
In this way, fixed i, j, k, we have the 3 variables $a_{i,j}, a_{j,k}, a_{i,k}$. Remember that $a_{i,j}, a_{j,k}, a_{i,k}$ only take the values 0 or 1. Also, let us note that the cases that lead to values of the variables for which cycles can be formed and that we must discard are $(a_{i,j}, a_{j,k}, a_{i,k}) = (0, 0, 1)$ and $(a_{i,j}, a_{j,k}, a_{i,k}) = (1, 1, 0)$.
In the case $(0, 0, 1)$ we would have that the city j is reached after the i, the k after the j, and yet the city k is reached rather than i, which is absurd. The case $(1, 1, 0)$ cannot be given either, since it reaches i before j and j before k, so it cannot be that we also reach k before i. We therefore must construct a penalty function so that for $f(a_{i,j}, a_{j,k}, a_{i,k})$ it holds that $f(0, 0, 1) > 0$, $f(1, 1, 0) > 0$ y $f(a_{i,j}, a_{j,k}, a_{i,k}) = 0$ for all other cases. A function that satisfies these conditions is from the Equation (26).

$$f(a_{i,j}, a_{j,k}, a_{i,k}) := a_{i,j}a_{j,k} - a_{i,j}a_{i,k} - a_{j,k}a_{i,k} + a_{i,k}^2. \tag{26}$$

then, adding to the cost function the Equation (27)

$$\lambda \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{N} (a_{i,j}a_{j,k} - a_{i,j}a_{i,k} - a_{j,k}a_{i,k} + a_{i,k}^2), \tag{27}$$

we will have that the best solutions will be those that comply with this constraint.

- Constraint 10: The objective we seek is to minimise vehicle travel time. What we could do is see how long each vehicle takes to complete the route and try to minimise as much of the time as possible. However, this function soon becomes complex so we have decided to develop a different idea that simplifies the process and smoothes the objective function. If we impose the condition that all vehicles travel less distance than the distance travelled by vehicle number 1, we will have that minimising the maximum of the distances will be equivalent to minimising the distance travelled by the first vehicle. We then have the following condition. For each $q \in \{2, \ldots, Q\}$:

$$\sum_{i=0}^{N+1} \sum_{j=0}^{N+1} d_{i,j}x_{i,j,1,q} \leq \sum_{i=0}^{N+1} \sum_{j=0}^{N+1} d_{i,j}x_{i,j,1,1}. \tag{28}$$

We transform this inequality into equality by taking once again $D_{max} := \sum_{i=0}^{n} \max_j\{d_{i,j}\}$ and the variables $b_{h,q}$ (the variables $b_{h,q}$ are like the sub tour's one in the MTZ slack variables and they are in their binary expression) in:

$$\sum_{i=0}^{N+1} \sum_{j=0}^{N+1} d_{i,j}x_{i,j,1,q} + \sum_{h=0}^{h_{max}} 2^h b_{h,q} - \sum_{i=0}^{N+1} \sum_{j=0}^{N+1} d_{i,j}x_{i,j,1,1} = 0. \tag{29}$$

Under these conditions the function to be minimized corresponds to:

$$\sum_{i=0}^{N+1} \sum_{j=0}^{N+1} d_{i,j}x_{i,j,1,1}. \tag{30}$$

This condition has the disadvantage that we are eliminating solutions where it is another vehicle that travels the longest distance. Let us explore how to avoid this problem and get more flexibility in the model to make it easier for the Quantum Annealing to find the optimum one. We can establish an auxiliary variable D and we set that the distance travelled by each vehicle must be less than this variable, that is to say:

$$\sum_{i=0}^{N+1}\sum_{j=0}^{N+1} d_{i,j}x_{i,j,1,q} \leq D \text{ , for all } q \in \{1,\ldots,Q\}. \tag{31}$$

The variable D is an integer, so we must treat it in some way in order to include it in the model. As we explained in the introduction of the section dedicated to the formulation of the MTZ model Section 5.2, it is convenient to try to avoid the binary representation of integer variables. To do so, we can express D as a combination of the distances between edges by taking $D = \sum_{i=0}^{N+1}\sum_{j=0}^{N+1} x_{i,j}b_{i,j}$. Thus after imposing the constraint (31) we have that the function to minimize is D.

Thanks to this modelling of the new VRP we have been able to reduce the number of variables required to the order of $5N^2Q$. However, we have managed to reduce it even further to $3N^2Q$, which is detailed in Appendix A.2. However, we have preferred to present this other model due to its easy understanding.

8. Results

To test the correct VRP model developed in QUBO, which minimises the maximum distance that all the vehicles travel, we will present some comparisons of the results obtained through the simulator of the different models that have been discussed in this paper.

The code has been implemented on the Ocean library [49] from D-Wave in python. The reader can find the code at [50].

Figure 2 offers a sample of our GPS formulation's results when using the D-Wave solver in different scenarios. We highlight some important cases that help us see the good functioning of the algorithm.

Figure 3 offers a sample of our VRP formulation's results based on the GPS when using the D-Wave solver in different scenarios. We highlight some important cases that help us see the good functioning of the algorithm. It is important to note that our algorithm minimizes the maximum distance travelled by all the vehicles (this is equivalent to reducing the time spanned by all cars). It is worth mentioning that the number of the qubits needed in the case $N = 8$ and $Q = 3$ is 1778. Where N is the number of cities and Q, the vehicles. In the discussion section, we will analyze this point and its impact on the topology of the QPU architecture and in this case of the D-Wave.

Let us observe in Tables 1–5 the comparison of the number of qubits, time during which the D-Wave Quantum Annealing simulator has been executed, and the length of the path found. The sign "-" represents that the algorithm did not find a possible way during the elapsed time (in minutes). In this examples, the cities which form the TSP to solve are the vertex of the regular polygon with these number of vertex.

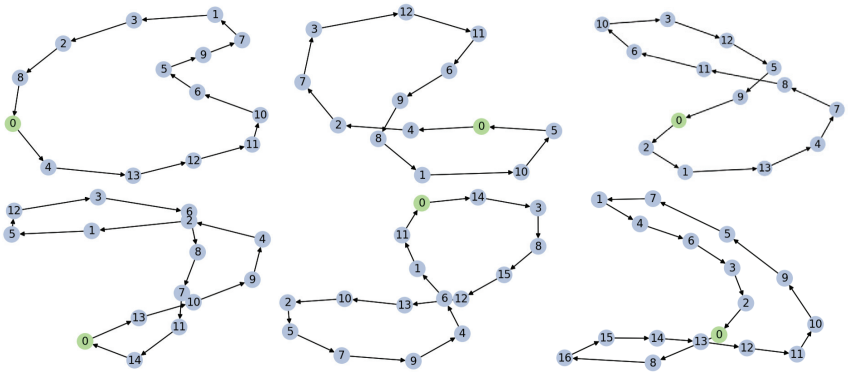


Figure 2. In these graphs, we can observe the algorithm's results in different scenarios of the GPS formulation. We can follow the correct scalability of the algorithm. We provide the code [50] to check its proper functioning and to allow others to simulate lower values or values higher than $N = 16$.

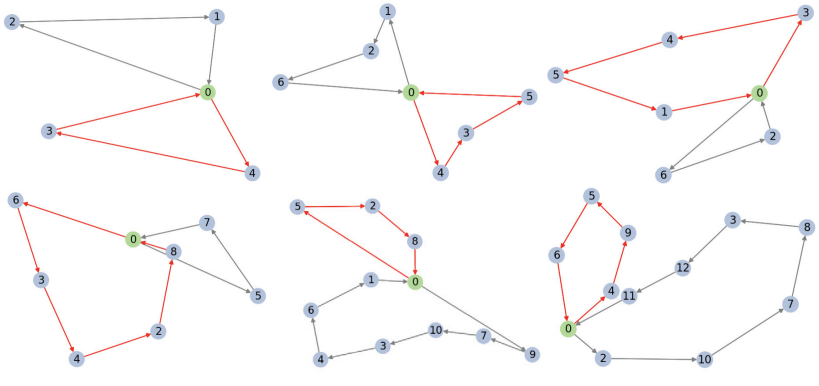


Figure 3. In these graphs, we can observe the algorithm's results in different scenarios of the VRP formulation. We can follow the correct scalability of the algorithm. We provide the code [50] to verify the proper functioning of the formulation. Vehicle number 1 is red, and the next is light-steel-blue. While the depot is the 0 node in pale-green colour, and the rest are represented in light-steel-blue. In this case, we have variables cities from 4 to 12 and using up to 2 vehicles. It is important to highlight that this VRP minimises the time travelled by the cars. The number of qubits used is 2418 to test the last case.

Table 1. A regular polygon layout has been taken where the cities occupy the positions of the nodes [50] for the elaboration of all tables. In this scenario of 4 cities, we set comparison with the 3 models, MTZ, native TSP and GPS. The comparison is based on the number of times to find the solution, the distance travelled, and the number of qubits. We can appreciate the good performance of our GPS model, and above all the savings it offers us in the number of qubits.

	GPS	**Native TSP**	**MTZ**
Number of qubits	75	100	140
Elapsed Time (min)	0.332	0.08	0.569
Path Length (m)	5.65	5.65	5.65

Table 2. A regular polygon layout has been taken where the cities occupy the positions of the nodes [50] for the elaboration of all tables. In this scenario of 6 cities, we set comparison with the 3 models, MTZ, native TSP and GPS. The comparison is based on the number of times to find the solution, the distance travelled, and the number of qubits. We can appreciate the good performance of our GPS model, and above all the savings it offers us in the number of qubits.

	GPS	**Native TSP**	**MTZ**
Number of qubits	147	294	266
Elapsed Time (min)	0.337	0.39	1.338
Path Length (m)	6.00	6.00	8.46

Table 3. A regular polygon layout has been taken where the cities occupy the positions of the nodes [50] for the elaboration of all tables. In this scenario of 8 cities, we set comparison with the 3 models, MTZ, native TSP and GPS. The comparison is based on the number of times to find the solution, the distance travelled, and the number of qubits. We can appreciate the good performance of our GPS model, and above all the savings it offers us in the number of qubits.

	GPS	**Native TSP**	**MTZ**
Number of qubits	243	648	522
Elapsed Time (min)	1.209	1.177	2.676
Path Length (m)	6.122	9.58	11.46

Table 4. A regular polygon layout has been taken where the cities occupy the positions of the nodes [50] for the elaboration of all tables. In this scenario of 10 cities, we set comparison with the 3 models, MTZ, native TSP and GPS. The comparison is based on the number of times to find the solution, the distance travelled, and the number of qubits. We can appreciate the good performance of our GPS model, and above all the savings it offers us in the number of qubits.

	GPS	**Native TSP**	**MTZ**
Number of qubits	363	1210	770
Elapsed Time (min)	3.316	3.087	4.175
Path Length (m)	12.51	10.978	-

Table 5. A regular polygon layout has been taken where the cities occupy the positions of the nodes [50] for the elaboration of all tables. In this scenario of 12 cities, we set comparison with the 3 models, MTZ, native TSP and GPS. The comparison is based on the number of times to find the solution, the distance travelled, and the number of qubits.

	GPS	**Native TSP**	**MTZ**
Number of qubits	507	2028	1066
Elapsed Time (min)	7.992	9.677	10.578
Path Length (m)	14.286	12.28	-

These results have been obtained using a simulator because we would require access to a quantum computer for a time similar to that needed to perform the simulations (in some cases more than an hour). However, it is the benefits of modelling with few qubits (such as GPS modelling) that will be much more notable when these problems are implemented on real quantum computers. Other studies that did not require many hours of the quantum computer were carried out on the D-Wave_2000Q_6. In the discussion section, we detail some interesting cases.

Discussion

Once the different models had been implemented, we achieved the following results. Through the results of the Figures 4–7 and Tables 1–5, the good performance of our formu-

lation compared to the general TSP [41] can be observed. An almost identical operation is seen with the generic TSP, except that we are improving at least the number of qubits for the same cases in our proposal. Although the time difference is not significant again, the difference between path lengths is. Let us remember that the advantage of the formulation in which we have worked is based on improving the number of qubits used. We then have that the larger the problems we are working on, the better this difference will be appreciated in the number of variables.

The MTZ model does not offer positive results. This is since *Annealing* presents many difficulties to find minimum expressions in which the representation of integers appears in their binary format. This is because although the numbers $2^k - 1$ and 2^k are close, they are not close in their binary form since they differ in k variables, so the *annealing* tends to present bad results. Apart from that adjusting, the Lagrange coefficient of this type of constraint is also a complicated task.

Native TSP and GPS modelling show better results. While it is true that general modelling gives slightly better results, it requires the use of a higher number of qubits. This may be since the function to be optimised for this model has a smaller number of local minima where the *Annealing* can get stuck or there can be a bad of the Lagrange coefficients.

The problem on which the simulations are carried out consists in finding the optimal path when the points are placed on the vertices of the regular polygons that have the same number of vertices as nodes in our problem.

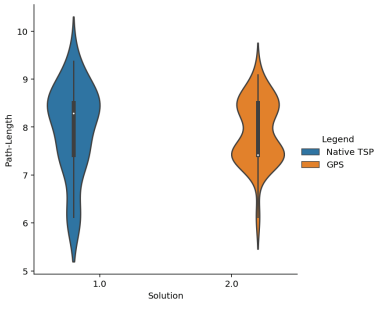


Figure 4. Path length comparison for $N = 9$. In this graph, we see how the length of the solution paths for the case of 9 Cities is very similar so that both models give good results.

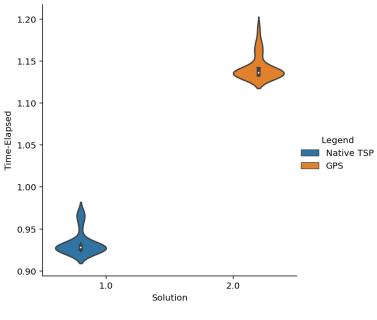


Figure 5. Time comparison for $N = 9$. This graph shows the time taken to carry out the executions in the case of 9 cities. Although it seems that there is a lot of difference, it only represents 10% of the total time, which, as we have seen in other experiences, is not significant.

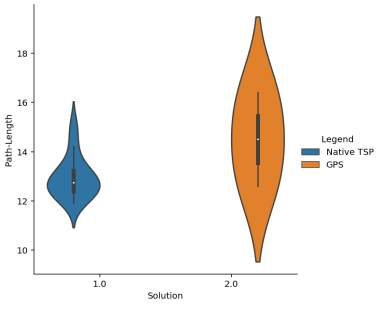


Figure 6. Path length comparison for $N = 11$. For the example of 11 cities, we can observe that the outcomes are quite similar. Although the time difference is not significant again, the difference between path lengths is. Let us remember that the advantage of the modelling we have worked is based on improving the number of qubits used. We then have that the larger the problems we are working on, the better that difference will be appreciated in the number of variables.

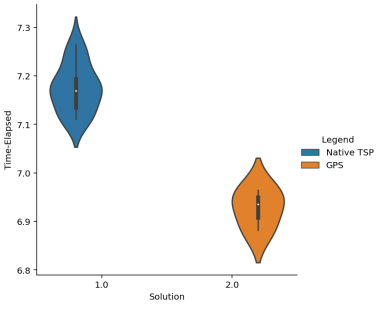


Figure 7. Time comparison for $N = 11$. For the example of 11 cities, we can observe that the outcomes are quite similar because although there is a mean difference of about 20 s between the results of both simulations, the experience with this problem and other similar ones is that this very small difference does not affect the results on the length of the solution path.

One of the behaviours and results that we believe is important to mention is the following. We realized that it is even more important to consider the number of edges that our model generates. The vertex/connections in a quantum computer are limited and define our quantum computer's typology and quality for error mitigation. Thus, a model that produces many edges (direct links) may request more from a computer than another which generates fewer. The Figure 8 offers us a comparative study between our GPS model and the native TSP. This figure shows the exponential behaviour and the number of interconnections that each model offers. Our model improves the number of qubits and gives us a great result reducing the number of connections a lot. The native TSP behaves as $0.8(N+2)^5$ while the GPS as $2(N+2)^3$.

One aspect of GPS worth commenting on here is to generalize it also to be used for the Cutting-plane method. We must change the current constraint (13) since this methodology only works with linear constraints. The way to do this is as follows. For each i, j, k:

- $x_{j,i,2} + x_{k,j,2} \leq 2x_{k,i,2} + w^1_{i,j,k}$
- $x_{j,i,2} + x_{k,j,2} \geq 2x_{k,i,2} - w^2_{i,j,k}$.

In these equations, the variables $w^p_{i,j,k}$ are auxiliaries. The purpose of these variables is to satisfy the said constrains. These two restrictions are satisfied by all cases of $(x_{j,i,2}, x_{k,j,2}, x_{k,i,2})$ except for $(0,0,1)$ (because it doesn't satisfy the second constraint) and $(1,1,0)$ (because it doesn't satisfy the first constraint).

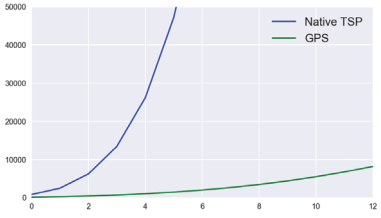


Figure 8. In this figure we can appreciate the exponential behaviour and the number of interconnections that each model offers. Our model (GPS) improves the number of qubits and gives us a great result reducing the number of connections a lot. The Native_TSP behaves as $0.8(N+2)^5$ while the GPS as $2(N+2)^3$.

9. Conclusions and Further Work

The importance of finding a good formulation in the QUBO model that minimises the number of variables to be used is crucial for the computing era we are in, as we have commented throughout this work. It is true that, although the technology of annealing-based quantum computers allows us to have much more qubits than gate-based computers, it remains a limitation and, therefore, a challenge to try to solve. Hence highlighting the importance of our research.

With this work, we offer a new formulation for the TSP called GPS and apply it to find an optimal formulation for the VRP that minimises the time the vehicles make their journey. We have also seen that the results of the D-Wave simulator solver are consistent with the expected solution. However, we consider it unnecessary to test it in gate-based quantum computers, given their limitations today in the number of qubits. Still, we emphasise that our current formulation is valid for such computers. The improvement in our models represents a fairly significant order of magnitude because we went from N^3 variables to $3N^2$. The Figures 9 and 10 summarises the major contribution of this article.

Our GPS formulation and the VRP proposal can help in optimisation problems when we want to reduce the number of variables and therefore reduce the number of qubits quite a bit. In addition, it is interesting in situations, such as the one raised in the future line of the article [14], by modelling some biological activities on selected sets of organic compounds as can be seen in [51], or resource optimization problems such as gasoline and aircraft travel. Another interesting application could be to compare GPS with the approach offered by this reference [52] using deep reinforcement learning to address combinatorial optimisation problems with feasibility constraints. This leads us to project on how to make this comparison in quantum computing using the proposal made in this reference [53].

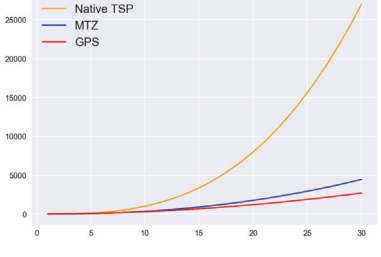


Figure 9. Comparison of the different models based on the number of qubits. This graph shows the behaviour and evolution of the numbers of qubits for each model. We see the best performance of our GPS model compared to the other models.

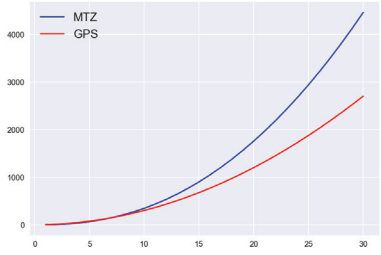


Figure 10. Benchmark between MTZ and GPS model based on the number of qubits. We can appreciate that for 30 cities, GPS model needs 2700 qubits while the MTZ 4458.

The results obtained from our VRP formulation and all the experiments carried out maintain the number of variables QN^2 and allow us to offer the community new formulations that minimise the time it takes for vehicles to travel.

Future work will apply the ideas developed in the QUBO model of these problems to similar ones. In particular, we will look for other variants of the TSP to use the modelling of this that we have carried out.

Author Contributions: Conceptualization, P.A.-A.; Methodology, P.A.-A. and S.G.-B. and G.A.-L.; Software, S.G.-B. and P.A.-A. and G.A.-L.; Validation, P.A.-A. and S.G.-B. and G.A.-L.; Formal Analysis, S.G.-B. and P.A.-A. and G.A.-L.; Investigation, S.G.-B. and P.A.-A.; Resources, P.A.-A.; Data Curation, P.A.-A.; Writing—Original Draft Preparation, S.G.-B. and P.A.-A.; Writing—Review and Editing, P.A.-A. and G.A.-L. and S.G.-B.; Visualization, P.A.-A. and G.A.-L. and S.G.-B.; Supervision, P.A.-A.; Project Administration, P.A.-A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: This article does not contain any studies with human or animal subjects.

Informed Consent Statement: Informed consent was obtained from all individual participants included in the study.

Data Availability Statement: Data sharing not applicable. No new dataset were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Appendix A.1. TSP Formulation N^2

There is a TSP model that requires N^2 variables, where these are the following:

$$x_{i,t} \text{ such as } i \in \{0, \ldots, N+1\} \text{ and } t \in \{0, \ldots, N+1\}. \tag{A1}$$

Under this formulation $x_{i,t} = 1$ denotes that the city i is reached at position t. The distance calculation function with this formulation is as follows

$$\sum_{i=0}^{N+1} \sum_{j=0}^{N+1} \sum_{t=0}^{N} d_{i,j} x_{i,t} x_{j,t+1}, \tag{A2}$$

where $d_{i,j}$ represents the distance between the node i and the node j. This expression has the problem that the distance formulation has terms of degree two and when trying to generalize this idea to other types of problems such as the VRP it will become a 4 degree constraint making use of a large number of auxiliary variables to convert it to QUBO type format.

Appendix A.2. Improved Model $3N^2Q$

In the previous modelling, we can improve the number of variables used from $5N^2Q$ to $3N^2Q$ since certain variables are redundant. Let us see how we can do this. Let us take the set of variables

$$x_{i,j,r,q} \text{ with } i < j \in \{0,\dots,N+1\}, r \in \{0,1,2\} \text{ and } q \in \{1,\dots,Q\}$$

In all of the modelling, the variables $x_{i,j,r}$ such that $i = j$ are not considered. Let us analyse the interpretation of each variable. For each edge (i,j), different cases depend on whether a vehicle passes through both cities, which city is visited before the other and whether the edge is travelled or not.

- $x_{i,j,0,q} = 1$ means that the city i is reached earlier than the j and the edge (i,j) is not travelled.
- $x_{i,j,1,q} = 1$ means that the vehicle q travels the cities i and j, it reaches the city i before the j and it travels the edge (i,j).
- $x_{i,j,2,q} = 1$ means that the city j is reached earlier than the i and the edge (j,i) is not travelled.

This new simplification keeps constraints (16), (18), (20), (21), (23) and (28) defined in the same way as the first proposal of the VRP formulation, so we will only focus on the changes of the remaining constraints:

- Constraint 1: For each i, j, q, one and only one of the possibilities must be met for r, so:

$$\text{For all } i,j,q\text{: } \sum_{r=0}^{2} x_{i,j,r,q} = 1, \tag{A3}$$

- Constraint 6: That the city i is reached before the city j does not depend on each vehicle. Therefore, for all the vehicles that either arrive at city i earlier than j, or arrive at city j earlier than i. Introducing the auxiliary variables $a_{i,j}$, we have the following constraint. For all $\quad i,j \in \{1,\dots,N\}$:

$$\sum_{q=1}^{Q} x_{i,j,0,q} + x_{i,j,1,q} = a_{i,j}Q. \tag{A4}$$

- Constraint 8: It must be fulfilled that either the vehicle pass through the city i before the j or arrive before to the city j than the i. Therefore, it must be verified that, for $\quad i \in \{0,\dots,N\}, \quad j \in \{1,\dots,N\} \quad$ and $\quad q \in \{1,\dots,Q\}$:

$$x_{i,j,0,q} + x_{i,j,1,q} = 1 - (x_{j,i,0,q} + x_{j,i,1,q}). \tag{A5}$$

- Constraint 9: If the city i is reached before j and the city j is reached before the city k, then the city i must be reached before the city k. This condition will prevent the vehicle from returning to a city it has already passed through and therefore prevents a cycle from forming.

$$\lambda \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}(a_{i,j}a_{j,k} - a_{i,j}a_{i,k} - a_{j,k}a_{i,k} + a_{i,k}^2), \tag{A6}$$

Appendix B. Restriction Penalty

Let us analyze the system that must be solved to build the penalty function from the Equation (13). Our penalty function $P(a_{i,j}, a_{j,k}, a_{i,k})$ must satisfy that $P(0,0,1) = 1$, $P(1,1,0) = 1$ and $P(a_{i,j}, a_{j,k}, a_{i,k}) = 0$ for the rest of the cases. Let us call the variables

$a_{i,j} = x$, $a_{j,k} = y$, $a_{i,k} = z$ to simplify the notation. Then, we arrive at the quadratic function P, as is demonstrated in the following:

$$P(x, y, z) = c_1x^2 + c_2xy + c_3xz + c_4y^2 + c_5yz + c_6z^2. \tag{A7}$$

Imposing the previous restrictions, we have the following system of equations.

- $P(0, 0, 1) = 1$ So that $c_6 = 1$.
- $P(0, 1, 0) = 0$ So that $c_4 = 0$.
- $P(0, 1, 1) = 0$ So that $c_5 + c_6 = 1 \Rightarrow c_5 = -1$.
- $P(1, 0, 0) = 0$ So that $c_1 = 0$.
- $P(1, 0, 1) = 0$ So that $c_1 + c_3 + c_6 = 0 \Rightarrow c_3 = -1$
- $P(1, 1, 0) = 1$ So that $c_2 = 1$.

So far, we have a system of six equations with six certain compatible unknowns. First, however, an additional restriction must be verified. Let us verify if it is met.

- $P(1, 1, 1) = 0$. $\sum_{i=1}^{6} c_i = 1 - 1 - 1 + 1 = 0$. So that indeed all the requirements are met.

We then have that the following function which is a penalty function for the constraint (13).

$$P(a_{i,j}, a_{j,k}, a_{i,k}) = a_{i,j}a_{j,k} - a_{i,j}a_{i,k} - a_{j,k}a_{i,k} + a_{i,k}^2 \tag{A8}$$

References

1. Gavish, B.; Graves, S.C. *The Travelling Salesman Problem and Related Problems*; Operations Research Center, Massachusetts Institute of Technology: Boston, MA, USA, 1978.
2. Hochba, D.S. Approximation algorithms for NP-hard problems. *ACM Sigact News* **1997**, *28*, 40–52. [CrossRef]
3. Lewis, M.; Glover, F. Quadratic unconstrained binary optimization problem preprocessing: Theory and empirical analysis. *Networks* **2017**, *70*, 79–97. [CrossRef]
4. Cipra, B. An introduction to the Ising model. *Am. Math. Mon.* **1987**, *94*, 937–959. [CrossRef]
5. McGeoch, C.; Cong, W. Experimental evaluation of an adiabiatic quantum system for combinatorial optimization. In Proceedings of the ACM International Conference on Computing Frontiers, Ischia, Italy, 14–16 May 2013; Volume 13.
6. McGeoch, C. Adiabatic quantum computation and quantum annealing: Theory and practice. *Synth. Lect. Quantum Comput.* **2014**, *5*, 1–93. [CrossRef]
7. Brooke, J.; Rosenbaum, D.; Aeppli, G. Quantum annealing of a disordered magnet. *Science* **1999**, *284*, 779–781. [CrossRef] [PubMed]
8. Born, M.; Fock, V. Beweis des Adiabatensatzes. *Z. Phys.* **1928**, *51*, 165–180. [CrossRef]
9. Farhi, E.; Gutmann, S.; Sipser, M. Quantum computation by adiabatic evolution. *arXiv* **2000**, arXiv:quant-ph/0001106.
10. Bian, Z.; Macready, F.; William, G.; Geordie, R. *The Ising Model: Teaching an old Problem New Tricks*; D-Wave Systems Inc.: Burnaby, BC, Canada, 2010; Volume 2.
11. Toth, P.; Vigo, D. *The Vehicle Routing Problem*; SIAM: Philadelphia, PA, USA, 2002.
12. Toth, P.; Vigo, D. Models, relaxations and exact approaches for the capacitated vehicle routing problem. *Discret. Appl. Math.* **2002**, *123*, 487–512. [CrossRef]
13. Ralphs, T.; Kopman, L.;Pulleyblank, W.; Trotter, L. On the capacitated vehicle routing problem. *Math. Program.* **2003**, *94*, 343–359. [CrossRef]
14. Atchade-Adelomou, P.; Alonso-Linaje, G.; Albo-Canals, J.; Casado-Fauli, D. qRobot: A Quantum Computing Approach in Mobile Robot Order Picking and Batching Problem Solver Optimization. *Algorithms* **2021**, *14*, 194 [CrossRef]
15. Miller, C.; Tucker, A.; Zemlin, R. Integer programming formulation of traveling salesman problems. *Oper. Res. Lett.* **1960**, *10*, 27–36. [CrossRef]
16. Boruvka, O. On a minimal problem. *Práce Moravské Pridovedecké Spolecnosti* **1926**, *3*, 37–58.
17. Boruvka, O. Prıspevek k rešenı otázky ekonomické stavby elektrovodnıch sıtı (contribution to the solution of a problem of economical construction of electrical networks). *Elektronickỳ Obzor* **1926**, *15*, 153–154.
18. Kruskal, J. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc. Am. Math. Soc.* **1956**, *7*, 48–50. [CrossRef]
19. Bellmore, M.; Nemhauser, G. The Traveling Salesman Problem: A Survey. *Oper. Res.* **1968**, *16*, 538–558. [CrossRef]
20. Lenstra, J.; Rinnooy Kan, H. Some Simple Applications of the Travelling Salesman Problem. *J. Oper. Res. Soc.* **1975**, *26*, 717–733. [CrossRef]
21. Adelomou, P.; Ribé, E.; Cardona, X. Formulation of the social workers' problem in quadratic unconstrained binary optimization form and solve it on a quantum computer. *J. Comput. Commun.* **2020**, *11*, 44–68. [CrossRef]
22. Lee, J.Y.; Shi, S.-Y.; Park, T.H.; Zhang, B.-T. Solving traveling salesman problems with DNA molecules encoding numerical values. *Biosystems* **2004**, *78*, 39–47. [CrossRef]

23. Ball, P. *DNA Computer Helps Travelling Salesman*; Springer Science and Business Media LLC: New York, NY, USA, 2000.
24. Pataki, G. Teaching integer programming formulations using the traveling salesman problem. *SIAM Rev.* **2003**, *45*, 116–123. [CrossRef]
25. Martoňák, R.; Santoro, G.; Tosatti, E. Quantum annealing of the traveling-salesman problem. *Phys. Rev. E* **2004**, *70*, 057701. [CrossRef]
26. Warren, R. Adapting the traveling salesman problem to an adiabatic quantum computer. *Quantum Inf. Process.* **2013**, *4*, 1781–1785. [CrossRef]
27. Warren, R. Small traveling salesman problems. *J. Adv. Appl. Math.* **2017**, *2*. [CrossRef]
28. Greco, F. *Traveling Salesman Problem*; BoD–Books on Demand: Norderstedt, Germany, 2008; ISBN 978-953-51-5750-2.
29. Reinelt, G. TSPLIB—A traveling salesman problem library. *ORSA J. Comput.* **1991**, *4*, 376–384. [CrossRef]
30. Crosson, E.; Harrow, A. Simulated quantum annealing can be exponentially faster than classical simulated annealing. In Proceedings of the 2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS), New Brunswick, NJ, USA, 9–11 October 2016; pp. 714–723.
31. Leung, K.; Jin, H.-D.; Xu, Z.-B. An expanding self-organizing neural network for the traveling salesman problem. *Neurocomputing* **2004**, *62*, 267–292. [CrossRef]
32. Cochrane, E.; Beasley, J. The co-adaptive neural network approach to the Euclidean travelling salesman problem. *Neural Netw.* **2003**, *10*, 1499–1525. [CrossRef]
33. Feld, S.; Gabor, T.; Seidel, C.; Neukart, F.; Galter, I.; Mauerer, W.; Linnhoff-Popien, C. A hybrid solution method for the capacitated vehicle routing problem using a quantum annealer. *Front. ICT* **2019**, *6*, 13. [CrossRef]
34. Irie, H.; Wongpaisarnsin, G.; Terabe, M.; Miki, A.; Taguchi, S. Quantum annealing of vehicle routing problem with time, state and capacity. In *International Workshop on Quantum Technology and Optimization Problems*; Springer: Cham, Switzerland, 2019; pp. 145–156.
35. Focacci, F.; Lodi, A.; Milano, M. A hybrid exact algorithm for the TSPTW. *INFORMS J. Comput.* **2002**, *4*, 403–417. [CrossRef]
36. Edelkamp, S.; Gath, M.; Cazenave, T.; Teytaud, F. Algorithm and knowledge engineering for the TSPTW problem. In Proceedings of the 2013 IEEE Symposium on Computational Intelligence in Scheduling (CISched), Singapore, 16–19 April 2013; pp. 44–51.
37. Atchade-Adelomou, P.; Golobardes-Ribé, E.; Vilasís-Cardona, X. Using the Variational-Quantum-Eigensolver (VQE) to Create an Intelligent Social Workers Schedule Problem Solver. In *Hybrid Artificial Intelligent Systems, Proceedings of the 5th International Conference, HAIS 2020, Gijón, Spain, 11–13 November 2020*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2020; pp. 245–260.
38. Atchade-Adelomou, P.; Golobardes-Ribe, E.; Vilasis-Cardona, X. Using the Parameterized Quantum Circuit combined with Variational-Quantum-Eigensolver (VQE) to create an Intelligent social workers' schedule problem solver. *arXiv* **2020**, arXiv:2010.05863.
39. Atchade-Adelomou, P.; Casado-Fauli, D.; Golobardes-Ribe, E.; Vilasis-Cardona, X. quantum Case-Based Reasoning (qCBR). *arXiv* **2021**, arXiv:2104.00409.
40. Applegate, D.; Cook, W. A computational study of the job-shop scheduling problem. *ORSA J. Comput.* **1991**, *3*, 149–156. [CrossRef]
41. Papalitsas, C.; Andronikos, T.; Giannakis, K.; Theocharopoulou, G.; Fanarioti, S. A QUBO model for the traveling salesman problem with time windows. *Algorithms* **2019**, *12*, 224. [CrossRef]
42. Boixo, S.; Rønnow, F.; Isakov, S.; Wang, Z.; Wecker, D.; Lidar, D.; Martinis, J.; Troyer, M. Evidence for quantum annealing with more than one hundred qubits. *Nat. Phys.* **2014**, *3*, 218–224. [CrossRef]
43. Preskill, J. Quantum Computing in the NISQ era and beyond. *Quantum* **2018**, *2*, 79. [CrossRef]
44. D-Wave Systems Inc. *Technical Description of the D-Wave Quantum Processing Unit*; D-Wave Systems Inc.: Burnaby, BC, Canada, 2020.
45. De Vogelaere, R. *Methods of Integration Which Preserve the Contact Transformation Property of the Hamilton Equations*; Technical Report; Department of Mathematics, University of Notre Dame: Notre Dame, IN, USA, 1956.
46. Marston, C.; Balint-Kurti, G. The Fourier grid Hamiltonian method for bound state eigenvalues and eigenfunctions. *J. Chem. Phys.* **1989**, *6*, 3571–3576. [CrossRef]
47. Shin, S.; Graeme, S.; Smolin, J.; Vazirani, U. How "quantum" is the D-Wave machine? *arXiv* **2014**, arXiv:1401.7087.
48. Lucas, A. Ising formulations of many NP problems. *Front. Phys.* **2014**, *2*, 5. [CrossRef]
49. Teplukhin, A.; Kendrick, B.; Tretiak, S.; Dub, P. Electronic structure with direct diagonalization on a D-wave quantum annealer. *Sci. Rep.* **2020**, *10*, 20753. [CrossRef]
50. Alonso-Linaje, G.; Atchade-Adelomou, P.; Gonzalez-Bermejo, S. Improvement in the Formulation of the TSP for Its Generalizations Type QUBO. 2021. Available online: https://github.com/pifparfait/GPS (accessed on 1 December 2021).
51. Jäntschi, L.; Katona, G.; Diudea, M. Modeling Molecular Properties by Cluj Indices. *Match* **2000**, *41*, 151–188.
52. Zhang, R.; Prokhorchuk, A.; Dauwels, J. Deep Reinforcement Learning for Traveling Salesman Problem with Time Windows and Rejections. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020. [CrossRef]
53. Atchade-Adelomou, P.; Alonso-Linaje, G. Quantum Enhanced Filter: QFilter. *arXiv* **2021**, arXiv:2104.03418.

mathematics

MDPI

Article

Quantum Weighted Fractional Fourier Transform

Tieyu Zhao [1,*], Tianyu Yang [2] and Yingying Chi [1]

[1] Information Science Teaching and Research Section, Northeastern University at Qinhuangdao, Qinhuangdao 066004, China; chiyingying@neuq.edu.cn
[2] Sports Department, Northeastern University at Qinhuangdao, Qinhuangdao 066004, China; yangtianyu@neuq.edu.cn
* Correspondence: zhaotieyu@neuq.edu.cn

Abstract: Quantum Fourier transform (QFT) is an important part of many quantum algorithms. However, there are few reports on quantum fractional Fourier transform (QFRFT). The main reason is that the definitions of fractional Fourier transform (FRFT) are diverse, while some definitions do not include unitarity, which leads to some studies pointing out that there is no QFRFT. In this paper, we first present a reformulation of the weighted fractional Fourier transform (WFRFT) and prove its unitarity, thereby proposing a quantum weighted fractional Fourier transform (QWFRFT). The proposal of QWFRFT provides the possibility for many quantum implementations of signal processing.

Keywords: quantum weighted fractional Fourier transform; quantum Fourier transform; quantum algorithm; quantum computing

MSC: 81-08

Citation: Zhao, T.; Yang, T.; Chi, Y. Quantum Weighted Fractional Fourier Transform. *Mathematics* **2022**, *10*, 1896. https://doi.org/10.3390/math10111896

Academic Editors: Fernando L. Pelayo and Mauro Mezzini

Received: 3 May 2022
Accepted: 31 May 2022
Published: 1 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

Feynman was the first to present the idea of quantum computing, that is, to directly use the state of microscopic particles to represent quantum information, which is considered to be the early prototype of the concept of quantum computing [1]. Subsequently, Deutsch formalized the concept of quantum computing, proposed the idea of a quantum Turing machine, and designed the first quantum parallel algorithm, which exhibited excellent performance beyond classical computing [2]. The proposal of Shor's algorithm caused researchers to realize that quantum computing had a natural parallel processing capability, which could introduce many disruptive technological innovations. Shor's algorithm states that a large number can be decomposed into the product of two prime factors in polynomial time. This greatly challenged the RSA (Rivest–Shamir–Adleman) encryption system, thus indicating that the RSA encryption system had been cracked in theory [3,4]. Grover's search algorithm convinced researchers of the power of quantum computing. Compared with the traditional search method, this algorithm can achieve the acceleration effect of square level [5]. Therefore, many improved Grover search algorithms have been proposed [6–10]. Meanwhile, quantum-inspired algorithms have also been proposed that can be simulated by classical computing [11–16]. Moreover, the quantum algorithm has been applied to solve linear systems of equations, which introduced new ideas for solving linear equations. This algorithm is also called the *HHL* algorithm [17]. The *HHL* algorithm has been widely used, and its improved algorithms have been continuously proposed [18–20]. Recently, quantum algorithms have been applied to solve differential equations [21–24]. A series of quantum computing technologies, such as quantum Fourier transform [25], quantum phase estimation [26], and the *HHL* algorithm, are called quantum basic linear algebra assembly [27]. At present, quantum computing has been widely used in cryptography, quantum simulation, machine learning, and other fields and shows a strong ability and great potential.

The Fourier transform plays an important role in the design of quantum algorithms, but little is known about the quantum algorithms of the fractional Fourier transform (*FRFT*). The initial definition of the *FRFT* was proposed in [28]. Its application provides a convenient technique for solving certain classes of ordinary and partial differential equations, which arise in quantum mechanics from classical quadratic Hamiltonians. The theoretical research of the *FRFT* has developed rapidly, and various definitions have been proposed, such as eigenvalue *FRFT* [29], weighted *FRFT* [30], and sampling *FRFT* [31]. These definitions are widely used in various fields of signal processing. So far, little is known about the reports and studies on the quantum fractional Fourier transform (*QFRFT*). The main reason is that the design of quantum algorithms should satisfy unitarity, and some FRFTs do not include unitarity. Thus, a quantum pseudo-fractional Fourier transform (*QPFRFT*) was proposed [32], and the authors showed that there was no *QFRFT*. However, we present a reformulation of the weighted fractional Fourier transform (*WFRFT*) and prove its unitarity, whereupon a quantum weighted fractional Fourier transform (*QWFRFT*) is proposed.

The remainder of this paper is organized as follows. The preliminary knowledge is described in Section 2. The unitarity of the WFRFT is proved in Section 3. Section 4 presents the *QWFRFT*. Finally, the conclusions are presented in Section 5.

2. Preparation

For a unitary matrix U, assuming that it has an eigenvector $|u\rangle$ and the corresponding eigenvalue $e^{2\pi i\varphi}$, $U|u\rangle = e^{2\pi i\varphi}|u\rangle$ is satisfied. Therefore, we can calculate φ through the phase estimation algorithm. The circuit of phase estimation is shown in Figure 1. It is not difficult to find that the quantum Fourier transform (*QFT*) is the key to phase estimation, and phase estimation is the key of many quantum algorithms.

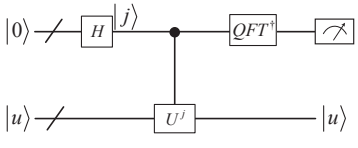


Figure 1. A circuit for phase estimation.

The importance of the *QFT* goes without saying. However, little is known about the report of the *QFRFT*. In 2012, Parasa et al. proposed a *QPFRFT* using multiple-valued logic [32]. The reason why researchers call it "pseudo" is that the FRFT used did not include unitarity. The FRFT was proposed by Bailey et al. [33], and its definition is as follows:

$$F^{\alpha}[k] = \sum_{j=0}^{N-1} f[j] \cdot \exp\left(2\pi i \cdot \frac{kj}{N} \cdot \alpha\right). \tag{1}$$

Parasa et al. pointed out: "It must be noted that unlike the discrete Fourier transform, the *FRFT* is not a unitary operation. More formally, this means that there exists no unitary operator which can implement the following quantum computational operation".

$$\sum_{j=0}^{N-1} f(j)|j\rangle \overset{NOTPOSSLBLE}{\rightarrow} \sum_{k=0}^{N-1} F^{\alpha}(k)|k\rangle. \tag{2}$$

Therefore, Parasa et al. explicitly state that it is not possible to define the *QFRFT*. However, the definitions of the *FRFT* are diverse, and the definition of one class of *WFRFT* includes unitarity. Hence, Parasa et al.'s statement that there is no *QFRFT* is not rigorous.

In 1995, Shih proposed the definition of a *WFRFT* [30]. The alpha-order *FRFT* of the function $f(t)$ can be expressed as

$$F^{\alpha}[f(t)] = \sum_{l=0}^{3} A_l(\alpha) f_l(t). \tag{3}$$

Here, $f_0(t) = f(t)$, $f_1(t) = F[f_0(t)]$, $f_2(t) = F[f_1(t)]$, and $f_3(t) = F[f_2(t)]$ (F denotes Fourier transform). The weighting coefficient $A_l(\alpha)$ is expressed as

$$A_l(\alpha) = \cos\left(\frac{(\alpha-l)\pi}{4}\right)\cos\left(\frac{2(\alpha-l)\pi}{4}\right)\exp\left(\frac{3(\alpha-l)i\pi}{4}\right), \tag{4}$$

where $l = 0, 1, 2, 3$.

3. Unitarity of Weighted Fractional Fourier Transform

A complex matrix U satisfies

$$UU^H = U^H U = I, \tag{5}$$

where H denotes the conjugate transpose, and I is the identity matrix. Then, matrix U is called a unitary matrix.

The discrete form of the *WFRFT* (Equation (3)) can be expressed as

$$DWFRFT = A_0(\alpha)\cdot I + A_1(\alpha)\cdot DFT + A_2(\alpha)\cdot DFT^2 + A_3(\alpha)\cdot DFT^3, \tag{6}$$

where $A_l(\alpha)$ is Equation (4), and DFT is the discrete Fourier transform. It is not easy to prove the unitarity of Equation (6). Therefore, we present the reformulation of the *WFRFT* and prove its unitarity. First, Equation (4) can be written as

$$\begin{aligned} A_l(\alpha) &= \cos\left(\tfrac{(\alpha-l)\pi}{4}\right)\cos\left(\tfrac{2(\alpha-l)\pi}{4}\right)\exp\left(\tfrac{3(\alpha-l)i\pi}{4}\right) \\ &= \tfrac{1}{2}\times\left[\exp\left(\tfrac{(\alpha-l)\pi i}{4}\right)+\exp\left(\tfrac{-(\alpha-l)\pi i}{4}\right)\right] \\ &\times\tfrac{1}{2}\times\left[\exp\left(\tfrac{2(\alpha-l)\pi i}{4}\right)+\exp\left(\tfrac{-2(\alpha-l)\pi i}{4}\right)\right]\times\exp\left(\tfrac{3(\alpha-l)i\pi}{4}\right) \\ &= \tfrac{1}{4}\left(1+\exp\left(\tfrac{2(\alpha-l)\pi i}{4}\right)+\exp\left(\tfrac{4(\alpha-l)\pi i}{4}\right)+\exp\left(\tfrac{6(\alpha-l)\pi i}{4}\right)\right) \\ &= \tfrac{1}{4}\sum_{k=0}^{3}\exp\left(\tfrac{2\pi i}{4}(\alpha-l)k\right) \\ &= \tfrac{1}{4}\sum_{k=0}^{3}\exp\left(\tfrac{2\pi i\alpha k}{4}\right)\exp\left(\tfrac{-2\pi i l k}{4}\right). \end{aligned} \tag{7}$$

Let $B_k^{\alpha} = \exp\left(\frac{2\pi i k\alpha}{4}\right)$; $k = 0, 1, 2, 3$; then, Equation (7) can be expressed as

$$\begin{pmatrix} A_0^{\alpha} \\ A_1^{\alpha} \\ A_2^{\alpha} \\ A_3^{\alpha} \end{pmatrix} = \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{pmatrix}\begin{pmatrix} B_0^{\alpha} \\ B_1^{\alpha} \\ B_2^{\alpha} \\ B_3^{\alpha} \end{pmatrix}. \tag{8}$$

We write Equation (6) as Equation (9).

$$DWFRFT = \left(I, DFT, DFT^2, DFT^3\right)\begin{pmatrix} A_0(\alpha) \\ A_1(\alpha) \\ A_2(\alpha) \\ A_3(\alpha) \end{pmatrix}. \tag{9}$$

Equation (8) is substituted into Equation (9), and we obtain

$$DWFRFT = \frac{1}{4}\left(I, DFT, DFT^2, DFT^3\right)\begin{pmatrix}1 & 1 & 1 & 1\\ 1 & -i & -1 & i\\ 1 & -1 & 1 & -1\\ 1 & i & -1 & -i\end{pmatrix}\begin{pmatrix}B_0^\alpha\\ B_1^\alpha\\ B_2^\alpha\\ B_3^\alpha\end{pmatrix}. \tag{10}$$

We let

$$\begin{cases} Y_0 = I + DFT + DFT^2 + DFT^3\\ Y_1 = I - i{\cdot}DFT - DFT^2 + i{\cdot}DFT^3\\ Y_2 = I - DFT + DFT^2 - DFT^3\\ Y_3 = I + i{\cdot}DFT - DFT^2 - i{\cdot}DFT^3 \end{cases} \tag{11}$$

Definition 1. *A reformulation of the DWFRFT.*

$$\begin{aligned} DWFRFT &= \tfrac{1}{4}(Y_0, Y_1, Y_2, Y_3)\begin{pmatrix}B_0^\alpha\\ B_1^\alpha\\ B_2^\alpha\\ B_3^\alpha\end{pmatrix}\\ &= \tfrac{1}{4}\left(Y_0B_0^\alpha + Y_1B_1^\alpha + Y_2B_2^\alpha + Y_3B_3^\alpha\right)\\ &= \tfrac{1}{4}\sum_{k=0}^{3} Y_kB_k^\alpha. \end{aligned} \tag{12}$$

where $B_k^\alpha = \exp\left(\frac{2\pi ik\alpha}{4}\right); k = 0, 1, 2, 3.$

Proposition 1. *Y_k are real symmetric matrices.*

Proof of Proposition 1. In Equation (11), I is the identity matrix, and DFT can be expressed as

$$DFT = \frac{1}{\sqrt{N}}\cdot\begin{pmatrix} u^{0\times0} & u^{0\times1} & \dots & u^{0\times(n-1)}\\ u^{1\times0} & u^{1\times1} & \dots & u^{1\times(n-1)}\\ \vdots & \vdots & \ddots & \vdots\\ u^{(n-1)\times0} & u^{(n-1)\times1} & \dots & u^{(n-1)\times(n-1)}\end{pmatrix}, \tag{13}$$

where $u = \exp(-2\pi i/N)$. Here, DFT is a symmetric matrix, so that DFT^2, DFT^3, and DFT^4 are also symmetric matrices. We know that the result of adding symmetric matrices is still a symmetric matrix. Therefore, Y_k are symmetric matrices (Equation (11)).

Next, we prove that Y_k are real matrices. The integer powers of the Fourier transform are shown in Figure 2. Here, DFT^2 and DFT^4 are real matrices; the matrix of DFT^2 is shown in Equation (14), and DFT^4 is the identity matrix $DFT^4 = DFT^0 = I$.

$$DFT^2 = \begin{pmatrix}1 & 0 & \dots & 0\\ 0 & 0 & \dots & 1\\ \vdots & \vdots & \iddots & \vdots\\ 0 & 1 & \dots & 0\end{pmatrix}. \tag{14}$$

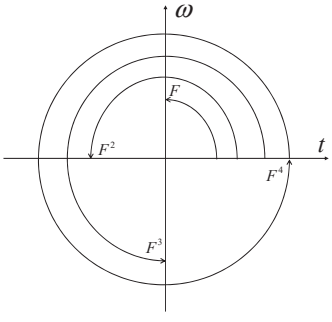


Figure 2. Time–frequency representation of Fourier transform.

Obviously, I and DFT^2 are real matrices. In Equation (13), each element of the DFT can be expressed as

$$u_{lk} = \exp(-2\pi ilk/N), \tag{15}$$

where $l = 0, 1, \ldots, n-1; k = 0, 1, \ldots, n-1$. Therefore, DFT^3 is an inverse Fourier transform, and each element of its matrix can be expressed as

$$w_{lk} = \exp(2\pi ilk/N), \tag{16}$$

where $l = 0, 1, \ldots, n-1$; $k = 0, 1, \ldots, n-1$. Thus, the result of $DFT + DFT^3$ is a real number,

$$\begin{aligned} w_{lk} + u_{lk} &= \exp(-2\pi ilk/N) + \exp(2\pi ilk/N) \\ &= \cos(2\pi lk/N) - i\sin(2\pi lk/N) + \cos(2\pi lk/N) + i\sin(2\pi lk/N) \\ &= 2\cos(2\pi lk/N). \end{aligned} \tag{17}$$

The result for $-iDFT + iDFT^3$ is

$$\begin{aligned} -iw_{lk} + iu_{lk} &= -i\exp(-2\pi ilk/N) + i\exp(2\pi ilk/N) \\ &= -i\cos(2\pi lk/N) - \sin(2\pi lk/N) + i\cos(2\pi lk/N) - \sin(2\pi lk/N) \\ &= -2\sin(2\pi lk/N). \end{aligned} \tag{18}$$

The result for $-DFT - DFT^3$ is

$$\begin{aligned} -w_{lk} - u_{lk} &= -\exp(-2\pi ilk/N) - \exp(2\pi ilk/N) \\ &= -\cos(2\pi lk/N) + i\sin(2\pi lk/N) - \cos(2\pi lk/N) - i\sin(2\pi lk/N) \\ &= -2\cos(2\pi lk/N). \end{aligned} \tag{19}$$

The result for $iDFT - iDFT^3$ is

$$\begin{aligned} iw_{lk} - iu_{lk} &= i\exp(-2\pi ilk/N) - i\exp(2\pi ilk/N) \\ &= i\cos(2\pi lk/N) + \sin(2\pi lk/N) - i\cos(2\pi lk/N) + \sin(2\pi lk/N) \\ &= 2\sin(2\pi lk/N). \end{aligned} \tag{20}$$

Therefore, for Equation (11), Y_k are real symmetric matrices. □

Proposition 2. *The weighted fractional Fourier transform is unitary.*

Proof of Proposition 2. By the proof of Proposition 1, we know that Y_k are real symmetric matrices; that is, $(Y_k)^H = Y_k$. Therefore, the conjugate transpose of the $DWFRFT$ is

$$\begin{aligned} (DWFRFT)^H &= \tfrac{1}{4}\left(Y_0B_0^{\alpha} + Y_1B_1^{\alpha} + Y_2B_2^{\alpha} + Y_3B_3^{\alpha}\right)^H \\ &= \tfrac{1}{4}\left(Y_0B_0^{-\alpha} + Y_1B_1^{-\alpha} + Y_2B_2^{-\alpha} + Y_3B_3^{-\alpha}\right). \end{aligned} \tag{21}$$

Thus, we obtain

$$\begin{aligned} DWFRFT\cdot(DWFRFT)^H &= \tfrac{1}{16}\left(Y_0B_0^{\alpha}+Y_1B_1^{\alpha}+Y_2B_2^{\alpha}+Y_3B_3^{\alpha}\right)\left(Y_0B_0^{-\alpha}+Y_1B_1^{-\alpha}+Y_2B_2^{-\alpha}+Y_3B_3^{-\alpha}\right) \\ &= \tfrac{1}{16}\sum_{k=0}^{3}\sum_{l=0}^{3}Y_kY_lB_k^{\alpha}B_l^{-\alpha}. \end{aligned} \tag{22}$$

Here,

$$Y_kY_l=\begin{cases} 0, k\neq l \\ Y_k^2, k=l \end{cases} \tag{23}$$

Then, Equation (22) is written as

$$DWFRFT\cdot(DWFRFT)^H=\frac{1}{16}\sum_{k=0}^{3}Y_k^2. \tag{24}$$

After calculation, we know that $Y_k^2=4Y_k$. Equation (25) is obtained.

$$DWFRFT\cdot(DWFRFT)^H=\frac{1}{4}\sum_{k=0}^{3}Y_k=\frac{1}{4}(Y_0+Y_1+Y_2+Y_3)=I. \tag{25}$$

Thus, the unitarity of the *WFRFT* is proved. □

We can also implement the new reformulation with the help of fast Fourier transform (*FFT*), and its implementation module is shown in Figure 3. The weighting coefficients are readjusted A_l^{α} in Figure 3; so, the computational complexity is $O(N\log N)$.

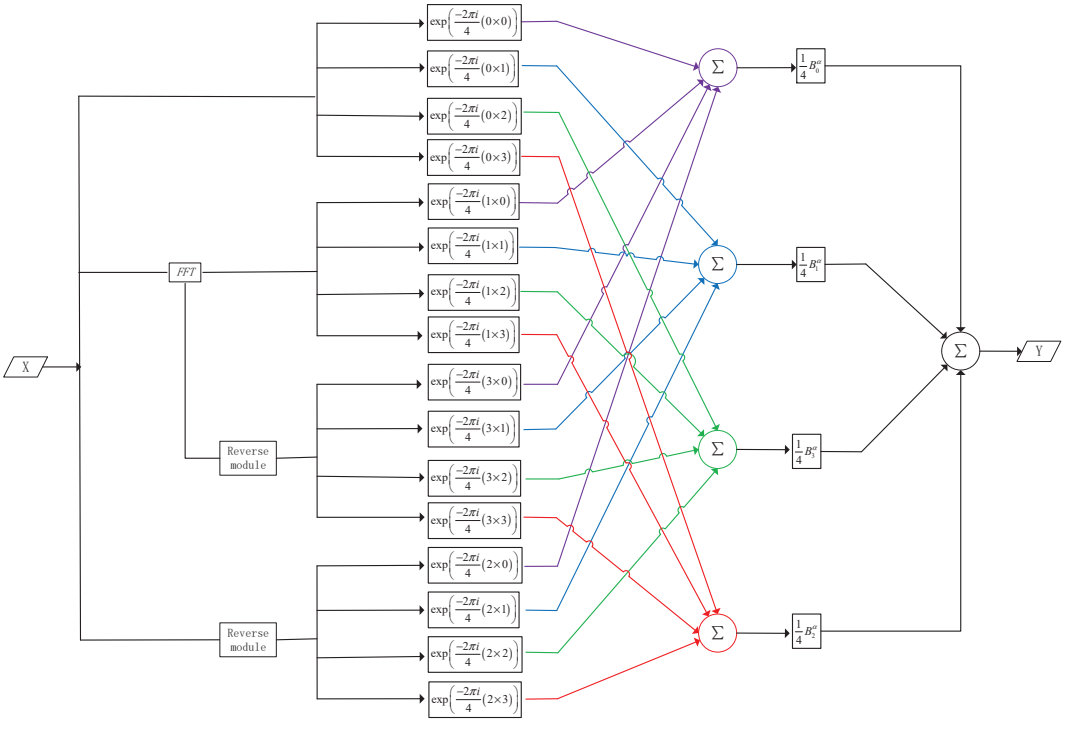


Figure 3. The reformulation of the *WFRFT* module.

4. Quantum Weighted Fractional Fourier Transform

In this section, we will present the *QWFRFT* with the help of the *QFT*. The *QFT* is an application of the classical Fourier transform to the amplitude of a quantum state. the vector $\boldsymbol{x}$ is transformed into the vector $\boldsymbol{y}$ by the classical Fourier transform,

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j u^{jk}; k = 0, 1, 2, \ldots, N-1 \tag{26}$$

where $u = e^{-2\pi i/N}$ and N is the signal length.

Similarly, *QFT* is applied to quantum state $|x\rangle = \sum_{j=0}^{N-1} x_j |j\rangle$ to obtain quantum state $|y\rangle = \sum_{k=0}^{N-1} y_k |k\rangle$, and we have

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j w_n^{jk}, \tag{27}$$

where $k = 0, 1, 2, \ldots, N-1$ and $w = e^{2\pi i/N}$. We note that Equation (27) is the inverse of the classical discrete Fourier transform; by convention, the *QFT* has the same effect as the inverse discrete Fourier transform.

In case that $|j\rangle$ is a basis state, the *QFT* can also be expressed as the map

$$QFT : |j\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} w^{jk} |k\rangle. \tag{28}$$

Equivalently, the *QFT* can be viewed as a unitary matrix acting on quantum state vectors, where the unitary matrix F_N is given by

$$F_N = \frac{1}{\sqrt{N}} \begin{pmatrix} w^{0\times 0} & w^{0\times 1} & \ldots & w^{0\times(n-1)} \\ w^{1\times 0} & w^{1\times 1} & \ldots & w^{1\times(n-1)} \\ \vdots & \vdots & \ddots & \vdots \\ w^{(n-1)\times 0} & w^{(n-1)\times 1} & \ldots & w^{(n-1)\times(n-1)} \end{pmatrix}. \tag{29}$$

Sine $N = 2^n$ and $w = e^{2\pi i/2^n}$. The electronic circuit of the *QFT* is shown in Figure 4.

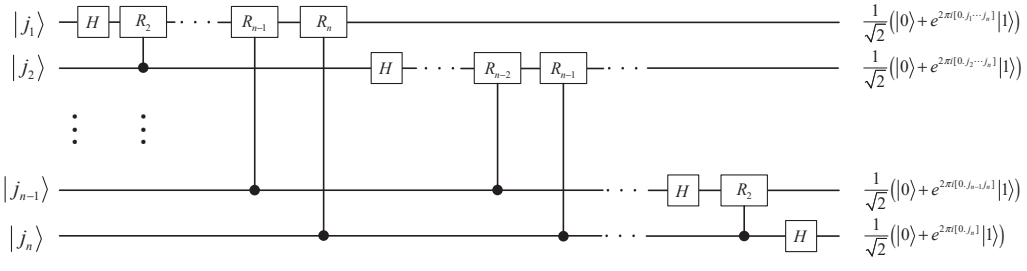


Figure 4. A circuit for the *QFT*.

Therefore, the *QFT* of the quantum state $|j\rangle = |j_1 j_2 \ldots j_n\rangle$ can be expressed as

$$QFT(|j_1 j_2 \ldots j_n\rangle) = \frac{1}{2^{n/2}} \left(|0\rangle + e^{2\pi i[0.j_n]}|1\rangle\right) \otimes \left(|0\rangle + e^{2\pi i[0.j_{n-1}j_n]}|1\rangle\right) \otimes \ldots \otimes \left(|0\rangle + e^{2\pi i[0.j_1 j_2 \ldots j_n]}|1\rangle\right), \tag{30}$$

where the binary of decimals can be expressed as

$$[0.j_1 j_2 \dots j_m] = \sum_{k=1}^{m} j_k 2^{-k}. \tag{31}$$

For instance, $[0.j_1] = j_1/2$ and $[0.j_1 j_2] = j_1/2 + j_2/2^2$. Then, the QFT can be further expressed as

$$QFT(|j_1 j_2 \dots j_n\rangle) = \frac{1}{2^{n/2}} \left(|0\rangle + w_1^{[j_n]}|1\rangle\right) \otimes \left(|0\rangle + w_2^{[j_{n-1} j_n]}|1\rangle\right) \otimes \dots \otimes \left(|0\rangle + w_n^{[j_1 j_2 \dots j_n]}|1\rangle\right). \tag{32}$$

Here, we use $[0.j_1 j_2 \dots j_m] = [j_1 j_2 \dots j_n]/2^m$, and $w_m = w^{-2m} = e^{2\pi i/2^m}$.

To implement the *QWFRFT*, we first present the integer powers (QFT^0, QFT^1, QFT^2, QFT^3) of the QFT.

1. We know that $QFT^0 = I$, and I is the identity matrix; obviously, this is a unitary operator. Then, its operation can be expressed as

$$|\alpha\rangle \bar{} I \bar{} |\beta_0\rangle$$

2. The QFT is a unitary operator. The Fourier transform of a quantum state $|\alpha\rangle$ can be expressed as

$$|\alpha\rangle \bar{} QFT \bar{} |\beta_1\rangle$$

3. The quadratic power of the QFT can be expressed as

$$QFT^2 = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \dots & 0 \end{pmatrix}$$

For the vector $(\alpha_0, \alpha_1, \dots, \alpha_{n-1})$, the transformation can be expressed as

$$(\alpha_0, \alpha_1, \dots, \alpha_{n-1}) \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \dots & 0 \end{pmatrix} = (\alpha_0, \alpha_{n-1}, \dots, \alpha_1)$$

In order to realize the quantum circuit of the above matrix, multiple swap gates are required. The swap gate of two quanta is shown in Figure 5.

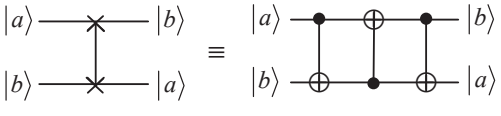


Figure 5. Swap gate.

Thus, for QFT^2, we provide quantum circuits of eight quantum states, as shown in Figure 6.

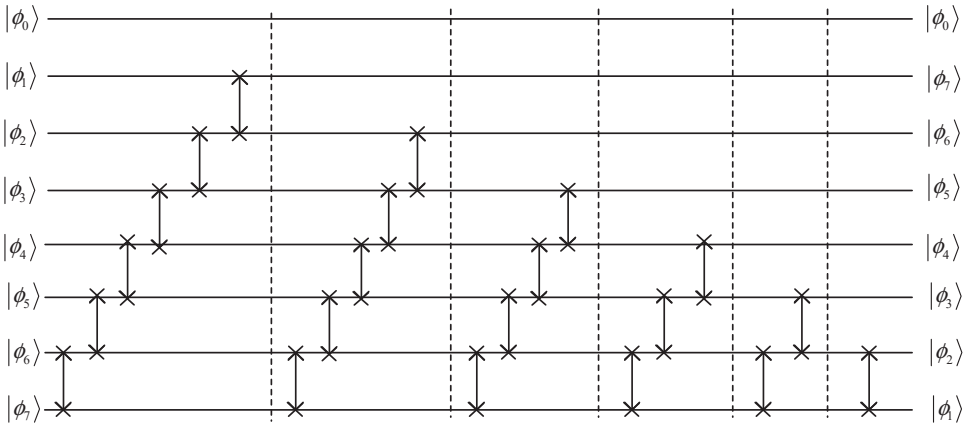


Figure 6. A circuit forthe QFT^2.

For a $2^n \times 2^n$ dimensional identity matrix, we can obtain the QFT^2 by row transformation, as shown in Figure 7.

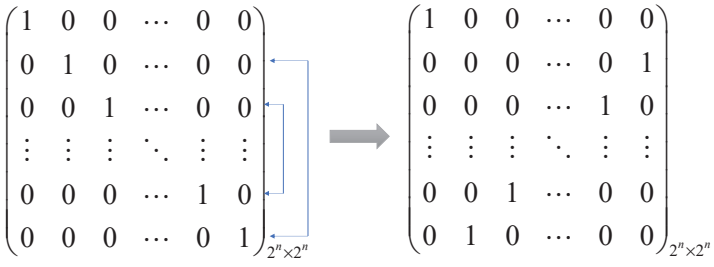


Figure 7. Matrix of the QFT^2.

Therefore, the quantum circuit of Figure 6 can be simplified as Figure 8.

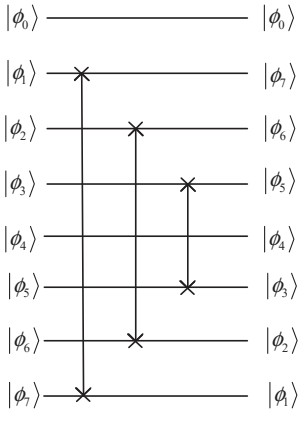


Figure 8. A circuit for the QFT^2.

Thus, the QFT^2 for quantum state $|\alpha\rangle$ can be expressed as

$$|\alpha\rangle^{-}QFT^{2-}|\beta_2\rangle$$

1. The third power of the *QFT*, which is equivalent to the inverse operation of the *QFT*, is also a unitary operator.

$$|\alpha\rangle^{-}QFT^{3-}|\beta_3\rangle$$

Therefore, the *QWFRFT* of the quantum state by Equation (10) can be expressed as

$$\begin{aligned} QWFRFT(|\alpha\rangle) &= \tfrac{1}{4}\Big(I(|\alpha\rangle), QFT(|\alpha\rangle), QFT^2(|\alpha\rangle), QFT^3(|\alpha\rangle)\Big)\begin{pmatrix}1 & 1 & 1 & 1\\ 1 & -i & -1 & i\\ 1 & -1 & 1 & -1\\ 1 & i & -1 & -i\end{pmatrix}\begin{pmatrix}B_0^\alpha\\ B_1^\alpha\\ B_2^\alpha\\ B_3^\alpha\end{pmatrix} \\ &= \tfrac{1}{4}(|\beta_0\rangle, |\beta_1\rangle, |\beta_2\rangle, |\beta_3\rangle)\begin{pmatrix}1 & 1 & 1 & 1\\ 1 & -i & -1 & i\\ 1 & -1 & 1 & -1\\ 1 & i & -1 & -i\end{pmatrix}\begin{pmatrix}B_0^\alpha\\ B_1^\alpha\\ B_2^\alpha\\ B_3^\alpha\end{pmatrix}. \end{aligned} \tag{33}$$

Equation (33) can be further written as

$$\begin{aligned} QWFRFT(|\alpha\rangle) &= \tfrac{1}{4}(|\beta_0\rangle, |\beta_1\rangle, |\beta_2\rangle, |\beta_3\rangle)\begin{pmatrix}1 & 1 & 1 & 1\\ 1 & -i & -1 & i\\ 1 & -1 & 1 & -1\\ 1 & i & -1 & -i\end{pmatrix}\begin{pmatrix}B_0^\alpha\\ B_1^\alpha\\ B_2^\alpha\\ B_3^\alpha\end{pmatrix} \\ &= \tfrac{1}{4}(|\beta_0\rangle, |\beta_1\rangle, |\beta_2\rangle, |\beta_3\rangle)\begin{pmatrix}\exp\left(\frac{-2\pi i 0\times 0}{4}\right) & \exp\left(\frac{-2\pi i 0\times 1}{4}\right) & \exp\left(\frac{-2\pi i 0\times 2}{4}\right) & \exp\left(\frac{-2\pi i 0\times 3}{4}\right)\\ \exp\left(\frac{-2\pi i 1\times 0}{4}\right) & \exp\left(\frac{-2\pi i 1\times 1}{4}\right) & \exp\left(\frac{-2\pi i 1\times 2}{4}\right) & \exp\left(\frac{-2\pi i 1\times 3}{4}\right)\\ \exp\left(\frac{-2\pi i 2\times 0}{4}\right) & \exp\left(\frac{-2\pi i 2\times 1}{4}\right) & \exp\left(\frac{-2\pi i 2\times 2}{4}\right) & \exp\left(\frac{-2\pi i 2\times 3}{4}\right)\\ \exp\left(\frac{-2\pi i 3\times 0}{4}\right) & \exp\left(\frac{-2\pi i 3\times 1}{4}\right) & \exp\left(\frac{-2\pi i 3\times 2}{4}\right) & \exp\left(\frac{-2\pi i 3\times 3}{4}\right)\end{pmatrix}\begin{pmatrix}B_0^\alpha\\ B_1^\alpha\\ B_2^\alpha\\ B_3^\alpha\end{pmatrix} \end{aligned} \tag{34}$$

where $B_k^\alpha = \exp\left(\frac{2\pi ik\alpha}{4}\right); k = 0, 1, 2, 3$. Then, Equation (34) can be written again as

$$\begin{aligned} QWFRFT(|\alpha\rangle) &= \tfrac{1}{4}\sum_{l=0}^{3}\sum_{k=0}^{3}|\beta_l\rangle \exp\left(\tfrac{-2\pi ilk}{4}\right)B_k^\alpha \\ &= \tfrac{1}{4}\sum_{l=0}^{3}\sum_{k=0}^{3}|\beta_l\rangle \exp\left(\tfrac{-2\pi ilk}{4}\right)\exp\left(\tfrac{2\pi ik\alpha}{4}\right) \\ &= \tfrac{1}{4}\sum_{l=0}^{3}\sum_{k=0}^{3}|\beta_l\rangle \exp\left(\tfrac{2\pi ik(\alpha-l)}{4}\right). \end{aligned} \tag{35}$$

With the help of the quantum artificial neural network (*QANN*), we are inspired to design a *QWFRFT*. Here, we first introduce the *QANN* [34,35]. If we use $\{|e_1\rangle, |e_2\rangle, \ldots, |e_M\rangle\}$ to denote the canonical basis for $\mathbb{C}^M$, then the quantum artificial neural network above can be rewritten as

$$Q(|x\rangle) = \sum_{k=1}^{M}\sum_{j=1}^{N}\left(\alpha_{j,k}^{(1)}\sigma_k\left(\left\langle w_{j,k}^{(1)}\middle|T|x\right\rangle + \theta_{j,k}^{(1)}\right) + i\alpha_{j,k}^{(2)}\sigma_k\left(\left\langle w_{j,k}^{(2)}\middle|T|x\right\rangle + \theta_{j,k}^{(2)}\right)\right)|e_k\rangle. \tag{36}$$

Put $y_{j,k}^{(i)} = \sigma_k\left(\sum_{t=1}^{n}\left\langle w_{j,k}^{(i)}(t)\middle|T|x_t\right\rangle + \theta_{j,k}^{(i)}\right)$ and $\left|\alpha_k^{(i)}\right\rangle = \sum_{j=1}^{N}\alpha_{j,k}^{(i)}y_{j,k}^{(i)}|e_k\rangle$. Then, a *QANN* can be illustrated by Figures 9 and 10 below.

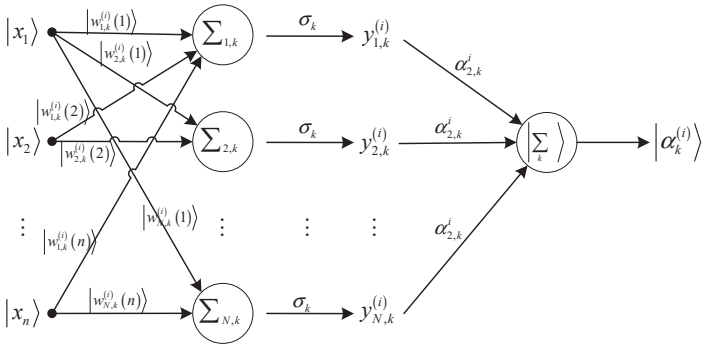


Figure 9. The output $\left|\alpha_k^{(i)}\right\rangle$ of a *QANN*, where $i = 1, 2; k = 1, 2, \ldots, M$.

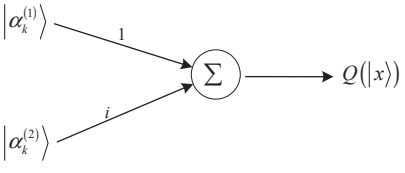


Figure 10. The output $Q(|x\rangle)$ of a *QANN*.

Thus, we can present the circuit of the *QWFRFT*, as shown in Figure 11.

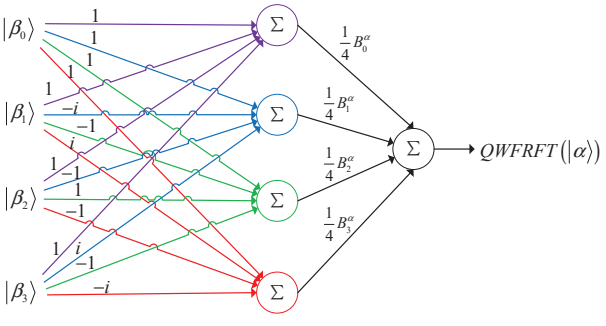


Figure 11. A circuit for the *QWFRFT*.

So far, we have completed the *QWFRFT* and circuit implementation. The work of this paper is a supplement to the work of Parasa et al. At one point, researchers pointed out that there is no quantum-weighted fractional Fourier transform [32]. However, our study illustrates the diversity of *FRFT* and proposes *QWFRFT*. Due to the characteristics of quantum parallelism, we believe that the *QWFRFT* has a wider application space.

At present, our method is only applicable to closed systems. The standard quantum theory has shown its limit to describe successfully experimental results. Counterintuitive results are obtained in different experiments [36,37]. The open system effects need to be further analyzed.

5. Conclusions

Unitarity is a prerequisite for the realization of quantum algorithms. In this paper, we proposed the reformulation of the *WFRFT*. The unitarity of the *WFRFT* was proved by means of the proposed reformulation. The *QFT* is an important part of the *QWFRFT*. Furthermore, we presented the integer power operation and quantum circuit of the *QFT*,

which lays the foundation for the *QWFRFT*. Finally, we designed the circuit of the *QWFRFT* with the help of a quantum artificial neural network and proposed the electronic circuit of the *QWFRFT*. The results of this paper show that there is a *QFRFT* algorithm, which lays the foundation for further research.

Author Contributions: Conceptualization, T.Z.; methodology, T.Z.; validation, T.Z., T.Y. and Y.C.; formal analysis, T.Z.; investigation, T.Z.; resources, T.Y.; writing—original draft preparation, T.Z.; writing—review and editing, T.Z.; project administration, T.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the Fundamental Research Funds for the Central Universities (N2123016); and the Scientific Research Projects of Hebei colleges and universities (QN2020511).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Feynman, R.P. Simulating physics with computers. *Int. J. Theor. Phys* **1982**, *21*, 467–488. [CrossRef]
2. Deutsch, D. Quantum theory, the Church–Turing principle and the universal quantum computer. *Proc. R. Soc. Lond. A Math. Phys. Sci.* **1985**, *400*, 97–117.
3. Shor, P.W. Algorithms for quantum computation: Discrete logarithms and factoring. In Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science, Santa Fe, NM, USA, 20–22 November 1994; pp. 124–134.
4. Shor, P.W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev.* **1999**, *41*, 303–332. [CrossRef]
5. Grover, L.K. A fast quantum mechanical algorithm for database search. In Proceedings of the 28th Annual ACM Symposium on the Theory of Computing, New York, NY, USA, 22–24 May 1996; pp. 212–219.
6. Biham, E.; Biham, O.; Biron, D.; Grassl, M.; Lidar, D.A. Grover's quantum search algorithm for an arbitrary initial amplitude distribution. *Phys. Rev. A* **1999**, *60*, 2742–2745. [CrossRef]
7. Boyer, M.; Brassard, G.; Høyer, P.; Tapp, A. Tight bounds on quantum searching. *Fortschr. Der Phys. Prog. Phys.* **1998**, *46*, 493–505. [CrossRef]
8. Grover, L.K. Quantum computers can search rapidly by using almost any transformation. *Phys. Rev. Lett.* **1998**, *80*, 4329–4332. [CrossRef]
9. GuiLu, L.; WeiLin, Z.; YanSong, L.; Li, N. Arbitrary phase rotation of the marked state cannot be used for Grover's quantum search algorithm. *Commun. Theor. Phys.* **1999**, *32*, 335–338. [CrossRef]
10. Long, G.L.; Li, Y.S.; Zhang, W.L.; Niu, L. Phase matching in quantum searching. *Phys. Lett. A* **1999**, *262*, 27–34. [CrossRef]
11. Narayanan, A.; Moore, M. Quantum-inspired genetic algorithms. In Proceedings of the IEEE International Conference on Evolutionary Computation, Nagoya, Japan, 20–22 May 1996; pp. 61–66.
12. Shenvi, N.; Kempe, J.; Whaley, K.B. Quantum random-walk search algorithm. *Phys. Rev. A* **2003**, *67*, 052307. [CrossRef]
13. Han, K.-H.; Kim, J.-H. Genetic quantum algorithm and its application to combinatorial optimization problem. In Proceedings of the IEEE Congress on Evolutionary Computation, La Jolla, CA, USA, 16–19 July 2000; pp. 1354–1360.
14. Chen, L.B. Mitochondrial membrane potential in living cells. *Annu. Rev. Cell Biol.* **1988**, *4*, 155–181. [CrossRef]
15. Høyer, P.; Špalek, R. Quantum fan-out is powerful. *Theory Comput.* **2005**, *1*, 81–103. [CrossRef]
16. Sun, J.; Wu, X.; Palade, V.; Fang, W.; Lai, C.-H.; Xu, W. Convergence analysis and improvements of quantum-behaved particle swarm optimization. *Inf. Sci.* **2012**, *193*, 81–103. [CrossRef]
17. Harrow, A.W.; Hassidim, A.; Lloyd, S. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* **2009**, *103*, 150502. [CrossRef] [PubMed]
18. Childs, A.M.; Kothari, R.; Somma, R.D. Quantum algorithm for systems of linear equations with exponentially improved dependence on precision. *SIAM J. Comput.* **2017**, *46*, 1920–1950. [CrossRef]
19. Clader, B.D.; Jacobs, B.C.; Sprouse, C.R. Preconditioned quantum linear system algorithm. *Phys. Rev. Lett.* **2013**, *110*, 250504. [CrossRef]
20. Wossnig, L.; Zhao, Z.; Prakash, A. Quantum linear system algorithm for dense matrices. *Phys. Rev. Lett.* **2018**, *120*, 050502. [CrossRef]
21. Arrazola, J.M.; Kalajdzievski, T.; Weedbrook, C.; Lloyd, S. Quantum algorithm for nonhomogeneous linear partial differential equations. *Phys. Rev. A* **2019**, *100*, 032306. [CrossRef]
22. Lloyd, S.; De Palma, G.; Gokler, C.; Kiani, B.; Liu, Z.-W.; Marvian, M.; Tennie, F.; Palmer, T. Quantum algorithm for nonlinear differential equations. *arXiv* **2020**, arXiv:2011.06571.

23. Childs, A.M.; Liu, J.-P. Quantum spectral methods for differential equations. *Commun. Math. Phys.* **2020**, *375*, 1427–1457. [CrossRef]
24. Childs, A.M.; Liu, J.-P.; Ostrander, A. High-precision quantum algorithms for partial differential equations. *Quantum* **2021**, *5*, 574. [CrossRef]
25. Coppersmith, D. An approximate Fourier transform useful in quantum factoring. *arXiv* **2002**, arXiv:quant-ph/0201067.
26. Nielsen, M.A.; Chuang, I.L. Quantum computation and quantum information. *Phys. Today* **2001**, *54*, 60.
27. Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N.; Lloyd, S. Quantum machine learning. *Nature* **2017**, *549*, 195–202. [CrossRef] [PubMed]
28. Namias, V. The fractional order Fourier transform and its application to quantum mechanics. *IMA J. Appl. Math.* **1980**, *25*, 241–265. [CrossRef]
29. Candan, C.; Kutay, M.A.; Ozaktas, H.M. The discrete fractional Fourier transform. *IEEE Trans. Signal Process.* **2000**, *48*, 1329–1337. [CrossRef]
30. Shih, C.C. Fractionalization of Fourier transform. *Opt. Commun.* **1995**, *118*, 495–498. [CrossRef]
31. Ozaktas, H.M.; Arikan, O.; Kutay, M.A.; Bozdagt, G. Digital computation of the fractional Fourier transform. *IEEE Trans. Signal Process.* **1996**, *44*, 2141–2150. [CrossRef]
32. Parasa, V.; Perkowski, M. Quantum pseudo-fractional fourier transform using multiple-valued logic. In Proceedings of the IEEE 42nd International Symposium on Multiple-Valued Logic, Victoria, BC, Canada, 14–16 May 2012; pp. 311–314.
33. Bailey, D.H.; Swarztrauber, P.N. The fractional Fourier transform and applications. *SIAM Rev.* **1991**, *33*, 389–404. [CrossRef]
34. Cao, H.; Cao, F.; Wang, D. Quantum artificial neural networks with applications. *Inf. Sci.* **2015**, *290*, 1–6. [CrossRef]
35. da Silva, A.J.; de Oliveira, W.R. Comments on "quantum artificial neural networks with applications". *Inf. Sci.* **2016**, *370*, 120–122. [CrossRef]
36. Eleuch, H.; Rotter, I. Nearby states in non-Hermitian quantum systems I: Two states. *Eur. Phys. J. D* **2015**, *69*, 229. [CrossRef]
37. Eleuch, H.; Rotter, I. Clustering of exceptional points and dynamical phase transitions. *Phys. Rev. A* **2016**, *93*, 042116. [CrossRef]

mathematics

MDPI

Article

Progress towards Analytically Optimal Angles in Quantum Approximate Optimisation

Daniil Rabinovich *, Richik Sengupta, Ernesto Campos, Vishwanathan Akshay and Jacob Biamonte

Laboratory of Quantum Algorithms for Machine Learning and Optimisation, Skolkovo Institute of Science and Technology, 3 Nobel Street, 121205 Moscow, Russia; r.sengupta@skoltech.ru (R.S.); ernesto.campos@skoltech.ru (E.C.); akshay.vishwanathan@skoltech.ru (V.A.); j.biamonte@skoltech.ru (J.B.)
* Correspondence: daniil.rabinovich@skoltech.ru

Abstract: The quantum approximate optimisation algorithm is a p layer, time variable split operator method executed on a quantum processor and driven to convergence by classical outer-loop optimisation. The classical co-processor varies individual application times of a problem/driver propagator sequence to prepare a state which approximately minimises the problem's generator. Analytical solutions to choose optimal application times (called parameters or angles) have proven difficult to find, whereas outer-loop optimisation is resource intensive. Here we prove that the optimal quantum approximate optimisation algorithm parameters for $p = 1$ layer reduce to one free variable and in the thermodynamic limit, we recover optimal angles. We moreover demonstrate that conditions for vanishing gradients of the overlap function share a similar form which leads to a linear relation between circuit parameters, independent of the number of qubits. Finally, we present a list of numerical effects, observed for particular system size and circuit depth, which are yet to be explained analytically.

Keywords: variatonal algorithms; QAOA; quantum circuit optimization

MSC: 81P68

Citation: Rabinovich, D.; Sengupta, R.; Campos, E.; Akshay, V.; Biamonte, J. Progress towards Analytically Optimal Angles in Quantum Approximate Optimisation. *Mathematics* **2022**, *10*, 2601. https://doi.org/10.3390/math10152601

Academic Editors: Fernando L. Pelayo and Mauro Mezzini

Received: 22 June 2022
Accepted: 23 July 2022
Published: 26 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

The field of quantum algorithms has dramatically transformed in the last few years due to the advent of a quantum to classical feedback loop: a fixed depth quantum circuit is adjusted to minimise a cost function. This approach partially circumvents certain limitations such as variability in pulse timing and requires shorter depth circuits at the cost of outer-loop training [1–6]. The most studied algorithm in this setting is the quantum approximate optimisation algorithm (QAOA) [7] which was developed to approximate solutions to combinatorial optimisation problem instances [8] i.e., MAX-k-SAT [9,10], MAX-Cut [7,11–16], and MAX-k-Colorable-Subgraph [17] instances. The algorithm has certain real-world applications, including finances [18] and might prove useful for general constraint optimisation [19].

The setting of QAOA is that of n qubits: states are represented as vectors in $V_n = [\mathbb{C}^2]^{\otimes n}$. We are given a non-negative Hamiltonian $P \in \mathrm{herm}_{\mathbb{C}}(V_n)$ and we seek the normalised ground vector $|t\rangle \in \arg\min_{\phi \in \{0,1\}^n} \langle \phi | P | \phi \rangle$.

QAOA might be viewed as a (time-variable fixed-depth) quantum split operator method. We let $\mathcal{V}(\gamma)$ be the propagator of P applied for time γ. We consider a second propagator $\mathcal{U}(\beta)$ generated by applying a yet-to-be-defined Hamiltonian H_x for time β. We start off in the equal superposition state $|+\rangle^{\otimes n} = 2^{-n/2}(|0\rangle + |1\rangle)^{\otimes n}$ and form a p-depth $\mathcal{U}, \mathcal{V}$ sequence:

$$|g_p(\boldsymbol{\gamma}, \boldsymbol{\beta})|^2 = |\langle t | \Pi_{k=1}^{p} [\mathcal{U}(\beta_k)\mathcal{V}(\gamma_k)] | + \rangle^{\otimes n}|^2. \quad (1)$$

The time of application of each propagator is varied to maximise preparation of the state $|t\rangle$. Finding $\boldsymbol{\gamma}, \boldsymbol{\beta}$ to maximise $|g_p(\boldsymbol{\gamma}, \boldsymbol{\beta})|$ has shown to be cumbersome. Even lacking such solutions, much progress has been made.

Recent milestones include experimental demonstration of $p = 3$ depth QAOA (corresponding to six tunable parameters) using a twenty three qubits [1] superconducting processor, universality results [20,21], as well as several results that aid and improve on the original implementation of the algorithm [11,12,17]. Towards practical realisation of the QAOA, trapped ion-based quantum computers have recently shown promising results, including demonstrations on up to forty qubits [2] and the potential to realise arbitrary combinatorial optimisation problems with all to all connectivity based on hardware-inspired modifications [22]. Although QAOA exhibits provable advantages such as recovering a near-optimal query complexity in Grover's search [23] and offers a pathway towards quantum advantage [13], several limitations have been discovered for low depth QAOA [9,24,25].

In the setting of maximum-constraint satisfiability (e.g., minimizing a Hamiltonian representing a function of type $f : \{0,1\}^n \to \mathbb{R}_+$), it has been shown that underparameterisation of QAOA sequences can be induced by increasing a problem instances constraint to variable ratio [9]. This effect persists in graph minimisation problems [26]. While this effect is perhaps an expected limitation of the quantum algorithm, parameter concentrations and noise-assisted training add a degree of optimism. QAOA exhibits parameter concentrations, in which training for some fraction of $\omega < n$ qubits provides a training sequence for n qubits [27]. Moreover, whereas layerwise training saturates for QAOA in which the algorithm plateaus and fails to reach the target, local coherent noise recovers layerwise training robustness [28]. Both concentrations and noise-assisted training imply a reduction in computational resources required in outer-loop optimisation.

Exact solutions to find the optimal parameters for QAOA have only been possible in special cases including, e.g., fully connected graphs [14–16] and projectors [27]. A general analytical approach which would allow for (i) calculation of optimal parameters, (ii) estimation of the critical circuit depth and (iii) performance guarantees for fixed depth remains open.

Here we prove that optimal QAOA parameters for $p = 1$ are related as $\gamma_1 = \pi - 2\beta_1$ and in the thermodynamic limit, we recover optimality as $\beta_1 n \to \pi$ and $\gamma_1 \to \pi$. We moreover demonstrate that conditions for vanishing gradients of the overlap function share a similar form which leads to a linear relation between circuit parameters, independent of the number of qubits. We hence devise an additional means to recover parameter concentrations [27] analytically. Finally, we present a list of numerical effects, observed for particular system size and circuit depth, which are yet to be explained analytically.

2. State Preparation with QAOA

We consider an n-qubit complex vector space $V_n = [\mathbb{C}^2]^{\otimes n} \cong \mathbb{C}^{2^n}$ with fixed standard computational basis $B_n = \{|0\rangle, |1\rangle\}^{\otimes n}$. For an arbitrary target state $|t\rangle \in B_n$ (equivalently $|t\rangle, t \in \{0,1\}^{\times n}$) we define propagators

$$\mathcal{U}(\beta) \equiv e^{-i\beta H_x}, \; \mathcal{V}(\gamma) \equiv e^{-i\gamma P}, \tag{2}$$

where $P = |t\rangle\langle t|$ and $H_x = \sum_{j=1}^{n} X_j$ is the one-body mixer Hamiltonian with X_j the Pauli matrix acting non-trivially on the j-th qubit. Here we focus on the state preparation, thus choosing the problem Hamiltonian to be a projector ($P^2 = P$) on an arbitrary bit string $|t\rangle$. We note that while the projector has only two energy levels, the effective Hamiltonian of the whole QAOA sequence has up to $n + 1$ distinct energy levels. In such settings, the propagator $\mathcal{V}(\gamma)$ acting on a superposition adds a phase $-\gamma$ to the component $|t\rangle$, while the propagator $\mathcal{U}(\beta)$ mixes the components' amplitudes.

A p-depth (p layer) QAOA circuit prepares a quantum state $|\psi\rangle$ as:

$$\left|\psi_p(\boldsymbol{\gamma},\boldsymbol{\beta})\right\rangle = \prod_{k=1}^{p}[\mathcal{U}(\beta_k)\mathcal{V}(\gamma_k)]|+\rangle^{\otimes n}, \tag{3}$$

where $\gamma_k \in [0, 2\pi)$, $\beta_k \in [0, \pi)$. The optimisation task is to determine QAOA optimal parameters for which the state prepared in (3) achieves maximum absolute value of the overlap $g_p(\boldsymbol{\gamma},\boldsymbol{\beta}) = \langle t|\psi_p(\boldsymbol{\gamma},\boldsymbol{\beta})\rangle$ with the target $|t\rangle$. In other words, we search for

$$(\boldsymbol{\gamma}_{opt},\boldsymbol{\beta}_{opt}) \in \arg\max_{\boldsymbol{\gamma},\boldsymbol{\beta}} \left|g_p(\boldsymbol{\gamma},\boldsymbol{\beta})\right|. \tag{4}$$

Note that the problem is equivalent to the minimisation of the ground state energy of Hamiltonian $P^{\perp} = \mathbb{1} - |t\rangle\langle t|$,

$$\min_{\boldsymbol{\gamma},\boldsymbol{\beta}} \left\langle\psi_p(\boldsymbol{\gamma},\boldsymbol{\beta})\right|P^{\perp}\left|\psi_p(\boldsymbol{\gamma},\boldsymbol{\beta})\right\rangle = 1 - \max_{\boldsymbol{\gamma},\boldsymbol{\beta}} \left|g_p(\boldsymbol{\gamma},\boldsymbol{\beta})\right|^2. \tag{5}$$

Remark 1 (Inversion symmetry)**.** *Under the affine transformation*

$$(\gamma, \beta) \to (2\pi - \gamma, \pi - \beta) \tag{6}$$

the absolute value of the overlap remains invariant as $g_p \to (-1)^n g_p^*$. Therefore, this narrows the search space to $\gamma_k \in [0, \pi)$, $\beta_k \in [0, \pi)$, whereas maximums inside the restricted region determine maximums in the composite space using Equation (6).

Proposition 1 (Overlap invariance)**.** *The overlap function $g_p(\boldsymbol{\gamma},\boldsymbol{\beta})$ is invariant with respect to $|t\rangle \in B_n$.*

Proof. Each $|t\rangle = |t_1 t_2 \dots t_n\rangle \in B_n$ determines a unitary operator $U = U^{\dagger} = \bigotimes_{j=1}^{n} X_j^{t_j}$. Hence, we have

$$\begin{aligned} g_p(\boldsymbol{\gamma},\boldsymbol{\beta}) =& \langle \mathbf{0}|U^{\dagger}\prod_{k=1}^{p} e^{-i\beta_k H_x} e^{-i\gamma_k U(|\mathbf{0}\rangle\langle\mathbf{0}|)U^{\dagger}}|+\rangle^{\otimes n} \\ =& \langle \mathbf{0}|U^{\dagger}\prod_{k=1}^{p} e^{-i\beta_k H_x}[Ue^{-i\gamma_k(|\mathbf{0}\rangle\langle\mathbf{0}|)}U^{\dagger}]|+\rangle^{\otimes n} \\ =& \langle \mathbf{0}|\prod_{k=1}^{p} e^{-i\beta_k H_x} e^{-i\gamma_k|\mathbf{0}\rangle\langle\mathbf{0}|}|+\rangle^{\otimes n}. \end{aligned} \tag{7}$$

The first equality follows from $U|\mathbf{0}\rangle = |t\rangle$ where $|\mathbf{0}\rangle = |0\rangle^{\otimes n}$. The second equality follows from the definition of the matrix exponential. The third equality follows as U commutes with H_x as does any analytic function of H_x, and $U|+\rangle^{\otimes n} = |+\rangle^{\otimes n}$. Thus, the overlap is seen to be independent of the target bit string $|t\rangle$. □

Remark 2. *Overlap invariance introduced in Proposition 1 shows that optimisation problems in Equations (4) and (5) do not depend on the target. Therefore, optimal parameters are the same for any target state. Thus, with no loss of generality we limit our consideration to the target $|t\rangle = |\mathbf{0}\rangle$.*

Preparation of state (3) requires a strategy to assign $2p$ variational parameters by outer-loop optimisation.

Remark 3 (Global optimisation)**.** *A strategy when all $2p$ parameters are optimised simultaneously which might provide the best approximation to prepare $|t\rangle$.*

Remark 4 (Layerwise training)**.** *Optimisation of parameters layer by layer. At each step of the algorithm, only one layer is optimised. After a layer is trained, a new layer is added and its parameters are optimised while keeping the parameters of the previous layers fixed.*

Global optimisation is evidently challenging for high-depth circuits. The optimisation can, in principle, be simplified by exploiting problem symmetries [29] and leveraging parameter concentrations [27,30]. Layerwise training might avoid barren plateaus [31] yet is known [28] to stagnate at some critical depth, past which additional layers (trained one at a time) do not improve overlap. Local coherent noise was found to re-establish the robustness of layerwise training [28].

3. $p = 1$ QAOA

For a single layer, the global and layerwise strategies are equivalent. Such a circuit was considered to establish parameter concentrations [27] analytically. The overlap was shown to be:

$$|g_1(\gamma,\beta)|^2 = \frac{1}{2^n}\left[1 + 2\cos^n\beta(\cos(\gamma - n\beta) - \cos n\beta) + 2\cos^{2n}\beta(1-\cos\gamma)\right]. \tag{8}$$

To find extreme points of (8) the authors in [27] set the derivatives with respect to γ and β to zero. This approach leads to solutions which contain maxima but also the minimum of the overlap (8). These must be carefully separated. Moreover, this approach ignores the operator structure of the overlap as presented here. For aesthetics, subscript *opt* in $\boldsymbol{\gamma}_{opt}$ and $\boldsymbol{\beta}_{opt}$ is further omitted.

Theorem 1. *Optimal $p = 1$ QAOA parameters relate as $\gamma = \pi - 2\beta$.*

Proof. To maximise the absolute value of the overlap

$$g \equiv g_1(\gamma,\beta) = \langle \mathbf{0}|e^{-i\beta H_x}e^{-i\gamma P}|+\rangle^{\otimes n}, \tag{9}$$

with $P = |\mathbf{0}\rangle\langle\mathbf{0}|$ we use the standard conditions $\dfrac{\partial(gg^*)}{\partial\gamma} = \dfrac{\partial(gg^*)}{\partial\beta} = 0$. Setting the first derivative to zero we arrive at

$$\langle \mathbf{0}|e^{-i\beta H_x}e^{-i\gamma P}P|+\rangle^{\otimes n}g^* = \langle +|^{\otimes n}Pe^{i\gamma P}e^{i\beta H_x}|\mathbf{0}\rangle g. \tag{10}$$

Using the explicit form of the projector and the fact that $\langle \mathbf{0}|e^{-i\beta H_x}|\mathbf{0}\rangle = \cos^n\beta$, equation (10) simplifies into

$$g = g^*e^{-2i\gamma} \Leftrightarrow ge^{i\gamma} = g^*e^{-i\gamma}, \tag{11}$$

which is equivalent to

$$\arg g = -\gamma. \tag{12}$$

Then the derivative of expression (9) with respect to β is set to zero and we arrive at

$$\langle \mathbf{0}|e^{-i\beta H_x}H_xe^{-i\gamma P}|+\rangle^{\otimes n}g^* = \langle +|^{\otimes n}e^{i\gamma P}H_xe^{i\beta H_x}|\mathbf{0}\rangle g. \tag{13}$$

Moving H_x next to its eigenstate $|+\rangle^{\otimes n}$ is compensated as follows:

$$\begin{aligned}&\langle \mathbf{0}|e^{-i\beta H_x}\{e^{-i\gamma P}H_x + (e^{-i\gamma}-1)[H_x,P]\}|+\rangle^{\otimes n}g^*\\ &= \langle +|^{\otimes n}\{H_xe^{i\gamma P} + [P,H_x](e^{i\gamma}-1)\}e^{i\beta H_x}|\mathbf{0}\rangle g.\end{aligned} \tag{14}$$

After simplification (see Remark 5) we arrive at

$$-gA = g^* A^* e^{-i\gamma}, \tag{15}$$

where $A = \langle + |^{\otimes n} [P, H_x] e^{i\beta H_x} |\mathbf{0}\rangle$. Now g^* is substituted from Equation (11) to establish

$$-e^{-i\gamma} A = A^*. \tag{16}$$

Thus, similar to Equation (12) we arrive at

$$\arg A = \frac{\gamma + \pi}{2}. \tag{17}$$

A is calculated as

$$A\sqrt{2^n} = \langle \mathbf{0} | (H_x - n) e^{i\beta H_x} |\mathbf{0}\rangle = -n \cos^{n-1} \beta e^{-i\beta}, \tag{18}$$

which shows that $\arg A = \pi - \beta$. Thus, from Equation (17) we arrive at

$$\pi - \beta = \frac{\gamma + \pi}{2}, \tag{19}$$

which finally establishes $\gamma = \pi - 2\beta$. □

Remark 5 (Trivial solutions)**.** *Equation* (14) *has three pathological solutions which must be ruled out: (i)* $\sin \frac{\gamma}{2} = 0$ *(which sets* $e^{i\gamma} - 1 = 0$*), (ii)* $\cos \beta = 0$ *(which sets* $A = 0$*), (iii)* $g(\gamma, \beta) = 0$*. All three cases imply* $|g(\gamma, \beta)| \leq g(0, 0)$*.*

Remark 6. *The zero derivative conditions result in* (11) *and* (15) *which have a similar form, viz.* $x = x^* e^{i\varphi}$*. The first condition* (11) *can be obtained without differentiation [28] using the explicit form of the overlap Equation* (9)

$$g\sqrt{2^n} = e^{-i\gamma} \cos^n \beta + (e^{-i\beta n} - \cos^n \beta), \tag{20}$$

and the fact that $\max_\gamma \left| Ae^{-i\gamma} + B \right| = |A| + |B|$ *for any* $A, B \in \mathbb{C}$*. Although the derivative with respect to* β *leads to the condition* (15)*, we find no way to recover this using elementary alignment arguments.*

Remark 7. *While optimal angle relation* $\gamma = \pi - 2\beta$ *has also been established in [27], here we demonstrate that it can result from certain ansatz symmetry, manifested in similar form of zero derivatives conditions* (11) *and* (15)*. This can provide useful insights to understand similar optimal angle dependency for deeper circuits (Section 4.2).*

To find optimal parameters one needs to solve the zero derivative conditions and then take solutions that deliver a global maximum to the overlap. For convenience, we substitute $\gamma = \pi - 2\beta$ to the overlap function (20), square it and after simplification arrive at

$$|g|^2 2^n = 1 + 4 \cos^{n+1} \beta (\cos^{n+1} \beta - \cos(n+1)\beta), \tag{21}$$

which is used to prove the next theorem.

Theorem 2. *The optimal* $p = 1$ *QAOA parameters converge as* $\beta n \to \pi$ *and* $\gamma \to \pi$ *when* $n \to \infty$*.*

Proof. Using the explicit form of the overlap (20), from Equation (11) one can establish

$$\mathrm{Im}[e^{i\gamma}(e^{-i\beta n} - \cos^n \beta)] = 0. \tag{22}$$

Substituting $\gamma = \pi - 2\beta$ one arrives at

$$\mathrm{Im}[e^{-i(n+2)\beta} - e^{-2i\beta}\cos^n\beta)] = 0, \tag{23}$$

which is equivalent to

$$\sin(n+2)\beta = \sin 2\beta \cos^n\beta. \tag{24}$$

We solve this equation in the limit $n \to \infty$. In this limit $\sin 2\beta \cos^n\beta \to 0$ independent of the value of β. Thus, the left-hand side of Equation (24) tends to zero. This implies that the leading order solution scales as

$$\beta = \frac{k\pi}{n+2} + o(n^{-1}) \tag{25}$$

where $k < n$ is a positive integer (in principle, n-dependent). To recover the optimal constant k we substitute Equation (25) to Equation (21) to obtain

$$|g|^2 2^n = 1 + 4\cos^{n+2}\frac{k\pi}{n+2}\left(\cos^n\frac{k\pi}{n+2} - (-1)^k\right) \tag{26}$$

up to $o(1)$ terms. Finally, as cosine is monotonously decreasing in the interval $[0, \pi)$ it is evident that the overlap maximises for the smallest odd constant $k = 1$. Therefore, the optimal parameter β is given by

$$\beta = \frac{\pi}{n+2} + o(n^{-1}) = \frac{\pi}{n} + o(n^{-1}), \tag{27}$$

which implies $n\beta \to \pi$ and thus $\gamma = \pi - 2\beta \to \pi$ when $n \to \infty$. □

Remark 8. *In Theorem 2 the leading order solutions were found for optimal parameters. Higher order corrections in n are found from Equation (24). For example, it is straightforward to show that*

$$\beta = \frac{\pi}{n} - \frac{4\pi}{n^2} + O(n^{-3}), \tag{28}$$

$$\gamma = \pi - \frac{2\pi}{n} + \frac{8\pi}{n^2} + O(n^{-3}). \tag{29}$$

Remark 9. *Expressions (28) and (29) are used to demonstrate parameter concentrations [27], i.e., the effect when optimal parameters for n and n + 1 qubits are polynomially close.*

Theorems 1 and 2 provide state of the art analytical results for state preparation with $p = 1$ depth QAOA circuit. For deeper circuits and more general settings, analysis becomes complicated and known results are mostly numerical. Therefore, below we provide a list of numerical effects for deeper circuits which lack analytical explanations.

4. Empirical Findings Missing Analytical Theory

4.1. Parameter Concentration in $p \geq 2$ QAOA

From expression (3), overlaps for circuits of different depths are related recursively as

$$g_{p+1}(\boldsymbol{\gamma}, \boldsymbol{\beta}, \gamma_{p+1}, \beta_{p+1}) = g_p(\boldsymbol{\gamma}, \tilde{\boldsymbol{\beta}}) + g_p(\boldsymbol{\gamma}, \boldsymbol{\beta})\cos^n\beta_{p+1}(e^{-i\gamma_{p+1}} - 1), \tag{30}$$

where $\tilde{\boldsymbol{\beta}} = (\beta_1 + \beta_{p+1}, \ldots, \beta_p + \beta_{p+1})$. This recursion was used in [27] for $p = 2$ where it was shown that in the thermodynamic limit $n \to \infty$ the zero derivative conditions let one obtain solutions for which $n\beta \to \pi$ and $\gamma \to \pi$. This establishes parameter concentrations. The effect was further confirmed numerically on up to $n = 17$ qubits and $p = 5$ layers. For arbitrary depth, parameter concentrations are conjectured, yet analytical confirmation remains open. The recursion (30) can be used in the suggested operator formalism to

derive a system of equations to calculate optimal parameters for circuits of arbitrary depth. In the suggested formalism the zero derivative conditions will contain expectations of propagators used in the circuit, and the system can be solved in the thermodynamic limit, albeit with a growing number of equations to satisfy.

4.2. Last Layer Behaviour

Theorem 1 establishes the linear relation between optimal parameters independent of the number of qubits n. Using a global training strategy for the same problem with $p \geq 2$ depth circuits, it was numerically observed [27] that optimal parameters depend on the depth, yet usually can be approximately described by some linear relation. In the present work, we have observed that the last layer is distinctively characterised by the very same linear relation $\gamma_p + 2\beta_p = \pi$ stated in Theorem 1. We numerically confirmed this up to $p = 5$ layers and $n = 17$ qubits, as shown in Figure 1. The effect remains unexplained analytically and could be the manifestation of some hidden ansatz symmetry.

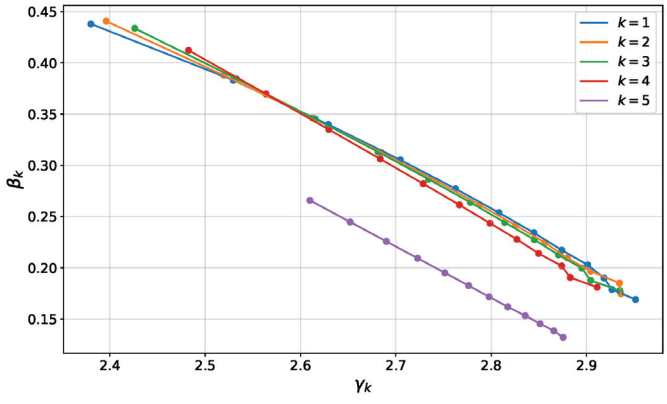


Figure 1. Optimal angles of $p = 5$ depth circuit for $n \in [6; 17]$. While the first layers can be approximately described by a linear relation, the last layer fits $\gamma_p + 2\beta_p = \pi$. Moreover, the values of the last layer's parameters are evidently distinct from the previous layers.

4.3. Saturation in Layerwise Training at $p = n$

It was demonstrated [28] that layerwise training *saturates*, meaning that past a critical depth p^*, overlap cannot be improved with further layer additions. Due to this effect, naive layerwise training performance falls below global training. Training saturation in layerwise optimisation was reported in [28] and confirmed up to $n = 10$ qubits. Most surprisingly, the saturation depth p^* was observed to be equal to the number of qubits n. Two effects remain unexplained analytically. Firstly does $p^* = n$. Secondly, could one go beyond the necessary conditions in [28] to explain saturations?

4.4. Removing Saturation in Layerwise Training

Any modification in the layerwise training process that violates the necessary saturation conditions can remove the system from its original saturation points. This idea was exploited in [28], where two types of variations were introduced for system sizes up to $n = 7$: (i) undertraining the QAOA circuit at each iteration and (ii) training in the presence of random coherent phase noise. Whereas both modifications (i) and (ii) removed saturations at $p = n$ yet the reason remains unexplained.

5. Conclusions

We have proven a relationship between optimal QAOA parameters for $p = 1$, and we recover optimal angles in the thermodynamic limit. We demonstrated the effect of

parameter concentrations for $p = 1$ QAOA circuits using an operator formalism. Compared to the explicit calculation where objective function gradients are set to zero, the operator approach exploits the ansatz symmetry in finding optimal parameters. The suggested approach can directly be adopted to find optimal parameters for the $p \geq 2$ QAOA circuit, with increasing complexity due to the larger number of parameters. Finally, we present a list of numerical effects, observed for particular system size and circuit depth, which are yet to be explained analytically. These unexplained effects include both limitations and advantages to QAOA. While difficult, adding missing theory to these subtle effects would improve our understanding of variational algorithms.

Author Contributions: Methodology, D.R., R.S., E.C., V.A. and J.B.; Writing—original draft, D.R., R.S., E.C., V.A. and J.B. All authors have read and agreed to the published version of the manuscript.

Funding: The authors acknowledge support from the research project Leading Research Center on Quantum Computing (agreement No. 014/20).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Harrigan, M.P.; Sung, K.J.; Neeley, M.; Satzinger, K.J.; Arute, F.; Arya, K.; Atalaya, J.; Bardin, J.C.; Barends, R.; Boixo, S.; et al. Quantum approximate optimization of non-planar graph problems on a planar superconducting processor. *Nat. Phys.* **2021**, *17*, 332–336. [CrossRef]
2. Pagano, G.; Bapat, A.; Becker, P.; Collins, K.; De, A.; Hess, P.; Kaplan, H.; Kyprianidis, A.; Tan, W.; Baldwin, C.; et al. Quantum approximate optimization of the long-range Ising model with a trapped-ion quantum simulator. *arXiv* **2019**, arXiv:1906.02700.
3. Guerreschi, G.G.; Matsuura, A.Y. QAOA for Max-Cut requires hundreds of qubits for quantum speed-up. *Sci. Rep.* **2019**, *9*, 6903. [CrossRef] [PubMed]
4. Butko, A.; Michelogiannakis, G.; Williams, S.; Iancu, C.; Donofrio, D.; Shalf, J.; Carter, J.; Siddiqi, I. Understanding quantum control processor capabilities and limitations through circuit characterization. In Proceedings of the 2020 International Conference on Rebooting Computing (ICRC), Atlanta, GA, USA, 1–3 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 66–75.
5. Biamonte, J. Universal variational quantum computation. *Phys. Rev. A* **2021**, *103*, L030401. [CrossRef]
6. Campos, E.; Nasrallah, A.; Biamonte, J. Abrupt transitions in variational quantum circuit training. *Phys. Rev. A* **2021**, *103*, 032607. [CrossRef]
7. Farhi, E.; Goldstone, J.; Gutmann, S. A Quantum Approximate Optimization Algorithm. *arXiv* **2014**, arXiv:1411.4028.
8. Niu, M.Y.; Lu, S.; Chuang, I.L. Optimizing qaoa: Success probability and runtime dependence on circuit depth. *arXiv* **2019**, arXiv:1905.12134.
9. Akshay, V.; Philathong, H.; Morales, M.E.; Biamonte, J.D. Reachability Deficits in Quantum Approximate Optimization. *Phys. Rev. Lett.* **2020**, *124*, 090504. [CrossRef]
10. Akshay, V.; Philathong, H.; Campos, E.; Rabinovich, D.; Zacharov, I.; Zhang, X.M.; Biamonte, J. On Circuit Depth Scaling For Quantum Approximate Optimization. *arXiv* **2022**, arXiv:2205.01698.
11. Zhou, L.; Wang, S.T.; Choi, S.; Pichler, H.; Lukin, M.D. Quantum Approximate Optimization Algorithm: Performance, Mechanism, and Implementation on Near-Term Devices. *Phys. Rev. X* **2020**, *10*, 021067. [CrossRef]
12. Brady, L.T.; Baldwin, C.L.; Bapat, A.; Kharkov, Y.; Gorshkov, A.V. Optimal Protocols in Quantum Annealing and Quantum Approximate Optimization Algorithm Problems. *Phys. Rev. Lett.* **2021**, *126*, 070505. [CrossRef] [PubMed]
13. Farhi, E.; Harrow, A.W. Quantum Supremacy through the Quantum Approximate Optimization Algorithm. *arXiv* **2016**, arXiv:1602.07674.
14. Farhi, E.; Goldstone, J.; Gutmann, S.; Zhou, L. The quantum approximate optimization algorithm and the sherrington-kirkpatrick model at infinite size. *arXiv* **2019**, arXiv:1910.08187.
15. Wauters, M.M.; Mbeng, G.B.; Santoro, G.E. Polynomial scaling of QAOA for ground-state preparation of the fully-connected p-spin ferromagnet. *arXiv* **2020**, arXiv:2003.07419.
16. Claes, J.; van Dam, W. Instance Independence of Single Layer Quantum Approximate Optimization Algorithm on Mixed-Spin Models at Infinite Size. *arXiv* **2021**, arXiv:2102.12043.
17. Wang, Z.; Rubin, N.C.; Dominy, J.M.; Rieffel, E.G. X Y mixers: Analytical and numerical results for the quantum alternating operator ansatz. *Phys. Rev. A* **2020**, *101*, 012320. [CrossRef]
18. Hodson, M.; Ruck, B.; Ong, H.; Garvin, D.; Dulman, S. Portfolio rebalancing experiments using the Quantum Alternating Operator Ansatz. *arXiv* **2019**, arXiv:1911.05296.

19. Tsoulos, I.G.; Stavrou, V.; Mastorakis, N.E.; Tsalikakis, D. GenConstraint: A programming tool for constraint optimization problems. *SoftwareX* **2019**, *10*, 100355. [CrossRef]
20. Lloyd, S. Quantum approximate optimization is computationally universal. *arXiv* **2018**, arXiv:1812.11075.
21. Morales, M.E.; Biamonte, J.; Zimborás, Z. On the universality of the quantum approximate optimization algorithm. *Quantum Inf. Process.* **2020**, *19*, 1–26. [CrossRef]
22. Rabinovich, D.; Adhikary, S.; Campos, E.; Akshay, V.; Anikin, E.; Sengupta, R.; Lakhmanskaya, O.; Lakhmanskiy, K.; Biamonte, J. Ion native variational ansatz for quantum approximate optimization. *arXiv* **2022**, arXiv:2206.11908.
23. Jiang, Z.; Rieffel, E.G.; Wang, Z. Near-optimal quantum circuit for Grover's unstructured search using a transverse field. *Phys. Rev. A* **2017**, *95*, 062317. [CrossRef]
24. Hastings, M.B. Classical and quantum bounded depth approximation algorithms. *arXiv* **2019**, arXiv:1905.07047.
25. Bravyi, S.; Kliesch, A.; Koenig, R.; Tang, E. Obstacles to State Preparation and Variational Optimization from Symmetry Protection. *Phys. Rev. Lett.* **2019**, *125*, 260505. [CrossRef]
26. Akshay, V.; Philathong, H.; Zacharov, I.; Biamonte, J. Reachability Deficits in Quantum Approximate Optimization of Graph Problems. *Quantum* **2021**, *5*, 532. [CrossRef]
27. Akshay, V.; Rabinovich, D.; Campos, E.; Biamonte, J. Parameter concentrations in quantum approximate optimization. *Phys. Rev. A* **2021**, *104*, L010401. [CrossRef]
28. Campos, E.; Rabinovich, D.; Akshay, V.; Biamonte, J. Training saturation in layerwise quantum approximate optimization. *Phys. Rev. A* **2021**, *104*, L030401. [CrossRef]
29. Shaydulin, R.; Wild, S.M. Exploiting symmetry reduces the cost of training QAOA. *IEEE Trans. Quantum Eng.* **2021**, *2*, 1–9. [CrossRef]
30. Streif, M.; Leib, M. Comparison of QAOA with quantum and simulated annealing. *arXiv* **2019**, arXiv:1901.01903.
31. Skolik, A.; McClean, J.R.; Mohseni, M.; van der Smagt, P.; Leib, M. Layerwise learning for quantum neural networks. *Quantum Mach. Intell.* **2021**, *3*, 1–11. [CrossRef]

mathematics

MDPI

Article

A Fast Quantum Image Component Labeling Algorithm

Yan Li [1,*], Dapeng Hao [2], Yang Xu [3] and Kinkeung Lai [4]

[1] School of Business Administration, Xi'an Eurasia University, Xi'an 710065, China
[2] School of Science, Xi'an Aeronautical University, Xi'an 710077, China; haodapeng@xaau.edu.cn
[3] School of Economics and Management, Xi'an Technological University, Xi'an 710021, China; xuyang@chd.eud.cn
[4] Department of Industrial and Manufacturing Systems Engineering, University of Hong Kong, Hong Kong 999077, China; mskklai@outlook.com
* Correspondence: happy3361@163.com

Abstract: Component Labeling, as a fundamental preprocessing task in image understanding and pattern recognition, is an indispensable task in digital image processing. It has been proved that it is one of the most time-consuming tasks within pattern recognition. In this paper, a fast quantum image component labeling algorithm is proposed, which is the quantum counterpart of classical local-operator technique. A binary image is represented by the modified novel enhanced quantum image representation (NEQR) and a quantum parallel-shrink operator and quantum propagate operator are executed in succession, to finally obtain the component label. The time complexity of the proposed quantum image component labeling algorithm is $O(n^2)$, and the spatial complexity of the quantum circuits designed is $O(cn)$. Simulation verifies the correctness of results.

Keywords: quantum image processing; image component labeling; local operator; Levialdi shrinking operator

MSC: 81P68

Citation: Li, Y.; Hao, D.; Xu, Y.; Lai, K. A Fast Quantum Image Component Labeling Algorithm. *Mathematics* **2022**, *10*, 2718. https://doi.org/10.3390/math10152718

Academic Editor: Jan Sładkowski

Received: 9 June 2022
Accepted: 28 July 2022
Published: 1 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

Image processing involves certain operations that help improve aesthetics and enhance comprehensibility of what the image conveys. This is widely used in environment, agriculture, military, industry, and medical sciences to extract valuable information. Due to the rapid development of information technology, data handled in image processing have undergone exponential growth. Usage of classical image processing has declined and therefore quantum image processing has emerged as a feasible way to solve the problems. The coherent superposition characteristics of the quantum state and other unique quantum mechanical principles are used for generating data processing capability in quantum image processing, which can accelerate the process significantly compared to classical algorithms.

In the last two decades or so, a large number of productive techniques have emerged for quantum image processing, which serves two purposes. The first is to construct models for representation of the digital image mainly including qubit lattice [1], entangled image [2], flexible representation of the quantum image (FRQI) [3], quantum log-polar image [4], a novel enhanced quantum representation of digital images (NEQR) [5], and Quantum Boolean image processing [6], a simple quantum representation of infrared images (SQR) [7] and some extensions from FRQI or NEQR [8–16]. The other is applications based on the above which vary with types of representation, such as geometric transform [17–19], image scaling [20–23], image scrambling [24,25], image segmentation [26–29], image edge extraction [30–35], image matching [36–38], image watermarking [39,40], and so on.

Although many issues are studied by researchers, as mentioned above, quantum image processing is still an emerging field and, compared with classical image processing, it is still in its infancy. To the best of our knowledge, image component labeling has not yet

been extended to the quantum imaging processing domain. Image component labeling is the most fundamental preprocessing required for image understanding, pattern recognition, and computer vision. By use of the labeling operation, a unique label is assigned to each connected region so that higher-level operations can process different regions separately. In some applications, image component labeling is still an active area of research in classical image processing, which has been proved to be one of the most time-consuming tasks in pattern recognition [41]. The parallel processing characteristic is the great advantage of quantum computation, which is one feasible way to accelerate image component labeling.

This paper proposes a fast quantum image component labeling algorithm based on a modified NEQR representation model for binary images, a quantum counterpart of classical local-operator techniques [42]. The quantum image component labeling algorithm consists of three main steps. Firstly, a binary image is represented by the quantum version using the modified NEQR model. Secondly, all pixels of the image are simultaneously worked upon by the quantum parallel-shrink operator several times until each black pixel changes to white, and the connectivity relations are reserved during the processing. Finally, the quantum label-propagate operator is executed on each pixel to restore the pixels with changed colors and each pixel assigns different numbers to different connected areas at the same time. The process is in reverse order to the image generated by the quantum parallel-shrink operations.

The rest of the paper is organized as follows. Section 2 briefly introduces classical local-operator techniques, giving a detailed example specifying how to operate the binary image to obtain the labels for the connected area. The purposed quantum version of the local-operator technique, as well as circuit design, is described in Section 3. Section 4 analyzes the circuit complexity. Simulation results based on the classical computer's Python software are given in Section 5. Finally, the conclusions are drawn in Section 6.

2. Local-Operator Technique

A connected region or component in a binary image is a maximal connected set of black pixels. The image component labeling algorithm assigns a unique label to each connected region in the image. Thus, in the labeled image, any two black pixels have the same label if and only if they lie in the same connected region. Local-operator techniques involve two types of local operations used for image region labeling: Parallel-shrink and Label-propagate. The two operators use local information from the neighborhood of a pixel to determine its new value.

2.1. Basic Definitions

The following basic definitions constitute some of the concepts required for Local-operator techniques [42]. Assume black pixels have value 1 and white pixels have value 0. Let $\boldsymbol{a} \in \{0,1\}^X$ denote the source binary image on the point set $\boldsymbol{X} = \{0 \le i \le n, 0 \le j \le m\}$ with $a(i,j)$ being the value at pixel $p(i,j)$. Two pixels are said to be *neighbors* if they share one edge, one vertex, or both. The pixels are chosen to be squares, and then a pixel may have either 4 or 8 neighbors in terms of edges or both edges and vertices. Two pixels $p(i_0, j_0)$ and $p(i_1, j_1)$ are called 4-neighbors if $|i_0 - i_1| + |j_0 - j_1| = 1$ and 8-neighbors if $\max\{|i_0 - i_1|, |j_0 - j_1|\} \le 1$. Let $\boldsymbol{C}$ be the *connectivity relation* defined on an image as follows: for all pairs of pixels, $p, q \in \boldsymbol{a}$ if and only if p and q are both black and are connected by a path in $\boldsymbol{a}$. The *internal distance* between two black pixels is defined as the length of a shortest 4- or 8-neighbor path connecting them within the component. The *internal diameter* of a connected component is defined as the maximum of lengths of all internal distances among all pairs of pixels within the component. Local-operator techniques use a part of 8-neighbor components of a given $n \times m$ binary image. The parallel-shrink operator uses neighborhood N_s shown in Figure 1 which is the set of points $\{p(i,j), p(i,j+1), p(i+1,j), p(i+1,j+1)\}$, and the label-propagate operator uses neighborhood N_p (Figure 1), which is the set of points $\{p(i,j), p(i,j-1), p(i-1,j), p(i-1,j-1)\}$.

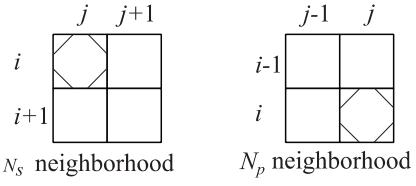


Figure 1. Neighborhoods.

2.2. The Parallel-Shrink Operator

The parallel-shrink operator φ_s was first designed by Levialdi [43] and it used the Heaviside operator H defined on N_s neighborhoods, such that, if $\boldsymbol{a}' = \varphi_s(\boldsymbol{a})$, then:

$$\boldsymbol{a}'(i,j) = H(H(\boldsymbol{a}(i,j) + \boldsymbol{a}(i,j+1) + \boldsymbol{a}(i+1,j) - 1) + H(\boldsymbol{a}(i,j) + \boldsymbol{a}(i+1,j+1))) \quad (1)$$

where H is the Heaviside operator defined by $H(t) = 0$ for $t \leq 0$ and $H(t) = 1$ if $t > 0$. Since $\boldsymbol{a}$ is a binary image, it is easy to express φ_s using the logic operations $\wedge(and)$ and $\vee(or)$ as follows:

$$\boldsymbol{a}'(i,j) = (\boldsymbol{a}(i,j) \wedge (\boldsymbol{a}(i,j+1) \vee \boldsymbol{a}(i+1,j) \vee \boldsymbol{a}(i+1,j+1))) \vee (\boldsymbol{a}(i,j+1) \wedge \boldsymbol{a}(i+1,j)) \quad (2)$$

The parallel-shrink operator shrinks the components toward the top left corner of the bounding rectangles of connected components. The two important properties of the shrinking procedure are as follows:

1. No connected component becomes disconnected.
2. No two disconnected components become connected in any step.

A component with an internal diameter r will shrink to a single black pixel after $r-1$ shrinking steps, and then disappear in the next shrinking step. After each shrinking operation, a different image is obtained and the result of applying the shrinking operation y times to the original image is called *partial result* y. Thus, we have a sequence of images $\boldsymbol{a}_y, \boldsymbol{a}_{y-1}, \cdots, \boldsymbol{a}_0$, $\boldsymbol{a}_0$ representing the initial binary image.

2.3. The Label-Propagate Operator

The label-propagate operators are applied in the reverse order to the images generated by the parallel-shrink operations. Suppose at a certain time of the label-propagate operator, black pixels of $\boldsymbol{a}_r$ are labeled with the correct labels and the labels can be used to label the black pixels of $\boldsymbol{a}_{r-1}$. The label-propagate operator then continues to label the black pixels until the image returns to the initial image $\boldsymbol{a}_0$.

Let l be a global variable that saves the maximum label in the image, the initial number is 0, and $l_r(i,j)$ is the label of pixel $p(i,j)$ in $\boldsymbol{a}_r$. The parallel-shrink operator φ_p is defined on N_p neighborhoods, such that, if $l_r = \varphi_p(l_{r+1})$, then:

$$l_{r+1} = \begin{cases} l\max_{r+1}(i,j) & \text{if } a_r(i,j) = 1 \text{ and } l\max_{r+1}(i,j) \neq 0 \\ l+1 & \text{if } a_r(i,j) = 1 \text{ and } l\max_{r+1}(i,j) = 0 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where $l\max_{r+1} = l_{r+1}(i,j) \vee l_{r+1}(i,j-1) \vee l_{r+1}(i-1,j) \vee l_{r+1}(i-1,j) \vee l_{r+1}(i-1,j-1)$, $\vee$ is a bit-wise logic or operation.

Note that, first, we must ensure that labels of the four neighbors for whom the value is not zero on N_p remain the same after Equation (3) is executed, then:

$$\begin{cases} l_r(i,j-1) = l_r(i,j) & \text{if } \boldsymbol{a}_r(i,j-1) = 1 \\ l_r(i-1,j) = l_r(i,j) & \text{if } \boldsymbol{a}_r(i-1,j) = 1 \\ l_r(i-1,j-1) = l_r(i,j) & \text{if } \boldsymbol{a}_r(i-1,j-1) = 1 \end{cases} \quad (4)$$

Second, we should backfill the value 1 of pixels based on *partial result y* during propagating.

2.4. A Simple Example of the Local-Operator Technique

In this subsection, we give a simple example to explain how the label-propagate operator functions. The binary image consists of two connected regions or components as shown in Figure 2a, and the maximum internal diameter of two connected regions is 3. The detailed process is described below.

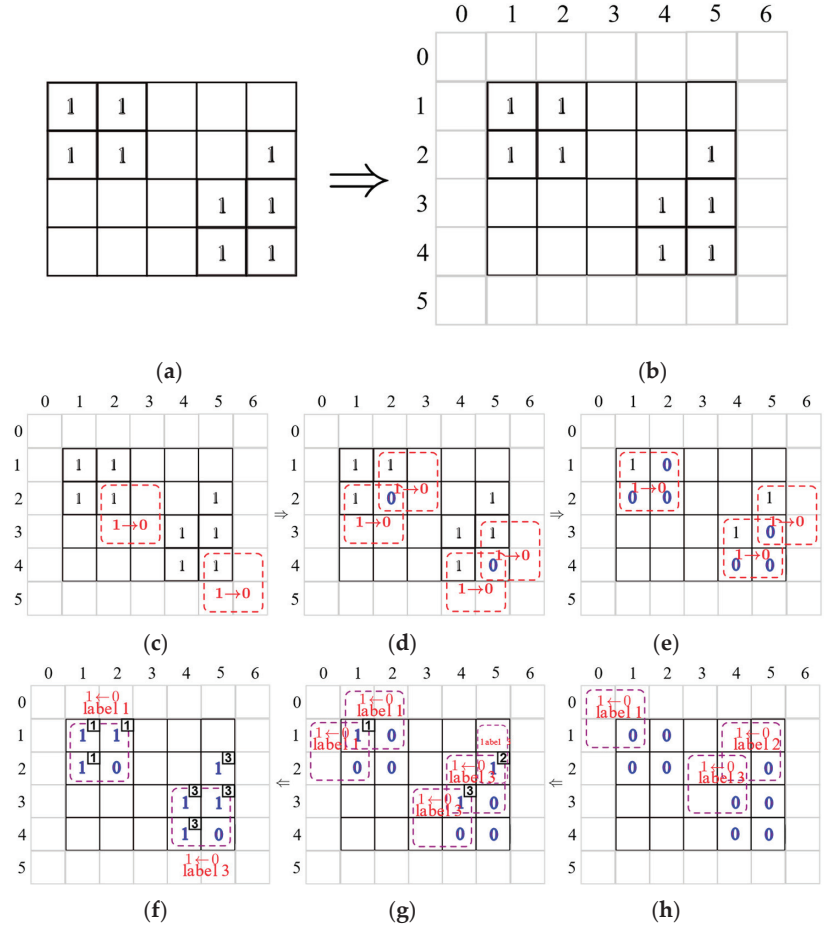


Figure 2. The simple example of local-operator techniques. (**a**) original image. (**b**) adding border. (**c**) partial result 1. (**d**) partial result 2. (**e**) partial result 3. (**f**) propagate 3. (**g**) propagate 2. (**h**) propagate 1.

Phase 1 Preparation: When every pixel has its N_s and N_p neighbors, the original image is extended by a 1-pixel outer border that pads the border of the image with white. The white pixels (0 value) are ignored in the board so that we can focus better on changes in the black regions (1 value), as shown in Figure 2b.

Phase 2 Parallel shrinking: In Phase 2, the parallel-shrink operator, (1) or (2), is performed on every pixel simultaneously. Since the maximum value of the *internal diameter* of the two connected regions is 3, the operator would be applied three times, and then all pixels of the image will change to 0 and that completes Phase 2. Figure 2c–e depict the

performance process. In Figure 2c–e, we only show the pixels that were changed using the red dotted box.

Phase 3 Label propagating: As mentioned before, label propagating is a reverse-order process, so Equations (3) and (4) are applied for the same number of times as Phase 2. The process order is indicated in Figure 2f–h using the left arrow, and the purple dotted boxes are used for marking the label change. In Figure 2h, $p(1,1)$, $p(2,5)$, and $p(3,4)$ are labeled with different numbers based on (3) at the same time. We use the row–column criterion to label the pixels. In Figure 2f, serial numbers of the pixels that have been labeled are represented in the upper left corner box, $p(1,2)$, $p(2,1)$, $p(3,5)$, $p(4,4)$ need new labels based on Equation (3), and $p(2,5)$ needs a change of the labeled number based on Equation (4) to keep consistent with $p(3,5)$'s N_p neighbors. In Figure 2f, the last two pixels, $p(2,2)$ and $p(4,5)$, are labeled, which means that label propagating is finished, and two different label numbers are obtained.

The example shows that the local-operator technique is a parallel method that is performed simultaneously on every pixel, so it only takes $O(n)$ time and is a type of fast image component labeling method.

3. Quantum Version of the Local-Operator Technique

In this section, a series of specific quantum circuits are designed to realize the local-operator technique.

3.1. Modified NEQR Model Representation of a Binary Image

The NEQR is a deterministic image retrieval model that facilitates the operations on an image, so we modified the model to represent the quantum image as required [5]. In the modified model, one qubit sequence is employed for storing the position information on the Cartesian coordinate system and another sequence represents the information required, including color, number of partial results, and label number. Two entangled qubit sequences are in superposition states.

The binary image with size $2^n \times 2^m$ can be represented by modified NEQR as the following equation:

$$\begin{aligned} |I\rangle &= \frac{1}{\sqrt{2^{m+n}}} \sum_{Y=0}^{2^m-1} \sum_{X=0}^{2^n-1} |f(Y,X)\rangle \otimes |YX\rangle \\ &= \frac{1}{\sqrt{2^{m+n}}} \sum_{Y=0}^{2^m-1} \sum_{X=0}^{2^n-1} |LPB\rangle \otimes |YX\rangle \end{aligned} \tag{5}$$

$$\begin{aligned} |YX\rangle &= |Y\rangle|X\rangle = |Y_m Y_{m-1} \cdots Y_0\rangle |X_n X_{n-1} \cdots X_0\rangle \\ |LPB\rangle &= |L\rangle|P\rangle|B\rangle = \left|L_j L_{j-1} \cdots L_0\right\rangle |P_i P_{i-1} \cdots P_0\rangle |B_0\rangle \end{aligned} \tag{6}$$

where $|YX\rangle$ address qubits represent the position information, X is row number, Y is column number, $|LPB\rangle$ work qubits represent the computation information, B is color, P is number of partial results, L is label number, and Y_i, X_i, L_i, P_i, $B_0 \in \{0,1\}$.

A quantum circuit equivalent to Equation (5) can be used to prepare the initial image state. Figure 3 is an example that presents the quantum circuit of Figure 2b, the address qubits are entangled using six Hadamard gates, and the information circuit of two connected regions are in dotted boxes.

3.2. Basic Quantum Functional Circuits

To realize the complex quantum circuit, a series of quantum functional operation modules are prepared for local-operator, including address shift operation module, logic operation module, assignment operation module, compare operation, full addition module, and full subtraction module.

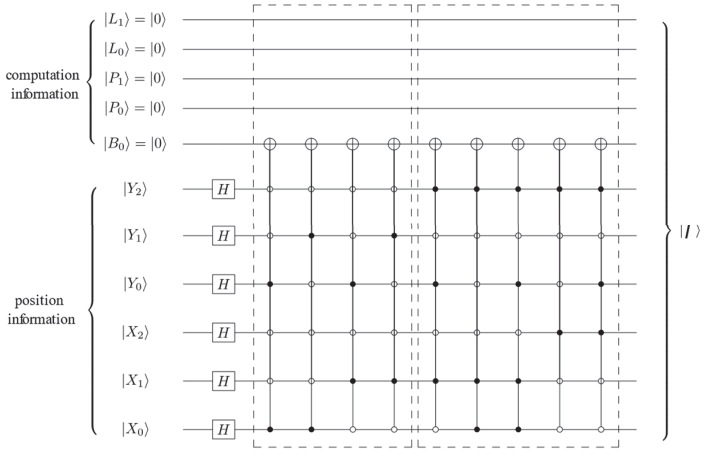


Figure 3. The quantum circuit of Figure 2b.

3.2.1. Address Shift Operation Circuits

It is an important operation for a large number of quantum algorithms to obtain the neighborhood information [34,35]. Shift transformation is a geometric operation that can be used to shift the whole image and it skillfully helps us to obtain the information of every pixel's adjacent information simultaneously. For example, make a one-unit shift right for an image, the pixel $p(x, y)$ will be transformed to $p(x + 1, y)$ and the value $a(x - 1, y)$ can be visited. Shift $-$ and $+$ are defined in Equations (7) and (8). When it is applied on x, the image moves left and right, and when it is applied on y, the image moves up and down.

$$\begin{aligned} S_{x\pm}(|I\rangle) &= \tfrac{1}{\sqrt{2^{m+n}}}\textstyle\sum_{Y=1}^{2^m-1}\sum_{X=1}^{2^n-1}|f(Y,X)\rangle\otimes S_{x\pm}(|YX\rangle) \\ &= \tfrac{1}{\sqrt{2^{m+n}}}\textstyle\sum_{Y=1}^{2^m-1}\sum_{X=1}^{2^n-1}|f(Y,X)\rangle\otimes|Y\rangle|(X\pm 1)\mathrm{mod}2^n\rangle \end{aligned} \tag{7}$$

$$\begin{aligned} S_{y\pm}(|I\rangle) &= \tfrac{1}{\sqrt{2^{m+n}}}\textstyle\sum_{Y=1}^{2^m-1}\sum_{X=1}^{2^n-1}|f(Y,X)\rangle\otimes S_{y\pm}(|YX\rangle) \\ &= \tfrac{1}{\sqrt{2^{m+n}}}\textstyle\sum_{Y=1}^{2^m-1}\sum_{X=1}^{2^n-1}|f(Y,X)\rangle\otimes|(Y\pm 1)\mathrm{mod}2^n\rangle|X\rangle \end{aligned} \tag{8}$$

Based on the above analysis, Equations (7) and (8) are equivalent to Equations (9) and (10), respectively.

$$S_{x\pm}(|I\rangle) = \frac{1}{\sqrt{2^{m+n}}}\sum_{Y=1}^{2^m-1}\sum_{X=1}^{2^n-1}|f(Y,X')\rangle\otimes S_{x\pm}(|YX\rangle) \tag{9}$$

$$S_{y\pm}(|I\rangle) = \frac{1}{\sqrt{2^{m+n}}}\sum_{Y=1}^{2^m-1}\sum_{X=1}^{2^n-1}|f(Y',X)\rangle\otimes S_{y\pm}(|YX\rangle) \tag{10}$$

where $X' = (X \mp 1)\mathrm{mod}2^n$ and $Y' = (Y \mp 1)\mathrm{mod}2^n$.

For a two-dimensional digital image, Figure 4 is an example for $S_{x\pm}$, and $S_{y\pm}$ is similar to $S_{x\pm}$**.** We call the shift transformation circuit an address shift operation circuit just like the address-of operator in classical programming.

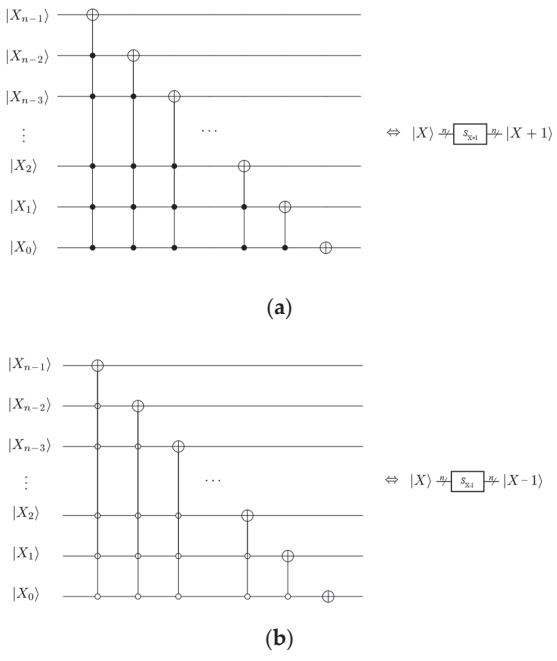


Figure 4. Address shift operation circuit. (**a**) Shift + operator circuit. (**b**) Shift − operator circuit.

3.2.2. Logic Operation Circuit

In reversible computing, the TOFFOLI gate is universal gate, and it can be used for simulating classical irreversible standard gates with ancilla qubits. Figure 5 is the circuit of the necessary logic operations for computation of Equations (2) and (3) for convenience, and the inputs and results are connected by black blocks. The following figures are used for the same notation.

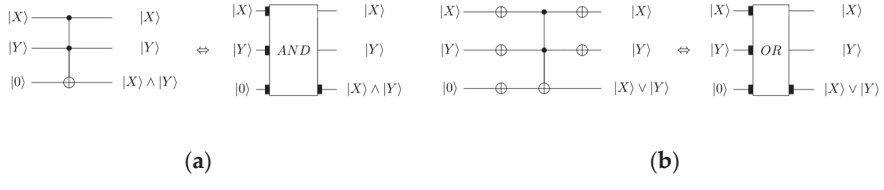


Figure 5. Logic operation circuit. (**a**) Logic operation ∧ (AND). (**b**) Logic operation ∨ (OR).

3.2.3. Control Assignment Operation Circuit

Assignment operation is the most basic statement in classical programming. To achieve the function, we design the control assignment operation circuit (Figure 6). The circuit consists of two parts, the first part is shown in a dotted block that is to clear the original data of the quantum wires $|X\rangle$ by using a group of swap gates, the original data are saved in the ancilla qubits. The second part assigns the value of $|Y\rangle$ to $|X\rangle$ using a group of CNOT gates. The whole circuits are controlled by using a CNOT gate, such that, if the control wire is 1, then assignment operation is executed. As described, the designed circuit is a reversible circuit, and, by using ancilla qubits, both $|X\rangle$ and $|Y\rangle$ are stored during the operation.

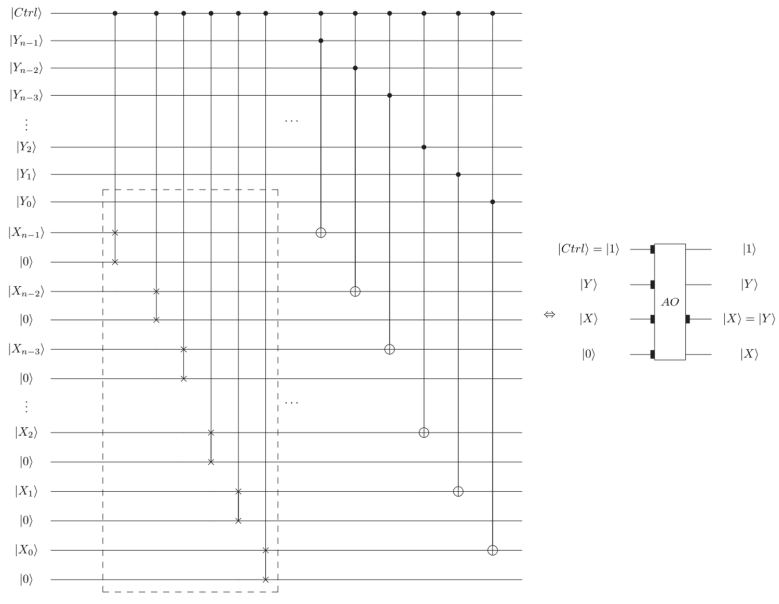


Figure 6. Assignment operation circuit.

3.2.4. Compare Operation Circuit

The compare operation circuit can be used to compare two n-qubits sequences $|X\rangle$ and $|Y\rangle$ illustrated in Figure 7 [32], where $|X\rangle = |X_{n-1}X_{n-2}\cdots X_1X_0\rangle$, $|Y\rangle = |Y_{n-1}Y_{n-2}\cdots Y_1Y_0\rangle$, $X_i, Y_i \in \{0,1\}$. The output two-qubit $|e_1e_0\rangle$ can be used to represent the result of comparison: if $e_1e_0 = 10$, then $X > Y$; if $e_1e_0 = 01$, then $X < Y$; if $e_1e_0 = 00$, then $X = Y$.

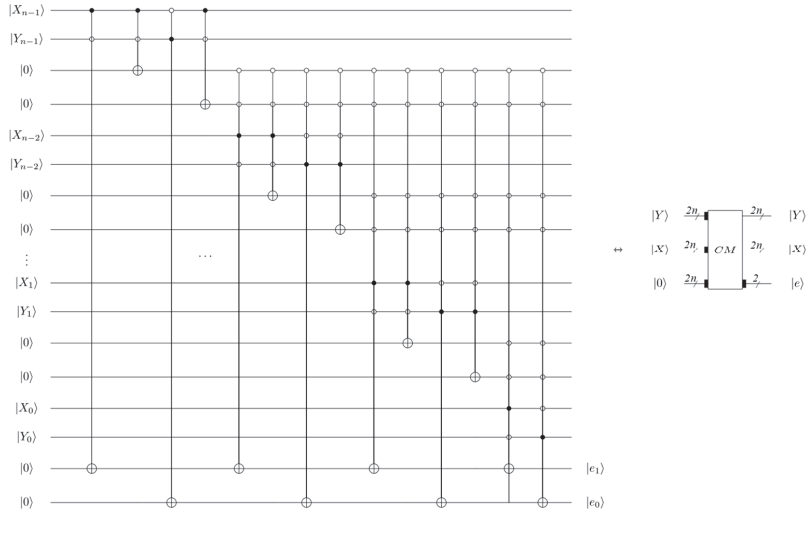


Figure 7. Compare operation circuit.

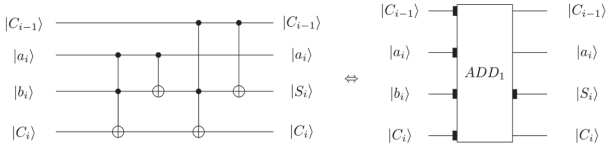
3.2.5. Full Addition Circuit

Full adder is an adder with carry, a common arithmetic unit in various algorithms. In this paper, 1-bit full addition circuit is composed of quantum CNOT and CCNOT gate, and

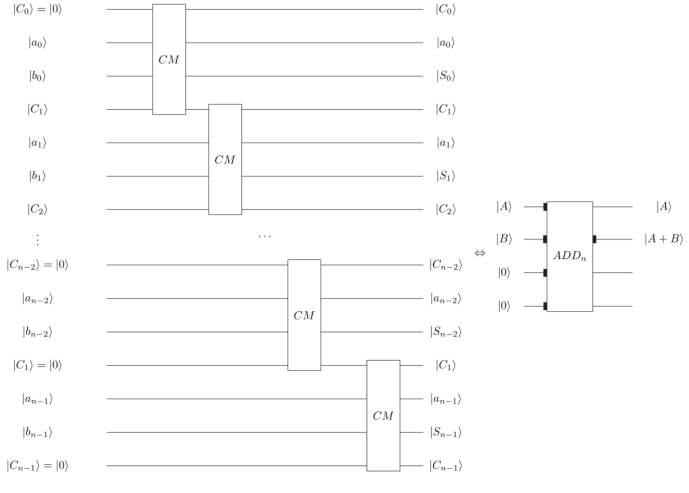
n-bit full addition circuit is a cascade of 1-bit full addition circuit as shown in Figure 8 [44]. $|C_{i-1}\rangle$ is the $(i-1)$th carry qubit, $|a_i\rangle$ is the ith augend number, $|b_i\rangle$ is the ith addend number, $|C_i\rangle$ is the ith carry qubit, and $|S_i\rangle$ is the sum of $a+b$. The relationships among $|a_i\rangle$, $|b_i\rangle$, $|C_{i-1}\rangle$, $|C_i\rangle$ and $|S_i\rangle$ are given by

$$|S_i\rangle = |a_i \oplus b_i \oplus C_{i-1}\rangle \tag{11}$$

$$|C_i\rangle = |a_i b_i + (a_i \oplus b_i)C_{i-1}\rangle = |a_i b_i \oplus (a_i \oplus b_i)C_{i-1}\rangle \tag{12}$$



(**a**)



(**b**)

Figure 8. Full addition circuit. (**a**) 1-bit full addition circuit. (**b**) n-bit full addition circuit.

3.2.6. Full Subtraction Circuit

The full subtractor circuit is similar to full addition circuit, the n-bit full subtraction circuit is a cascade of the 1-bit full subtraction circuit, as shown in Figure 9 [44]. $|B_{i-1}\rangle$ is the $(i-1)$th borrow qubit, $|a_i\rangle$ is the ith minuend number, $|b_i\rangle$ is the ith subtrahend number, $|B_i\rangle$ is the ith borrow qubit, and $|D_i\rangle$ is the ith qubit of difference $a-b$. The relationships among $|a_i\rangle$, $|b_i\rangle$, $|B_{i-1}\rangle$, $|B_i\rangle$, and $|D_i\rangle$ are given by

$$|D_i\rangle = |a_i \oplus b_i \oplus B_{i-1}\rangle \tag{13}$$

$$|B_i\rangle = |b_i B_{i-1} + \overline{a}_i(b_i + B_{i-1})\rangle = |b_i B_{i-1} \oplus \overline{a}_i(b_i \oplus B_{i-1})\rangle \tag{14}$$

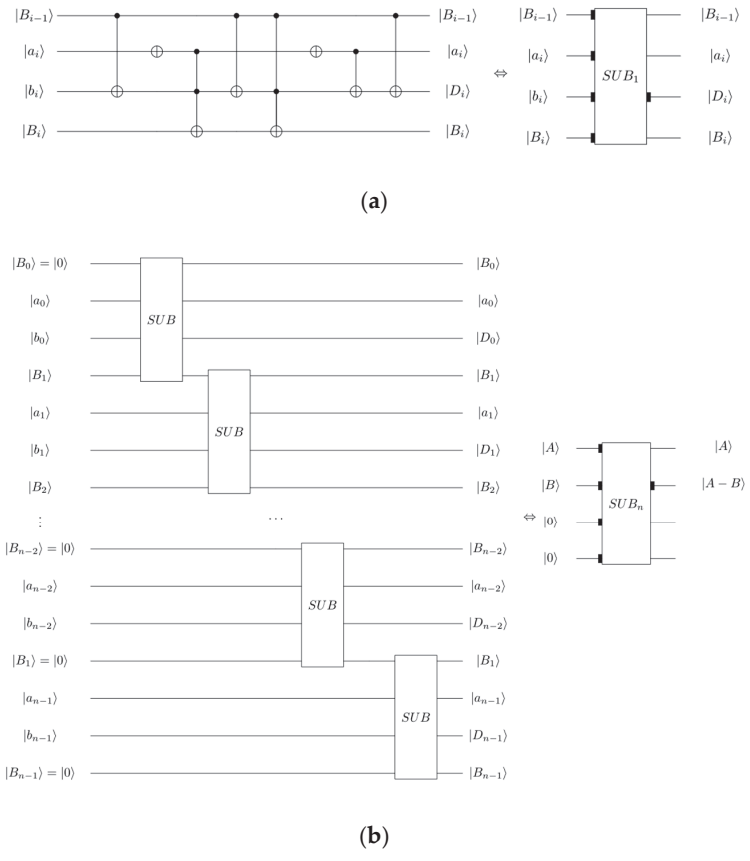


Figure 9. Full subtractor circuit. (**a**) 1-bit full subtractor circuit. (**b**) n-bit full subtractor circuit.

3.3. Implementing Quantum Image Component Labeling

The entire workflow of the proposed procedure can be accomplished in four steps. First, the classical color digital image is converted into binary format and a 1-pixel outer border as described in Section 2.4 is added. Next, the quantum image $|I\rangle$ is prepared as the input image using a modified NEQR model. The quantum circuit is similar to Figure 3. Then, the quantum counterpart of parallel shrinking is executed repeatedly until color value of all pixels is zero in the image. Finally, the quantum counterpart of label propagating is executed repeatedly for labeling regions or components in the image, and the number of executions is the same as in Step 3. Similar to its classical counterpart, the most important steps of quantum image component labeling are quantum parallel shrinking and quantum label propagating.

3.3.1. Quantum Parallel Shrinking

Quantum parallel shrinking is to repeat the following three steps until all pixels in the image are zero where pixel zero is the background color.

In the first step, neighborhood $\boldsymbol{N}_s$ is obtained through address shift operation circuits, the designed circuit is shown in Figure 10, and the color information is saved in the ancilla qubits. Note that the state of $|I\rangle$ should return to the initial value after using the address shift operation circuit four times, for later use.

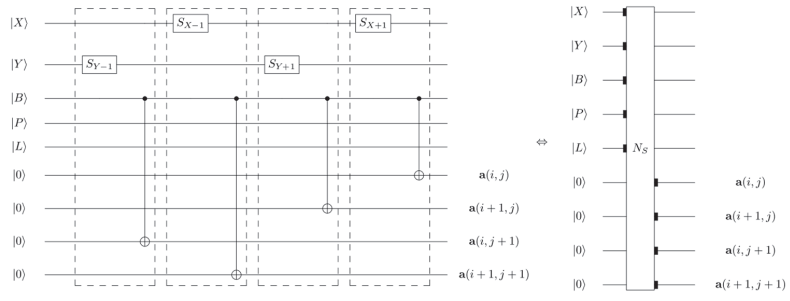


Figure 10. Obtaining of information of N_S.

The second step, the Levialdi operator, is implemented through Logic operation circuits, and the designed circuit is shown in Figure 11. The number of ancilla qubits used is the same as the number of Logic operators.

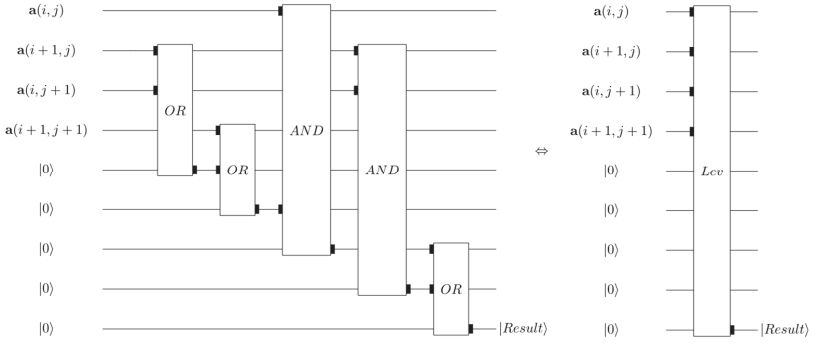


Figure 11. Execution of the Levialdi operator.

The third step, the number of *partial results*, is stored in $|P\rangle$ through the full addition circuit, and the designed circuit is shown in Part 3 of Figure 12. If $|result\rangle = |0\rangle$ and $|B\rangle = |1\rangle$, then $P = P + 1$. Fourthly, the value of pixels $|B\rangle$ is updated through CNOT gate (Part 4 of Figure 12). Finally, the number of loops $i = i + 1$ through the full addition circuit is shown in Part 5 of Figure 12. The circuit of Equation (2) is shown in Figure 12.

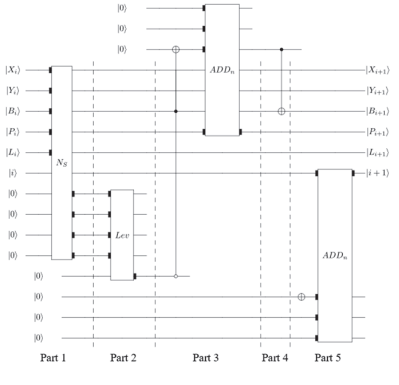


Figure 12. The circuit of Equation (2).

3.3.2. Quantum Label Propagating

Quantum label propagating is to repeat the following four steps until the manipulated image becomes the original image.

The first step, neighborhood N_p, is obtained through address shift operation circuits, the designed circuit is shown in Figure 13, and label numbers $|L_i\rangle$ are saved in the ancilla qubits.

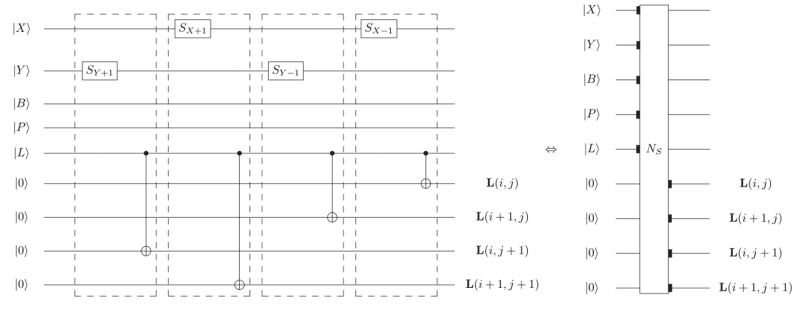


Figure 13. Obtaining of information of N_p.

The second step, the computation of $l$max, is implemented through Logic operation circuits, and the designed circuit is shown in Figure 14.

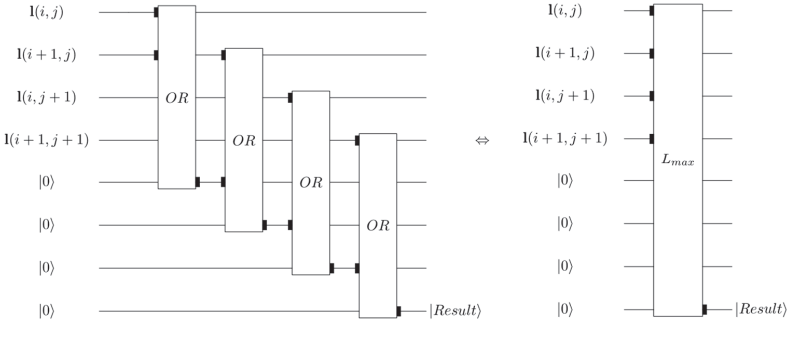


Figure 14. The computation of $l$max.

In the third step, the global variable i $-$ 1 and Equation (3) are implemented through the designed circuit shown in Figure 15. The condition of Equation (3) is realized in part 1, $|P_i\rangle$, and $|i\rangle$ as input to comparer CM1, and $P_i = 1$ means that the color of the pixel changes from 1 to 0, so $P_{i-1} = 1$ is equivalent to $a(i,j) = 1$. Comparer CM2 is used to determine if $l\text{max}_i = 0$. If $P_{i-1} = i$ and $l\text{max}_i = 0$, part 2 is executed, the global variable $l = l + 1$, then assign l to $l_{i-1}(i,j)$ using control assignment operator AO. If $P_{i-1} = i$ and $l\text{max}_i \neq 0$, part 3 is executed, and the control assignment operator AO is used to implement $l_{i-1}(i,j) = l\text{max}_i$.

In the fourth step, Equation (4) is implemented through the designed circuit shown in Figure 16. $l_{i-1}(i,j)$ is stored in ancilla qubits using CNOT gate in part 1. In part 2, $l_{i-1}(i,j-1)$ is obtained using address shift operator S_{Y-1}, the condition of $a_{i-1}(i,j-1)$ is realized using comparer CM, and then equation $l_r(i,j-1) = l_r(i,j)$ is implemented using control assignment operator AO. Part 3 and part 4 use the same circuits as part 2 to achieve the same assignment function. In part 5, the initial state is restored.

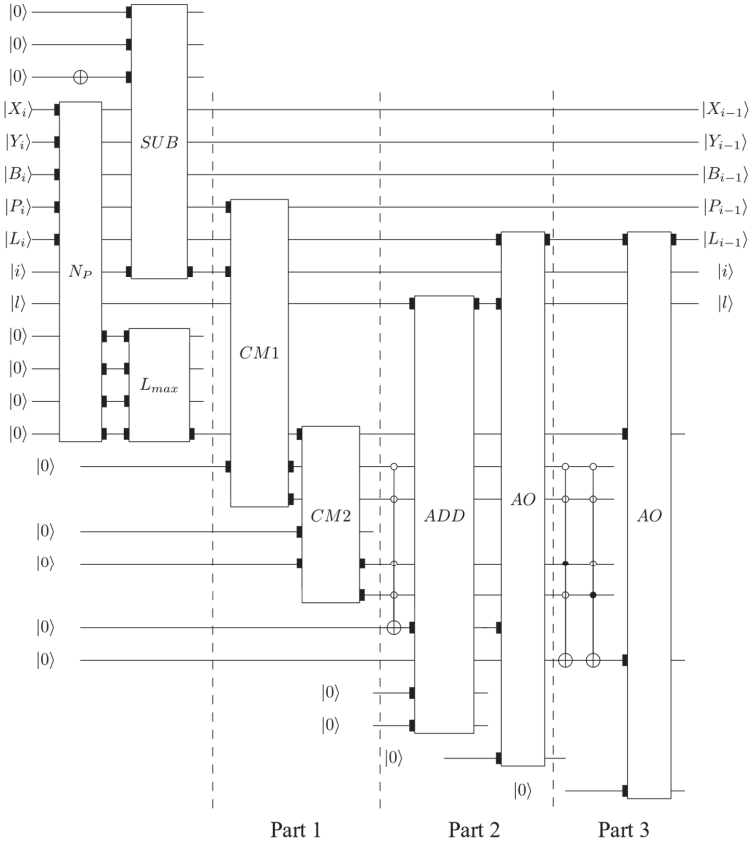


Figure 15. The quantum circuit of Equation (3).

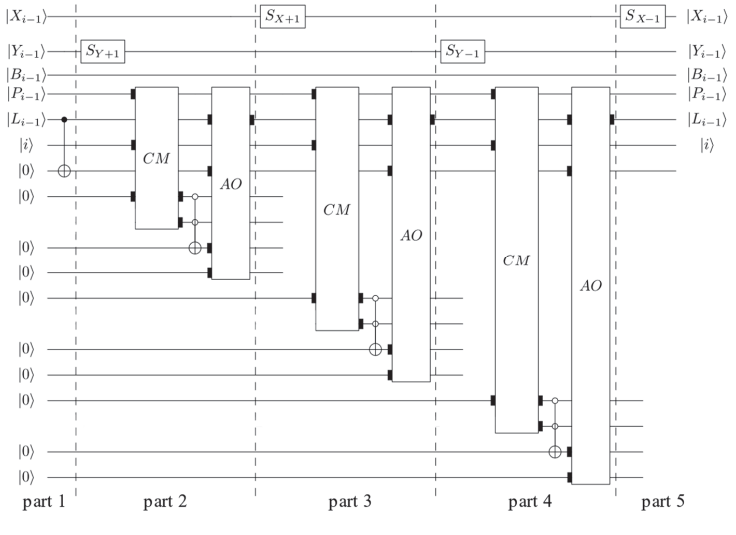


Figure 16. The quantum circuit of Equation (4).

4. Quantum Circuit Complexity Analysis

Section 2 introduced the classical local-operator technique, label propagating, and parallel shrinking executed the same number of times, depending on the Manhattan diameter of the largest component, so its time complexity is just $O(n)$. Thanks to Levialdi's pioneering work, the classical local-operator technique is much faster than scan type algorithms $O(n^2)$ [41].

In this section, we focus on the quantum circuit complexity. The time complexity depends on the number of elementary gates used. The elementary gates include the NOT gate, Hadamard gate, CNOT gate, and 2×2 unitary operator. The time complexity of elementary gates is 1. The spatial complexity mainly refers to the ancilla qubits employed in the circuits.

4.1. Time Complexity

From [45], one Toffoli gate can be further approximately simulated by six CNOT gates and [46] points out that an n-controlled NOT (n-CNOT) gate is equivalent to $2(n-1)$ Toffoli gates and 1 CNOT gate. One SWAP gate is equivalent to three CNOT gates.

Section 3.2 introduced six basic quantum functional circuits, and label propagating and parallel shrinking circuits are composed of the functional circuits. One address shift operation circuit costs the time complexity of $O(n^2)$. Logic operation $\wedge$ is one Toffoli gate, and the time complexity is $O(6)$. Logic operation $\vee$ needs five NOT gates and one Toffoli gate, and the time complexity is $O(11)$. Control Assignment Operation needs $2n$ SWAP gates and n Toffoli gates, and the time complexity is $O(6n + 6n) = O(12n)$. Compare Operation needs 4 Toffoli gates and no more than 4n n-CNOT gates, and the time complexity is $O(24 + 4n \times (2(n-1))) \approx O(n^2)$. The 1-bit full addition circuit needs two Toffoli gates and two CNOT gates, and the n-bit full addition circuit needs $(n-1)$ 1-bit full addition circuit, so the n-bit full addition circuit's time complexity is $O((n-1) \times (12+2)) = O(14n-14)$. The 1-bit full subtractor circuit needs two NOT gates, four CNOT gates, and two Toffoli gates, the n-bit full subtractor circuit needs $(n-1)$ 1-bit full subtractor circuits, and the time complexity of the n-bit full subtractor circuit is $O((n-1) \times (2+4+12)) = O(18n-18)$.

Considering a $2^n \times 2^m$ binary image, $n > m$, the time complexity is analyzed as follows:

4.1.1. Quantum Parallel Shrinking

From Figures 10 and 11, the quantum parallel shrinking circuit is composed of four address shift operation circuits, three Logic operation $\wedge$, two Logic operation $\vee$, and four CNOT, so the time complexity is $O(4n^2 + 18 + 2 \times 11 + 4) \approx O(4n^2)$.

4.1.2. Quantum Label Propagating

From Figures 13–15, the quantum label propagating circuit is composed of four address shift operations, four Logic operation $\wedge$, one n-bit full subtractors, two compare operation, one n-bit full addition circuit, and two assignment operations, so the time complexity is:

$$O(4n^2 + 24 + (18n - 18) + 2n^2 + (14n - 14) + 24n) = O(6n^2 + 56n - 8) \approx O(6n^2)$$

According to the above analysis, time complexity of the proposed quantum image component labeling algorithm is $O(10n^2) \approx O(n^2)$, which is only the second-order polynomial function of image size.

4.2. Spatial Complexity

Table 1 shows that the number of ancilla qubits in basic quantum functional circuits is a linear function of image size. Therefore, ancilla qubits of the proposed quantum image component labeling algorithm is also a linear function of image size. That is, the spatial complexity of the quantum circuits designed in this paper is $O(cn)$.

Table 1. Number of ancilla qubits used by the basic quantum functional circuits.

No.	Function	Ancilla Qubits
1	Address Shift Operation Circuit	0
2	Logic Operation Circuit	1
3	Control Assignment Operation Circuit	$n+1$
4	Compare Operation Circuit	$2n$
5	Full Addition Circuit	$2n+1$
6	Full Subtraction Circuit	$2n+1$

Although quantum complexity is discussed, lots of quantum algorithms still cannot be applied to the quantum computer or a quantum simulator. Quantum computers and quantum simulations are in their infancy, quantum computing will be limited to about 10 qubits in quantum computer, and a quantum simulator can only operate at most 30 qubits. LaRose [47] has given a detailed explanation of quantum software platforms. Therefore, in the next section, we give the use of matrix calculation to complete the algorithm simulation, which is also a common practice at present.

5. Simulation on Classical Computer

This section describes simulations of the quantum image component labeling algorithm on a classical computer, while quantum computers are currently not at hand. The simulations were run on a classical computer with Inter (R) Core (TM) i7-7500U @2.70 GHz 8.0 GB RAM and 64-bit operating system. The simulations are based on linear algebra with complex vectors as quantum states and unitary matrices as unitary transforms with calculations performed using Python 3.9.

In order to compare with classical algorithms, YACCLAB [48] (Yet Another Connected Components Labeling Benchmark) is used, which is an open-source C++ benchmarking framework for component labeling. YACCLAB allows researchers to test classical component labeling algorithms under the same environment and with the same collection of datasets, which provides a rich and varied dataset that includes both synthetic and real images and lots of well-written programs for classical algorithms.

In the experiments, the library Boost.python is used for Python calling C++, and the quantum image component labeling algorithm is compared with CT [49] (Contour Tracing approach), SAUF [50] (Scan plus Array-based Union-Find algorithm), and NULL, which are three classical component labeling types in YACCLAB. The NULL is a fake algorithm that performs the basic assignment operation defining a lower bound limit of the execution time. The experimental results (Figure 17) show that the proposed algorithm is better than the two classical algorithms, and the execution time is the average time for labeling on the image dataset.

To visually present the result of the proposed algorithm, the two images in traffic scenarios are used for testing. The first image is a traffic sign (Figure 18), the size of the image is 256×265, the second image (Figure 19) is the license plate held by the author, the size is 654×220. In the experiment, we used different colors to distinguish different components. The traffic sign and the license plate have execution time of 0.21 ms and 0.26 ms, respectively. Two tests in the experiment verify the correctness of the quantum image component labeling algorithm in this paper.

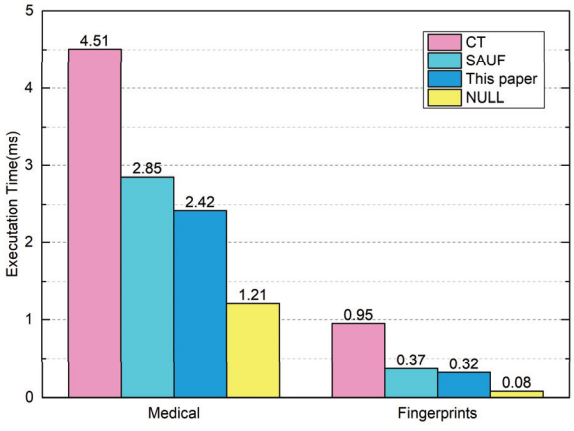


Figure 17. Experimental comparison of CF, SAUF, and NULL.

(**a**) (**b**) (**c**)

Figure 18. The example of traffic signs. (**a**) original image. (**b**) binary image. (**c**) component labeling.

(**a**) (**b**) (**c**)

Figure 19. The example of license plates, "陕" in figure means Shanxi Province. (**a**) original image. (**b**) binary image. (**c**) component labeling.

6. Conclusions

In recent years, with the sharp increase in image data processing, the problem of real-time processing has become a limitation in classical image processing. An image component labeling algorithm is an important pre-processing operation in many image processing algorithms. However, in quantum image processing, component labeling algorithm has not been reported in extant literature. In this paper, we develop a quantum version of the image component labeling algorithm which makes full use of quantum parallelism. Firstly, the modified NEQR model is used to represent the information of binary image. Secondly, basic function circuits are prepared. Thirdly, the quantum circuits of parallel shrinking and label propagating are designed by using the basic function circuits. The purposed circuits can process information of all the pixels simultaneously, which can improve the efficiency of image preprocessing.

Quantum image processing applications have developed only in recent years. The results obtained in this paper could be used in more quantum image processing algorithms. In the future, we will be working to develop new quantum image analysis algorithms based on the quantum component labeling algorithm, especially in the fields of transportation, logistics, and robot navigation.

Author Contributions: Conceptualization, Y.L. and D.H; methodology, K.L.; software, D.H.; validation, Y.X.; writing—original draft preparation, Y.L.; writing—review and editing, D.H.; supervision, K.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors are thankful to the editor and the reviewers for their valuable comments and suggestions that helped to improve the quality of the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Venegas-Andraca, S.E.; Bose, S. Storing, Processing, and Retrieving an Image Using Quantum Mechanics. *Quantum Inf. Comput.* **2003**, *5105*, 137–147.
2. Venegas-Andraca, S.E.; Ball, J. Processing images in entangled quantum systems. *Quantum Inf. Processing* **2010**, *9*, 1–11. [CrossRef]
3. Le, P.Q.; Dong, F.; Hirota, K. A flexible representation of quantum images for polynomial preparation, image compression, and processing operations. *Quantum Inf. Processing* **2011**, *10*, 63–84. [CrossRef]
4. Zhang, Y.; Lu, K.; Gao, Y.; Xu, K. A novel quantum representation for log-polar images. *Quantum Inf. Processing* **2013**, *12*, 3103–3126. [CrossRef]
5. Zhang, Y.; Lu, K.; Gao, Y.; Wang, M. NEQR: A novel enhanced quantum representation of digital images. *Quantum Inf. Processing* **2013**, *12*, 2833–2860. [CrossRef]
6. Mastriani, M. Quantum Boolean image denoising. *Quantum Inf. Processing* **2015**, *14*, 1647–1673. [CrossRef]
7. Yuan, S.; Mao, X.; Xue, Y.; Chen, L.; Xiong, Q.; Compare, A. SQR: A simple quantum representation of infrared images. *Quantum Inf. Processing* **2014**, *13*, 1353–1379. [CrossRef]
8. Sun, B.; Iliyasu, A.; Yan, F.; Dong, F.; Hirota, K.J. An RGB multi-channel representation for images on quantum computers. *J. Adv. Comput. Intell. Intell. Inform.* **2013**, *17*, 404–415. [CrossRef]
9. Li, H.-S.; Zhu, Q.; Zhou, R.-G.; Song, L.; Yang, X.-J. Multi-dimensional color image storage and retrieval for a normal arbitrary quantum superposition state. *Quantum Inf. Processing* **2014**, *13*, 991–1011. [CrossRef]
10. Sang, J.; Wang, S.; Li, Q. A novel quantum representation of color digital images. *Quantum Inf. Processing* **2017**, *16*, 42. [CrossRef]
11. Abdolmaleky, M.; Naseri, M.; Batle, J.; Farouk, A.; Gong, L.-H. Red-Green-Blue multi-channel quantum representation of digital images. *Optik* **2017**, *128*, 121–132. [CrossRef]
12. Xu, G.; Xu, X.; Wang, X.; Wang, X. Order-encoded quantum image model and parallel histogram specification. *Quantum Inf. Processing* **2019**, *18*, 346. [CrossRef]
13. Khan, R.A. An improved flexible representation of quantum images. *Quantum Inf. Processing* **2019**, *18*, 201. [CrossRef]
14. Wang, L.; Ran, Q.; Ma, J.; Yu, S.; Tan, L. QRCI: A new quantum representation model of color digital images. *Opt. Commun.* **2019**, *438*, 147–158. [CrossRef]
15. Chen, G.-L.; Song, X.-H.; Venegas-Andraca, S.E.; El-Latif, A.; Ahmed, A. QIRHSI: Novel quantum image representation based on HSI color space model. *Quantum Inf. Processing* **2022**, *21*, 5. [CrossRef]
16. Zhu, H.-H.; Chen, X.-B.; Yang, Y.-X. Image preparations of multi-mode quantum image representation and their application on quantum image reproduction. *Optik* **2022**, *251*, 168321. [CrossRef]
17. Le, P.Q.; Iliyasu, A.M.; Dong, F.; Hirota, K. Strategies for designing geometric transformations on quantum images. *Theor. Comput. Sci.* **2011**, *412*, 1406–1418. [CrossRef]
18. Wang, J.; Jiang, N.; Wang, L. Quantum image translation. *Quantum Inf. Processing* **2015**, *14*, 1589–1604. [CrossRef]
19. Le, P.Q.; Iliyasu, A.M.; Dong, F.; Hirota, K. Fast Geometric Transformations on Quantum Images. *Int. J. Appl. Math.* **2010**, *40*, 3.
20. Jiang, N.; Wang, J.; Mu, Y. Quantum image scaling up based on nearest-neighbor interpolation with integer scaling ratio. *Quantum Inf. Processing* **2015**, *14*, 4001–4026. [CrossRef]
21. Sang, J.; Wang, S.; Niu, X. Quantum realization of the nearest-neighbor interpolation method for FRQI and NEQR. *Quantum Inf. Processing* **2016**, *15*, 37–64. [CrossRef]
22. Li, P.; Liu, X. Bilinear interpolation method for quantum images based on quantum Fourier transform. *Int. J. Quantum Inf.* **2018**, *16*, 1850031. [CrossRef]
23. Zhou, R.-G.; Cheng, Y.; Liu, D. Quantum image scaling based on bilinear interpolation with arbitrary scaling ratio. *Quantum Inf. Processing* **2019**, *18*, 267. [CrossRef]
24. Jiang, N.; Wang, L. Analysis and improvement of the quantum Arnold image scrambling. *Quantum Inf. Processing* **2014**, *13*, 1545–1551. [CrossRef]
25. Zhou, R.-G.; Sun, Y.-J.; Fan, P. Quantum image Gray-code and bit-plane scrambling. *Quantum Inf. Processing* **2015**, *14*, 1717–1734. [CrossRef]

26. Caraiman, S.; Manta, V.I. Histogram-based segmentation of quantum images. *Theor. Comput. Sci.* **2014**, *529*, 46–60. [CrossRef]
27. Caraiman, S.; Manta, V.I. Image segmentation on a quantum computer. *Quantum Inf. Processing* **2015**, *14*, 1693–1715. [CrossRef]
28. Wang, X.; Yang, C.; Xie, G.-S.; Liu, Z. Image thresholding segmentation on quantum state space. *Entropy* **2018**, *20*, 728. [CrossRef]
29. Li, P.; Shi, T.; Zhao, Y.; Lu, A. Design of threshold segmentation method for quantum image. *Int. J. Theor. Phys.* **2020**, *59*, 514–538. [CrossRef]
30. Zhang, Y.; Lu, K.; Gao, Y. QSobel: A novel quantum image edge extraction algorithm. *Sci. China Inf. Sci.* **2015**, *58*, 1–13. [CrossRef]
31. Fan, P.; Zhou, R.-G.; Hu, W.W.; Jing, N. Quantum image edge extraction based on Laplacian operator and zero-cross method. *Quantum Inf. Processing* **2019**, *18*, 27. [CrossRef]
32. Fan, P.; Zhou, R.-G.; Hu, W.; Jing, N. Quantum image edge extraction based on classical Sobel operator for NEQR. *Quantum Inf. Processing* **2019**, *18*, 24. [CrossRef]
33. Zhou, R.-G.; Yu, H.; Cheng, Y.; Li, F.-X. Quantum image edge extraction based on improved Prewitt operator. *Quantum Inf. Processing* **2019**, *18*, 261. [CrossRef]
34. Li, P.; Shi, T.; Lu, A.; Wang, B. Quantum implementation of classical Marr–Hildreth edge detection. *Quantum Inf. Processing* **2020**, *19*, 64. [CrossRef]
35. Chetia, R.; Boruah, S.; Sahu, P. Quantum image edge detection using improved Sobel mask based on NEQR. *Quantum Inf. Processing* **2021**, *20*, 21. [CrossRef]
36. Jiang, N.; Dang, Y.; Wang, J. Quantum image matching. *Quantum Inf. Processing* **2016**, *15*, 3543–3572. [CrossRef]
37. Dang, Y.; Jiang, N.; Hu, H.; Zhang, W. Analysis and improvement of the quantum image matching. *Quantum Inf. Processing* **2017**, *16*, 269. [CrossRef]
38. Luo, G.; Zhou, R.-G.; Liu, X.; Hu, W.; Luo, J. Fuzzy matching based on gray-scale difference for quantum images. *Int. J. Theor. Phys.* **2018**, *57*, 2447–2460. [CrossRef]
39. Heidari, S.; Naseri, M. A novel LSB based quantum watermarking. *Int. J. Theor. Phys.* **2016**, *55*, 4205–4218. [CrossRef]
40. Hu, W.; Zhou, R.-G.; Luo, J.; Liu, B. LSBs-based quantum color images watermarking algorithm in edge region. *Quantum Inf. Processing* **2019**, *18*, 16. [CrossRef]
41. He, L.; Ren, X.; Gao, Q.; Zhao, X.; Yao, B.; Chao, Y. The connected-component labeling problem: A review of state-of-the-art algorithms. *Pattern Recognit.* **2017**, *70*, 25–43. [CrossRef]
42. Alnuweiri, H.M.; Prasanna, V.K. Parallel architectures and algorithms for image component labeling. *IEEE Trans. Pattern Anal. Mach. Intell.* **1992**, *14*, 1014–1034. [CrossRef]
43. Levialdi, S. On shrinking binary picture patterns. *Commun. ACM* **1972**, *15*, 7–10. [CrossRef]
44. Cheng, K.-W.; Tseng, C.-C. Quantum full adder and subtractor. *Electron. Lett.* **2002**, *38*, 1343–1344. [CrossRef]
45. Barenco, A.; Bennett, C.H.; Cleve, R.; DiVincenzo, D.P.; Margolus, N.; Shor, P.; Sleator, T.; Smolin, J.A.; Weinfurter, H. Elementary gates for quantum computation. *Phys. Rev. A* **1995**, *52*, 3457. [CrossRef] [PubMed]
46. Nielsen, M.A.; Chuang, I. *Quantum Computation and Quantum Information*, 10th ed.; Cambridge University Press: Cambridge, UK, 2010.
47. LaRose, R. Overview and comparison of gate level quantum software platforms. *Quantum* **2019**, *3*, 130. [CrossRef]
48. Bolelli, F.; Cancilla, M.; Baraldi, L.; Grana, C.J. Toward reliable experiments on the performance of connected components labeling algorithms. *J. Real Time Image Processing* **2020**, *17*, 229–244. [CrossRef]
49. Chang, F.; Chen, C.-J.; Lu, C.-J. A linear-time component-labeling algorithm using contour tracing technique. *Comput. Vis. Image Underst.* **2004**, *93*, 206–220. [CrossRef]
50. Wu, K.; Otoo, E.; Suzuki, K. Optimizing two-pass connected-component labeling algorithms. *Pattern Anal. Appl.* **2009**, *12*, 117–135. [CrossRef]

MDPI
St. Alban-Anlage 66
4052 Basel
Switzerland
Tel. +41 61 683 77 34
Fax +41 61 302 89 18
www.mdpi.com

Mathematics Editorial Office
E-mail: mathematics@mdpi.com
www.mdpi.com/journal/mathematics

Academic Open
Access Publishing

www.mdpi.com

ISBN 978-3-0365-7747-0