AFRL-RI-RS-TR-2020-019

DESIGN AND IMPLEMENTATION OF QUANTUM OPTIMIZATION METHODS

CLARKSON UNIVERSITY

FEBRUARY 2020

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nations. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2020-019 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /
KRISTI MEZZANO
Work Unit Manager

/ S /
LAUREN HUIE-SEVERSKY
Technical Advisor, Computing
and Communications Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
FEBRUARY 2020	FINAL TECHNICAL REPORT	MAY 2018 – AUG 2019

4. TITLE AND SUBTITLE

DESIGN AND IMPLEMENTATION OF QUANTUM OPTIMIZATION METHODS

5a. CONTRACT NUMBER
N/A

5b. GRANT NUMBER
FA8750-18-1-0104

5c. PROGRAM ELEMENT NUMBER
62788F

6. AUTHOR(S)

Christino Tamon

5d. PROJECT NUMBER
CYDT

5e. TASK NUMBER
CL

5f. WORK UNIT NUMBER
RK

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Clarkson University
8 Clarkson Ave
Potsdam NY 13699-5815

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/RITQ
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RI

11. SPONSOR/MONITOR'S REPORT NUMBER
AFRL-RI-RS-TR-2020-019

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The project explores the applications of quantum computing ideas to optimization problems in machine learning related to partially observable Markov decision process. The problem of approximating the optimal policy for the expected discounted reward in an infinite horizon is proved to be decidable for quantum observable Markov decision process. This provides a relevant example where the quantum optimization problem is tractable while other related problems are known to be undecidable. A generalization of this quantum Markov model has potential applications for quantum control problems under noisy communication channels. The project also includes an educational component to develop a quantum computing course and a software development component to develop a prototype for a graphical quantum circuit simulator.

15. SUBJECT TERMS

Quantum Computing, Markov Model, Quantum Circuits, Optimal Control Theory, Dynamic Programming

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			**KRISTI MEZZANO**
U	U	U	UU	28	19b. TELEPHONE NUMBER (*Include area code*) **N/A**

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

1. SUMMARY

The project explores the application of quantum computing ideas to optimization problems relevant to machine learning. The project consists of a research component and an educational component. In the research component, the focus is on the complexity of optimization problems related to partially observable Markov decision processes. In the educational component, the focus is on the curriculum development and offering of a quantum computing course at the upper undergraduate and beginning graduate levels. As an additional component, the project also includes the design and development of a prototype for a graphical simulator for quantum circuits.

2. INTRODUCTION

We outline some background on the research and educational components of this project in the following subsections.

2.1 Research Background

Reinforcement learning is an area of machine learning whose main focus is on problems related to sequential decision making under uncertainty. This is strongly connected to mathematical areas such as optimal control theory and dynamic programming which were studied and developed by Bellman, Dynkin, Blackwell, and others in the early 1960s; see Bertsekas and Shreve [1]. More recently, reinforcement learning is relevant for applications involving autonomous robots, self-driving vehicles, and others. A comprehensive treatment of reinforcement learning can be found in Sutton and Barto [2].

The framework of reinforcement learning consists of a game where an agent interacts with a stochastic environment. The environment is represented by a collection of Markov chains sharing a common state space. In this sequential game, at each step, the agent chooses an action which selects a specific Markov chain. The environment will then stochastically update the state using the chosen Markov chain. The agent receives a reward based on the current state and action taken. It is typically assumed that the reward function is known to the agent. The goal of the agent is to maximize the total expected reward accumulated over the duration or horizon of the game (which may be finite or infinite).

Two particular models of interest are *Markov decision process* (MDP) and its more realistic variant called *partially observable Markov decision process* (POMDP). In the MDP model, the state of the environment is visible to the agent, while in the POMDP model, the state information is hidden. But, we assume that the environment emits a stochastic signal that depends on its current state.

A standard approach to handle POMDPs is to represent the hidden state information as a probability distribution over the states and viewing it as a MDP over the space of all probability distributions. The transitions on this MDP is defined by a Bayes update based on the observed output signal. This MDP is known as the *belief* Markov decision process. Under this reduction, methods for solving MDPs can be utilized. A drawback of this approach is that the state space becomes uncountably infinite (even if the original state space is finite).

A policy for the agent is a function that determines which action to choose at each step given the current history of the game. The main optimization problem is to search for the optimal policy which obtains the maximum total expected reward. In a seminal work, Blackwell [3] proved that there always exists a stationary (or time-independent) optimal policy if the space of actions and output signals are both finite. This seminal result of Blackwell was proved under the generous assumption that the state space is a Borel space. This assumption is standard for analyses involving probability (or measurable) spaces.

Subsequently, Sondik [4] showed that, under additional assumptions, the optimal policy admits a simple and compact representation as a piecewise-linear and convex function. This latter observation is important for subsequent algorithmic developments in the area.

In this project, we consider the following two problems. The first problem is to determine if there are quantum speedups in solving the optimization problem of finding the optimal policy in POMDPs. The second problem is to study natural generalizations of POMDPs in a quantum setting. For the first problem, we outline several natural (albeit modest) quantum speedups that can be obtained on standard algorithms for POMDPs. For the second problem, we propose a new and more general model for a quantum POMDP which extends a known model described by Barry et al. [5]. We also explore methods to solve the optimization problem for POMDPs in the quantum setting.

2.2 Educational Environment

As part of the project, we plan to develop a course on quantum computing which is aimed for upper-level undergraduate and beginning graduate students. One of the main goals is to create a course that is accessible for students from computer science, mathematics, physics, and engineering. We would like to offer a modern course in quantum computing which strongly conveys the exciting recent developments in quantum computing but provides a solid foundation for students to explore further directions (either academically or otherwise).

We draw on our past experiences in teaching two similar courses at Clarkson University. This includes a graduate-level course in quantum computation and a graduate-level course on quantum cryptography. But, the new course will incorporate some new insights and tools that had been developed in recent years for vertically integrating the quantum computing knowledge into the classroom.

3. METHODS, ASSUMPTIONS, AND PROCEDURES

In this section, we describe some of the research methodology used and some of the educational development approaches taken in the project.

3.1 Research Methodology

We summarize and briefly describe known methods for optimization problems related to Markov decision process and partially observable Markov decision process. In subsequent sections, we will consider quantum solutions and quantum generalizations related to these models.

We provide a brief description of a partially observable Markov decision process (POMDP). It is defined by the following components:

1. A finite set S of states of the environment.
2. A finite set of possible actions that the agent can take. Each action controls a specific Markov chain over the set of states. Each Markov chain describes a stochastic transition from every state to a collection of next possible states.
3. A finite set of possible output signals. To each state, the environment assigns a stochastic output map which emits a specific output signal with a given probability. If this output map reveals the underlying state (with probability one), then we simply obtain a Markov decision process.
4. A reward function R that assigns a real number to each state. We assume that the reward function has a bounded range.
5. A discount factor γ that is a non-negative real number strictly less than one. This is a numeric scaling factor that adjusts the importance of past rewards. In summing the total rewards, we apply the discount factor geometrically to each term in the summation. This guarantees that the sum is bounded (even in the case of an infinite horizon when the game continues forever).

The history (or trajectory) of the game is a sequence (possibly infinite) of state, action and output triplets. To each finite initial segment of the history, we can compute the associated probability and total reward values. In this manner, the expected total reward can be evaluated given a fixed initial state and the policy used.

A fundamental technique for solving the optimal policy problem is to consider a *value function* over the states. This value function encodes the total expected reward obtained starting from any initial state. The value function is defined recursively as

$$V(s) = R(s) + \gamma \mathbb{E}\big[V\big(P(s,a)\big)\big] \tag{1}$$

Here, the expectation is taken with respect to the stochastic output map, s is the initial state, a is the action chosen, and $P(s,a)$ denotes the next state.

The stochastic transition to the next state is induced by the Markov chain associated with the chosen action and the observed stochastic output. For a stationary policy, we may assume that the action chosen above is determined by the policy.

Blackwell [3] proved that the value function V^* of the optimal policy satisfies Bellman's optimality equation given by

$$V^*(s) = sup_a\{R(s) + \gamma\mathbb{E}[V^*(P(s,a))]\} \quad (2)$$

The maximization is taken over all possible actions. From the optimal value function, we can derive the optimal policy itself by always choosing the maximizing action in the above equation. This is the standard method used to recover the optimal solution in dynamic programming problems.

The idea behind one of the main algorithms for computing the optimal value function is to iteratively improve the current value function using the Bellman update operator. This operator is defined by

$$\mathfrak{T}(V)(s) = sup_a\{R(s) + \gamma\mathbb{E}[V(P(s,a))]\} \quad (3)$$

Here, $\mathfrak{T}$ is an operator that maps a value function to an improved value function. The first key observation is that the optimal value function is a unique fixed-point of the operator $\mathfrak{T}$. The second key observation is that $\mathfrak{T}$ is a contractive operator. These two observations guarantee the convergence of the fixed-point iteration using the operator $\mathfrak{T}$ starting from an arbitrary value function. These fundamental observations were proved by Blackwell [3] under very general conditions on the state space.

The fixed-point iteration algorithm above is also known as the Value Iteration algorithm. Sondik [4] made a crucial observation about the representation of the optimal policy function. Since the average reward is a linear function over the states, the Bellman update equation shows that the optimal value function is the maximum of a collection of linear functions; in particular, it is a piecewise-linear convex function. Under certain assumptions, the optimal policy has a compact and finite representation (even in the infinite horizon case).

The works of Blackwell [3] and Sondik [4] provided the mathematical and algorithmic foundations of stochastic optimal control theory and reinforcement learning. Another foundational work in reinforcement learning is the seminal paper by Madani, Hanks, and Condon [6]. In the latter paper, the authors proved that most relevant problems related to partially observable Markov decision process (which include probabilistic planning) are undecidable. In particular, they proved that the problem of computing the optimal policy of a POMDP over an infinite horizon under the expected discounted criterion is undecidable. But, they observed that approximating the optimal policy to within arbitrary accuracy is decidable (due to the fixed-point iteration).

3.2 Educational Approach

Given that the quantum computing course that we developed is aimed at students from different backgrounds and disciplines, we made several basic assumptions. We assume that the students have foundational knowledge in the following areas:

- Calculus, linear algebra, and discrete mathematics. These will be typically satisfied by science and engineering students with junior standing. Calculus and discrete mathematics are basic required knowledge for a certain level of mathematical maturity, while knowledge of linear algebra is important to understand rudimentary quantum theory.
- Computer science: theoretical and applied (programming skills). We placed stronger emphasis on the applied computer science knowledge since some of the basic theoretical computer science concepts will be reviewed in the course. More specifically, we assume the students are comfortable with (or can learn on their own) programming languages such as Python, Java, or related ones.

The course uses the classic textbook by Chuang and Nielsen's "Quantum Computation and Quantum Information" (Cambridge University Press, 2001). We are of the opinion that this book still has an impressive coverage of topics and is sufficiently accessible to students. The list of topics covered in the book is the most natural progression for a course in quantum computation: basic linear algebra for quantum theory, quantum circuits, early quantum algorithms and Shor's algorithm, Grover search, quantum information theory, and additional topics. Moreover, the book includes self-contained background chapters in linear algebra and theoretical computer science.

The course includes homework assignments that cover the theoretical aspects of the course and also some programming components which relied on publicly available quantum software development platforms. The latter include the popular platforms such as pyquil (Rigetti) and qiskit (IBM). Some of the students had also explored other platforms such as Leap (Dwave), Liquid (Microsoft), and Quipper (based on the Haskell programming language).

4. RESULTS AND DISCUSSION

In this section, we summarize our main research findings and educational outcomes. We also summarize our software development project for a graphical quantum circuit simulator.

4.1 Research Findings

We describe some results and observations on quantum speedups for optimization problem on classical POMDPs and some generalizations on the quantum POMDP model.

Quantum Speedups.

We observe some modest theoretical quantum speedups on the optimization problem on classical POMDPs.

1. *Maximum Finding*. By Sondik's observation (Sondik [4]), the optimal policy value function is a piecewise-linear convex function that can be represented as the maximum of a collection of linear functions. More specifically, the optimal policy value is a function defined over the belief states (which are probability distributions over the states of the POMDP) given by

$$V(p) = \max_a \langle w_a, p \rangle \tag{4}$$

   Thus, the evaluation of this policy function involves a maximum finding over a collection of normal vectors (which define the hyperplanes). Here, we can apply an algorithm due to Durr and Hoyer [7] which uses Grover search to find the maximum of a collection of items. This algorithm achieves the same quadratic speedup as Grover search. To apply this idea, we need to implement a quantum blackbox oracle which computes the above inner product for each candidate linear function. The method of Durr and Hoyer converts this oracle to the standard Grover oracle which represents marked elements in a collection.

2. *Matrix inversion*. An alternative algorithm to the Value Iteration (as described in the previous section) is Policy Iteration. The idea behind this algorithm is to search for the optimal policy by explicitly improving the policy itself and using the value function as a measure of progress in the Bellman update. Given a fixed policy, its value function can be computed by solving a linear system of equations or by matrix inversion. So, assuming the belief space is discretized appropriately, the value function may be written as

$$V = (I - \gamma P)^{-1} R \tag{5}$$

   Here, we may apply the quantum algorithm due to Harrow, Hassidim and Lloyd [8]. But, the potential speedups here are conditional on several assumptions required in the HHL algorithm. Among some of these assumptions, we mention the following. First, we need to load the reward vector into quantum memory. Second, we require that the matrix that we are inverting is sparse (for Hamiltonian simulation) and has sufficiently good condition

number. Third, we assume that the entries of this matrix is accessible via a blackbox oracle. Fourth, the output value function is given as a quantum state that has to be measured.

Remark. The quantum speedup offered by the maximum finding for evaluating the value function is more concrete than the potential speedup in matrix inversion (using the HHL algorithm) for the second application. It is unclear if the HHL algorithm will provide a substantial speedup for this particular application due to the strong assumptions placed on this algorithm.

Quantum Markov models.

We outline some results in generalizing the POMDP model to the quantum setting. Barry et al. [5] defined an interesting quantum variant of Markov decision processes. We describe their model in what follows and then provide our generalization.

Similar to the classical model of POMDP, the quantum POMDP model of Barry et al. [5] consists of similar components. We describe their model in the following.

Definition A. (Quantum Observable Markov Decision Process [5]) Assume that Σ is a finite set of (input) actions and Δ is a finite set of observed (output) symbols. A quantum observable Markov decision process (QOMDP) is defined as $(S, \Sigma, \Delta, \Lambda, R, \rho_0, \gamma)$. We define each component in the following:

1. The set of states is represented by the set of all density matrices over a *d*-dimensional complex Euclidean space:

$$S = \{\rho \in M_d(\mathbb{R}) | \rho \succeq 0, Tr(\rho) = 1\} \tag{6}$$

   Each quantum state is a density matrix (a positive semidefinite matrix of unit trace). This is a generalization of the classical case since a density matrix (which represents a mixed quantum state) is a generalization of a probability distribution over the computational basis states.

2. A finite collection of superoperators

$$\{\Lambda_a | \Lambda_a \text{ is a superoperator, for } a \in \Sigma\} \tag{7}$$

   where each superoperator corresponds to an action that the agent chooses. Each *superoperator* is defined by its Kraus decomposition

$$\Lambda_a(\rho) = \sum_{b \in \Delta} \Lambda_{a,b} \rho \Lambda_{a,b}^{\dagger} \tag{8}$$

   where the sum ranges over the set of outcomes. If the above superoperator is chosen, the output received by the agent consists of the signal $b \in \Delta$ along with the post-measurement quantum state (see Figure 1)

$$\frac{\Lambda_{a,b} \rho \Lambda_{a,b}^{\dagger}}{Tr\left(\Lambda_{a,b} \rho \Lambda_{a,b}^{\dagger}\right)} \tag{9}$$

3. A *reward* operator R.

   Given a quantum state, the reward value is given as the average of the observable with respect to the quantum state:

$$R(\rho) = Tr(R\rho) \tag{10}$$

4. A distinguished *start state* ρ_0.

5. A *discount factor* γ that is a nonnegative real number strictly less than one.

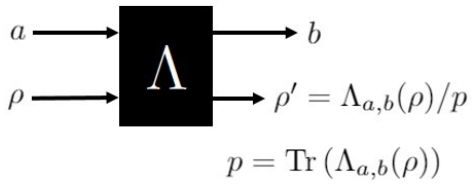

Figure 1. Superoperator

The goal of the **optimization** problem related to a QOMDP is to find a policy which determines a sequence of actions for the agent so that the total discounted reward is maximized:

$$\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t\right] \tag{11}$$

Here, the sequence of rewards $R_t = R(\rho_t)$ obtained by the policy is a sequence of random variables induced by the sequence of states during every step. Also, note that we focus on the infinite horizon case where the interaction between the agent and the environment continues for an arbitrarily long time. In fact, we study the limiting case where time approaches infinity.

Another related problem is the **planning** problem where the objective is to find a policy whose sequence of actions will reach a specific target state of the environment. We may view this planning problem as a *reachability* problem.

In what follows, we will reformulate the QOMDP model in a generalized framework. We will need to assume some notions from the theory of quantum information (see Wilde [9] and Watrous [10]). Our quantum POMDP model shares some of the basic underlying components with the model of Barry et al. [5], such as the set of mixed quantum states (density matrices), the finite set of input actions, the finite set of output observed symbols, and the discount factor. But, it differs in some of the other components.

Definition B. (Quantum Partially Observable Markov Decision Process) Assume that Σ is a finite set of (input) actions and Δ is a finite set of observed (output) symbols. A quantum partially observable Markov decision process (QPOMDP) is defined as $(S, \Sigma, \Delta, \Phi, \Omega, R, \rho_0, \gamma, C_i, C_o)$ where each component is given in the following:

1. The set of states is represented by the set of all density matrices over a *d*-dimensional complex Euclidean space:

$$S = \{\rho \in M_d(\mathbb{R}) | \rho \succeq 0, Tr(\rho) = 1\} \quad (12)$$

This is similar to the case for QOMDP.

2. A finite set of actions where each action corresponds to a quantum channel

$$\{\Phi_a | \Phi_a \text{ is a quantum channel, for } a \in \Sigma\} \quad (13)$$

A quantum channel is a completely positive and trace preserving linear map acting on density matrices (or mixed quantum states). It is a generalization of classical Markov chains (which act on probability distributions). The precise framework which we used is the notion of a *conditional quantum channel* (see Wilde [9], page 160). A conditional quantum channel is similar to a controlled unitary operation but where the unitary operation is replaced with the more general concept of a quantum channel. Formally, we may view a conditional channel acting on a density matrix as an operation which prepares a classical register (the choice or control) and a quantum register (which contains the application of a specific chosen quantum channel to the input density matrix):

$$\Phi(\rho) = \sum_a |a\rangle\langle a| \otimes \Phi_a(\rho) \quad (14)$$

Given a classical choice specified in the first register, the second register contains the required chosen quantum state obtained from applying the specified quantum channel to the input mixed state. See Figure 2.

3. A finite set of output signals where each output corresponds to a quantum instrument.

More formally, a *quantum instrument* is a collection of quantum operations (completely positive and trace nonincreasing linear maps)

$$\{\Omega_b | \Omega_b \text{ is a quantum operation, for } b \in \Delta\} \quad (15)$$

whose sum forms a quantum channel (see Watrous [10], page 111).

A quantum instrument is a generalization of a classical stochastic process over a finite collection of elements. Similarly to a conditional quantum channel, we may view a quantum instrument as an operation which prepares a classical register (containing the classical outcome) and a quantum register (which contains the application of a specific quantum operation to the input density matrix):

$$\Omega(\rho) = \sum_b |b\rangle\langle b| \otimes \Omega_b(\rho) \quad (16)$$

Upon measuring the first register, the second register contains the post-measurement quantum state (after renormalization). Note that a quantum instrument also represents a quantum measurement (see Watrous [10]). See Figure 2.

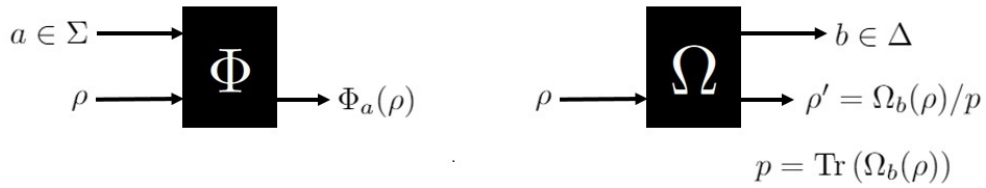

Figure 2. Conditional Quantum Channel and Quantum Instrument

4. A reward operator R that is an *observable* (which is a Hermitian operator).

   The assumption that the operator is Hermitian ensures that the eigenvalues are real-valued. Given a quantum state, the reward value is given as the average of the observable with respect to the quantum state (viewed as a trace inner product between the operator and the quantum state):

   $$Tr(R\rho) = \langle R, \rho \rangle \tag{17}$$

   This generalizes the classical notion of the average reward which is the inner product between the reward vector and a probability distribution

   $$\mathbb{E}[R(x)] = \sum_x R(x)p(x) \tag{18}$$

5. A classical channel $C_o: \Delta \to \Delta_0$ whose input is the output of the quantum instrument and whose output ranges over a possibly different alphabet Δ_0.

   This classical channel models the process that transmits information from the environment to the agent. In Barry et al. [5], this channel is the noiseless channel. But this classical channel may capture cases where the output signal received by the agent is noisy.

6. A classical channel $C_i: \Sigma \to \Sigma_0$ whose input is the action chosen by the agent and whose output is used by the environment to select the quantum channel.

   This classical channel models the process that transmits information from the agent to the environment. It may be used to capture cases where the choice of actions of the agent is corrupted by noise.

Given that the current quantum state is ρ, if the agent chooses action a, then the QPOMDP outputs the symbol b with probability $p = Tr(\Omega_b \Phi_a(\rho))$ which results in a post-measurement state of $\rho' = \Omega_b \Phi_a(\rho)/p$. See Figure 3.

Given noisy channels C_i and C_o, it is possible that although the agent chooses action a, the QPOMDP receives the action a' and outputs symbol b with probability $q = Tr(\Omega_b \Phi_{a'}(\rho))$ which results in a post-measurement state of $\Omega_b \Phi_{a'}(\rho)/q$, and the agent receives the output b'. Here, we assume that $a' = C_i(a)$ and $b' = C_o(b)$.

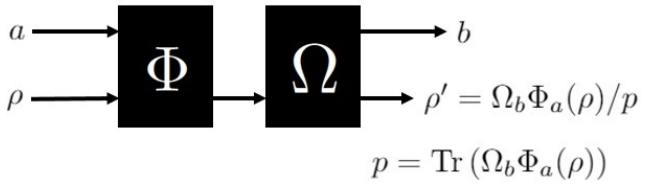

Figure 3. Composition of Conditional Channel and Quantum Instrument

Example: We consider a simple example of a quantum observable Markov decision process defined over the Bloch sphere. There are two possible actions given by $\Sigma = \{a_0, a_1\}$ and two possible outputs given by $\Delta = \{b_{-1}, b_{+1}\}$. The conditional quantum channel and the quantum instruments are defined below.

The conditional quantum channel is given by

$$\Phi_{a_0}(\rho) = \frac{1}{2} R_x\left(\frac{\pi}{3}\right) \rho R_x^\dagger\left(\frac{\pi}{3}\right) + \frac{1}{2}\rho \tag{19}$$

$$\Phi_{a_1}(\rho) = \frac{1}{2} R_y\left(\frac{\pi}{3}\right) \rho R_y^\dagger\left(\frac{\pi}{3}\right) + \frac{1}{2}\rho \tag{20}$$

and the quantum instrument is given by the two components

$$\Omega_{b_1}(\rho) = \frac{1}{2} R_z\left(\frac{\pi}{3}\right) \rho R_z^\dagger\left(\frac{\pi}{3}\right) \tag{21}$$

$$\Omega_{b_{-1}}(\rho) = \frac{1}{2} R_z\left(-\frac{\pi}{3}\right) \rho R_z^\dagger\left(-\frac{\pi}{3}\right) \tag{22}$$

Here, $R_x(\theta) = \exp(-i\theta X/2)$, $R_y(\theta) = \exp(-i\theta Y/2)$, and $R_z(\theta) = \exp(-i\theta Z/2)$ are the standard Bloch rotations around the respective axes.

Figure 4 shows the possible trajectories of the first few steps of this QOMDP.

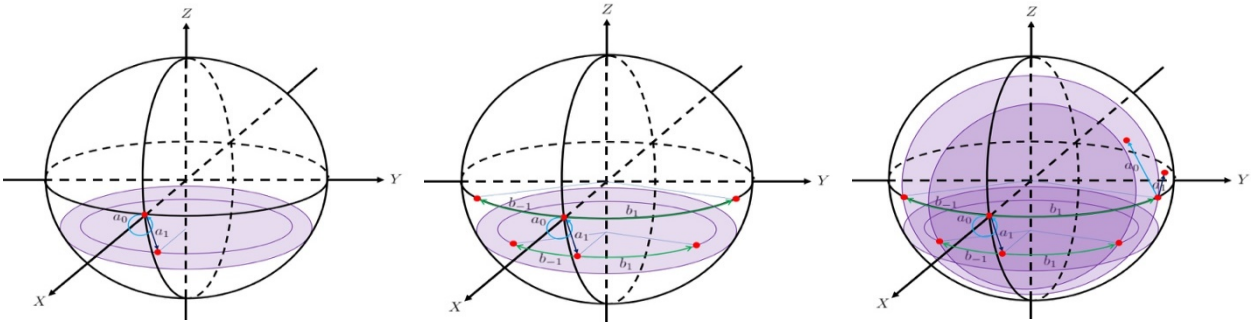

Figure 4. Example of a Bloch sphere trajectory

Technical Observations.

In the context of the quantum Markov models above, we observe the following results.

Fact 1: The *quantum observable Markov decision process* (QOMDP) model (from Definition A) is a special case of a *quantum partially observable Markov decision process* (QPOMDP) (from Definition B).

Proof: This follows from two facts. First, we use the noiseless (or identity) classical channels for the input and output channels in a QPOMDP to simulate the communication between the agent and the environment in a QOMDP. Second, we observe that a collection of superoperators (as described in Barry et al. [5]) is equivalent to a composition of a conditional quantum channel and a quantum instrument. This is a corollary of our observation on the equivalence between quantum Moore transducer and quantum Mealy transducer (see Fact 4 below). See Figure 5.

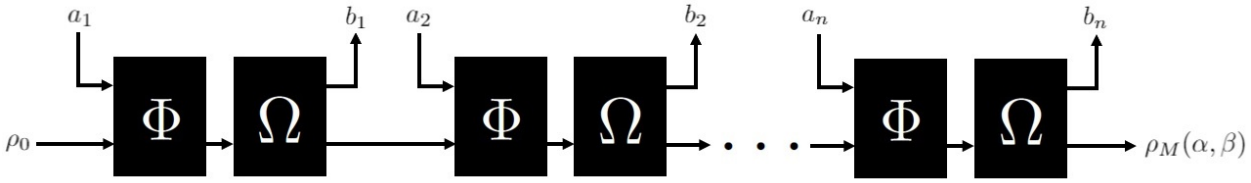

Figure 5. Trajectory of a Quantum Observable Markov Decision Process

Fact 2: The *hidden quantum Markov model* defined by Monras et al. [11] is a special case of a *quantum partially observable Markov decision process* (QPOMDP) (from Definition B).

Proof: This follows by using a trivial conditional channel. More specifically, a trivial conditional channel is defined over a unary input (action) alphabet. So, there is only one option for the action (that is, no choice from the perspective of the agent). This describes the framework of a hidden Markov model where there is a single hidden underlying Markov chain with a stochastic output mapping from each state. The hidden quantum Markov model is a generalization of the classical hidden Markov model by using a quantum channel in place of a Markov chain. See Figure 6.

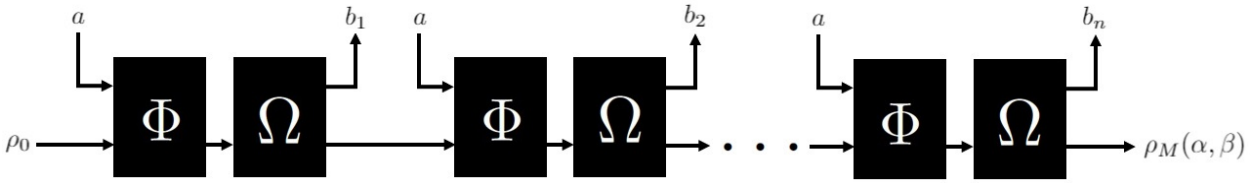

Figure 6. Hidden Quantum Markov Model

Fact 3*: Quantum Markov chain* is a special case of a *quantum partially observable Markov decision process* (from Definition B).

Proof: This follows by using a trivial conditional quantum channel (only one action is available) and a trivial quantum instrument. A trivial quantum instrument may be defined in several ways.

First, it may be defined over a unary output alphabet. This supports the case of a unitary (unmeasured) quantum walk since the measurement is deterministic (the output is unique). Second, it may be defined over an alphabet that corresponds to the computational basis states. The supports the case of a measured quantum walk where we alternate a unitary evolution with a measurement operator. Finally, we may define the quantum instrument to support partial measurement of the quantum walk by using a non-unary output alphabet whose size is strictly smaller than the number of basis states. See Figure 7.

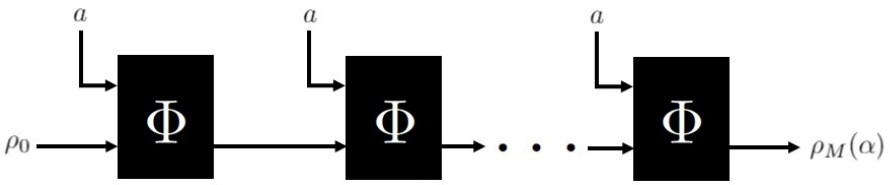

Figure 7. Quantum Markov Chain

Remark 1: We remark that quantum Markov chains are relevant for quantum algorithms since Grover search can be viewed as a quantized Markov chain on graphs.

Remark 2: The QPOMDP model generalizes the quantum observable Markov decision process (QOMDP) model since the latter is a perfect information model where the quantum state of the environment is known to the agent at each given step. This holds since the dynamics of the environment is deterministic over the set of density matrices. In this sense, QOMDP is a generalization of the belief MDP (see Sondik [4]). By introducing the noisy output channel, the dynamics in the new model is stochastic instead of deterministic. Although the set of density matrices is closed under convex combinations, partial information about the noise may be useful.

Barry et al. [5] defined a special variant of a QOMDP called goal-state QOMDP which is relevant to study planning problems. We describe this model in the following.

Definition C. (Barry et al. [5]) A *goal-state quantum observable Markov decision process* is defined as $(S, \Sigma, \Delta, \Lambda, \rho_0, \rho_g)$ where the first five components are similar to the definition in a QOMDP and the last component is an absorbing goal state. The goal state is *absorbing* in that for each action *a* and each output *b*, we have

$$\frac{\Lambda_{a,b}\rho_g\Lambda_{a,b}^{\dagger}}{Tr\left(\Lambda_{a,b}\rho_g\Lambda_{a,b}^{\dagger}\right)} = \rho_g \tag{23}$$

The objective in a planning problem related to a goal-state QOMDP is to design a policy which reaches the goal state (from the start state) with probability one.

In what follows, we define quantum analogues of the classical finite transducer (which is a finite automata with output). There are two standard models of finite transducers, namely, the Moore machine and the Mealy machine. In the Moore machine, the output is associated with a state, while in the Mealy machine, the output is associated with a state and an action (or a transition). Based on these two different models, we define two quantum analogues (which will be related closely to the QPOMDP and the QOMDP models, respectively).

Definition D. A *quantum Moore transducer* is defined as the tuple $(S, \Sigma, \Delta, \Phi, \Omega, \rho_0, \Pi)$ where the first six components are similar to the components of a quantum partially observable Markov decision process (Definition B) and the last component is a projection operator to an accepting or goal subspace.

Definition E. A *quantum Mealy transducer* is defined as the tuple $(S, \Sigma, \Delta, \Lambda, \rho_0, \Pi)$ where the first five components are similar to the components of a quantum observable Markov decision process (Definition A) except that we view Λ as a collection of quantum instruments, and the last component is a projection operator to an accepting or goal subspace.

Remark 3: The goal-state QOMDP is a special case of a quantum Mealy transducer by using the rank-one projector onto the goal state in the latter model. Moreover, note that a superoperator is a quantum instrument. The QOMDP model is defined based on the notion of a "conditional quantum instrument." We point out that this notion can be formalized by decoupling it into a conditional quantum channel and a quantum instrument.

Fact 4: (see [13], Theorem 3.6) The quantum Moore transducer model is equivalent to the quantum Mealy transducer model.

Proof: This follows from the equivalence between a set of superoperators and a composition of a conditional quantum channel with a quantum instrument. For one direction, given a quantum Moore machine $(S, \Sigma, \Delta, \Phi, \Omega, \rho_0, \Pi)$, we construct a quantum Mealy machine by defining $\Lambda_{a,b} = \Omega_b \circ \Phi_a$ for each action a and each output signal b. Let $\Lambda_a(\rho) = \sum_b |\, b\rangle\langle b| \otimes \Lambda_{a,b}(\rho)$ be a quantum instrument associated to action a. It can be shown that this defines an equivalent quantum Mealy machine. For the opposite direction, given a quantum Mealy machine $(S, \Sigma, \Delta, \Lambda, \rho_0, \Pi)$, we define a quantum channel using $\Phi_a(\rho) = \sum_b |\, b\rangle\langle b| \otimes \Lambda_{a,b}(\rho)$ and let $\Omega_b = |b\rangle\langle b| \otimes I$. Then, we construct a conditional quantum channel using the former and note the latter sums to a quantum instrument. This defines an equivalent quantum Moore machine.

Computational Complexity. We discuss the complexity of several computational problems related to Markov decision processes. Our focus will be on three types of problems: reachability, non-occurrence, and policy existence. These problems given as decision (yes/no) problems are defined in the following.

REACHABILITY

Input: A (classical or quantum) Moore machine and a threshold probability value.

Question: Is there a sequence of input actions where the probability that the Moore machine reaches the accepting subspace is greater or equal to the given threshold probability value?

NON-OCCURRENCE

Input: A (classical or quantum) Moore machine and a threshold probability value.

Question: Are there a sequence of input actions and a sequence of output observations where the probability that the Moore machine reaches the accepting subspace and generates the sequence of outputs is less or equal to the given threshold probability value?

POLICY EXISTENCE

Input: A (classical or quantum) partially observable Markov decision process and a threshold value.

Output: Is there a policy whose value is greater or equal to the given threshold value?

We summarize the complexity of these computational problems in Table 1. The italicized quantum results are immediate corollaries of the classical undecidable results. Our contributions are described in Theorem 1 and Theorem 4 from [13].

Table 1. Complexity of Computational Problems

Problem	**Classical Complexity**	**Quantum Complexity**
Reachability	Undecidable [6]	*Undecidable*
Perfect Reachability	Decidable (folklore)	Undecidable [5]
Non-Occurrence	Undecidable (Theorem 1, [13])	*Undecidable*
Perfect Non-Occurrence	Decidable [12]	Undecidable [12]
Policy Existence	Undecidable [6]	*Undecidable*
Approximate Policy Existence	Decidable [3,4,6]	Decidable (Theorem 4, [13])

Remark 4: The classical Reachability problem (also called the *Probabilistic Planning* problem) for goal-state POMDP was proved to be undecidable by Madani et al. [6]. This immediately implies the undecidability of the quantum version since the quantum model includes the classical model as a special case. This observation was pointed out by Barry et al. [5].

On the other hand, the Perfect Reachability problem where the threshold probability value is one admits different complexities. Although the classical problem is decidable (this is a folklore result but a proof also appeared in [5]), the quantum problem was proved to be undecidable by Barry et al. [5].

As shown in Figure 8, the goal of the policy for the planning problem is to find a sequence of actions for which the goal state is reached from the start state.

The undecidability of Reachability problem of QOMDP is based on the Matrix Mortality problem (used by Eisert et al. [12]) which asks if, given a finite sequence of integer matrices, there is a sequence of matrices (where each matrix is an element of the input collection) whose product is

the zero matrix. It is known that the undecidable Post Correspondence Problem (PCP) is reducible to the Matrix Mortality problem.

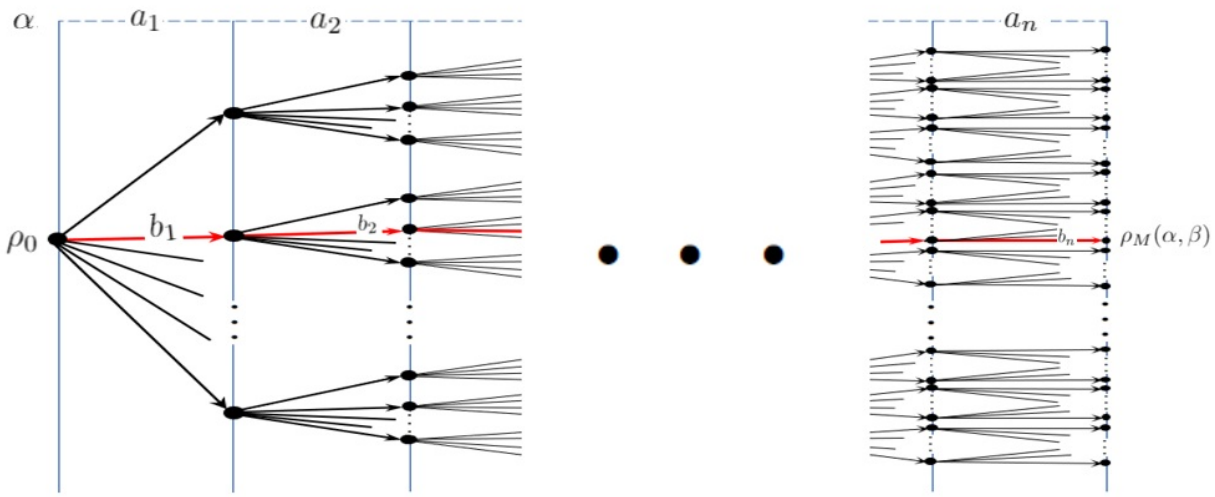

Figure 8. A Policy Tree of Trajectories

Remark 5: The Perfect Non-Occurrence problem asked if there is a sequence that will never appear as output of a Moore machine. Eisert et al. [12] studied the quantum version of this problem and proved that it is undecidable. In contrast, they proved that the classical problem is decidable.

For the general Non-Occurrence problem, we show the following.

Theorem 1. (see [13]) The classical Non-Occurrence is undecidable.

Proof: (Sketch) The idea is to reduce goal-state reachability to non-occurrence. Given that the goal-state reachability problem is undecidable (Madani et al. [6]), this immediately shows the undecidability of the non-occurrence problem. For the reduction, given a goal-state reachability question on a Moore machine, we construct another Moore machine where we add an extra sink state reachable only from the goal state and an extra output symbol that is observable only from the extra sink state. We show that the goal state is reachable in the original machine with probability at least p if and only if the second machine outputs a sequence containing the extra symbol with probability at most 1-p.

Remark 6: In the Policy Existence problem, the goal is to find a policy which maximizes the average total discounted reward. Madani et al. [6] proved that the classical problem is undecidable. But, they observed that approximate version of this problem is decidable due to the results of Blackwell [3] and Sondik [4]. We describe some of the basic ideas behind the approximation algorithm. For this, we need to recall some standard concepts from real analysis.

Definition F. In a topological space X, a collection of subsets of X is called a σ-algebra if it contains X and it is closed under complementation and countable unions. The smallest such σ-algebra which contains all open sets of X is called the Borel algebra of X. The elements of this Borel algebra is called the Borel subsets of X. A *Borel set* is a Borel subset of a complete separable metric space.

In Blackwell [3], a *Markov decision process* is defined as a tuple $(S, \Sigma, P, R, p_0, \gamma)$ where S is a Borel set, Σ is a finite set of actions, P is a collection of Markov chains indexed by the actions, R is a bounded real-valued reward function, p_0 is a start state, and γ is a discount factor. These components are similar to the ones used to definite partially observable Markov decision process.

The history at time *n* is given by $H_n = (S \times \Sigma)^{n-1} \times S$. A *policy* is a sequence $\pi = (\pi_n)$ of maps where π_n is a conditional distribution over the set of actions given the history at time *n*. A policy is called *stationary* if each π_n is equal to the same deterministic map from the set of states to the set of actions. The *value function* of a policy for a Markov decision process is defined to be

$$V(p_0) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(X_t)\right] \tag{24}$$

where the rewards are defined over the infinite sequence of state trajectory which begins with the start state.

The value function of a policy is a bounded real-valued map over the set of states S. Consider the operator which maps a value function to another value function given as follows:

$$T(V) = \sup_a T_a(V) \tag{25}$$

where

$$T_a(V)(p) = R(p) + \gamma \mathbb{E}\big[V\big(P(p,a)\big)\big] \tag{26}$$

Theorem. (Blackwell [3]) The operator T is γ-Lipschitz (that is γ-contractive) and has a unique fixed point $V^\star$ so that for any V_0 we have $||T^n(V_0) - V^\star|| \leq \gamma^n ||V_0 - V^\star||$ (which follows from the Banach fixed point theorem).

Theorem. (Blackwell [3]) A policy is optimal if and only if its value function is a fixed point of T. Moreover, there exists an optimal stationary policy.

We apply Blackwell's theorems to quantum observable Markov decision process.

Theorem 2. (see [13]) Any quantum observable Markov decision process has an optimal stationary policy.

Proof: (Sketch) It suffices to show that the set of density matrices over a finite dimensional Euclidean space forms a Borel space, and then apply Blackwell's theorem. We first observe that the space of complex finite dimensional matrices is a complete separable metric space (with a norm induced by the Hilbert-Schmidt trace inner product). Since this space is closed, it is Borel.

Next, we describe an important observation due to Sondik [4] which we extend to the case of quantum observable Markov decision process.

Theorem 3. (see [13], Theorem 6.6) The value function of a quantum observable Markov decision process obtained from $T^n(V_0)$ is piecewise linear and convex.

Proof: We prove this by induction on the number of iterations. Without loss of generality, we may assume that the quantum observable Markov decision process is given as a quantum Moore machine $(S, \Sigma, \Delta, \Phi, \Omega, \rho_0, \Pi)$. Further, assume that we have the following Kraus decomposition for each conditional quantum channel

$$\Phi_a(\rho) = \sum_i K_{a,i} \rho K_{a,i}^\dagger \tag{27}$$

Also, assume that each of the component of the quantum instrument satisfies

$$\Omega_b(\rho) = L_b \rho L_b^\dagger \tag{28}$$

Recall that the reward operator is given by the trace inner product

$$Tr(R\rho) = \langle R, \rho \rangle \tag{29}$$

So, we have

$$T(V)(\rho) = \sup_a \left(\langle R, \rho \rangle + \gamma \sum_b p_b V\left(\Omega_b \circ \frac{\Phi_a(\rho)}{p_b} \right) \right) \tag{30}$$

Here, p_b denotes the trace of $\Omega_b \circ \Phi_a(\rho)$.

Assume inductively that the value function is piecewise linear and convex. Thus,

$$T(V)(\rho) = \sup_a \left(\langle R, \rho \rangle + \gamma \sum_b p_b \sup_c \left\langle R_c, \Omega_b \circ \frac{\Phi_a(\rho)}{p_b} \right\rangle \right) \tag{31}$$

This can be written as

$$T(V)(\rho) = \sup_a \left(\langle R, \rho \rangle + \gamma \sum_b \sup_c \left\langle \tilde{R}_{abc}, \rho \right\rangle \right) = \sup_d \left\langle \tilde{R}_d, \rho \right\rangle \tag{32}$$

So, this shows that the next value function is a piecewise linear (due to the inner product) and convex (due to the supremum).

The previous theorem shows that the value function admits a compact representation which can be used to design an approximation algorithm for the optimal policy. We first state formally the approximation problem.

APPROXIMATE POLICY EXISTENCE

Input: A quantum observable Markov decision process and $\epsilon \in (0,1)$.

Output: Is there a policy whose value is within an additive factor of ϵ from the optimal policy value?

Theorem 4. (see [13], Theorem 6.7) The problem of approximating the optimal policy value of a quantum observable Markov decision process is decidable.

Proof: Let V_0 be an arbitrary initial value function. We repeat

$$V_{n+1} = T(V_n) \tag{33}$$

until

$$\|V_{n+1} - V_n\| \leq \frac{(1-\gamma)\epsilon}{\gamma} \tag{34}$$

When this process halts, we have

$$\|V^\star - V_{n+1}\| \leq \|V^\star - T(V_{n+1})\| + \|T(V_{n+1}) - V_{n+1}\| \tag{35}$$

This is bounded by

$$\|T(V^\star) - T(V_{n+1})\| + \|T(V_{n+1}) - V_{n+1}\| \leq \gamma\|V^\star - V_{n+1}\| + \gamma\|V_{n+1} - V_n\| \tag{36}$$

So, this shows that

$$\|V^\star - V_{n+1}\| \leq \epsilon \tag{37}$$

The iteration will halt since T is contractive with modulus γ. Thus, after n steps where

$$n \sim \ln\left(\frac{1}{\epsilon}\right) \tag{38}$$

the fixed point iteration halts. Moreover, we can check the termination condition since the value function is a compact piecewise linear and convex map (which consists of finitely many components).

Remark 7: Given that Madani et al. [6] proved that most computational problems related to Markov decision processes are undecidable, the previous theorem provides one of the few examples of a tractable problem related to quantum Markov decision process.

4.2 Educational Outcomes

The course CS469/569 "Quantum Information and Computation" was offered at Clarkson University during the Spring 2019 semester (taught by Christino Tamon). The textbook used was "Quantum Computation and Quantum Information" by Nielse and Chuang [14]. The course webpage is available at:

https://afsweb.clarkson.edu/class/cs469/469-s19.html

The class was attended by 11 undergraduate students and 3 graduate students. The class met twice weekly (75 minutes each) for 13 weeks. There were 2 exams and 1 final project along with 6 homework assignments.

The list of topics covered include:

- *Basics of quantum information*. This covers Chapters 1 and 2. Some motivating examples, such as teleportation and non-local games, were given.
- *Quantum circuits*. This covers the basic parts of Chapter 4.
- *Shor's algorithm*. This covers Chapter 5 and the algorithms of Deutsch-Josza, Bernstein-Vazirani, Simon and Shor. The complete analyses of all algorithms were covered except that some number theoretic details of Shor's algorithm were not covered (since some students have not taken a number theory course).
- *Grover search*. This covers Chapter 6 which includes a comprehensive discussion of Grover's algorithm (viewed algebraically and geometrically).
- *Phase estimation*. This was covered as part of the proof of Shor's algorithm. But, here we show how this useful method is applied in the Harrow-Hassidim-Lloyd (HHL) algorithm for solving linear systems of equations (under certain very strong assumptions). The proof of the HHL algorithm was sketched but the full details were left as a possible project topic.
- *Noisy quantum theory*. We cover the basics of mixed states (density matrices), quantum channels, and quantum instruments. Although the text by Chuang and Nielsen contained a coverage of these topics, we also supplement these with material from the book by Wilde. We motivate this topic by considering noise in quantum computation.

The homework assignments were mainly theoretical exercises related to the lectures. The last assignment asked the students to explore some available quantum software platforms and their support for the basic quantum algorithms.

For the final project, students may work in groups and can choose a topic they wish to explore. The range of topics chosen by the students include amplitude amplification, topological quantum computation, divergence measures in quantum information, time-travel quantum computing, optimization for quantum circuits, and quantum cryptography.

4.3 Software Development Project

As an additional component of the project, we (Massimiliano Cutugno, Joshua Gordon, and Christino Tamon) designed and developed a graphical quantum circuit simulator. The simulator is written in Java which provides a built-in cross-platform support (since Java was designed to be a cross platform programming language). The basic graphical tool supports standard unitary quantum circuits and simple measurements (not unlike the IBM graphical tool). Our tool has potential features that might be useful to support a more general quantum computational model.

We currently use a git-based private repository (Bitbucket) for our software development. Our tool is called *QuaCC* (Quantum Circuit Compiler) and the latest publicly available version of our project can be found at:

https://github.com/MassiCuts/Quantum-Circuit

5. CONCLUSIONS

In this section, we provide some conclusions to our research, educational, and software development efforts in this project.

For the research component, we observed modest theoretical speedups for the classical optimization problem on partially observable Markov decision processes. The first type of speedup is based on the Grover search for maximum finding. This is used for evaluating the policy function in the value iteration algorithm. The second type of speedup is based on the HHL algorithm (which requires some strong assumptions for its application). This is used in matrix inversion for the policy iteration algorithm. Both speedups are heavily theoretical in nature and might be not practically relevant for the near future.

Our generalization of the quantum observable Markov decision process model (of Barry et al. [5]) is based on several standard notions from quantum information theory. These include conditional channels and quantum instruments. In our model, we combine these with two classical channels connecting the agent and the environment. These classical channels model possibly noisy communications between the agent and the environment. This is arguably more realistic for certain applications such as long-distance control of an autonomous robot in an unfamiliar domain. We plan to pursue this line of research for future work.

For the educational component, we developed and offered a quantum computing course which covered the fundamental topics in the area and incorporated connections with some recent developments (in both theory and practice). For future offerings of the course, we plan to spend more time covering the available quantum software platforms. As an example, given that one of our required computer science courses is offered in Haskell, we may include a coverage of Quipper (a Haskell-based quantum computing platform) which has one of the more impressive support for developing quantum algorithms. We might also increase the coverage of the use of platforms such as pyquil (Rigetti) and qiskit (IBM).

For the software development part, we have developed a prototype for a graphical tool that is useful for developing and testing quantum circuits. It also provides a useful front-end for pyquil. This tool is not dissimilar to some of the graphical tools currently available but it has potential to support noisy computation. We plan to continue maintaining and upgrading our software project in the future.

6. REFERENCES

[1] Bertsekas, D. and Shreve, S., **Stochastic Optimal Control: The Discrete Time Case**, Academic Press, 1978.

[2] Sutton, R. and Barto, A., **Reinforcement Learning: An Introduction**, second edition, The MIT Press, 2018.

[3] Blackwell, D., "Discounted Dynamic Programming," Ann. Math. Statist., 36 (1965), pp. 226-235.

[4] Sondik, E., "The Optimal Control of Partially Observable Markov Processes over the Infinite Horizon: Discounted Costs," Operations Research, 26:2 (1978), pp. 282-304.

[5] Barry, J., Barry, D., and Aaronson, S., "Quantum Partially Observable Markov Decision Processes," Physical Review A, 90 (2014), pp. 032311.

[6] Madani, O., Hanks, S., and Condon, A., "On the Undecidability of Probabilistic Planning and Related Stochastic Optimization Problems," Artificial Intelligence, 147 (2003), pp. 5-34.

[7] Durr, C. and Hoyer, P., "A Quantum Algorithm for Finding the Minimum," available online at arXiv:quant-ph/9607014.

[8] Harrow, A., Hassidim, A., and Lloyd, S., "Quantum Algorithm for Solving Linear Systems of Equations," Physical Review Letters, 15 (2009), pp. 150502.

[9] Wilde, M., **Quantum Information Theory**, Cambridge University Press, 2013.

[10] Watrous, J., **Theory of Quantum Information**, Cambridge University Press, 2018.

[11] Monras, A., Beige, A., and Wiesner, K., "Hidden Quantum Markov models and non-adaptive read-out of many body states," Applied Mathematics and Computational Science, 3 (2011), pp. 93-122.

[12] Eisert, J., Muller, M., and Gogolin, C., "Quantum measurement occurrence is undecidable," Physical Review Letters, 108 (2012), 260501.

[13] Tamon, C., and Xie, W., "A note on quantum Markov models," available online at arXiv:1911.01953 [quant-ph].

[14] Nielsen, M., and Chuang, I., **Quantum Computation and Quantum Information**, Cambridge University Press, 2000.

7. LIST OF SYMBOLS, ABBREVIATIONS AND ACRONYMS

MDP – Markov Decision Process

POMDP – Partially Observable Markov Decision Process

QOMDP – Quantum Observable Markov Decision Process