Optimization Methods for Mixed-Integer Control Problems in Complex Systems

by

Xinyu Fei

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Industrial and Operations Engineering and Scientific Computing)
in the University of Michigan
2024

Doctoral Committee:

Professor Siqian Shen, Chair
Assistant Professor Albert S. Berahas
Associate Professor Viswanath Nagarajan
Professor Lei Ying

Xinyu Fei

xinyuf@umich.edu

ORCID iD: 0000-0001-7010-8664

ACKNOWLEDGEMENTS

This acknowledgment demonstrates part of my deep appreciation to everyone who walks alongside me and empowers me in this journey.

First and foremost, I express my deepest gratitude to my advisor, Prof. Siqian Shen who has been an extraordinary mentor in both my academic career and personal growth. Her creativity, dedication, and resilience have shaped me into a better student, researcher, and person. I am deeply thankful for her wisdom, encouragement, and for showing me the heights that passion and hard work can reach.

I am exceptionally grateful to the members of my dissertation committee members, Prof. Albert S. Berahas, Prof. Viswanath Nagarajan, and Prof. Lei Ying. Their diverse perspectives and collective wisdom have greatly enhanced the quality of my work. My heartfelt thanks go to my collaborators, Dr. Lucas Brady, Dr. Sven Leyffer, Dr. Jeffery Larson, Prof. Yiheng Feng, Prof. Yafeng Yin, Prof. Henry Liu, who have contributed significantly to my professional development through their thoughtful feedback. I have also been deeply benefitted to work closely with Prof. Xian Yu, Dr. Xingmin Wang, Dr. Huiwen Jia, Gongyu Chen on different research topics. My sincere appreciation extends to my intern managers, Dr. Phillip Kriett and Dr. Andy Johnson, who have been offering valuable industry insights that have informed my research.

I am profoundly grateful to my best friend, Ransheng, for her unwavering support and encouragement. I would also like to thank my friends, Tongxin, Yisha, Qianwei, Jingwen, Moyan, Zhongzhu, Yaohui, Arlen, Lauren, Kati, Miao, Haotian. Their endless support and friendship have given me strength and happiness.

I owe a debt of gratitude to my parents, Rongxuan Zheng and Jineng Fei, whose love and sacrifices have laid the foundations of my achievements. A special note of thanks to my boyfriend, Depu Meng, for his love, understanding, and the countless ways he has contributed to my personal and academic life.

In sum, this dissertation stands as a testament to the collective efforts and contributions of all who have been part of my journey. Many thanks to everyone who has touched my life in the course of my doctoral studies.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

LIST OF ALGORITHMS

ALGORITHM

LIST OF APPENDICES

APPENDIX

ABSTRACT

Applying sophisticated optimization methods to control problems in various systems offers a structured approach for achieving system efficiency and reliability, while ensuring the feasibility of physics-based constraints. This dissertation primarily focuses on developing optimization algorithms for control problems in complex systems with mixed-integer variables and parameter uncertainties. In Chapter 1, we provide a detailed introduction about the problem motivation and summarize our problems and contributions. Specifically, we develop new optimization models and algorithms for network-based traffic control and binary control in quantum systems, detailed as follows.

Traffic signal control is an effective way of mitigating traffic congestion, where its complexity escalates in large cities due to numerous intersections and varying traffic conditions. Advancements in communication technologies within transportation infrastructures have made distributed traffic signal control at the network level possible and worth investigating. In Chapter 2, we build a stochastic optimization model for network-level traffic signal control under traffic demand uncertainty and solve it using decentralized algorithms. We compare the results with state-of-the-art traffic control methods via testing instances of real-world traffic networks and data.

Quantum computing and quantum systems provide a novel way of significantly accelerating computation and their operations largely rely on quantum pulse control optimization to attain desired states or status. The optimization of binary quantum pulse control can potentially enhance the performance of classical quantum variational techniques and improve solution quality. In Chapter 3, we study a discrete-valued binary quantum control problem and propose an algorithmic framework to solve the problem with penalty on switches. In Chapter 4, we explore the binary quantum control problem in a continuous time horizon by optimizing both control functions and control time intervals. Moreover, the time-varying noise in quantum systems and the wide use of inhomogeneous quantum ensembles highlight the need for quantum controls considering uncertainties. In Chapter 5, we develop a stochastic optimization model for the binary quantum control problem with uncertain Hamiltonian controllers and solve it using gradient-based methods with rounding algorithms. We test the performance of our approaches and benchmark with state-of-the-art control methods in all

three chapters. Lastly, in Chapter 6, we conclude the dissertation and state future research directions.

CHAPTER 1

Introduction

1.1 Background

Optimal control serves as the backbone for optimizing the performance and reliability of real-world systems [30, 31]. By making high-quality control decisions, we can significantly enhance system efficiency, mitigate associated risks, and achieve desired outcomes. The main focus of this dissertation is to explore and develop optimization methods for mixed-integer control problems in complex systems with three main challenges. First, all the systems have difficult physical constraints and mixed integer and continuous variables, resulting in computational complexity and the difficulty of attaining optimal solutions. Second, many control problems in real-world systems are large-scale, making centralized solution approaches impractical due to high computational costs. Third, real-world systems commonly include data collection errors, system noise, and fluctuations, which introduce various uncertainties that must be accounted for in our optimization approach. All these issues lead to extreme challenges and difficulties in solving mixed-integer control problems, especially under uncertainties, requiring the development of innovative optimization methods to design high-quality and robust controls.

1.1.1 Optimization Methods for Control Problems

Optimization methods are powerful solution approaches that capture the physical constraints and mixed-integer variables in systems, and integrate their physical nature into mathematical formulations that can be solved effectively. Therefore, employing various optimization methods for complex system control enables more accurate, robust, and scalable control solutions. In the literature, the Alternating Direction Method of Multipliers (ADMM), Benders decomposition, and the trust-region method are notably used due to their distinctive advantages. In the following paragraphs, we review these three widely recognized optimization algorithms, which will form the foundation of different algorithms we develop in this dissertation.

ADMM algorithm The ADMM algorithm [25] is known for its distributed optimization capabilities, which is crucial in the case that the decision variables are separated across different subsystems. The ADMM algorithm decomposes a larger problem into smaller subproblems, making it more manageable and scalable for real-world applications. The method is particularly beneficial in control optimization tasks where a distributed solution approach is required due to the decentralized nature of the systems [see e.g., 10, 58, 207].

The ADMM algorithm is initially developed for solving convex optimization problems which can be decomposed into small subproblems [25]. Specifically, the ADMM algorithm studies the following problem with decision variables $x \in \mathbb{R}^m,\ z \in \mathbb{R}^n$:

$$\begin{aligned} \min \quad & f(x) + g(z) && (1.1a) \\ \text{s.t.} \quad & Ax + Bz = c && (1.1b) \end{aligned}$$

where $A \in \mathbb{R}^{p\times m},\ B \in \mathbb{R}^{p\times n}, c \in \mathbb{R}^p$ are parameters. The algorithm formulates the augmented Lagrangian function as

$$L_\rho(x, z, \lambda) = f(x) + g(z) + y^T(Ax + Bz - c) + \frac{\rho}{2}\|Ax + Bz - c\|_2^2, \tag{1.2}$$

and consists of the following steps at iteration $t + 1$:

$$\begin{aligned} x^{t+1} &= \arg\min_x L_\rho(x, z^t, y^t) && (1.3a) \\ z^{t+1} &= \arg\min_z L_\rho(x^{t+1}, z, y^t) && (1.3b) \\ y^{t+1} &= y^t + \rho(Ax^{t+1} + Bz^{t+1} - c). && (1.3c) \end{aligned}$$

Boyd et al. [25] prove that both the objective value and solutions obtained by the ADMM algorithm converge to the optimal value and solutions for convex problems. We incorporate the ADMM algorithm into our control algorithms designed in Chapters 2– 3.

Benders decomposition By structurally separating problems into a master problem and multiple sub-problems, Benders decomposition [18] provides a systematic way of solving two-stage stochastic programs and is recognized for its ability to handle uncertainties that are common in many real-world control optimization problems [see e.g., 21, 69, 142].

Benders decomposition was first proposed by Benders [18] to solve mixed-integer linear programs. When the integer variables are fixed, the resulting problem is a linear program with strong duality which can be used to develop cuts [see e.g., 23, 90, 170, 187]. Specifically, we consider a general two-stage stochastic mixed-integer linear program with finite sampled realizations of uncertain parameters ξ. The first stage decision variable $x \in X$ is decided before we realize the uncertainty ξ where $X \subseteq \mathbb{R}^m$ or $X \subseteq \mathbb{Z}^m$ is the feasible region for x. The second stage decision variable $y \in \mathbb{R}^n$ is a continuous variable decided after we realize the uncertainty ξ. We denote K as the number of finite samples and $\xi_k = [W_k, h_k, T_k, q_k],\ k = 1, \ldots, K$ as specific realizations of the uncertainty ξ. We set p_k as the occurrence probability of scenario k such that $\sum_{k=1}^{K} p_k = 1$. The two-stage stochastic program has the following form:

$$\begin{aligned} \min \quad & c^T x + \sum_{k=1}^{K} p_k Q(x, \xi_k) && (1.4a) \\ \text{s.t.} \quad & x \in X && (1.4b) \end{aligned}$$

where X is a constraint set without uncertainty and $Q(x, \xi_k)$ is the optimal value of the following second-stage problem:

$$\begin{aligned} \min \quad & q_k^T y_k && (1.5a) \\ \text{s.t.} \quad & T_k x + W_k y_k = h_k,\ k = 1, \ldots, K && (1.5b) \\ & y_k \geq 0,\ k = 1, \ldots, K. && (1.5c) \end{aligned}$$

The traditional Benders decomposition algorithm solves a sequence of relaxed master problems and subproblems. We create new variable $\theta = (\theta_1, \ldots, \theta_K) \in \mathbb{R}^K$ and define a relaxed master problem at iteration t as

$$\begin{aligned} (\text{RMP}^t) \quad \min \quad & c^T x + \sum_{k=1}^{K} p_k \theta_k && (1.6a) \\ \text{s.t.} \quad & x \in X && (1.6b) \\ & (x, \theta) \in \Sigma^{t-1}, && (1.6c) \end{aligned}$$

where Σ^{t-1} is the set of Benders cuts as linear functions of x and θ generated up to the current iteration t from sample-based subproblems ($\Sigma^0 = \emptyset$). With dual variable π_k corresponding to constraints (1.5b), the subproblems are defined as the linear programming dual of the second-stage problems:

$$
\begin{aligned}
(\mathrm{SP}^k) \quad \max \quad & \pi_k^T(h_k - T_k x) && (1.7\mathrm{a}) \\
\text{s.t.} \quad & W_k \pi_k \leq q_k && (1.7\mathrm{b})
\end{aligned}
$$

Given a first-stage variable solution $\hat{x}^t$, one can solve subproblems (SP^k) with $x = \hat{x}^t$. Let $V^{k,t}$ and $R^{k,t}$ be the set of extreme points and rays of the feasible region of (SP^k) in iteration t, respectively ($V^{k,0} = R^{k,0} = \emptyset$). We start to update the set of Benders cuts from $\Sigma^t = \Sigma^{t-1}$. If the problem (SP^k) is unbounded, it means that the original second-stage problem is infeasible with fixed $\hat{x}^t$, therefore we update the Benders cuts Σ^t by adding feasibility cuts derived from $R^{k,t}$:

$$
\Sigma^t = \Sigma^t \cup \{(h_k - T_k x)^T \rho_k \geq 0,\ \rho_k \in R^{k,t}\} \tag{1.8}
$$

If the problem (SP^k) is bounded, let $Q_{k,t}$ be the optimal value. When $\theta_k^t < Q_{k,t}$, it means that the current solution $(\hat{x}^t, \hat{\theta}^t)$ is not optimal for the original problem. Therefore we update the Benders cuts Σ^t by adding optimality cuts derived from $V^{k,t}$:

$$
\Sigma^t = \Sigma^t \cup \{\theta_k \geq (h_k - T_k x)^T \nu_k,\ \nu_k \in V^{k,t}\} \tag{1.9}
$$

We go through all the subproblems for $k = 1, \ldots, K$ to update the Benders cuts set Σ^t. Then we go to the next iteration $t+1$ and solve (RMP^{t+1}). The optimal value of every (RMP^t) provides a lower bound to the original two-stage stochastic program and the combination of optimal values of the subproblems (SP^k), $c^T\hat{x}^t + \sum_{k=1}^K p_k Q_{k,t}$ provides an upper bound. The algorithm converges in a finite number of iterations because of the finite number of subproblems [187]. We propose our distributed algorithm for a two-stage stochastic program based on Benders decomposition in Chapter 3.

Trust-region method The Trust-Region method [47] is a well-established optimization technique used for solving problems with nonlinear constraints and objective functions [see e.g., 19, 42, 81, 217]. The method defines and adjusts the region where the estimated model function is trusted to represent the original objective function accurately and solves a series of subproblems [153].

Specifically, we consider a nonlinear optimization problem $\min_{x \in \mathbb{R}^n} f(x)$. The trust-region method starts from an initial feasible point x^0. At each iteration t, a model function $m^t(p)$

is designed to approximate the change of $f(x)$ as

$$m^t(p) = f(x^{t-1}) + p^T \nabla f(x^{t-1}) + \frac{1}{2} p^T \nabla^2 f(x^{t-1}) p, \tag{1.10}$$

where $\nabla f(x^{t-1})$ and $\nabla^2 f(x^{t-1})$ represent the gradient and the Hessian matrix of f, respectively. We solve the following subproblem to minimize the model function within a trust region with radius Δ^t:

$$\min_{p \in \mathbb{R}^n} \quad m^t(p) \tag{1.11a}$$

$$\text{s.t.} \quad \|p\| \leq \Delta^t. \tag{1.11b}$$

The choice of the radius Δ^t is decided by the agreement between the model function m^t and the objective function f at previous iterations. Given an optimal solution p^t of the subproblem, the predicted reduction is defined as $m^t(0) - m^t(p^t)$ and the actual reduction is defined as $f(x^{t-1}) - f(x^{t-1} + p^t)$. The ratio ρ^t is defined as the actual reduction divided by the predicted reduction. If ρ^t is high, it indicates that the model function m^t yields a good approximation to the objective function at the new point, then the radius Δ^t will be broadened, otherwise, the radius Δ^t will be narrowed. If the actual reduction is larger than a threshold, the current point x^t is updated by $x^t = x^{t-1} + p^t$, otherwise, it will keep the same as $x^t = x^{t-1}$. With proper update rules of trust region radius and assumptions of the objective function $f(x)$, the trust-region method has global converge to stationary points [153]. It has also been extended to solve nonlinear constrained optimization problems [35]. In Chapter 3, we design our improvement heuristic based on the trust-region method as a part of our algorithmic framework.

However, applying one type of the aforementioned methods is not sufficient for optimizing large-scale mixed-integer control problems in practice. For example, there are no general guarantees for the ADMM algorithm's convergence in non-convex problems [134]; the Benders decomposition can be computationally expensive with numerous added cuts and constraints in master problems and subproblems [73, 90]; the performance of the trust-region method in non-convex problems is highly dependent on its initial points for non-convex problems, which can limit its applicability [47]. Given these challenges, we propose algorithmic frameworks that integrate and enhance existing methods. In this dissertation, we consider two applications of control optimization: (i) signal control in traffic systems and (ii) pulse control in quantum systems, each with its unique challenges and opportunities.

1.1.2 Control in Traffic Systems

Traffic congestion, a common issue in growing cities worldwide, causes serious travel delays and environmental problems [88, 113, 186]. Among various solutions proposed to mitigate congestion, such as congestion pricing and road expansion, traffic signal control stands out for its effectiveness by optimizing the timing of green and red signals at intersections. However, the traffic signal control problem becomes notably challenging in large cities due to the numerous intersections, increasing the need for scalability of solution methods. Moreover, traffic conditions vary significantly during different time periods and days while changing a signal control plan requires complex tasks [238, 243]. Therefore, designing a traffic signal control plan that considers the uncertainties of traffic conditions and is applicable to large-scale traffic networks is valuable and challenging.

Traffic signal control is based on the modeling of traffic flow [120], consisting of models simulating the behavior of an individual vehicle by using dynamic variables to describe states [see e.g., 71, 149, 165, 230], and models simulating the traffic situation using variables to describe overall states of all vehicles [see e.g., 86, 126, 161, 237]. Daganzo [49] first develops a cell transmission model (CTM) to model traffic flows on highways, which divides roads into homogeneous sections and considers the states of each section across discrete time steps [3]. Lo et al. [132] apply the CTM on urban traffic networks and develop a dynamic traffic signal control model. Among all of the models, the CTM can describe traffic flows more precisely, with an acceptable computational complexity, and therefore, our optimization framework for the traffic signal control problem is based on the CTM.

Existing traffic signal control systems can be divided into three main categories: 1) fixed-time; 2) actuated; and 3) adaptive control according to their responsiveness and flexibility. Fixed-time traffic signal control utilizes offline optimization algorithms based on previously observed traffic data to set a predetermined signal timing plan [see e.g., 6, 7, 130, 148, 223]. and the most widely-used plan is generated by Webster's formula [223]. Muralidharan et al. [148] discusses the widespread use of fixed-time signal control due to its low cost and the challenges it faces. Actuated traffic signal control collects real-time data from sensors and applies pre-defined logic to decide signals. Adaptive control performs instant adjustments to traffic signal plans according to real-time traffic situations [see e.g., 1, 22, 67, 140, 192]. By Tang et al. [203], a large proportion of the traffic intersections in the US are still controlled by fixed-time traffic signals due to the absence of detectors. Even in areas with detectors, fixed-time control might remain in use, especially in congested urban areas. Therefore, we focus on designing fixed-time signal control under uncertainties in this dissertation.

1.1.3 Control in Quantum Systems

Quantum computing [151] is a cutting-edge field that applies the principles of quantum mechanics to perform faster computations and tackles complex problems unmanageable by classical computers. A vital component in quantum computing is quantum pulse control optimization, which aims to manipulate the evolution of quantum systems effectively to achieve desired quantum states or operations. Quantum pulse control was initially used in quantum chemistry but later found use in designing quantum circuits and creating new quantum algorithms [see e.g., 13, 100, 162, 233]. In this dissertation, we mainly focus on discrete binary quantum control with restricted feasible regions derived by linear constraints, which corresponds to the widely-used quantum approximate optimization algorithm [59] and other variational quantum algorithms [20, 37]. Moreover, high switching frequencies are undesirable in practical applications because they can damage the equipment in quantum systems. Furthermore, recent developments highlight the need for quantum control under uncertainties due to the presence of time-varying noise in quantum systems [see e.g., 50, 78, 79, 102, 167] and the common use of inhomogeneous quantum ensembles, which comprise multiple quantum systems with varied system parameters [see e.g., 39, 118, 141, 155, 168]. These underline the significance of binary quantum pulse control optimization and the extensions with fewer switches and uncertainties. The nonconvexity, uncertainty, binary variables, and restricted feasible regions lead to extreme challenges and difficulties in solving binary quantum control problems.

Various control methods have been employed to address challenges in quantum pulse control. The gradient ascent pulse engineering (GRAPE) algorithm is a widely-used method that estimates control functions with piece-wise constant functions and applies gradient-based methods to optimize them [see e.g., 52, 95, 109, 173]. Another classical method is the chopped random basis (CRAB) technique that describes the control space by a series of basis functions and optimizes the coefficients [see e.g., 36, 56, 147, 198]. Other methods for quantum control optimization include Lyapunov-based techniques [see e.g., 45, 46, 105, 106] and Krotov method [see e.g., 104, 144, 159, 196, 197, 204], and model-free methods such as evolution algorithm [232, 239], and reinforcement learning [32, 55, 152].

However, all the aforementioned control algorithms in quantum computing are designed for continuous controls, leaving a gap in the study of binary controls. Some literature consider the QAOA algorithm as a type of binary control problem with only two controllers, named bang-bang control, and propose solution methods [11, 124], but they are limited to situations with just two controllers. To the best of our knowledge, the paper by Vogt and Petersson [217] is the only work studying a binary quantum control problem, but it lacks the generality of quantum systems. In this dissertation, we explore a more general binary quantum control problem, extending our study to reduce switches and consider uncertainties, and propose various efficient quantum control design algorithms. Our research aims to fill in the existing knowledge gaps and provide a more comprehensive understanding of binary controls in quantum systems.

1.2 Dissertation Overview

The remainder of the dissertation is organized as follows.

In Chapter 2, we study a coordinated traffic signal control problem in large-scale traffic networks, aiming to maximize vehicle throughput on roads or networks under uncertain traffic demand and vehicle turning. We build a two-stage stochastic mixed-integer linear program with finite samples of uncertain parameters and develop a spatial-temporally distributed algorithm combining Bender decomposition and ADMM with an optimality guarantee. Through testing on corridors and grid networks with synthetic and real-world traffic data, our results show that considering traffic uncertainties significantly improves signal control quality, and our distributed algorithm can quickly find high-quality signal plans for multiple intersections in complex road networks. The work in Chapter 2 has been published in Fei et al. [66].

In Chapter 3, we focus on a deterministic discrete binary quantum pulse control optimization problem. We develop a generic mixed-integer non-convex optimization model and extend it to handle additional side constraints and reduce the number of switches. We adjust the widely recognized gradient ascent pulse engineering (GRAPE) algorithm and introduce a new ADMM algorithm to solve the continuous relaxation of the model. We employ rounding techniques to obtain binary control solutions and prove the theoretical upper bounds of the rounding error. Furthermore, we design a modified trust-region method to improve the controls. Our numerical studies on diverse quantum control examples demonstrate that our algorithms obtain high-quality control results with few switches. The work in Chapter 3 has been published in Fei et al. [64].

In Chapter 4, we consider a binary quantum pulse control optimization problem in a continuous time horizon. Unlike the optimization problem with time discretization in Chapter 3, we develop a new algorithmic framework that not only optimizes control functions but also time points switching among controllers. Specifically, we develop two heuristic methods to obtain controller sequences from continuous discretized controls balancing the quality of controls and the number of switches. Then, we build and solve a generic switching time optimization model with given Hamiltonian quantum controllers as parameters to optimize the switches of control functions. Additionally, we introduce a new technique to accelerate time-evolution simulations in classical computers by pre-computing the eigenvalues of a small number of Hamiltonian matrices. Our numerical simulations indicate that our switching time optimization framework obtains controls with higher quality and similar switches within dramatically less computational time compared to the discretized model. The work in Chapter 4 has been published online in Fei et al. [65].

In Chapter 5, we concentrate on a discrete-valued binary quantum pulse control optimization problem considering the uncertainty of Hamiltonian controllers. We extend our previous deterministic model proposed in Chapter 3 to a mixed-integer stochastic program with a sample-based reformulation aiming to optimize both risk-neutral and risk-averse measurements. We explore the differentiability of the new sample-based objective function and apply two gradient-based algorithms to solve the continuous relaxation. We apply and improve a sum-up-rounding technique to round continuous relaxation solutions and prove the gaps between continuous and binary controls. Through numerical experiments on various examples of quantum pulse optimization, we evaluate the impact of uncertainties in quantum systems and demonstrate that the controls derived from our stochastic optimization model outperform the controls of the deterministic model significantly in terms of quality and robustness.

In Chapter 6, we conclude the work in this dissertation and discuss future research directions. We are interested in combining mixed-integer programs with learning techniques for optimal control problems. Extending our algorithms to other areas such as quantum algorithm design and route recommendation is another potential direction.

1.3 Main Contributions

Overall, the contributions of this dissertation are outlined below. First, we consider a network-level traffic signal control problem with various uncertainties and formulate it as a two-stage mixed-integer stochastic program. We propose a spatial-temporally distributed algorithm to accelerate computation and guarantee optimality. Our numerical results show that our distributed algorithm handles problem sizes unmanageable by a centralized solver and reduces traffic delays. Second, we bridge the gap between optimization methods and binary quantum pulse control problems. We propose a generic model with restricted linear feasible regions and develop an algorithmic framework to solve the model by time discretization. The numerical results indicate that our methods obtain high-quality binary control sequences and prevent frequent switching. Third, we derive a new algorithm optimizing both control functions and switching time points. With the pre-computation of Hamiltonian controllers, we accelerate time-evolution simulations on classical computers. This method obtains controls with higher quality and similar switching frequencies within significantly less computational time. Fourth, we build a stochastic optimization model with a sample-based formulation for the binary quantum control under time-varying noise and propose an algorithm combining gradient-based methods and rounding techniques. We also provide a theoretical discussion on the choices of risk measure function and gaps of rounded solutions. We analyze control performance under various examples and highlight the benefits of considering uncertainties in quantum systems.

CHAPTER 2

Traffic Signal Control under Stochastic Traffic Demand and Vehicle Turning via Decentralized Decomposition Approaches

2.1 Introductory Remarks

In the past few decades, city scales have increased significantly and as a result, privately owned vehicles increase, resulting in rapidly growing congestion issues in cities of all sizes worldwide. According to Schrank et al. [186], in 2017, traffic congestion caused urban Americans to travel extra 8.8 billion hours and to purchase extra 3.3 billion gallons of fuel. Taking the city of Detroit in the United States (US) as an example, in 2019, each driver lost 39 hours on the road on average due to traffic delay [113]. Various approaches have been proposed for addressing traffic congestion issues on road networks, including congestion pricing, road expansion, and traffic signal control. Among them, traffic signal control can effectively mitigate congestion by optimizing the traffic signal timing parameters (cycle, green split, and offset) at signalized intersections without major changes to the existing infrastructure [88].

Existing traffic signal control systems can be divided into three main categories: 1) fixed-time; 2) actuated; and 3) adaptive control according to their responsiveness and flexibility. Both actuated and adaptive are real-time control strategies that can react to time-varying traffic demand and outperform fixed-time control in most cases. However, a large proportion of the traffic intersections in the US are still controlled by fixed-time traffic signals due to the absence of detectors [203]. Even with detectors on the road, fixed-time control might still be used in congested urban areas. For fixed-time traffic signals, a whole day is split into different time of day (TOD) intervals and each of the TOD uses a traffic signal timing plan with fixed parameters (cycle, split, and offset). However, traffic conditions are stochastic and can vary within the same time periods [235, 238], and changing a signal timing plan is infrequent as it requires a series of labor works such as monitoring the traffic systems, tuning the parameters, and setting new parameters to the controller [243]. Therefore, a fixed-time traffic signal timing plan is required to accommodate different traffic conditions within a certain time of day.

In this chapter, we focus on the problem of optimizing fixed-time traffic signal parameters of a large-scale network considering stochastic traffic input. Based on the Cell Transmission Model (CTM) [49], this problem is formulated as a mixed-integer stochastic linear programming. There are different types of network topology in the real world including isolated intersections, corridors, and grid networks [209]. Our proposed method aims to deal with large-scale networks and does not limit the network topology to a certain type. Such a centralized formulation based on CTM can better incorporate the coordination among intersections even for a complicated network topology while traditional methods can only deal with traffic signals in a corridor [209].

For the traffic demand, our proposed method takes the link flow and turning ratios as the direct input and both parameters can be stochastic. Even without detectors, link flow and turning ratios can be estimated from other data sources. For example, vehicle trajectory data that can be collected through different resources (e.g., ride-hailing services and navigation systems) is an ideal alternative, which is more scalable and almost available at every signalized intersection. Limited by the low penetration rate, it can hardly support real-time applications but is more applicable for offline parameter estimation [221, 241, 245, 246, 247]. Other than the vehicle trajectory data, if origin-destination (OD) demand is given, an equilibrium model can be used to obtain the link flow and turning ratios [166, 188]. As aforementioned, traffic demand is stochastic and time-varying and all measurements are also prone to errors and noises [43, 235]. By taking a stochastic traffic demand into consideration, the resulting traffic signal timing plan is more robust and works well for more different cases.

Our goal is to establish an optimization-based control paradigm to speed up solutions to large-scale network-level traffic signal control *in a distributed manner*. We demonstrate the results by testing a diverse set of instances, generated using synthetic data and also real-world road networks and traffic data. Via out-of-sample tests, we show that the signal control plans produced by our methods perform consistently better than plans solved using off-the-shelf optimization solvers, or models that do not consider uncertainty. Our decomposition and decentralized algorithms can significantly reduce computational time and produce reliable traffic signal plans for moderate-sized city networks.

The remainder of the chapter is organized as follows. In Section 2.2, we review the most relevant literature on traffic signal control and optimization methods used in this chapter. In Section 2.3, we formulate the traffic signal control problem under uncertain demand and turning ratios as a two-stage stochastic mixed-integer program (MIP). In Section 2.4, we combine the alternating direction method of multipliers (ADMM) and Benders decomposition algorithms to develop spatially distributed algorithms for solving the stochastic MIP in a decentralized manner. In Section 2.5, we present numerical results of diverse instances with different uncertainty settings. In Section 2.6, we conclude the chapter and state future research directions.

2.2 Literature review

Traffic flow models have different scales from microscopic to macroscopic [see, e.g., 86, 120]. A microscopic model simulates the behavior of individual vehicles using dynamic variables to describe states such as location and velocity of vehicles [165, 230]. A meso- or macro-scale model simulates traffic conditions using variables that describe the overall states of all vehicles, including traffic density, volume, and average speed [86]. Daganzo [49] first develops a cell transmission model (CTM) to model traffic flows on highways. As a mesoscopic traffic model, the CTM divides roads into homogeneous sections and considers states of each section across discrete time steps. We refer interested readers to Adacher and Tiriolo [3] for a comprehensive review of different CTM variants and appropriate situations for using them. Compared to solving microscopic traffic models, computing macroscopic models is much more efficient because of the aggregated traffic states and the resulting fewer number of variables and dynamic equations. In this chapter, our optimization framework is based on the CTM for traffic condition modeling. Next, in Section 2.2.1, we review the most relevant work on distributed traffic signal control, especially with respect to ADMM. Section 2.2.2 is a literature review of stochastic traffic signal control and Section 2.2.3 includes contributions of this chapter.

2.2.1 Distributed Traffic Signal Control

Many existing studies formulate the traffic signal control problem to optimize the network performance based on a certain traffic flow model. Lo [131] builds an MIP based on the CTM for deriving traffic signal control policies but does not consider traffic demand or vehicle turning uncertainties. The model is only tested on corridors instead of general road networks, mainly due to the exponentially increased number of variables and constraints in the latter case. Indeed, solving an MIP requires gathering global traffic information in a road network and becomes extremely difficult as the network size increases. On the other hand, each intersection is physically isolated, making it natural to consider decentralized traffic signal control. Advancements in parallel computing, sensing, and wireless communication technologies allow efficient implementations of decentralized control at individual intersections, leading to more research focusing on how to design parallel and distributed traffic signal control schemes. For example, Al Islam and Hajbabaie [4] present a distributed-coordinated approach for signal timing optimization in connected urban street networks. They first, formulate the traffic signal control problem at each intersection as an MIP. Then, after separately solving the MIP associated with each intersection, all the intersections communicate with each other to improve their local traffic control strategies (guided by heuristics), obtain estimated traffic flows, and update parameters in their MIPs for the next time period. Tajalli et al. [202] also formulate the traffic signal control problem as an MIP based on CTM for each intersection but consider the coordination of intersections in a certain time period by penalizing the difference between estimated and real traffic flow in the objective function. Furthermore, a series of literature follows the procedure that first fixes traffic signals for each intersection and runs simulations on the overall traffic network, and then updates traffic signals according to the estimated information from the simulation [see, e.g., 82, 125, 201]. All of the aforementioned papers fail to ensure the optimality of the solution and do not consider traffic uncertainties.

ADMM was initially developed to solve convex programs that can be decomposed into multiple sub-clusters and the objective function is the summation of functions related to each sub-cluster. For convex programs, it is proved that both the objective value and solutions obtained by ADMM converge to the optimum [25]. Timotheou et al. [207] propose a distributed algorithm that decomposes a traffic network into individual intersections for traffic signal control and they apply ADMM to obtain the coordination between intersections as well as solving the MIP for each intersection individually. However, vehicle turning is not considered in their model and the work does not take into account the stochasticity of traffic demand or vehicle turning.

Another class of literature is about the max pressure control, which is originally studied in the communication network for routing and scheduling [150, 205]. Varaiya [211] firstly introduces the pressure-based method to solve the distributed traffic signal control problem by assigning traffic signals according to queue lengths of different directions. The max pressure control is derived based on the store-and-forward model and proved to be throughput-optimal, which means it can stabilize the network queue lengths when the traffic demand is within the network capacity. Many studies have been conducted to improve the max pressure control from different perspectives [112, 123, 222, 229]. For example, Zaidi et al. [240] modify the original max pressure method to conduct traffic signal control taking fixed and adapt routing into consideration in a general traffic network. Levin et al. [112] propose to utilize a cyclic phase structure which is easier to be implemented in the real world.

However, the store-and-forward model has some strong assumptions which essentially ignore the vehicle length and travel time along the link [2]. Besides, the max pressure control only ensures the stability of the network by showing that the overall network queue lengths are bounded. According to Little's law [129], bounded total queue lengths guarantee a bounded total delay but not the optimal minimum delay. By taking the real-time traffic state (e.g., queue length) as the input, the implementation of the max pressure control has a high requirement for real-time traffic monitoring and detection, which is not available for a large proportion of signalized intersections. As aforementioned, this chapter focuses on generating background or fixed-time traffic signal timing for those intersections without real-time monitoring capability.

2.2.2 Traffic Signal Control under Uncertainties

Starting from Heydecker [85] investigating the impact of uncertain traffic flows and showing that using the average flow results in unsatisfactory performance, a series of studies have been conducted on proposing traffic signal timing plan under flow fluctuations. Yin [235] and Zhang et al. [243] consider fixed-time signal timing plans for road corridors under stochastic traffic demand. They ensure that the probability of the travel delay exceeding a given threshold is sufficiently small and apply heuristics to solve the stochastic MIPs. Based on their work, Shirke et al. [191] propose a metaheuristic approach to design robust traffic signal timing plans efficiently. Another class of existing literature focuses on optimizing the average performance given the distribution of uncertain parameters. For example, Tong et al. [208] propose a two-stage stochastic linear program (LP) for controlling traffic signals at one intersection to minimize the expected traffic delay under uncertain traffic demand. Li et al. [122] extend the previous work and formulate a two-stage stochastic LP for coordinated traffic signal control on a corridor. However, the models in both Tong et al. [208] and Li et al. [122] relax the integrality constraints. Li et al. [121] propose a two-stage MIP and apply a heuristic stochastic gradient method to solve the model. Moreover, in all these related work, the authors only consider one intersection or corridors instead of complex road networks. For general traffic networks, Chiou [41] develop a two-stage mixed-integer nonlinear program based on user equilibrium instead of the CTM and apply a Quasi-Newton method to solve the model in centralization.

2.2.3 Main Contributions

Compared to the prior work, the main contributions of this chapter are threefold. First, we consider a fixed-time traffic signal control problem and formulate it on general grid networks as an MIP based on the CTM, which takes into account the queue propagation along links and optimizes traffic delay directly. We consider both vehicle movements along corridors and turning movements of vehicles, to obtain better coordination between intersections within a traffic network. Second, we extend the deterministic MIP to a stochastic one by incorporating various types of input parameter uncertainties, such as stochastic traffic demand and turning ratios. Third, we speed up the computation of the formulation by integrating Benders decomposition and ADMM approaches. Specifically, we ignore the traffic flow relationship between intersections and obtain a signal timing plan for each intersection. Then, we coordinate the information of neighboring intersections and update the parameters of each intersection. Such a distributed computational framework can best utilize advanced technologies in intelligent transportation systems such as distributed micro-computers – that can be installed at traffic intersections. In Table 2.1, we compare our work with the most relevant papers, in terms of modeling methods, instance scales, types of decisions, assumptions on parameters, and solution approaches.

Table 2.1: Comparison between our work and other traffic signal control papers.

Paper	Model	Scale	Decision	Parameter	Approach
Our work	Two-stage stochastic MIP based on the CTM	network	fixed-time	stochastic traffic demand and turning ratio	decentralized
Lo [131]	Deterministic MIP	corridor	real-time	deterministic	centralized
Yin [235], Zhang et al. [243], Shirke et al. [191]	Stochastic MIP	corridor	fixed-time	stochastic traffic demand	centralized
Tong et al. [208]	Two-stage stochastic LP	isolated intersection	real-time	stochastic traffic demand	centralized
Li et al. [122]	Two-stage stochastic LP	corridor	real-time	stochastic traffic demand	centralized
Li et al. [121]	Two-stage stochastic MIP	corridor	real-time	stochastic traffic demand	centralized
Chiou [41]	Two-stage stochastic MIP	network	fixed-time	stochastic traffic demand	centralized
Tajalli et al. [202], Timotheou et al. [207]	Deterministic MIP	network	real-time	deterministic	decentralized
Varaiya [211], Zaidi et al. [240]	Pressure-based method	network	real-time	deterministic	decentralized

2.3 A Stochastic MIP for Traffic Signal Control

We define the notation and introduce CTM in Section 2.3.1. We then formulate the fixed-time traffic signal control as a two-stage stochastic MIP using CTM in Section 2.3.2, where we aim to maximize the expected total throughput (i.e., the number of vehicles going through the whole traffic network). The model takes the initial traffic condition, samples, and probabilities of source demand and vehicle turning ratios as input parameters and computes optimal signal control plans at all intersections.

2.3.1 Problem Description and Notation

Structure of the CTM Consider a road network $G(V, E)$ where V is the set of nodes and E is the set of arcs. Here the nodes are signalized intersections and the arcs refer to road segments connecting pairs of intersections. In this chapter, we focus on road networks in which at most four arcs are connected to the same node, representing North, South, East, and West directions and therefore the underlying network is a grid. Each arc is partitioned into homogeneous sections called cells, where the length of each cell is the distance traveled by a vehicle at normal speed without traffic congestion in a unit of time. Let $\mathcal{C} = \mathcal{E} \cup \mathcal{O} \cup \mathcal{D} \cup \mathcal{I} \cup \mathcal{M} \cup \mathcal{V}$ be the set of cells where $\mathcal{E}$ is the set of ordinary cells, $\mathcal{O}$ the set of origin cells, $\mathcal{D}$ the set of destination cells, $\mathcal{I}$ the set of intersection cells, $\mathcal{M}$ the set of merge cells and $\mathcal{V}$ the set of diverge cells. Ordinary cells have both inflow and outflow of vehicles from other cells. Origin cells receive exogenous inflow and destination cells send outflow traffic outside the network. Intersection cells are cells where vehicles can choose to turn left, go straight, or turn right. Merge cells receive inflow traffic from more than one cell, and diverge cells send outflow traffic to more than one cell. Let $d(c)$ be the set of cells sending their outflow to a cell $c \in \mathcal{C}$ and $p(c)$ be the set of cells receiving their inflow from a cell $c \in \mathcal{C}$. Let $\mathcal{R} = \{1, 2, \ldots, N_I\}$ be the set of intersections where N_I denotes the total number of intersections.

A signal phase of a signalized intersection is a representation of moving directions that can allow vehicles to pass the intersection at the same time. Let $\mathcal{F}_i = \{1, 2, \ldots, |\mathcal{F}_i|\}$ be the set of indices of all the phases of a signalized intersection i for all $i \in \mathcal{R}$. Let I_{ij} be the set of the intersection cells of intersection i and phase j for all $i \in \mathcal{R}$ and $j \in \mathcal{F}_i$. We discretize the whole time horizon into T time steps. Figure 2.1 depicts the way of transforming one intersection and its related arcs into cells in CTM. We refer the interested readers to the legend in Figure 2.1 for different types of the aforementioned cells in the CTM and the traffic flow.

Input Parameters Let n_c^{init} be the number of initial vehicles inside each cell $c \in \mathcal{C}$, and let $\beta_{cc'}$ denote the turning ratio of a diverging cell $c \in \mathcal{V}$ moving towards the direction of an intersection cell $c' \in d(c)$. Notice that $\sum_{c' \in d(c)} \beta_{cc'} = 1$ for each $c \in \mathcal{V}$. Let Q_{ct} and N_{ct} respectively represent the maximum number of vehicles that can flow through and reside in cell $c \in \mathcal{C}$, and W_{ct} be the ratio between the shock-wave propagation speed and the flow-free speed of a cell c during each time interval $[t, t+1)$ for $t \in \{1, \ldots, T\}$. Moreover, D_{ct} denotes the number of vehicles entering an origin cell $c \in \mathcal{O}$ during time $[t, t+1)$ (i.e., the source demand considered in this chapter). Parameter $G_{\min}$ and $G_{\max}$ indicate the minimum and maximum green time, meaning that any green signal set at time t cannot change during time $[t, t + G_{\min})$ and the green signal of the same direction cannot last for more than $G_{\max}$ time [242], respectively. Let N_{cy} be the maximum number of cycles in the whole time horizon where traffic signal plans are the same in each cycle. We use $n^{\text{init}}, \beta, Q, N, W, D$ to represent

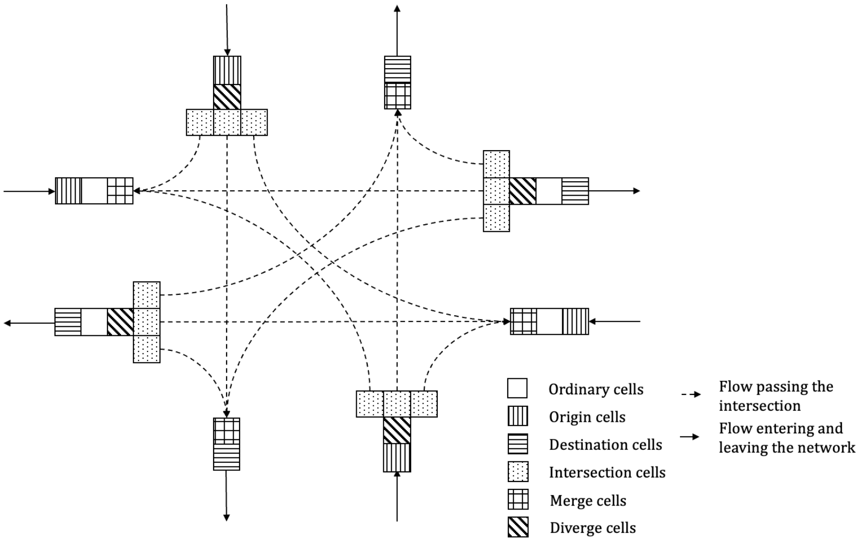


Figure 2.1: A CTM-based transformation of an intersection.

the vector forms of all the above parameters, respectively. In addition, we introduce a sufficiently large parameter U, a sufficiently small parameter ϵ, and a weight parameter α to formulate constraints and the objective function in our stochastic MIP described later.

We denote $\xi = [D, \beta]$ as the overall uncertain parameter and P as the probability distribution of ξ, which is assumed known and can be derived from historical data. Without loss of generality, we assume a discrete distribution P and a finite set Ξ of realizations such that $\Xi = \{\xi^1, \ldots, \xi^K\}$ and each realization ξ^k is associated with probability p^k such that $\sum_{k=1}^{K} p^k = 1$.

Decision Variables We determine the traffic signal control plan by finding the beginning time and ending time when the traffic signal is green for each phase and each cycle. Correspondingly, we define a continuous variable l_i as the cycle length and a continuous variable o_i as the offset of each intersection $i \in \mathcal{R}$. Variable g_{ij} indicates the interval length when the traffic signal of an intersection i and a phase j is green, and continuous variables b_{ijm} and e_{ijm} indicate the beginning and ending green time at intersection i, phase j during cycle m, for each $i \in \mathcal{R}$, $j \in \mathcal{F}_i$, and $m \in \{1, \ldots, N_{\text{cy}}\}$. For $t \in \{1, \ldots, T\}$, we define binary variables z_{1ijmt} and z_{2ijmt} to describe the relationship between the time step t, the beginning green time b_{ijm} and the ending green time e_{ijm} of the intersection i, phase j and cycle m, of which the details are provided in constraints (2.1a)–(2.1c). For each cell $c \in \mathcal{C}$ and time $t \in \{1, \ldots, T\}$, we define continuous variables y_{ct} and n_{ct} as the number of vehicles leaving and inside cell c during time $[t, t+1)$, respectively. We use $l, o, b, e, g, z_1, z_2, y, n$ to represent the vector forms of all the above continuous and binary decision variables.

2.3.2 Two-stage Stochastic Optimization Model

We consider a two-stage stochastic optimization approach for controlling traffic signals, where in the first stage, we determine traffic-signal related decisions, such as the cycle length l, offset o, start and end of green time intervals b, e and auxiliary binary variables z_1, z_2; in the second stage, for each realized sample $\xi^k = [D^k, \beta^k]$, we can use network flow constraints to build a linear program to determine the actual number of vehicles leaving and inside cells, given by values of variables y^k and n^k, respectively. Table 2.2 provides a summary of the definitions of all the sets, parameters, and variables used in this chapter.

Table 2.2: A summary of notation of the stochastic traffic signal control problem.

Parameters	
$\mathcal{C}$	Set of all cells (where $\mathcal{C} = \mathcal{O} \cup \mathcal{D} \cup \mathcal{I} \cup \mathcal{M} \cup \mathcal{V}$)
$\mathcal{O},\ \mathcal{D},\ \mathcal{I},\ \mathcal{M},\ \mathcal{V}$	Set of origins, destinations, intersection, merge, diverge cells
N_I	The total number of intersections
$\mathcal{R}$	Set of intersections (where $\lvert\mathcal{R}\rvert = N_I$)
$\mathcal{F}_i$	Set of signal phases at intersection i
T	The total number of time steps
N_{cy}	The total number of cycles in the traffic signal timing plan
$G_{\min},\ G_{\max}$	Minimum and maximum green time
$d(c)$	Cells receiving inflow from $c \in \mathcal{C}$
$p(c)$	Cells sending outflow to $c \in \mathcal{C}$
n_c^{init}	Number of initial vehicles inside cell $c \in \mathcal{C}$
Q_{ct}, N_{ct}	Maximum number of vehicles flowing through and reside in cell $c \in \mathcal{C}$ during time $[t, t+1)$ for $t \in \{1, \dots, T\}$
W_{ct}	Ratio between shock-wave and free-flow speed of cell $c \in \mathcal{C}$ during time $[t, t+1)$ for $t \in \{1, \dots, T\}$
D_{ct}^k	Source demand realization at origin $c \in \mathcal{O}$ during time $[t, t+1)$ in scenario $k = 1, \dots, K$
$\beta_{cc'}^k$	Turing ratio realization from cell $c \in \mathcal{V}$ to cell $c' \in d(c)$ in scenario $k = 1, \dots, K$
Decision variables	
$l_i,\ o_i$	Cycle length and offset of intersection $i \in \mathcal{R}$
g_{ij}	Green length of intersection $i \in \mathcal{R}$ at phase $j \in \mathcal{F}_i$
$b_{ijm},\ e_{ijm}$	Beginning and ending time of the green phase $j \in \mathcal{F}_i$ in cycle $m = 1, \dots, N_{\text{cy}}$ of intersection $i \in \mathcal{R}$
$z_{1ijmt},\ z_{2ijmt}$	Binary variables describing the relationship between $b_{ijm},\ e_{ijm}$ and time $[t, t+1)$ for $t \in \{1, \dots, T\}$
$y_{ct}^k,\ n_{ct}^k$	Number of vehicles leaving and inside cell $c \in \mathcal{C}$ during time $[t, t+1)$ for $t \in \{1, \dots, T\}$ in scenario $k = 1, \dots, K$

We first formulate all the constraints in the first stage for configuring a feasible traffic signal timing plan, and denote the feasible region as:

$$X = \Big\{ \quad -U \cdot z_{1ijmt} \le b_{ijm} - t \le U(1 - z_{1ijmt}) - \epsilon,\ \forall i \in \mathcal{R},\ \forall j \in \mathcal{F}_i,\ m = 1, \dots, N_{\text{cy}},\ t = 1, \dots, T \tag{2.1a}$$

$$-U \cdot z_{2ijmt} + \epsilon \le t - e_{ijm} \le U(1 - z_{2ijmt}),\ \forall i \in \mathcal{R},\ \forall j \in \mathcal{F}_i,\ m = 1, \dots, N_{\text{cy}},\ t = 1, \dots, T \tag{2.1b}$$

$$\textstyle\sum_{j \in \mathcal{F}_i} (z_{1ijmt} + z_{2ijmt}) \le \lvert\mathcal{F}_i\rvert + 1,\ \forall i \in \mathcal{R},\ m = 1, \dots, N_{\text{cy}},\ t = 1, \dots, T \tag{2.1c}$$

$$o_i \le l_i,\ \forall i \in \mathcal{R} \tag{2.1d}$$

$$b_{i1m} = l_i \cdot (m - 1) - o_i,\ \forall i \in \mathcal{R},\ m = 1, \dots, N_{\text{cy}} \tag{2.1e}$$

$$e_{ijm} = b_{ijm} + g_{ij},\ \forall i \in \mathcal{R},\ \forall j \in \mathcal{F}_i,\ m = 1, \dots, N_{\text{cy}} \tag{2.1f}$$

$$b_{ijm} = e_{ij-1m},\ \forall i \in \mathcal{R},\ \forall j \in \mathcal{F}_i / \{1\},\ m = 1, \dots, N_{\text{cy}} \tag{2.1g}$$

$$l_i = \textstyle\sum_{j \in \mathcal{F}_i} g_{ij},\ \forall i \in \mathcal{R} \tag{2.1h}$$

$$G_{\min} \le g_{ij} \le G_{\max},\ \forall i \in \mathcal{R},\ \forall j \in \mathcal{F}_i \tag{2.1i}$$

$$z_{1ijmt},\ z_{2ijmt} \in \{0, 1\},\ \forall i \in \mathcal{R},\ \forall j \in \mathcal{F}_i,\ m = 1, \dots, N_{\text{cy}},\ t = 1, \dots, T \Big\}. \tag{2.1j}$$

Constraints (2.1a)–(2.1b) describe the relationship between each time step t, the start time b and the end time e of the time interval when the traffic signal of each phase is green. For each intersection $i \in \mathcal{R}$, each phase $j \in \mathcal{F}_i$ and each cycle $m \in \{1, \dots, N_{\text{cy}}\}$ at each time step t, $z_{1ijmt} = 1$ if $t \geq b_{ijm}$ and $z_{1jmt} = 0$ otherwise. Similarly, $z_{2ijmt} = 1$ if $t \leq e_{ijmt}$ and $z_{2ijmt} = 0$ otherwise. Constraints (2.1c) indicate that at each time step, there is only one phase with green light. Constraints (2.1d) indicate that the offset should be less than the cycle length. Constraints (2.1e)–(2.1g) detail the steps for computing the start and the end of the time interval at each cycle, when the traffic signal of each phase is green based on the cycle length and the length of the interval. Constraints (2.1h) indicate that the sum of the green time over all phases should be equal to the cycle length. Constraints (2.1i) bound the green time from below and above. Constraints (2.1j) require z_1 and z_2 being binary valued.

The overall two-stage stochastic MIP is formulated as:

$$\min_{l,g,b,e,o,z_1,z_2,y,n} \quad \sum_{k=1}^{K} p^k \left(-\sum_{c \in \mathcal{D}} \sum_{t=1}^{T} n_{ct}^k + \alpha \sum_{c \in \mathcal{C}} \sum_{t=1}^{T} (T-t) y_{ct}^k \right) \tag{2.2a}$$

$$\text{s.t.} \quad (l, o, g, b, e, z_1, z_2) \in X$$

$$y_{ct}^k \leq n_{ct}^k, \ \forall c \in \mathcal{C}, \ t = 1, \dots, T, \ k = 1, \dots, K \tag{2.2b}$$

$$y_{ct}^k \leq Q_{ct}, \ \forall c \in \mathcal{E} \cup \mathcal{O} \cup \mathcal{M} \cup \mathcal{V}, \ t = 1, \dots, T, \ k = 1, \dots, K \tag{2.2c}$$

$$y_{ct}^k \leq \sum_{m=1}^{N_{\text{cy}}} (z_{1ijmt} + z_{2ijmt} - 1) Q_{ct}, \ \forall c \in \mathcal{I}_{ij}, \ \forall i \in \mathcal{R}, \ \forall j \in \mathcal{F}_i, \ t = 1, \dots, T, \ k = 1, \dots, K \tag{2.2d}$$

$$y_{ct}^k \leq Q_{c't}, \ \forall c \in \mathcal{C}/\mathcal{V}, \ \forall c' \in d(c), \ t = 1, \dots, T, \ k = 1, \dots, K \tag{2.2e}$$

$$\beta_{cc'}^k y_{ct}^k \leq Q_{c't}, \ \forall c \in \mathcal{V}, \ \forall i \in \mathcal{R}, \ \forall j \in \mathcal{F}_i, \ \forall c' \in d(c) \cap I_{ij}, \ t = 1, \dots, T, \ k = 1, \dots, K \tag{2.2f}$$

$$y_{ct}^k \leq W_{c't}(N_{c't} - n_{c't}^k), \ \forall c \in \mathcal{C}/\mathcal{V}, \ \forall c' \in d(c), \ t = 1, \dots, T, \ k = 1, \dots, K \tag{2.2g}$$

$$\beta_{cc'} y_{ct}^k \leq W_{c't}(N_{c't} - n_{c't}^k), \ \forall c \in \mathcal{V}, \ \forall c' \in d(c), \ t = 1, \dots, T, \ k = 1, \dots, K \tag{2.2h}$$

$$n_{ct+1}^k = n_{ct}^k + \sum_{c' \in p(c)} y_{c't}^k - y_{ct}^k, \ \forall c \in \mathcal{C}/\mathcal{O}/\mathcal{I}, \ t = 1, \dots, T, \ k = 1, \dots, K \tag{2.2i}$$

$$n_{ct+1}^k = n_{ct}^k + D_{ct}^k - y_{ct}^k, \ \forall c \in \mathcal{O}, \ t = 1, \dots, T, \ k = 1, \dots, K \tag{2.2j}$$

$$n_{ct+1}^k = n_{ct}^k + \sum_{c' \in p(c)} \beta_{c'c}^k y_{c't}^k - y_{ct}^k, \ \forall c \in \mathcal{I}, \ t = 1, \dots, T, \ k = 1, \dots, K \tag{2.2k}$$

$$n_{c1}^k = n_c^{\text{init}}, \ \forall c \in \mathcal{C}, \ k = 1, \dots, K \tag{2.2l}$$

$$y_{ct}^k \geq 0, \ n_{ct}^k \geq 0, \ \forall c \in \mathcal{C}, \ t = 1, \dots, T, \ k = 1, \dots, K, \tag{2.2m}$$

where in the objective function (2.2a) we minimize the negative value of the expected vehicle throughput of the network over all time steps plus a so-called CTM objective term $\sum_{c \in \mathcal{C}} \sum_{t=1}^{T} (T-t) y_{ct}^k, \ \forall k = 1, \dots, K$. This will force y_{ct}^k for each cell $c \in \mathcal{C}$ at each time step $t \in \{1, \dots, T\}$ to obtain the minimum of the right-hand sides of constraints (2.2b)–(2.2h), and then to ensure all vehicles traveling forward as much as possible. (Each y_{ct}^k is weighted by the remaining $T - t$ time steps.) The parameter α is set to balance between these two objectives, where a larger α can eliminate the vehicle-holding problem to a certain extent. Constraints (2.2b)–(2.2m) involve both first-stage variables z_1, z_2 and second-stage recourse variables y^k, n^k, and establish the fundamental relationships in CTM. Specifically, constraints (2.2b) indicate that the number of vehicles leaving a cell c is limited by the number of vehicles inside cell c. Constraints (2.2c)–(2.2d) imply that the number of vehicles leaving a cell c is limited by the flow capacity of cell c. Notice that for an intersection cell, its capacity is determined by the related traffic signal, such that if the signal is red, then the capacity should be zero. Constraints (2.2e)–(2.2f) indicate that the number of vehicles leaving a cell c is also limited by the flow capacity of its processing cell c'. Notice that a diverging cell has more than one processing cell and the number of vehicles entering each processing cell is estimated by the turning ratio. Constraints (2.2g)–(2.2h) indicate that the number of vehicles leaving a cell c should be limited by the number of vehicles that can enter its processing cells $d(c)$. Constraints (2.2i)–(2.2k) are flow conservation equations for cells, enforcing that the difference between the number of vehicles in a cell c between two consecutive time steps t and $t+1$ equals to the number of vehicles coming from preceding cells minus the number of vehicles leaving cell c during time $[t, t+1)$. Notice that the number of vehicles entering each origin cell is the source demand. Constraints (2.2l) present the initial number of vehicles inside each cell. Constraints (2.2m) indicate that the number of vehicles leaving and inside each cell should be non-negative.

2.4 Decentralized and Decomposition Algorithms

The scalability issue of solving the two-stage stochastic MIP mainly comes from the number of intersections and time steps. In this section, we propose algorithms for solving Model (2.2) using Benders decomposition combined with spatial and temporal decomposition. We propose an exact ADMM-based spatially decentralized Benders algorithm and prove the optimality in Section 2.4.1. We develop a heuristic temporal decomposition technique to further reduce the computational time in Section 2.4.2.

2.4.1 ADMM-based Spatially Decentralized Benders Algorithm

At each iteration of the standard Benders algorithm, we solve the relaxed master problem (first-stage problem), the subproblem (second-stage problem), and add derived constraints to the relaxed master problem (Shapiro et al. [187]). We present the details of applying the Benders algorithm to solve our problem directly in Appendix A.1. However, the computational complexity of solving first-stage problems and second-stage problems increases significantly as network sizes and time steps increase. Therefore, we propose a distributed algorithm that solves first-stage problems as well as second-stage problems separately for each intersection based on the partition of the network. In Section 2.4.1.1 we propose a distributed formulation of first-stage and second-stage problems. In Section 2.4.1.2 we solve the second-stage problems for each intersection individually by applying ADMM [25]. In Section 2.4.1.3 we generate optimality cuts based on the optimal value and solutions obtained by ADMM and prove that the objective value obtained by our proposed algorithm converges to the optimal objective value.

2.4.1.1 Distributed Formulation

We partition the network into N_I areas where each area contains only one intersection. For each area with an intersection i for $i \in \mathcal{R}$, let $\mathcal{C}_i, \mathcal{E}_i, \mathcal{O}_i, \mathcal{D}_i, \mathcal{M}_i, \mathcal{V}_i$ be the corresponding sets of all cells, ordinary cells, origin cells, destination cells, merge cells, and diverge cells, respectively. Because signal constraints (2.1a)–(2.1j) can be written separately for each intersection, by defining variables θ_i^k for each scenario $\xi^k,\ k = 1, \ldots, K$ and each intersection $i \in \mathcal{R}$, we formulate an intersection-based relaxed master problem as follows.

$$
\begin{aligned}
(\mathbf{RMP}_i)\ \ \min \quad & \sum_{k=1}^{K} p^k \theta_i^k && (2.3\text{a})\\
\text{s.t.} \quad & \text{Constraints (2.1a)–(2.1j) corresponding to the intersection } i \\
& (z_{1i}, z_{2i}, \theta_i) \in \Sigma_i(z_{1i}, z_{2i}, \theta_i), && (2.3\text{b})
\end{aligned}
$$

where $\Sigma_i(z_{1i}, z_{2i}, \theta_i)$ is the set of cuts as linear functions of z_{1i}, z_{2i} generated up to the current iteration. The problems $(\mathbf{RMP}_i)$ allow us to solve the relaxed master problem for each area separately which only contains one intersection $i \in \mathcal{R}$.

Next, we consider the distributed formulation of second-stage subproblems. For each intersection $i \in \mathcal{R}$, we partition the set of cells $\mathcal{C}_i$ into two parts – one consists of all the cells where the constraints of these cells are related to cells in other intersections, called boundary cells and the other consists of all the remaining cells, called internal cells [207]. For each area containing intersection $i \in \mathcal{R}$, let $\mathcal{B}_i^I$ be the set of input boundary cells receiving inflow traffic from a cell of a neighboring area, and $\mathcal{B}_i^O$ be the set of output boundary cells sending outflow traffic to a cell of a neighboring area. In a centralized stochastic programming model, constraints (2.2g) and (2.2i) are related to boundary cells. We rewrite these constraints separately for boundary cells and internal cells. Notice that in a grid network setting, for each boundary cell $c \in \mathcal{B}_i^O \cup \mathcal{B}_i^I$, there is only one cell receiving the inflow traffic from c and one cell sending the outflow traffic to c, denoted by d_c and p_c such that $d(c) = \{d_c\}$ and

$p(c) = \{p_c\}$.

For each intersection $i \in \mathcal{R}$, the constraints related to boundary cells include variables corresponding to other intersections. For each $k = 1, \ldots, K$ and $i \in \mathcal{R}$, we introduce new decision variable vectors $\tilde{y}_i^k \in \mathbb{R}^{|\mathcal{B}_i^I| \times T}$ and $\tilde{n}_i^k \in \mathbb{R}^{|\mathcal{B}_i^O| \times T}$ to estimate the corresponding y^k and n^k of cells of neighboring intersections. Constraints (2.2g)–(2.2h) are rewritten as equality constraints for boundary cells by defining auxiliary variables $s_{ct}^k \geq 0$ for each cell $c \in \mathcal{B}_i^I \cup \mathcal{B}_i^O$, $\forall i \in \mathcal{R}$, at each time step $t = 1, \ldots, T$ and for each scenario $k = 1, \ldots, K$. Let $s^k = [s_{ct}^k,\ c \in \cup_{i \in \mathcal{R}}(\mathcal{B}_i^I \cup \mathcal{B}_i^O),\ t = 1, \ldots, T]^\mathsf{T}$, $\forall k = 1, \ldots, K$. Given the solutions $\hat{z}_{1i},\ \hat{z}_{2i}$ of the first-stage problems $(\mathbf{RMP}_i)$, $i \in \mathcal{R}$, we substitute $z_1 = \hat{z}_{1i},\ z_2 = \hat{z}_{2i}$ into constraints (2.2d) and formulate the second-stage problem as a distributed formulation:

$$(\mathbf{SP\text{-}D}^k)\ \min \quad -\sum_{i \in \mathcal{R}} \sum_{c \in \mathcal{D}_i} \sum_{t=1}^{T} n_{ct}^k - \alpha \sum_{i \in \mathcal{R}} \sum_{c \in \mathcal{C}_i} \sum_{t=1}^{T} (T-t) y_{ct}^k \tag{2.4a}$$

$$\text{s.t.} \quad \text{Constraints (2.2b)–(2.2m) for the scenario } \xi^k,\ \forall i \in R,\ \forall c \in \mathcal{C}_i / \mathcal{B}_i^I / \mathcal{B}_i^O \tag{2.4b}$$

$$s_{ct}^k \geq 0,\ \forall i \in \mathcal{R},\ \forall c \in \mathcal{B}_i^I \cup \mathcal{B}_i^O,\ \forall c' \in d(c),\ t = 1, \ldots, T \tag{2.4c}$$

$$y_{ct}^k + s_{ct}^k = W_{d_c t}(N_{d_c t} - \tilde{n}_{ict}^k),\ \forall i \in \mathcal{R},\ \forall c \in \mathcal{B}_i^O,\ t = 1, \ldots, T \tag{2.4d}$$

$$n_{ct+1}^k = n_{ct}^k + \tilde{y}_{ict}^k - y_{ct}^k,\ \forall i \in \mathcal{R},\ \forall c \in \mathcal{B}_i^I,\ t = 1, \ldots, T \tag{2.4e}$$

$$\tilde{y}_{ict}^k = y_{p_c t}^k,\ \forall i \in \mathcal{R},\ \forall c \in \mathcal{B}_i^I,\ t = 1, \ldots, T \tag{2.4f}$$

$$\tilde{n}_{ict}^k = n_{d_c t}^k,\ \forall i \in \mathcal{R},\ \forall c \in \mathcal{B}_i^O,\ t = 1, \ldots, T. \tag{2.4g}$$

Constraints (2.4b) correspond to constraints (2.2b)–(2.2m) for the internal cells under scenario ξ^k. Constraints (2.4d) and (2.4e) refer to the equality form of constraints (2.2g) and (2.2i) related to boundary cells given ξ^k. Constraints (2.4f) and (2.4g) indicate that for each intersection $i \in \mathcal{R}$, $\tilde{y}_i^k$ and $\tilde{n}_i^k$ should be equal to the value of y^k and n^k of cells of neighboring intersections receiving flow from or sending flow to boundary cells.

The objective function and constraints (2.4b)–(2.4e) are separable for each intersection $i \in \mathcal{R}$. Because for each cell $c \in \mathcal{B}_i^I \cup \mathcal{B}_i^O$, the cell sending or receiving flow p_c or d_c belong to other intersections, constraints (2.4f)–(2.4g) are not separable. We relax the linear constraints corresponding to all the original variables y^k, n^k, s^k and estimate variables $\tilde{y}^k, \tilde{n}^k$ using a Lagrangian penalty function and propose an ADMM algorithm to solve the second-stage problem $(\mathbf{SP\text{-}D}^k)$ in the next section.

2.4.1.2 ADMM for Second-stage Problems

In the distributed formulation, each intersection is considered as a block and the model contains several additional coupled (linear) constraints between pairs of intersection blocks. We apply ADMM to solve each second-stage problem ($\mathbf{SP\text{-}D}^k$), $\forall k = 1, \ldots, K$. Denote $\mathbf{Y}_i$ as the convex feasible region of $[y_i^k, n_i^k, s_i^k]$ defined by constraints (2.4b)–(2.4c) for each intersection $i \in \mathcal{R}$. Introducing dual variables κ_{ict}^k, λ_{ict}^k, μ_{ict}^k, ν_{ict}^k for constraints (2.4d), (2.4e), (2.4f), and (2.4g), respectively, and using $\kappa_i^k, \lambda_i^k, \mu_i^k, \nu_i^k$ to represent vector forms of these dual variables, we define an augmented dual Lagrangian function with a feasible set $\mathbf{Y}_i$ for each intersection $i \in \mathcal{R}$ as follows. (Here L represents the Lagrangian penalty parameter.)

$$\begin{aligned}\mathcal{L}_i^k(y_i^k, n_i^k, s_i^k, \tilde{y}_i^k, \tilde{n}_i^k, \kappa_i^k, \lambda_i^k, \mu_i^k, \nu_i^k) = &- \sum_{c\in\mathcal{D}_i}\sum_{t=1}^{T} n_{ct}^k - \alpha\sum_{c\in\mathcal{C}_i}\sum_{t=1}^{T}(T-t)y_{ct}^k \\ &+ \sum_{c\in\mathcal{B}_i^O}\sum_{t=1}^{T}\kappa_{ict}^k(y_{ct}^k + s_{ct}^k - W_{d_ct}(N_{d_ct} - \tilde{n}_{ict}^k)) \\ &+ \sum_{c\in\mathcal{B}_i^I}\sum_{t=1}^{T}\lambda_{ict}^k(n_{ct+1}^k - n_{ct}^k - \tilde{y}_{ict}^k + y_{ct}^k) \\ &+ \sum_{c\in\mathcal{B}_i^I}\sum_{t=1}^{T}\mu_{ict}^k(\tilde{y}_{ict}^k - y_{p_ct}^k) + \sum_{c\in\mathcal{B}_i^O}\sum_{t=1}^{T}\nu_{ict}^k(\tilde{n}_{ict}^k - n_{d_ct}^k) \\ &+ \frac{L}{2}\sum_{c\in\mathcal{B}_i^O}\sum_{t=1}^{T}\|y_{ct}^k + s_{ct}^k - W_{d_ct}(N_{d_ct} - \tilde{n}_{ict}^k)\|^2 \\ &+ \frac{L}{2}\sum_{c\in\mathcal{B}_i^I}\sum_{t=1}^{T}\|n_{ct+1}^k - n_{ct}^k - \tilde{y}_{ict}^k + y_{ct}^k\|^2 \\ &+ \frac{L}{2}\sum_{c\in\mathcal{B}_i^I}\sum_{t=1}^{T}\|\tilde{y}_{ict}^k - y_{p_ct}^k\|^2 + \frac{L}{2}\sum_{c\in\mathcal{B}_i^O}\sum_{t=1}^{T}\|\tilde{n}_{ict}^k - n_{d_ct}^k\|^2.\end{aligned} \tag{2.5}$$

The subproblem ($\mathbf{SP\text{-}D}^k$) can be rewritten as the minimization problem of the augmented Lagrangian function with a feasible set composed by constraints (2.4b)–(2.4e). Following the definitions of dual variables, the augmented Lagrangian function is the summation of functions $\mathcal{L}_i^k$ for all intersections $i \in \mathcal{R}$, i.e.,

$$\mathcal{L}^k(y^k, n^k, s^k, \tilde{y}^k, \tilde{n}^k, \kappa^k, \lambda^k, \mu^k, \nu^k) = \sum_{i\in\mathcal{R}}\mathcal{L}_i^k(y_i^k, n_i^k, s_i^k, \tilde{y}_i^k, \tilde{n}_i^k, \kappa_i^k, \lambda_i^k, \mu_i^k, \nu_i^k). \tag{2.6}$$

The minimization problem of $\mathcal{L}^k$ is equivalent to a series of problems that minimize $\mathcal{L}_i^k$ for each intersection $i \in \mathcal{R}$.

For each intersection $i \in \mathcal{R}$, ADMM consists of three main steps: (i) updating original variables y_i^k, n_i^k, s_i^k by solving the minimization problem of the Lagrangian function $\mathcal{L}^k$ with fixed estimates variables $\tilde{y}_i^k,\ \tilde{n}_i^k$ and dual variables, (ii) updating estimates variables $\tilde{y}_i^k, \tilde{n}_i^k$ by solving the minimization problem of the Lagrangian function $\mathcal{L}^k$ with fixed original variables $y_i^k,\ n_i^k,\ s_i^k$ and dual variables, and (iii) updating dual variables $\kappa_i^k, \lambda_i^k, \mu_i^k, \nu_i^k$ by gradient ascent. Specifically, in iteration $l+1$, we update variables based on current values y^{kl}, n^{kl}, s^{kl},$\tilde{y}^{kl}, \tilde{n}^{kl}$,$\kappa^{kl}$,$\lambda^{kl}, \mu^{kl}, \nu^{kl}$ as follows.

$$[y_i^{kl+1}, n_i^{kl+1}, s_i^{kl+1}] = \underset{[y_i^k, n_i^k, s_i^k] \in \mathbf{Y}_i}{\arg\min}\ \mathcal{L}_i^k(y_i^k, n_i^k, s_i^k, \tilde{y}^{kl}, \tilde{n}^{kl}, \kappa_i^{kl}, \lambda_i^{kl}\mu_i^{kl}, \nu_i^{kl}),\ \forall i \in \mathcal{R}, \tag{2.7a}$$

$$[\tilde{y}_i^{kl+1}, \tilde{n}_i^{kl+1}] = \underset{\tilde{y}_i^k, \tilde{n}_i^k}{\arg\min}\, \mathcal{L}_i^k(y_i^{kl+1}, n_i^{kl+1}, s_i^{kl+1}, \tilde{y}_i^k, \tilde{n}_i^k, \kappa_i^{kl}, \lambda_i^{kl}, \mu_i^{kl}, \nu_i^{kl}),\ \forall i \in \mathcal{R}, \tag{2.7b}$$

$$\kappa_{ict}^{kl+1} = \kappa_{ict}^{kl} + L(y_{ct}^{kl+1} + s_{ct}^{kl+1} - W_{dct}(N_{dct} - \tilde{n}_{ict}^{kl+1})),\ \forall i \in \mathcal{R},\ \forall c \in \mathcal{B}_i^O,\ t = 1, \ldots, T, \tag{2.7c}$$

$$\lambda_{ict}^{kl+1} = \lambda_{ict}^{kl} + L(n_{ct+1}^{kl+1} - n_{ct}^{kl+1} - \tilde{y}_{ict}^{kl+1} + y_{ct}^{kl+1}),\ \forall i \in \mathcal{R},\ \forall c \in \mathcal{B}_i^I,\ t = 1, \ldots, T, \tag{2.7d}$$

$$\mu_{ict}^{kl+1} = \mu_{ict}^{kl} + L(\tilde{y}_{ct}^{kl+1} - y_{pct}^{kl+1}),\ \forall i \in \mathcal{R},\ \forall c \in \mathcal{B}_i^I,\ t = 1, \ldots, T, \tag{2.7e}$$

$$\nu_{ict}^{kl+1} = \nu_{ict}^{kl} + L(\tilde{n}_{ct}^{kl+1} - n_{dct}^{kl+1}),\ \forall i \in \mathcal{R},\ \forall c \in \mathcal{B}_i^O,\ t = 1, \ldots, T. \tag{2.7f}$$

Because the procedure of updating variables is separable for each intersection, the computation can be conducted in parallel for different intersections in each iteration, which can speed up the computation drastically. In practice, each subproblem can be solved by local computers installed at each intersection and then communicate with each other to update the duals, to fully utilize computing technologies in connected transportation systems.

2.4.1.3 ADMM-based Spatially Decentralized Benders Algorithm

We use the same formulation of ($\mathbf{RMP}_i$) as first-stage problems for all the intersections $i \in \mathcal{R}$ and solve them directly to obtain solutions $\hat{z}_{1i}, \hat{z}_{2i}, \theta_i^k$. Given first-stage solutions, for each $k = 1, \ldots, K$, we solve the second-stage problem by ADMM and obtain the optimal values $\hat{\mathcal{L}}_i^k(\hat{z}_{1i}, \hat{z}_{2i})$ and the optimal dual solutions $\hat{\sigma}_{ct}^k,\ c \in \mathcal{I}_{ij},\ j \in \mathcal{F}_i,\ t = 1, \ldots, T$ (associated with constraints (2.2b)) for each intersection $i \in \mathcal{R}$. We generate a linear optimality cut for θ_i^k as:

$$\theta_i^k \geq \hat{\mathcal{L}}_i^k(\hat{z}_{1i}, \hat{z}_{2i}) - \sum_{j \in \mathcal{F}_i} \sum_{c \in \mathcal{I}_{ij}} \sum_{m=1}^{N_{cy}} \sum_{t=1}^{T} (\hat{z}_{1ijmt} + \hat{z}_{2ijmt} - 1) Q_{ct} \hat{\sigma}_{ct}^k$$

$$+\sum_{j\in\mathcal{F}_i}\sum_{c\in\mathcal{I}_{ij}}\sum_{m=1}^{N_{cy}}\sum_{t=1}^{T}(z_{1ijmt}+z_{2ijmt}-1)Q_{ct}\hat{\sigma}_{ct}^{k}. \tag{2.8}$$

Theorem 2.4.1. *The optimality cut* (2.8) *is a valid cut.*

Proof. For each $k=1,\dots,K$ and $i\in\mathcal{R}$, let $\hat{\tilde{y}}_i^k,\hat{\tilde{n}}_i^k,\hat{\kappa}_i^k,\hat{\lambda}_i^k,\hat{\mu}_i^k,\hat{\nu}_i^k$ be the optimal solutions of variables $\tilde{y}_i^k,\tilde{n}_i^k,\kappa_i^k,\lambda_i^k,\mu_i^k,\nu_i^k$ in ADMM. From the convergence property of ADMM on convex problems, we have

$$\hat{\mathcal{L}}_i^k(\hat{z}_{1i},\hat{z}_{2i})=\min_{[y_i^k,n_i^k,s_i^k]\in\mathbf{Y}_i}\mathcal{L}_i^k(y_i^k,n_i^k,s_i^k,\hat{\tilde{y}}_i^k,\hat{\tilde{n}}_i^k,\hat{\kappa}_i^k,\hat{\lambda}_i^k,\hat{\mu}_i^k,\hat{\nu}_i^k). \tag{2.9}$$

Consider the minimization problem on the right-hand side of (2.9). Associate dual variables ρ_{ct}^k, σ_{ct}^k, $\pi_{cc't}^k$, $\gamma_{cc't}^k$, δ_{ct}^k, τ_c^k to constraints (2.2b), (2.2c)–(2.2d), (2.2e)–(2.2f), (2.2g)–(2.2h), (2.2i)–(2.2k), and (2.2l), respectively. The dual Lagrangian function of this minimization problem is in the form of

$$\begin{aligned}
&\tilde{\mathcal{L}}_i^k(\hat{z}_{1i},\hat{z}_{2i},y_i^k,n_i^k,s_i^k,\rho^k,\sigma^k,\pi^k,\gamma^k,\delta^k,\tau^k)\\
=&\mathcal{L}_i^k(y_i^k,n_i^k,s_i^k,\hat{\tilde{y}}_i^k,\hat{\tilde{n}}_i^k,\hat{\kappa}_i^k,\hat{\lambda}_i^k,\hat{\mu}_i^k,\hat{\nu}_i^k)+\sum_{c\in\mathcal{C}_i/\mathcal{I}_i}\sum_{t=1}^{T}(Q_{ct}-y_{ct}^k)\sigma_{ct}^k\\
&+\sum_{j\in\mathcal{F}_i}\sum_{c\in\mathcal{I}_{ij}}\sum_{m=1}^{N_{cy}}\sum_{t=1}^{T}((\hat{z}_{1ijmt}+\hat{z}_{2ijmt}-1)Q_{ct}-y_{ct}^k)\sigma_{ct}^k+\sum_{c\in\mathcal{C}_i}\sum_{c'\in d(c)}\sum_{t=1}^{T}(Q_{c't}-y_{ct}^k)\pi_{cc't}^k\\
&+\sum_{c\in\mathcal{C}_i/\mathcal{B}_i^O}\sum_{c'\in d(c)}\sum_{t=1}^{T}(W_{c't}N_{c't}-y_{ct}^k-W_{c't}n_{ct}^k)\gamma_{cc't}^k\\
&+\sum_{c\in\mathcal{O}_i}\sum_{t=1}^{T}(D_{ct}^k-y_{ct}^k+n_{ct}^k-n_{ct+1}^k)\delta_{ct}^k+\sum_{c\in\mathcal{C}_i}(n_c^{\text{init}}-n_{c1}^k)\tau_c^k.
\end{aligned} \tag{2.10}$$

Because the minimization problem of $\mathcal{L}_i^k$ is a semi-definite quadratic program, the strong duality holds. Therefore,

$$\hat{\mathcal{L}}_i^k(\hat{z}_{1i},\hat{z}_{2i})=\max_{\rho^k\geq0,\sigma^k\geq0,\pi^k\geq0,\gamma^k\in\Gamma^k,\delta^k,\tau^k}\ \min_{y_i^k,n_i^k,s_i^k}\tilde{\mathcal{L}}_i^k(\hat{z}_{1i},\hat{z}_{2i},y_i^k,n_i^k,s_i^k,\rho^k,\sigma^k,\pi^k,\gamma^k,\delta^k,\tau^k), \tag{2.11}$$

where Γ^k is the feasible set of γ^k such that $\gamma_{cc't}\geq0,\ \forall c\in\mathcal{C}_i/\mathcal{B}_i^o,\ c'\in d(c),\ t=1,\dots,T$. Furthermore, the feasibility set of $\rho^k,\sigma^k,\pi^k,\gamma^k,\delta^k,\tau^k$ generated by $\min_{y_i^k,n_i^k,s_i^k}\tilde{\mathcal{L}}_i^k$ does not depend on first-stage decision variables. Substituting (2.11) into the cut (2.8), we can verify that the cut is valid. □

The procedure of ADMM-based spatially decentralized Benders algorithm is described in Algorithm 2.1.

Algorithm 2.1: An ADMM-based Spatially Decentralized Benders Algorithm for solving Model (2.2).

```
for $i \in \mathcal{R}$ do
    Initialize ($\mathbf{RMP}_i$) with $\Sigma_i = \emptyset$.
end
while the termination criteria is not satisfied do
    for $i \in \mathcal{R}$ do
        Solve ($\mathbf{RMP}_i$) to obtain optimal solution $(\hat{z}_{1i}, \hat{z}_{2i}, \hat{\theta}_i)$.
    end
    for $k = 1, \dots, K$ do
        Solve ($\mathbf{SP\text{-}D}^k$) by ADMM in parallel according to Section 2.4.1.2 and obtain
          the optimal value $\mathcal{L}_i^{k*}$ and optimal dual variables $\hat{\sigma}$.
    end
    for $i \in \mathcal{R}$ do
        if $\hat{\theta}_i^k < \mathcal{L}_i^{k*}$ then
            Add optimality cut (2.8) to $\Sigma_i$.
        end
    end
end
Return the objective value as $\sum_{k=1}^{K} p^k \sum_{i \in \mathcal{R}} \hat{\theta}_i^k$ and the solutions of ($\mathbf{RMP}_i$), $i \in \mathcal{R}$.
```

Remark 2.4.1. *Because the added cuts are all valid, the summation of the objective values of (**RMP**$_i$) for all $i \in \mathcal{R}$ provides a lower bound of the original stochastic programming model. The following theorem guarantees the optimality of the ADMM-based decentralized Benders algorithm. Furthermore, the summation of $\hat{\mathcal{L}}_i^k$ provides an upper bound of the original stochastic programming model since it is the objective value of a feasible solution. We terminate the algorithm when the gap between the upper and lower bounds is sufficiently small.*

Theorem 2.4.2. *With the assumption that the objective function of the relaxed master problem is separable for the intersections, when the algorithm terminates, the centralized relaxed master problem* (**RMP**) *is equivalent to the union of distributed relaxed master problems* (**RMP**$_i$), $i \in \mathcal{R}$. *Furthermore, Benders-ADMM converges to the optimal solutions and the optimal objective value.*

Proof. For each particular first-stage solution $\hat{z}_1, \hat{z}_2$, we denote $\theta^k \geq F_D^k(\hat{z}_1, \hat{z}_2)$ as the optimality cut generated by Benders, and $\theta_i^k \geq F_{D_i}^k(\hat{z}_{1i}, \hat{z}_{2i})$ as the cut generated by Benders-

ADMM. It holds that

$$F_D^k(\hat{z}_1, \hat{z}_2) = \sum_{i \in \mathcal{R}} F_{D_i}^k(\hat{z}_{1i}, \hat{z}_{2i}). \tag{2.12}$$

When the algorithm terminates, for any solution $(z_1^*, z_2^*, \theta^{k*})$ of (**RMP**), we solve the second-stage subproblems by ADMM given (z_1^*, z_2^*) and let $\theta_i^{k*} = \mathcal{L}_i^*$. From the procedure of Benders-ADMM and equation (2.12), $(z_{1i}^*, z_{2i}^*, \theta_i^{k*})$ is a solution to ($\mathbf{RMP}_i$) for each intersection $i \in \mathcal{R}$. On the other hand, for any solution $(z_{1i}^*, z_{2i}^*, \theta_i^{k*})$ of ($\mathbf{RMP}_i$), we can construct a solution $\prod_{i\in\mathcal{R}} z_{1i}^*, \prod_{i\in\mathcal{R}} z_{2i}^*, \sum_{i\in\mathcal{R}} \theta_i^{k*}$ for (**RMP**) because of the Benders procedures and equation (2.12). It is well known that Benders decomposition algorithm converges to the optimal solutions [187] and therefore, Benders-ADMM also converges to the optimal solutions and the optimal objective value. □

Remark 2.4.2. *ADMM-based spatially decentralized Benders algorithm (Algorithm 2.1) can be also employed when the partitioned areas contain multiple intersections.*

2.4.2 Heuristic Temporal Decomposition

Our ADMM-based spatially decentralized Benders algorithm solves the relaxed mater problems and subproblems in a distributed manner, which reduces the computational time and guarantees optimality. To further accelerate the convergence of our Benders algorithm, we introduce a heuristic temporal decomposition technique to pre-determine the cycle length in this section. The heuristic technique sacrifices the global optimality but still guarantees the optimality under a certain cycle length constraint. For the first-stage problem of each intersection $i \in \mathcal{R}$, the computational complexity mainly depends on the number of time steps, T.

Lemma 2.4.1. *For each $i \in \mathcal{R}$, the number of variables and constraints in ($\boldsymbol{RMP}_i$) grows as $O(T^2)$ with the number T of time steps.*

Proof. The number of variables in ($\mathbf{RMP}_i$) is $2|\mathcal{F}_i|N_{\text{cy}}T + 2|\mathcal{F}_i|N_{\text{cy}}$. As $|\mathcal{F}_i|$ does not depend on T and N_{cy} grows linearly with T, the number of variables grows as $O(T^2)$. Similarly, the number of constraints is $O(|\mathcal{F}_i|N_{\text{cy}}T)$, which also grows as $O(T^2)$. This completes the proof. □

To reduce the size of the first-stage problem, we pre-determine the cycle lengths l_i as $\hat{l}_i$ for each $i \in \mathcal{R}$ based on the source demand volume according to Koonce and Rodegerdts [99]. Since the traffic signal control plan is the same for each cycle, the signal status of time step $(m-1)\hat{l}_i + t'$ is exactly the same as t' for all the cycles $m = 1, \ldots, N_{\text{cy}}$. Therefore, we redefine binary variables $z_{1ijmt},\ z_{2ijmt}$ for each intersection $i \in \mathcal{R}$, each phase $j \in \mathcal{F}_i$, only for the first two cycles $m = 1,\ 2$ and time steps $t = 1, \ldots, \hat{l}_i$. The first-stage problems (**RMP**$_i$) and (**RMP**) are reformulated by setting $N_{\text{cy}} = 2$ and $T = \hat{l}_i$. In the following lemma, we show that it is sufficient to describe the signal status at each time step $t = 1, \ldots, T$ with the redefined decision vectors $z_{1i},\ z_{2i}$ for the first two cycles and the first $\hat{l}_i$ time steps.

Lemma 2.4.2. *For each intersection $i \in \mathcal{R}$, with given cycle length $\hat{l}_i$, the flow upper bound constraints for cells controlled by signals* (2.2d) *are equivalent to*

$$y_{ct}^k \le \sum_{m=1}^{2} (z_{1ijmt'} + z_{2ijmt'} - 1) Q_{ct},\ \forall c \in \mathcal{I}_{ij},\ \forall i \in \mathcal{R},\ \forall j \in \mathcal{F}_i,\ t = 1, \ldots, T,\ k = 1, \ldots, K, \tag{2.13}$$

where $t' = (t-1)$ mod $\hat{l}_i + 1$ is the index of time step corresponding to time step t.

Proof. Recalling the definition of binary variables $z_1,\ z_2$, we have that $\sum_{m=1}^{N_{\text{cy}}} (z_{1ijmt} + z_{2ijmt} - 1) = 1$ if and only if the signal of phase j at intersection i is green at time step t. Because the signal timing plan is the same for each cycle, we know that

$$\sum_{m=1}^{N_{\text{cy}}} (z_{1ijmt} + z_{2ijmt} - 1) = \sum_{m=1}^{N_{\text{cy}}} (z_{1ijmt'} + z_{2ijmt'} - 1),\ t = 1, \ldots, T,\ t' = (t-1) \bmod \hat{l}_i + 1. \tag{2.14}$$

Next, we show that the right-hand-side is equivalent to taking the summation with $N_{\text{cy}} = 2$. From constraints (2.1d)–(2.1h), given $l_i = \hat{l}_i$, we have

$$-\hat{l}_i \le b_{i11} \le 0,\ \forall i \in \mathcal{R}; \tag{2.15a}$$

$$e_{i|\mathcal{F}_i|2} \ge \hat{l}_i,\ \forall i \in \mathcal{R}. \tag{2.15b}$$

The time horizon we consider is $\{1, \ldots, \hat{l}_i\}$, which is a subset of $[b_{i11}, e_{i|\mathcal{F}_i|2}]$. Hence, according to the definition of $z_1,\ z_2$, we have that $z_{1ijmt'} = z_{2ijmt'} = 0$ for all the cycles $m \ge 2$. □

Theorem 2.4.3. *By pre-determining the cycle length, the problem size of reformulated first-stage problems (**RMP**$_i$) for each $i \in \mathcal{R}$ only grows linearly with the cycle length. The statement also holds for the reformulated centralized first-stage problem (**RMP**).*

Proof. Combining Lemma 2.4.1 and Lemma 2.4.2, we show that the number of variables reduces to $4|\mathcal{F}_i|\hat{l}_i + 4|\mathcal{F}_i|$ and number of constraints reduce to $O(|\mathcal{F}_i|\hat{l}_i)$. It is clear that the number of variables and the number of constraints in ($\mathbf{RMP}_i$) are both $O(\hat{l}_i)$. Because the number of variables and constraints in ($\mathbf{RMP}$) is the summation of the number of variables and constraints in ($\mathbf{RMP}_i$) for all $i \in \mathcal{R}$, the conclusion holds if we pre-determine the cycle lengths in ($\mathbf{RMP}$). □

With heuristic pre-determined cycle lengths and reformulated first-stage and second-stage problems, Benders-ADMM only ensures convergence to the optimal solutions and optimal objective value of the original centralized model with additional signal cycle length constraints $l_i = \hat{l}_i, \ \forall i \in \mathcal{R}$.

Remark 2.4.3. *The pre-determined cycle length for all the intersections are the same [99]. The cycle length l^{fix} is set as the mean value of $l_i, i \in \mathcal{R}$ computed by the following formula [223].*

$$l_i = \left\lceil \frac{|\mathcal{F}_i| * 7.5 + 5}{1 - \sum_{j=1}^{4} D_j/Q_j} \right\rceil \tag{2.16}$$

where D_j and Q_j are the source demand and maximum flow capacity of the direction corresponding to a phase j.

2.5 Numerical Studies

We apply algorithms proposed in Section 2.4 to solve the traffic signal control problem on instances of randomly generated grid networks and real-world traffic networks. In Section 2.5.1, we introduce the experimental design. In Section 2.5.2 and Section 2.5.3, we present the computational results to demonstrate the efficacy of our approaches.

2.5.1 Experimental Design

We introduce the setting of warm-up initialization in our experiments in Section 2.5.1.1. We present the network design and parameter setting of randomly generated grid network instances and real-world traffic network instances in Section 2.5.1.2 and Section 2.5.1.3, respectively. We discuss the metrics and procedure of our out-of-sample evaluation in Section 2.5.1.4.

2.5.1.1 Warm-up Initialization

We introduce a warm start technique according to Webster [223] to simulate the initialized state of traffic networks and obtain the number n_c^{init} of vehicles inside each cell $c \in \mathcal{C}$.

We define a fixed traffic signal time plan and compute the number of vehicles in each cell $c \in \mathcal{C}$ by solving the model with an objective function as the second term of (2.2a) and constraints (2.2b)–(2.2m). For each each phase $j \in \mathcal{F}_i$, the green time of a phase j is defined as

$$g_j^{\text{fix}} = \frac{D_j}{\sum_{j=1}^{4} D_j} l^{\text{fix}}, \tag{2.17}$$

where D_j is the source demand of the direction corresponding to a phase j and l^{fix} is the cycle length. For each cell $c \in \mathcal{C}$, we set n_c^{init} as the number of vehicles inside c at the last time step.

2.5.1.2 Randomly Generated Grid Networks

We conduct numerical studies on randomly generated grid networks with the size $N_{\text{row}} \times N_{\text{col}}$, where N_{row} is the number of rows and N_{col} is the number of intersections in each row that has the same structure of intersections and road segments. We define the set of phases $\mathcal{F}_i$ for all the intersections $i \in \mathcal{R}$ as $\mathcal{F}_i = \{1, 2, 3, 4\}$, where $j = 1$ means turning left in East-West direction, $j = 2$ means going straight or turning right in East-West direction, $j = 3$ means turning left in North-South direction, and $j = 4$ means going straight or turning right in North-South direction. We set the values of input parameters as follows. Let "veh" denote the number of vehicles. For each intersection cell $c \in \mathcal{I}$ and time step $t = 1, \dots, T$, $Q_{ct} = 1.5$ veh and $N_{ct} = 6$ veh, meaning that at most 1.5 and 6 vehicles can flow through and reside in an intersection cell at any time, respectively. For the other cells $c \in \mathcal{C}/\mathcal{I}$ and time step $t = 1, \dots, T$, $Q_{ct} = 3$ veh and $N_{ct} = 12$ veh, meaning that at most 3 and 12 vehicles can flow through and reside in other cells at each time. The ratio of shock-wave speed over free-flow speed is the same for all the cells in $\mathcal{C}$ for each $t \in \{1, \dots, T\}$, which is set as $W = 1/3$. The initialized number of vehicles n^{init} is generated by the warm-start technique described in Section 2.5.1.1. The minimum green time $G_{\min} = 6$ seconds and the maximum green time $G_{\max} = 75$ seconds. The whole time horizon is half an hour and has 600 time steps. We set the weight parameter $\alpha = 0.001$ in the objective function.

We generate random samples of source demand and turning ratios as follows. We assume uniform arrivals of vehicles during the half-an-hour time horizon, and therefore values of D_{ct} are the same for all time steps $t = 1, \dots, T$ for any origin cell $c \in \mathcal{O}$. The source demand for each cell $c \in \mathcal{O}$ follows a truncated Normal distribution defined on $[0, \infty)$ shown in Table 2.3, where Column "SD/Mean" represents the ratio between standard deviation and the mean value. The unit of source demand is the number of vehicles per hour (veh/h).

The turning ratios of all the diverge cells $c \in \mathcal{D}$ follow truncated Normal distribution

Table 2.3: Instances of randomly generated source demand.

Instance	Mean (veh/h)	SD/Mean
1	200 (E-W), 50 (S-N)	2
2	200 (E-W), 50 (S-N)	3
3	200 (E-W), 50 (S-N)	4
4	400 (E-W), 100 (S-N)	2
5	400 (E-W), 100 (S-N)	3
6	400 (E-W), 100 (S-N)	4

E-W: direction of east and west;
S-N: direction of south and north;

defined on $[0, 1]$ with mean values $= 0.15,\ 0.72,\ 0.13$, representing the ratio of turning left, going straight and turning right, respectively. The ratio between the standard deviation and the mean value is set to 0.3. We test our approaches on randomly generated grid networks having sizes 4×4, 2×8, 6×6 and 10×10 and the distribution instance #4 of source demand given in Table 2.3. We test the other types of distributions in Table 2.3 only for the 4×4 grid network.

We generate 100 in-sample scenarios to formulate the stochastic optimization model. As a benchmark, we firstly use Gurobi to solve the stochastic programming model directly. Then, we employ Benders with/without temporal decomposition described in Section 2.4.2 to solve the model. Furthermore, based on the temporal decomposition, we apply Benders and the ADMM-based spatially decentralized Benders. For the 4×4 grid network, we build a deterministic counterpart by setting the source demand and turning ratios as the expectation and solve the model by applying ADMM-based spatially decentralized Benders with temporal decomposition.

2.5.1.3 Real-world Traffic Networks

We also test all the algorithms on a real traffic network based on the road network of Downtown Ann Arbor (a city in Michigan, United States). This traffic network contains 14 corridors, 37 signalized intersections, and 27 unsignalized intersections with stop signs and one-way roads. We present the map of our considered network area and the neighboring area in Figure 2.2. Notice that the test instance is a typical downtown area with grid traffic networks with multiple arterials in each direction. We mark the main arterials in the direction of East and West by red lines, including Miller Avenue, Huron Street, Washington Street, and Liberty Street. We mark the main arterials in the direction of North and South by purple lines, including Ashley Street, Main street, Division Street, and State Street. The downtown area of Ann Arbor is mainly the center area of our network, near the intersections of main arterials, which are marked by blue text. In addition, we mark the area of the

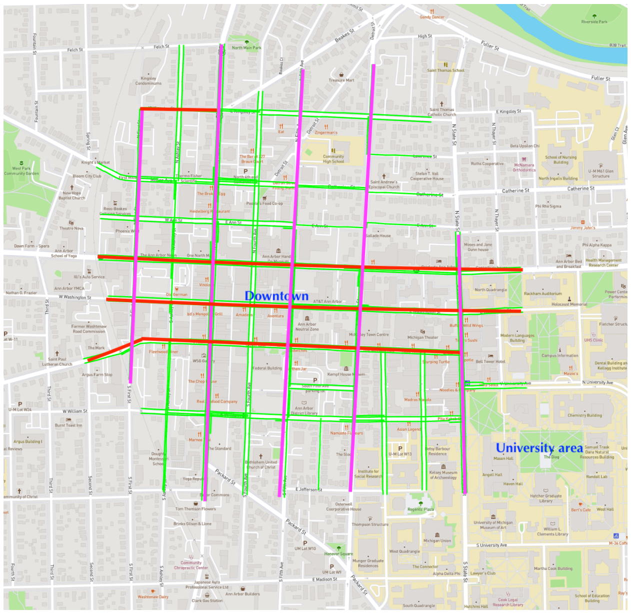


Figure 2.2: Map of Ann Arbor downtown.

University of Michigan as the Southeast area of the traffic network. In the morning, people mostly travel from West to East for studying and working at the university. In the afternoon, people mostly travel from East to West for going back home.

All the parameters, including mean values of source demand and turning ratios, are computed based on collected real-world traffic data. We test two instances with high average source demand during morning and afternoon peak hours where we set the SD/Mean ratio = 2 and an instance with low average source demand during off-peak hours with SD/Mean ratio = 3. The ratio between standard deviation and mean of turning ratios is 0.3. We generate 10 scenarios to formulate the stochastic optimization model and apply ADMM-based spatially decentralized Benders with temporal decomposition to solve the model. To construct a deterministic counterpart, we use the mean values of the source demand and turning ratios as input parameters and apply ADMM-based spatially decentralized Benders with temporal decomposition.

Remark 2.5.1. *We set a heuristic lower bound of the green time that the green length of each phase j should be no less than 60% of g_j^{fix} and apply early termination to make the solutions more practical. Notice that our algorithm still works without these settings.*

2.5.1.4 Out-of-sample Evaluation

Metrics for Evaluation After obtaining a traffic signal control plan, we evaluate its performance by conducting CTM simulation according to Daganzo [49]. Let y^*_{ct} be the obtained solution of the number of vehicles leaving a cell $c \in \mathcal{C}$ at a time step $t = 1, \ldots, T$. We define a link as a set of cells that belong to the same road segment connecting two neighboring intersections. For each link starting from a cell c_1 and ending at a cell c_2, we define the cumulative number of vehicles at time step $t = 1, \ldots, T$ of the inflow and outflow as $CN_{\text{in}}(t)$ and $CN_{\text{out}}(t)$. The computation of these two metrics follows:

$$CN_{\text{in}}(t) = \sum_{t'=1}^{t} y^*_{proc(c_1)t} \tag{2.18a}$$

$$CN_{\text{out}}(t) = \sum_{t'=1}^{t} y^*_{c_2 t} \tag{2.18b}$$

We define the cumulative number of vehicles of outflow at time step t as $CN^*(t)$ by assuming free-flow speed, and compute the value as

$$CN^*(t) = CN_{\text{in}}(t - N_c), \tag{2.19}$$

where N_c is the number of cells contained in the link. The performance of a traffic signal control plan is evaluated for every link during the T time steps by the total travel time $\sum_{t=1}^{T}(CN_{\text{in}}(t) - CN_{\text{out}}(t))$ and the total delay $\sum_{t=1}^{T}(CN^*(t) - CN_{\text{out}}(t))$. We also compute the average travel time and the average delay of each vehicle. In addition, the total number of vehicles traveling through a traffic network during the time horizon is the sum of $CN_{\text{out}}(T)$ for all the links ending at destination cells.

Out-of-sample Evaluation Procedures For both randomly generated grid networks and the real-world traffic network, we generate 5 replications of instances with the same parameter settings, each having 100 independently identically distributed scenarios with the same distribution as the one used in in-sample computation. We conduct the out-of-sample tests based on CTM simulation for different traffic signal control plans on these replications. We present the averages of the in-sample and out-of-sample objective values, the average travel delay, and the total throughput across the total 500 scenarios. For the real-world traffic network, we also compare the performance with our baseline solution that sets the green time of each phase as g_j^{fix}.

All the numerical experiments of randomly generated grid networks are conducted on Windows Server 2012 R2 Standard with 128 GB RAM and 2.20 GHz processor. All the numerical experiments of real-world traffic networks are conducted on a Windows computer with 32 GB RAM and 3.60 GHz processor.

2.5.2 Results of Randomly Generated Grid Networks

We present the computational time results in Section 2.5.2.1 and solution performance in Section 2.5.2.2.

2.5.2.1 Computational Time Comparison

We set the time limit for Gurobi to 7200 seconds. If we do not apply any decomposition schemes and directly solve the problem, Gurobi is not able to provide a feasible solution or even an upper bound of the objective value within the time limit. When using the generic Benders without temporal decomposition, we are not able to solve the first-stage problem within the time limit, and thus it cannot provide a feasible solution to the second-stage problem. Based on these results, the temporal decomposition is necessary.

We present the computational time results of the three algorithms with the temporal decomposition in Table 2.4, where we vary network sizes. In the table, N_{row} and N_{col} represent the number of intersections of each row and column in the grid networks while "Benders" and "Benders-ADMM" represent Benders decomposition algorithm and ADMM-based spatially decentralized Benders algorithm, respectively. In the table, "MP-min", "MP-max", "MP-A", "SP-min", "SP-max", "SP-A" stand for the minimum, maximum and average computational time in seconds of solving first-stage master problems and second-stage subproblems during all the iterations, respectively. If an algorithm is not able to return a feasible solution due to the time limit, we mark the related results as "-" in the table.

Table 2.4: CPU time of different algorithms for solving grid networks with various sizes.

N_{row}	N_{col}	Benders					
		MP-min (s)	MP-max (s)	MP-A(s)	SP-min (s)	SP-max(s)	SP-A(s)
4	4	1.18	123.99	54.98	2005.10	6905.05	4632.75
2	8	0.96	76.20	33.78	1687.63	4834.43	3252.43
6	6	1.78	685.31	132.60	17810.42	43607.42	34603.78
10	10	-	-	-	-	-	-
N_{row}	N_{col}	Benders-ADMM					
		MP-min (s)	MP-max (s)	MP-A(s)	SP-min (s)	SP-max(s)	SP-A(s)
4	4	1.73	46.22	18.81	539.44	594.26	571.98
2	8	1.68	33.96	14.71	295.12	519.81	426.21
6	6	4.26	101.51	42.43	1060.78	1685.95	1404.67
10	10	13.73	77.55	45.64	2780.25	3682.51	3231.38

The results show that the CPU time varies significantly during the iterations of different algorithms. With added cuts, the computational time of the first-stage problem increases drastically. The computational time of second-stage problems varies depending on first-stage solutions. The results also indicate that for every instance, Benders-ADMM performs significantly better than Benders when solving first-stage problems since the size of first-stage problems is reduced by the spatial decomposition. For second-stage problems, Benders-ADMM outperforms Benders since parallel computing is able to be applied so that each intersection can solve the second-stage problem at the same time. Moreover, for all the algorithms, the computational time increases when the network size is larger. For networks with the same number of intersections, the algorithms take less time to solve the model of an asymmetric network than a symmetric one.

2.5.2.2 Evaluation Results

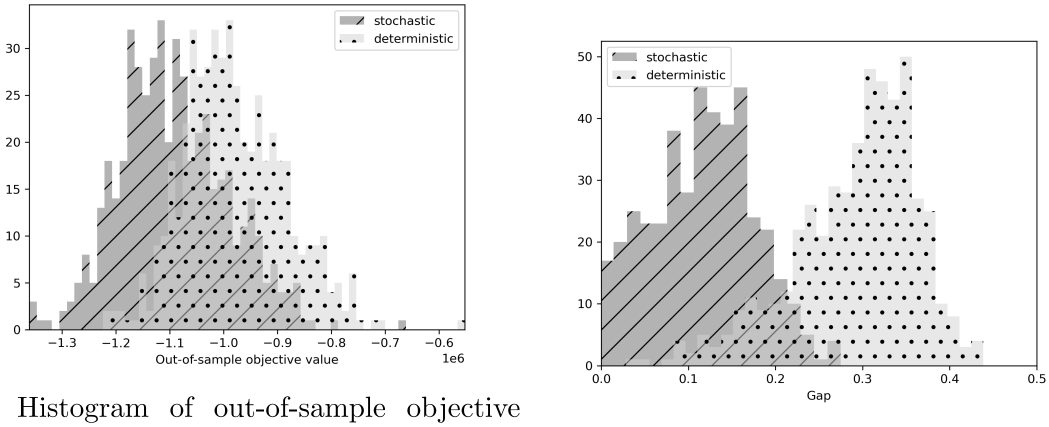
Overall objective values We use Benders-ADMM (Algorithm 2.1) to solve the deterministic (Deter) and stochastic (SP) MIP models and present the objective values of both models in Table 2.5. Column "Mean" presents mean values of source demand and Column "SD/Mean" presents the ratios between standard deviations and demand mean values of the solutions to different types of instances. Columns "In-sample Obj" and "Out-of-sample Obj" present the in-sample and out-of-sample objective values. Column "Gap" presents the gaps between in-sample and out-of-sample objectives. In all cases, gaps of the stochastic MIP model are smaller than gaps of the deterministic one, indicating that the stochastic approach better describes the real traffic conditions. When we increase demand mean values, all the objective values increase since there are more vehicles entering the network. When the deviations increase, objective gaps of the deterministic model increase while the stochastic model can still maintain relatively low gaps, showing its solution robustness against parameter uncertainty.

Table 2.5: In-sample and out-of-sample objective values of randomly generated grid networks.

Mean (veh/h)	SD/	In-sample Obj (veh·s)		Out-of-sample Obj (veh·s)		Gap	
	Mean	Deter	SP	Deter	SP	Deter	SP
200 (E-W), 50 (S-N)	2	−420597.37	−556225.39	−597997.23	−596944.23	29.65%	6.81%
200 (E-W), 50 (S-N)	3	−420597.37	−649147.73	−715975.68	−727529.79	41.25%	10.76%
200 (E-W), 50 (S-N)	4	−420597.37	−751408.39	−794377.55	−818953.97	47.05%	8.23%
400 (E-W), 100 (S-N)	2	−687354.94	−865289.83	−981940.02	−994091.40	30.00%	12.96%
400 (E-W), 100 (S-N)	3	−687354.94	−986676.06	−978864.21	−1092679.91	29.77%	9.69%
400 (E-W), 100 (S-N)	4	−687354.94	−1044571.92	−1122114.76	−1179086.03	38.74%	9.87%

Figure 2.3 shows the histograms of out-of-sample objective values and the gaps for the instance with 400 veh/h in East-West direction and SD/Mean ratio = 3. Here the rectangles filled with slashes are associated with the stochastic model and the ones with dots are associated with the deterministic model. In most cases, the objective value and the gap of the stochastic model are smaller than the ones of the deterministic model. The deviation

of the gap of the stochastic model is also less than the deterministic model, showing the solution robustness of the former.



(a) Histogram of out-of-sample objective value

(b) Histogram of objective gap

Figure 2.3: Out-of-sample performance and gap results of the instance with demand mean as 400 (E-W), 100 (S-N) and SD/Mean ratio as 3

Other performance metrics We use travel delay and throughput as the metrics to evaluate signal timing plans. In Table 2.6, we present their values in out-of-sample tests of signal timing plans obtained from the stochastic and deterministic models. Column “Gap” presents the gaps of the corresponding metrics between the two types of models. In most cases, the travel delay of the stochastic model is less than the one of the deterministic counterpart and the throughput of the stochastic one is larger. Therefore, the signal timing plans obtained by the stochastic model outperform the ones of the deterministic model, demonstrating the importance and benefits of considering data uncertainties in traffic signal control. When the demand mean value increases, more improvements are brought by the stochastic model in most cases. For the same mean value, the largest improvement of the stochastic model is often attained when the traffic network is not too idle or too congested (i.e., SD/Mean = 3).

We show the distribution of out-of-sample results in Table 2.7, where we present the standard deviation (SD) and ratio between standard deviation and mean (SD/Mean) of travel delay (D) and throughput (T), indicated by “SD-D”, “SD-T”, “SD/Mean-D”, “SD/Mean-T”, respectively. The ratio between standard deviation and mean of travel delay is less than the ratio of the throughput. In most cases, when the mean value and deviation of the demand increase, there exist more samples where the traffic network is fully congested, leading to less deviation of traffic delay and throughput. Figure 2.4 shows the histograms of out-of-sample

Table 2.6: Out-of-sample evaluation of solutions for randomly generated grid networks with varying demand mean values and SD/Mean ratios.

Mean (veh/h)	SD/	Average Travel Delay (s)			Total	Total Throughput (veh)		
	Mean	Deter	SP	Gap	Arrival (veh)	Deter	SP	Gap
200 (E-W), 50 (S-N)	2	29.36	30.00	-2.14%	2105.61	1869.71	1880.02	0.55%
200 (E-W), 50 (S-N)	3	60.20	54.76	9.04%	2872.96	2280.55	2346.54	2.89%
200 (E-W), 50 (S-N)	4	96.81	87.24	8.24%	3658.03	2536.44	2583.79	1.87%
400 (E-W), 100 (S-N)	2	96.67	92.31	4.70%	4195.30	2966.80	3030.21	1.80%
400 (E-W), 100 (S-N)	3	179.12	149.07	16.57%	5825.33	3191.74	3351.85	4.68%
400 (E-W), 100 (S-N)	4	207.95	197.14	5.20%	7300.13	3364.67	3535.66	5.08%

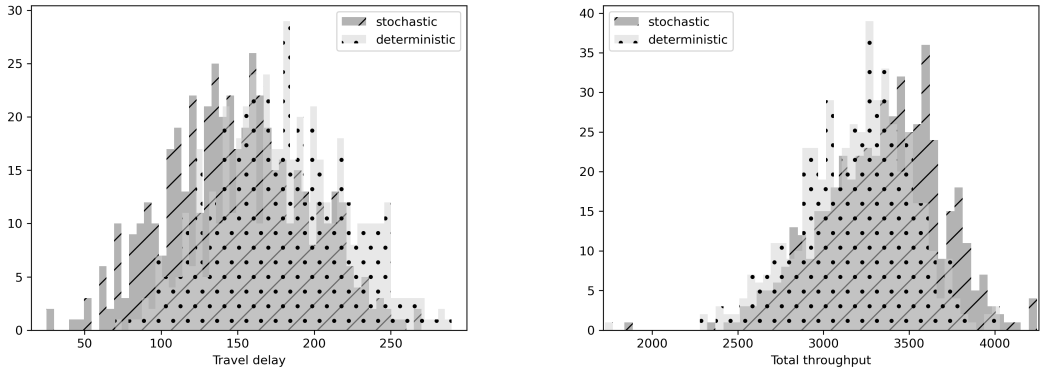
tests on the instance with 400 vehicles per hour in the East-West direction and SD/Mean ratio being 3. In most cases, the stochastic model outperforms the deterministic model in the out-of-sample tests.

Table 2.7: Standard deviation of out-of-sample evaluation of solutions for randomly generated grid networks with varying demand mean values and SD/Mean ratios

Mean of Demand (veh/h)	SD/Mean of	SD-D (s)		SD-T (veh)		SD/Mean-D		SD/Mean-T	
	Demand	Deter	SP	Deter	SP	Deter	SP	Deter	SP
200 (E-W), 50 (S-N)	2	15.03	14.28	312.95	325.95	0.51	0.48	0.17	0.17
200 (E-W), 50 (S-N)	3	29.75	27.52	334.76	366.29	0.49	0.50	0.15	0.16
200 (E-W), 50 (S-N)	4	37.64	36.83	328.85	308.30	0.39	0.42	0.13	0.12
400 (E-W), 100 (S-N)	2	35.09	34.17	336.78	362.72	0.36	0.37	0.11	0.12
400 (E-W), 100 (S-N)	3	42.18	44.82	318.49	346.43	0.24	0.30	0.10	0.10
400 (E-W), 100 (S-N)	4	48.63	48.11	277.71	325.89	0.23	0.24	0.08	0.09

2.5.3 Results of Real-world Traffic Networks

In this section, we present the results of the ADMM-based spatially decentralized Benders on real-world instances. Because solving the model without temporal decomposition is computationally impractical, we pre-determine the cycle length heuristically. In our numerical experiments, we show that the performance of signal timing plans is not sensitive to the cycle length and set the cycle length as the average value of each intersection's cycle length, computed by the rule in Remark 2.4.3. Our numerical results show that the signal timing plan with an in-sample scenario size $K = 10$ obtains the best performance and 500 scenarios are enough for simulating the uncertain traffic reality in the out-of-sample test. The gap between in-sample and out-of-sample tests is around 20%, mainly because we terminate early in solving second-stage problems by ADMM allowing at most 20% gap between the obtained objective value and the optimal one. The details of sensitivity analysis for cycle length, the in-sample and out-of-sample scenario sizes are presented in Appendix A.2.1.



(a) Histogram of traffic delay

(b) Histogram of throughput

Figure 2.4: Out-of-sample delay and throughput results of the instance with demand mean as 400 (E-W), 100 (S-N) and SD/Mean ratio as 3.

We select both instances of morning peak and off-peak hours of the real-world network, to present the computational time and solution performance evaluation in Section 2.5.3.1 and Section 2.5.3.2, respectively. Similar results of the instance of afternoon peak hours are presented in Appendix A.2.3.

2.5.3.1 Results of Morning Peak Hours

We present the computational time of the deterministic and stochastic models in seconds in Table 2.8. The results show that our ADMM-based spatially decentralized Benders algorithm is able to solve both models within an acceptable time limit.

Table 2.8: CPU time results of the traffic network of Downtown Ann Arbor during morning peak hours.

	MP-min (s)	MP-max (s)	MP-A (s)	SP-min(s)	SP-max (s)	SP-A (s)
Deterministic	0.24	0.79	0.48	41.35	41.66	41.54
Stochastic	1.55	2.67	1.94	447.30	474.69	464.51

We present the in-sample objective values, out-of-sample objective values, and gaps between the deterministic and stochastic models in Table 2.9. To evaluate the traffic timing plans, we also present the average travel delay, total arrival, and total throughput in the second half of the table. In Columns "Average Delay" and "Total Throughput", the percentages in Row "Deterministic" show the improvements of relative metrics compared to the baseline, and the percentages in the row "Stochastic" show the improvements of related metrics shown in each column as compared to the deterministic model, respectively.

Table 2.9: Out-of-sample evaluation results of Downtown Ann Arbor during morning peak hours.

	In-sample Obj (veh·s)	Out-of-sample Obj (veh·s)	Gap
Deterministic	−1498923.97	−1939263.92	22.70%
Stochastic	−1540021.95	−2023592.97	23.88%
	Average Delay (s)	Total Arrival (veh)	Total Throughput (veh)
Baseline	352.76	7956.79	3295.30
Deterministic	297.40 (15.69%)	7956.79	3789.91 (15.01%)
Stochastic	276.33 (7.62%)	7956.79	4009.33 (5.47%)

The results in Table 2.9 show that although the gaps between in-sample and out-of-sample objective values of deterministic and stochastic models are similar, the stochastic model outperforms the deterministic counterpart in terms of the average delay and total throughput. Therefore, it is valuable to take into account uncertainties in real-world traffic signal control. We show that both deterministic and stochastic models significantly outperform the baseline in delays and throughputs, with at most 15.69% and 21.67% improvement, respectively, illustrating the advantages of considering the coordination between different intersections.

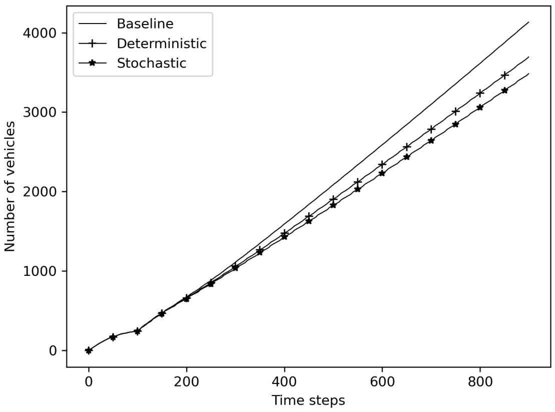
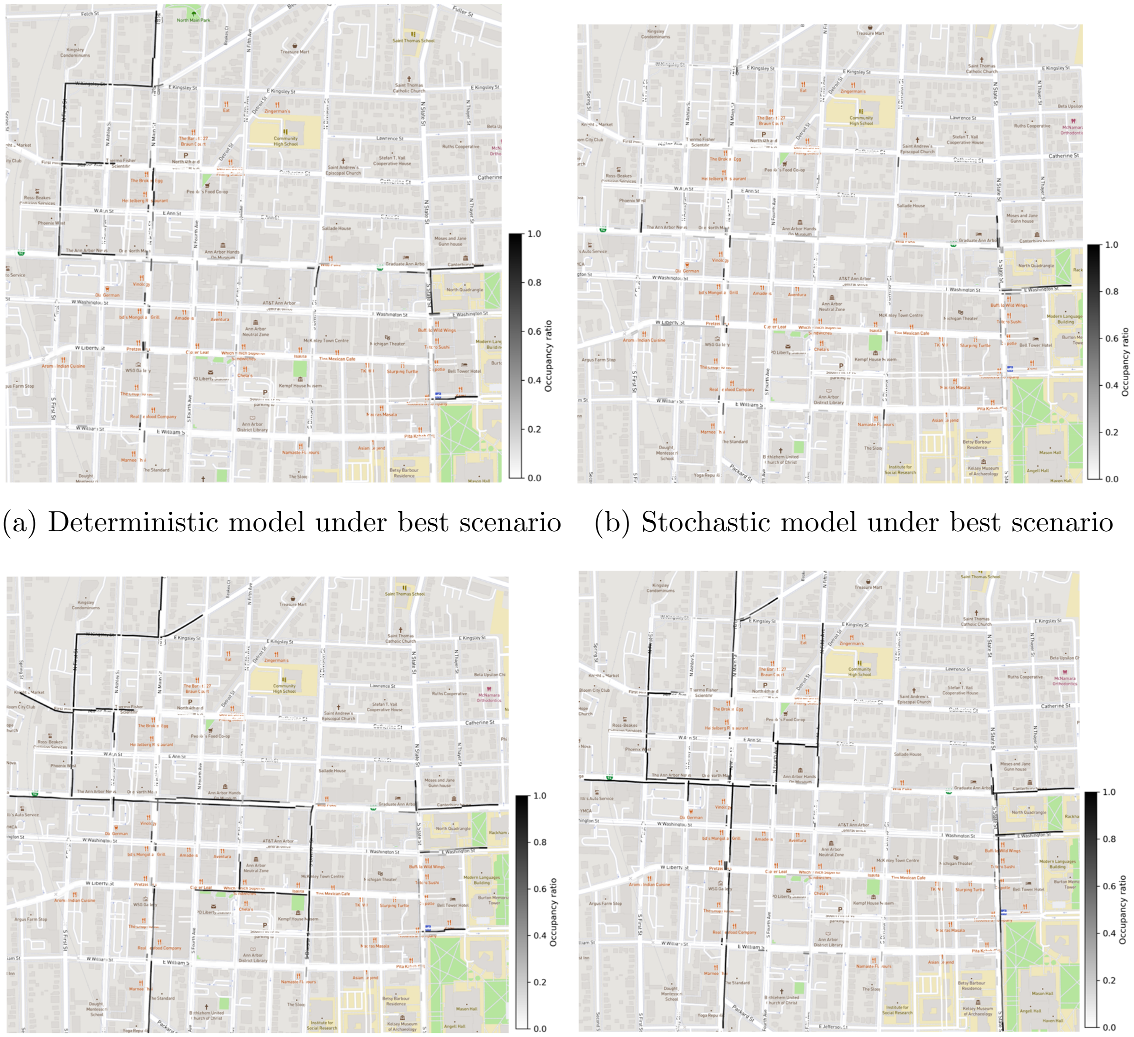


Figure 2.5: Average number of vehicles in the traffic network during morning peak hours for all the scenarios.

We visualize the average number of vehicles in the network across all the scenarios over time in Figure 2.5. The line represents the baseline, the line marked by plus signs represents the deterministic model and the line marked by stars represents the stochastic model. We find that the number of vehicles in the traffic network increases the fastest in the baseline setting, while it increases the slowest in the stochastic model, which also suggests the benefits of stochastic models in preventing congestion. We also provide the visualizations of the number of vehicles under the best and worst scenarios with respect to arrival and delay in the Appendix A.2.2.



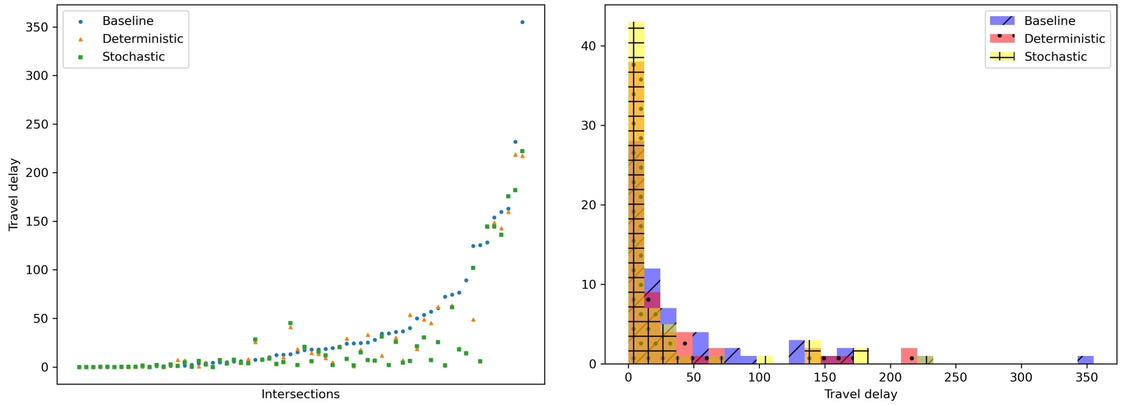
(a) Deterministic model under best scenario (b) Stochastic model under best scenario

(c) Deterministic model under worst scenario (d) Stochastic model under worst scenario

Figure 2.6: Spatial distribution of vehicles at time step 800 during morning peak hours.

Figure 2.6 provides the snapshots of the spatial distribution of the number of vehicles of deterministic and stochastic models under scenarios with the minimum and maximum delay (i.e., the best scenario and worst scenario) at time step $t = 800$. In the figure, we visualize the occupancy ratio, which stands for the ratio between the number of vehicles and the maximum allowed number of vehicles in the cell. The road segments are darker if the occupancy ratio is higher, meaning worse congestion. Figures 2.6a and 2.6b show less congestion given by the stochastic model under the best scenario. Comparing Figures 2.6c and 2.6d, although both models have congestion, there is a smaller number of congested roads given by the signal timing plan produced by the stochastic model. The figures also show that for the deterministic model, the congestion in the East-West direction is worse while for the stochastic model, the congestion in the North-South direction is worse.

We present the traffic delay of solutions using the three different methods (i.e., Baseline, Deterministic, and Stochastic) for each intersection in Figure 2.7a. In addition, we present the histograms of the traffic delay for all the intersections in Figure 2.7b.



(a) Delay of all the methods.

(b) Histogram statistics of the delay.

Figure 2.7: Average travel delay for all the intersections during morning peak hours.

In Figure 2.7, not all the delays at intersections are reduced. Compared to the baseline, the delays at 60% of the intersections are reduced or unchanged by the deterministic model, and 69% are reduced or unchanged by the stochastic model. The histograms show that the maximum delays at all intersections by the deterministic and stochastic models are significantly lower compared to the baseline, showing that considering coordination leads to avoiding over-saturation. The stochastic model shows the best performance by obtaining the highest counts of intersections for the minimum delay group, which is less than 11 seconds. The spatial distribution of intersections with delay reduced or unchanged is presented in Appendix A.2.2.

2.5.3.2 Results of Off-peak Hours

The computational time of the deterministic and stochastic models under off-peak hours are presented in Table 2.10. Both models can be solved within acceptable computational time. Compared to the results under peak hours, it takes less time to solve the models with lower demand.

We compare the performance of the baseline, the deterministic model, and the stochastic model in Table 2.11. We also show the solution improvements of the deterministic model

Table 2.10: CPU time results of the traffic network of Downtown Ann Arbor during off-peak hours.

	MP-min (s)	MP-max (s)	MP-A (s)	SP-min(s)	SP-max (s)	SP-A (s)
Deterministic	0.13	0.40	0.23	40.70	42.36	41.28
Stochastic	0.89	2.72	1.79	424.50	450.26	439.44

compared to the baseline solution, and the solution improvements of the stochastic model compared to the deterministic one, both by percentage. For all the signal timing plans, the average delay of off-peak hours is less than the one of peak hours.

Table 2.11: Out-of-sample evaluation results of Downtown Ann Arbor during off-peak hours.

	In-sample Obj (veh·s)	Out-of-sample Obj (veh·s)	Gap
Deterministic	−580971.48	−1121107.27	48.17%
Stochastic	−592697.54	−1140783.78	48.04%
	Average Delay (s)	Total Arrival (veh)	Total Throughput (veh)
Baseline	117.99	2905.69	2040.82
Deterministic	59.36 (49.69%)	2905.69	2419.07 (18.53%)
Stochastic	51.19 (13.77%)	2905.69	2466.60 (1.96%)

Comparing Table 2.11 with Table 2.9, the gaps between in-sample and out-of-sample objective values of the two models are larger than the gaps during peak hours since the deviation of the source demand is higher. The improvements of the deterministic and stochastic models are both more significant in terms of average delay while the improvement of throughput is similar compared to the results of peak hours. We visualize the average number of vehicles in the network across all the scenarios over time in Figure 2.8. The increase rates of the number of vehicles of all models are slower than the results of peak hours since the mean source demand of off-peak hours is smaller. In Figure 2.9, we visualize the number of vehicles in the network of the scenarios with the minimum delay and maximum delay of the deterministic model, stochastic model, and baseline setting. Figure 2.9a shows that under the scenario with minimum delay, the number of vehicles in the traffic network keeps stable for both deterministic and stochastic models, and there are fewer vehicles of the stochastic model. Figure 2.9b shows that under the scenario with maximum delay, the number of vehicles of all the settings increases and the increasing rate of the stochastic model is slower than the other two settings. We also provide the visualizations of the number of vehicles under the best and worst scenarios with respect to arrival in Appendix A.2.2.

Figure 2.10 provides the snapshots of the spatial distribution of the number of vehicles of the deterministic and stochastic models under the best and worst scenario at time step $t = 800$. There are fewer congested intersections in the results of the stochastic model under both scenarios, compared to the peak hours. While the solutions returned by the stochastic

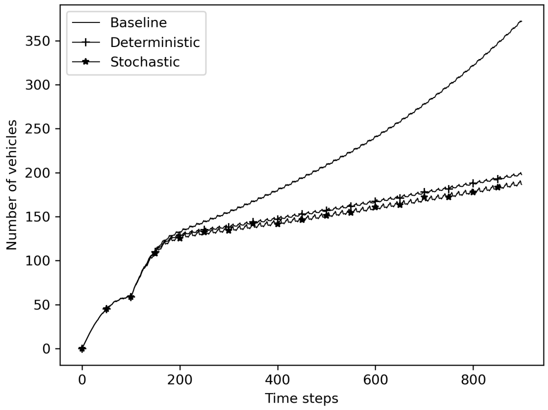
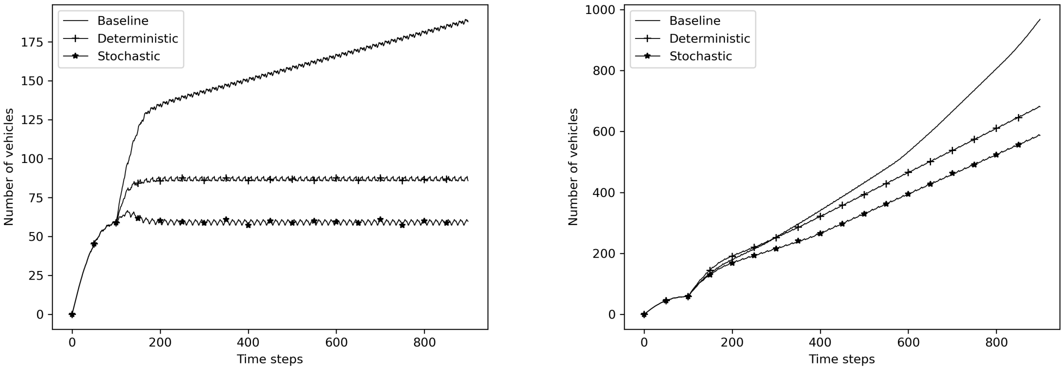


Figure 2.8: Average number of vehicles in the traffic network during afternoon peak hours for all the scenarios.



(a) Scenario with minimum delay

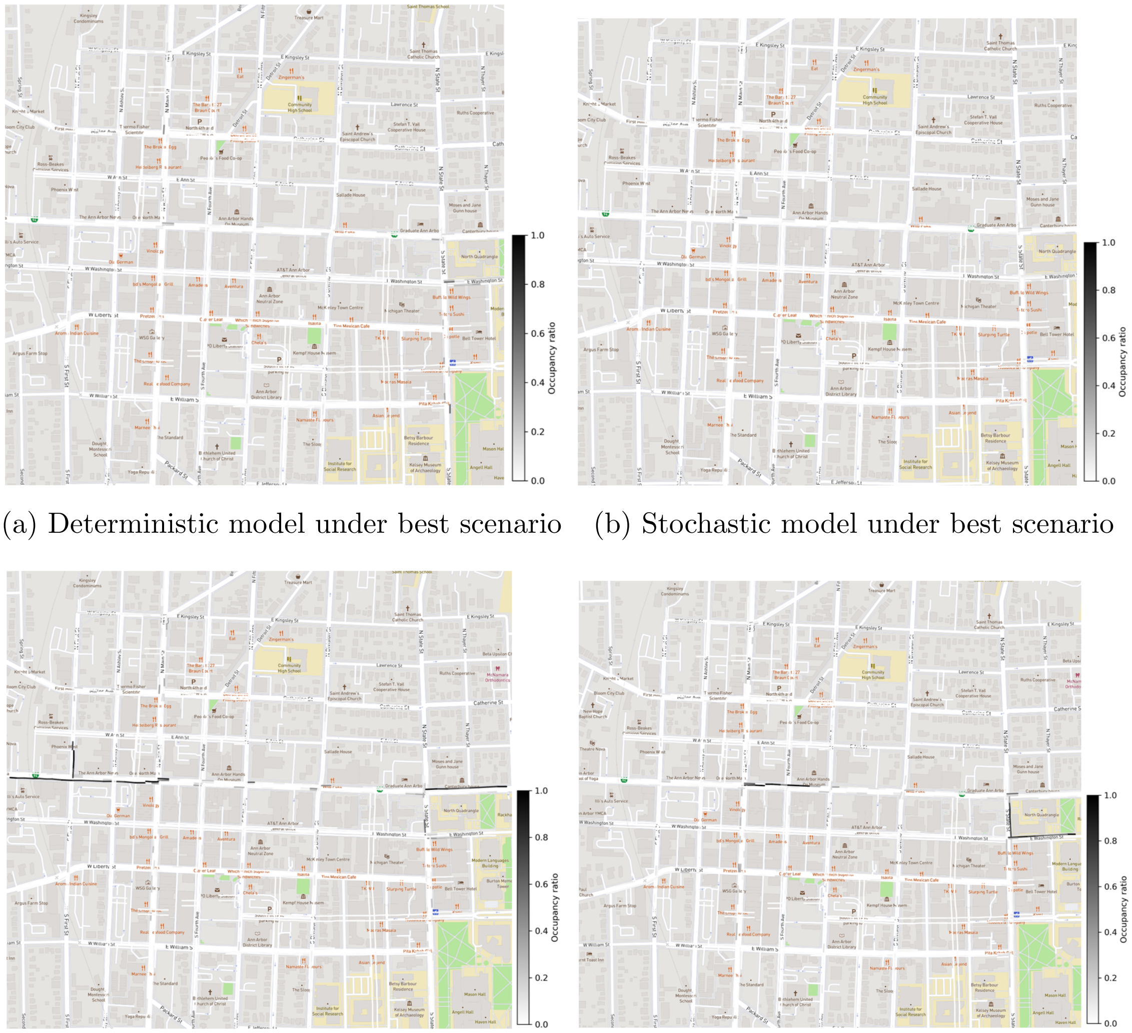
(b) Scenario with maximum delay

Figure 2.9: Number of vehicles in the Downtown Ann Arbor traffic network during off-peak hours.

model can lead to almost fully empty roads, the deterministic counterpart still leaves several roads congested.

We present the average traffic delay of solutions given by the three methods during off-peak hours for each intersection in Figure 2.11a, and the traffic delay for all intersections in Figure 2.11b. All the legends are the same as in Figure 2.7 for the morning peak hours instance.

Compared to baseline, the delays at 64% of the intersections are reduced or unchanged by the deterministic model, and at 70% intersections are reduced or unchanged by the stochastic model. These percentages are higher compared to the ones during morning peak



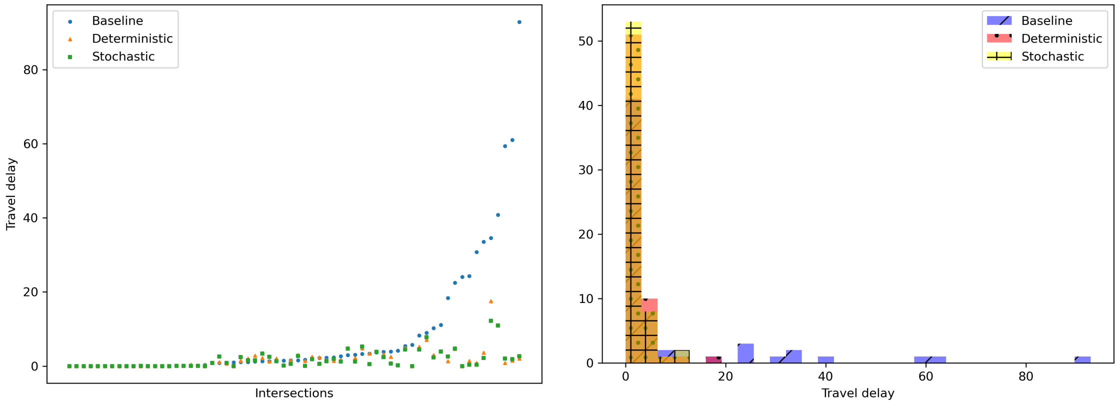
(a) Deterministic model under best scenario (b) Stochastic model under best scenario

(c) Deterministic model under worst scenario (d) Stochastic model under worst scenario

Figure 2.10: Spatial distribution of vehicles at time step 800 during off-peak hours.

hours. The histograms show that both centralized models improve the maximum delays at all intersections compared to the baseline case. The stochastic model obtains the most number of intersections in the minimum delay group, which is less than 2.5 seconds during off-peak hours, demonstrating the benefits of considering coordination and uncertainties. We present the spatial distribution of intersections with delay reduced or unchanged in Appendix A.2.2.

Based on all the above results, we show the benefits of considering travel demand and turning ratio uncertainties as well as the coordination of intersections in traffic signal control with real-world data and road networks. Our models work well for real-world instances during both peak hours and off-peak hours and our stochastic model is more appropriate for instances with high standard deviations of travel demand.



(a) Delay of all the methods.

(b) Histogram statistics of the delay.

Figure 2.11: Average travel delay for all the intersections during off-peak hours.

2.6 Concluding Remarks

In this chapter, we built a CTM-based MIP for traffic signal control and extended the deterministic model to a two-stage stochastic optimization model by considering random source demand and turning ratios. We proposed efficient algorithms for solving the models and overcoming the scalability difficulties. Our algorithm not only reduced the computational time but also ensured the optimality for the non-convex model with mixed-integer variables.

With the numerical results obtained from randomly generated grid networks and real-world traffic networks, we first showed the reduction of the computational time of our algorithm. Then we demonstrated the benefits of considering stochastic traffic demand. Furthermore, we illustrated the advantages of our model to consider the coordination of all the intersections in a real-world traffic network. We noted that the parameters in the model need fine-tuning in practice, and this can be achieved easily since our decision models are solved offline. The proposed method can be used by traffic engineers to generate the fixed-time or background signal timing plan by taking stochastic traffic demand as input in forms of distribution or different scenarios, which is more robust than only considering the average traffic volume.

There are several limitations and corresponding possible directions for future research. Firstly, our proposed formulation only optimizes traffic signal timing parameters based on a pre-determined phase sequence and structure. It will be more flexible if the phase structure (i.e., lead and lag phase) can also be considered as a decision variable. Secondly, like most existing traffic signal control literature [131, 211], vehicle re-routing and induced demand due to the change of the traffic signal timing plan is not explicitly included in this work.

The overall traffic control problem can also be formulated as a more complicated bi-level programming problem by taking vehicle re-routing into consideration. We leave this for future study. In addition, with the development of connected and autonomous vehicles (CAV), the distribution of traffic demand can be directly estimated from real-world data. The combination of data collected from CAV and our proposed approach provides a complete framework for traffic signal optimization in practice. Furthermore, demographics and socioeconomics can be used to predict long-term traffic demand, providing the potential to estimate traffic flow distribution better, which is helpful in guiding the planning and operation of urban traffic optimization.

CHAPTER 3

Binary Control Pulse Optimization for Quantum Systems

3.1 Introductory Remarks

Quantum control [29, 169, 224] is a rich field that started with applications in quantum chemistry [89, 101, 163, 189, 190] but has since expanded to other subfields such as atomic, molecular, and optical physics [76, 95, 210]. Quantum control theory has been used in the quantum information community for a long time, mostly for designing pulse controls and gates in quantum devices [54, 77, 96, 143, 154, 157, 158, 212, 220]; but more recently, researchers have considered using control theories and optimization for the high-level design of quantum algorithms [13, 26, 27, 127, 137, 213, 234]. In this chapter, we focus primarily on the quantum information applications of control theory, but our methods are generalizable to other settings. We do not focus on the analytic aspects of quantum control but on the more mechanistic aspects of how to apply optimization techniques to the problem of designing and finding solutions to a variety of quantum control problems.

Quantum computing architecture falls into different categories, such as quantum circuits, which utilize discrete unitary gates to control a system, and quantum annealing [60, 93], which uses smooth Hamiltonian evolution instead. Quantum control is already familiar within quantum circuit architecture design as a method for designing quantum gates, and several of our examples described in Section 3.2 belong to this category. Within the framework of quantum algorithms, quantum control is often used in variational quantum algorithms, which can straddle the line between discrete gates and smooth evolution. This chapter focuses mainly on discrete binary control, which is more applicable to a quantum gate architecture. Variational techniques have already been implemented experimentally (see, e.g., [84, 156]), and our techniques can potentially enhance the performance of the classical variational loop on such near-term devices.

Most quantum optimal control algorithms are based on gradient descent for better convergence than using gradient-free algorithms [153]. Khaneja et al. [95] proposed the gradient ascent pulse engineering (GRAPE) algorithm for designing pulse sequences in nuclear magnetic resonance. The authors approximated the control function by a piecewise constant function and evaluated the explicit derivative. Based on the GRAPE algorithm, Larocca and Wisniacki [109] proposed a K-GRAPE algorithm by using a Krylov subspace to estimate quantum states during the process of time evolution. Brady et al. [26] applied an analytical framework based on gradient descent to discuss the optimal control procedure of a special control problem that minimizes the energy of a quantum state with a combination of two Hamiltonians. Another class of quantum optimal control algorithms are based on the chopped random basis (CRAB) optimal control technique, which describes the control space by a series of basis functions and optimizes the corresponding coefficients [36, 56]. Sørensen et al. [198] combined the GRAPE algorithm and CRAB algorithm to achieve better results and faster convergence. However, all these algorithms are designed for unconstrained continuous quantum control problems.

In this work, we focus on a quantum pulse optimization problem having binary control variables and restricted feasible regions derived by linear constraints. These constraints describe a so-called bang-bang control in a model that corresponds to a quantum circuit design similar to that in the quantum approximate optimization algorithm (QAOA) [59] and other variational quantum algorithms [20, 37]. Some literature formulates the QAOA algorithm into bang-bang control problems with a fixed evolution time and investigates the performance of multiple methods including Stochastic Descent and Pontryagin's minimum principle for optimizing control models [11, 124]. However, they only consider quantum systems with two controllers, while we propose a more general solution framework for quantum systems with multiple controllers. In this chapter, we introduce four quantum control examples: (i) energy minimization problem, (ii) controlled NOT (CNOT) gate estimation problem, (iii) NOT gate estimation problem with the energy leakage, and (iv) circuit compilation problem. The nonconvexity, binary variables, and restricted feasible sets in these examples lead to extreme challenges and difficulties in solving the related binary quantum control problems.

Methods for binary control mainly include genetic algorithms [146], branch-and-bound [110, 114], and local search [216]. To the best of our knowledge, genetic algorithms have not been applied to binary quantum control, but Zahedinejad et al. [239] showed that such algorithms fail to obtain high-quality solutions even to continuous quantum control problems. Branch-and-bound can find high-quality solutions for binary optimal control, but the long computational time makes the algorithm intractable and hard to use for large-scale practical problems. Vogt and Petersson [216] introduced a trust-region method [153] to solve the single flux quantum control problem with binary controls. We extend their approach by allowing the addition of constraints, such as min-up-time and max-switching constraints to control the number of switches.

The main contributions of this chapter are as follows. First, we propose a generic model including continuous and discretized versions for binary quantum control problems with linear constraints indicating that at each time step there can be only one active control. Second, we introduce an exact penalty function for linear constraints and develop an algorithmic framework combining the GRAPE algorithm and rounding techniques to solve it. Third, to prevent chattering on controls, we propose a new model that includes a total variation (TV) regularizer, and then we propose a new approach based on the alternating direction method of multipliers (ADMM) to solve this model. Fourth, we introduce a modified trust-region method to improve the solutions obtained from these algorithms. Compared with other methods, our algorithmic framework can obtain high-quality binary control sequences and prevent frequent switches. We demonstrate the performance on multiple quantum control examples with various objective functions and controllers.

The chapter is organized as follows. In Section 3.2 we construct generic models of quantum binary control problems using continuous and discretized formulations. We also introduce the corresponding specific formulations for four quantum control examples. In Section 3.3 we review the GRAPE algorithm, utilize a penalty function to relax certain linear constraints, and introduce our rounding techniques. In Section 3.4 we propose and solve the penalty model with the TV regularizer and the corresponding ADMM algorithm. In Section 3.5 we derive trust-region subproblems and propose an approximate local-branching method based on the trust-region algorithm. In Section 3.6 we present numerical simulations for multiple instances of the three quantum control examples in Section 3.2 to demonstrate the computational efficacy and solution performance of our models and algorithms. In Section 3.7 we summarize our work and propose future research directions.

3.2 Formulations for Quantum Control

Consider a quantum system with q qubits and a time horizon $[0, t_f]$, where $t_f > 0$ is defined as the evolution time. Let $H^{(0)} \in \mathbb{C}^{2^q \times 2^q}$ be the intrinsic Hamiltonian. Let N be the number of control operators and $H^{(j)} \in \mathbb{C}^{2^q \times 2^q},\ j = 1, \ldots, N$ be the given control Hamiltonians. Let $X_{\text{init}},\ X_{\text{targ}} \in \mathbb{C}^{2^q \times 2^q}$ be the initial and target unitary operators of the quantum system, respectively. Define variables $u_j(t) \in \{0, 1\},\ j = 1, \ldots, N,\ \forall t$ as the real-valued control functions at time t, and use u to represent the corresponding vector form. Define variables $H(t) \in \mathbb{C}^{2^q \times 2^q}$ as the time-dependent control Hamiltonian function and $X(t) \in \mathbb{C}^{2^q \times 2^q}$ as the time-dependent unitary quantum operator function. A generic quantum control problem for optimizing a function of quantum operators is formulated as

$$
\begin{aligned}
(P) \quad \min \quad & F(X(t_f)) && \text{(3.1a)}\\
\text{s.t.} \quad & H(t) = H^{(0)} + \sum_{j=1}^{N} u_j(t) H^{(j)},\ \forall t \in [0, t_f] && \text{(3.1b)}\\
& \frac{d}{dt} X(t) = -iH(t)X(t),\ \forall t \in [0, t_f] && \text{(3.1c)}\\
& X(0) = X_{\text{init}} && \text{(3.1d)}\\
& \sum_{j=1}^{N} u_j(t) = 1,\ \forall t \in [0, t_f] && \text{(3.1e)}\\
& u_j(t) \in \{0, 1\},\ j = 1, \dots, N. && \text{(3.1f)}
\end{aligned}
$$

The objective function (3.1a) is a general function with respect to the final unitary operator $X(t_f)$. We assume that F is a differentiable function. In Sections 3.2.1–3.2.4, we provide the specific formulations of the objective function for four quantum control examples. Constraint (3.1b) describes the computation of the time-dependent Hamiltonian function based on the intrinsic Hamiltonian, control Hamiltonians, and control functions. Constraint (3.1c) is the ordinary differential equation describing the time evolution of the operator of the quantum system with the same units for energy and frequency, setting $\hbar = 1$. Constraint (3.1d) is the initial condition of the operator. Constraint (3.1e) indicates that the summation of all the control functions should be one at all times, which is described by the Special Ordered Set of Type 1 (SOS1) property in optimal control theory [181]. In binary control, this SOS1 property guarantees that only one control field will be active at a time that mimics the bang-bang nature of some quantum applications such as the quantum approximate optimization algorithm and the variational quantum eigensolver (VQE). Constraints (3.1f) require all time-based values of the control functions to be feasible.

Following the time discretization process in [95], we explicitly integrate constraint (3.1c). In particular, we divide the time horizon $[0, t_f]$ into T time intervals $[t_{k-1}, t_k),\ k = 1, \dots, T$. We consider time intervals with an equal length represented by Δt, but our work readily extends to nonuniform discretizations. For each controller $j = 1, \dots, N$ and each time step $k = 1, \dots, T$, we define discretized binary control variables as $u_{jk} \in \{0, 1\}$. For each time step $k = 1, \dots, T$, we define the discretized time-dependent Hamiltonians as $H_k \in \mathbb{C}^{2^q \times 2^q}$ and the quantum operators as $X_k \in \mathbb{C}^{2^q \times 2^q}$. The differential equation (3.1c) is thus approximated by

$$\frac{d}{dt}X(t) = -iH_kX(t),\ \forall t \in [t_{k-1}, t_k),\ k = 1, \ldots, T. \tag{3.2}$$

For each $k = 1, \ldots, T$, the linear differential equation (3.2) is a Schrödinger equation of a unitary operator, and we obtain an explicit solution as

$$X(t) = \exp\left\{-iH_k(t - t_{k-1})\right\} X(t_{k-1}),\ \forall t \in [t_{k-1}, t_k),$$

because H_k is time independent. We then obtain a discretized quantum control problem (DQCP) as a discretization of the differential control model (P) using explicit solutions on each interval $[t_{k-1}, t_k)$ for unitary operators as

$$(DQCP) \quad \min_{u,X,H} \quad F(X_T) \tag{3.3a}$$

$$\text{s.t.} \quad H_k = H^{(0)} + \sum_{j=1}^{N} u_{jk}H^{(j)},\ k = 1, \ldots, T \tag{3.3b}$$

$$X_k = e^{-iH_k\Delta t}X_{k-1},\ k = 1, \ldots, T \tag{3.3c}$$

$$X_0 = X_{\text{init}} \tag{3.3d}$$

$$\sum_{j=1}^{N} u_{jk} = 1,\ k = 1, \ldots, T \tag{3.3e}$$

$$u_{jk} \in \{0, 1\},\ j = 1, \ldots, N,\ k = 1, \ldots, T. \tag{3.3f}$$

The objective function (3.3a) is the objective function (3.1a) evaluated at the approximated final operator $X_T \approx X(t_f)$. Constraints (3.3b)–(3.3f) are the discretized formulations of constraints (3.1b)–(3.1f). In the following sections we use this discretized formulation to develop our algorithms. We present the following discussion on the relationship between the continuous and discretized formulation.

Remark 3.2.1. *Problem* (DQCP) *is equivalent to* (P) *for piecewise-constant controls $u_j,\ j = 1, \ldots, N$. In addition, we have*

$$\int_0^{t_k} u_j(\tau)d\tau = \sum_{\tau=1}^{k} u_{j\tau}\Delta t,\ j = 1, \ldots, N,\ k = 1, \ldots, T. \tag{3.4}$$

Remark 3.2.2. *The model can be generalized to instances with multiple active controllers in any time interval by considering each possible combination of the controllers. For an instance with L control Hamiltonians $H^{(1)}, \ldots, H^{(L)}$, we consider $N = 2^L$ combinations*

$H^{(l_1,\dots,l_k)}$, $\forall \{l_1,\dots,l_k\} \subseteq \{1,\dots,L\}$ *and define control functions for each combination. Thus, the original problem is converted to a new problem with only one active controller in any time interval.*

In this chapter we consider four examples with different objective functions and control Hamiltonians. In the following sections we introduce the continuous model of four example problems in quantum control. They can be formulated as discretized models (DQCP) following the above discretization process.

3.2.1 Energy Minimization Problem

Consider a quantum spin system consisting of q qubits, no intrinsic Hamiltonian, and two control Hamiltonians $H^{(1)}$, $H^{(2)}$. The initial operator X_{init} is a 2^q-dimensional identity matrix. Let $|\psi_0\rangle$ be the ground state of the first control Hamiltonian $H^{(1)}$. The ground state energy of this quantum system is the minimum eigenvalue of the control Hamiltonian $H^{(2)}$, represented by $E_{\min}$ in our model. Since the ground state energy differs across problem instances, it is important to compare performance consistently across these instances. Therefore, we maximize the approximation ratio, defined as the achieved energy divided by the true ground state energy. The problems we study are well defined so that the initial state of the system has zero energy with respect to $H^{(2)}$; furthermore, the ground state energy will always be a large negative number. Therefore, the approximation ratio reflects the energy level achieved by our procedure from the initial energy of zero to the true ground state energy. Because our procedure can improve the state relative only to its initial configuration, a negative approximation ratio is not possible.

To ensure consistency with the objective functions of our other two examples, we minimize one minus the approximation ratio. Because the true ground state energy is negative, this formula is equivalent to minimizing the achieved energy. The continuous-time formulation is

$$\begin{aligned}
\min \quad & 1 - \langle\psi_0| X(t_f)^\dagger H^{(2)} X(t_f) |\psi_0\rangle / E_{\min} && \text{(3.5a)}\\
\text{s.t.} \quad & H(t) = u_1(t)H^{(1)} + u_2(t)H^{(2)},\ \forall t \in [0, t_f] && \text{(3.5b)}\\
& H^{(1)} = -\sum_{i=1}^{q} \sigma_i^x,\ H^{(2)} = \sum_{ij} J_{ij}\sigma_i^z\sigma_j^z && \text{(3.5c)}\\
& \text{Constraints (3.1c)–(3.1e)} && \\
& u_1(t),\ u_2(t) \in \{0,1\}, && \text{(3.5d)}
\end{aligned}$$

where $\cdot^\dagger$ represents the conjugate transpose of a matrix and $\langle\cdot|$ represents the conjugate transpose of a quantum state vector $|\cdot\rangle$. The matrix $[J_{ij}],\ i,j = 1,\dots,q$ is the adjacency matrix of a randomly generated graph with q nodes, and $\sigma_i^x,\ \sigma_i^z$ are Pauli matrices of qubit i.

3.2.2 CNOT Gate Estimation Problem

Our second example is defined on an isotropic Heisenberg spin-1/2 chain with length 2 according to [160]. For our study we consider just the unitary evolution with no energy leakage. The quantum system includes 2 qubits, an intrinsic Hamiltonian $H^{(0)}$, and two control Hamiltonians $H^{(1)}, H^{(2)}$. The initial operator X_{init} is a 4-dimensional identity matrix and the target operator is the matrix formulation of the CNOT gate represented by $X_{\text{CNOT}} \in \mathbb{C}^{4\times 4}$. This problem's continuous-time formulation is

$$\begin{aligned}
\min_{u(t),X(t),H(t)} \quad & 1 - \frac{1}{4}\left|\operatorname{tr}\left\{X_{\text{CNOT}}^{\dagger}X(t_f)\right\}\right| && (3.6a)\\
\text{s.t.} \quad & H(t) = H^{(0)} + u_1(t)H^{(1)} + u_2(t)H^{(2)},\ \forall t \in [0,t_f] && (3.6b)\\
& H^{(0)} = \sigma_1^x\sigma_2^x + \sigma_1^y\sigma_2^y + \sigma_1^z\sigma_2^z && (3.6c)\\
& H^{(1)} = \sigma_1^x,\ H^{(2)} = \sigma_1^y && (3.6d)\\
& \text{Constraints (3.1c)–(3.1d)} && \\
& u_1(t),\ u_2(t) \in \{0,1\}, && (3.6e)
\end{aligned}$$

where $\sigma_i^x,\ \sigma_i^y,\ \sigma_i^z,\ i = 1,2$ are Pauli matrices of two qubits. The goal is to minimize the objective function, defined as the infidelity between the final operator and the target operator.

3.2.3 NOT Gate Estimation with Energy Leakage

We consider an anharmonic multilevel system that is widely used in quantum experiments to implement physical qubits. Following [145], we consider the lowest three levels of an energy spectrum with only nearest-level coupling for a single qubit $q = 1$. The first two levels comprise the qubit and the third level models the energy leakage. Under this setting, the system contains an intrinsic Hamiltonian $H^{(0)}$ and two control Hamiltonians $H^{(1)},\ H^{(2)}$. The initial operator X_{init} is a 3-dimensional identity matrix, and the target operator is the matrix formulation of the NOT gate corresponding to the lowest three levels of the energy spectrum represented by $X_{\mathrm{NOT}} \in \mathbb{C}^{3\times 3}$, which is

$$X_{\mathrm{NOT}} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{3.7}$$

The continuous-time formulation of this problem is

$$\begin{aligned}
\min_{u(t),X(t),H(t)} \quad & 1 - \frac{1}{2}\left|\mathrm{tr}\left\{X_{\mathrm{NOT}}^{\dagger} X(t_f)\right\}\right| && \text{(3.8a)}\\
\text{s.t.} \quad & H(t) = H^{(0)} + u_1(t)H^{(1)} + u_2(t)H^{(2)},\ \forall t \in [0, t_f] && \text{(3.8b)}\\
& H^{(0)} = \mu_1|1\rangle\langle 1| + \mu_2|2\rangle\langle 2| && \text{(3.8c)}\\
& H^{(1)} = \omega_1(|0\rangle\langle 1| + |1\rangle\langle 0|) + \omega_2(|1\rangle\langle 2| + |2\rangle\langle 1|) && \text{(3.8d)}\\
& H^{(2)} = \omega_1(i|0\rangle\langle 1| - i|1\rangle\langle 0|) + \omega_2(i|1\rangle\langle 2| - i|2\rangle\langle 1|) && \text{(3.8e)}\\
& \text{Constraints (3.1c)–(3.1d)} && \\
& u_1(t),\ u_2(t) \in \{0, 1\}, && \text{(3.8f)}
\end{aligned}$$

where $|j\rangle$ represents a vector with the j-th element as 1 and other elements as 0. The parameters $\mu_1,\ \mu_2$ weigh the relative strength of transitions and the parameters $\omega_1,\ \omega_2$ correspond to the drive frequency. We present the specific values of the parameters used in numerical simulations in Section 3.6.2.

3.2.4 Circuit Compilation Problem

For a general quantum algorithm, each quantum circuit needs to be compiled to a representation imposed by specific controllers and constraints in order to execute the algorithm on specific quantum devices. We take quantum circuits generated by the unitary coupled-cluster single-double (UCCSD) method [15, 177] for estimating the ground state energy of molecules in quantum chemistry as examples. We consider the Hamiltonian controllers in the gmon-qubit system [40] with q qubits. Each qubit has a flux-drive controller and a charge-drive controller. In addition, there is a rectangular-grid topology with nearest-neighbor connectivity denoted by E, and each pair of connected qubits $(j_1, j_2) \in E$ has a controller. Define $w_{j_1 j_2}(t)$ as control functions for the connected qubits controllers and use $w(t)$ to represent the corresponding vector form of $w_{j_1 j_2}(t)$. The initial operator X_{init} is a 2^q-dimensional identity matrix. The target operator is the matrix formulation of the quantum circuit generated by UCCSD for specific molecules represented by $X_{\mathrm{UCCSD}} \in \mathbb{C}^{2^q \times 2^q}$. We refer interested readers to Gokhale et al. [75] for more details. The continuous-time formulation of this problem is

$$\begin{aligned}
\min \quad & 1-\frac{1}{2^q}\left|\operatorname{tr}\left\{X^\dagger_{\mathrm{UCCSD}}X(t_f)\right\}\right| && \text{(3.9a)}\\
\text{s.t.} \quad & H(t)=\sum_{j=1}^{2q}u_j(t)H^{(j)}+\sum_{(j_1,j_2)\in E}w_{j_1j_2}(t)H^{(j_1j_2)},\ \forall t\in[0,t_f] && \text{(3.9b)}\\
& H^{(2j-1)}=J_c\sigma^x_j,\ j=1,\dots,q && \text{(3.9c)}\\
& H^{(2j)}=J_f\begin{pmatrix}0&0\\0&1\end{pmatrix},\ j=1,\dots,q && \text{(3.9d)}\\
& H^{(j_1j_2)}=J_e\sigma^x_{j_1}\sigma^x_{j_2},\ (j_1,j_2)\in E && \text{(3.9e)}\\
& \text{Constraints (3.1c)–(3.1e)} && \\
& u_j(t)\in\{0,1\},\ w_{j_1j_2}(t)\in\{0,1\},\ j=1,\dots,2q,\ (j_1,j_2)\in E, && \text{(3.9f)}
\end{aligned}$$

where $J_c,\ J_f,\ J_e$ are parameters corresponding to the quantum system and σ^x_j is the Pauli matrix of qubit j. We finish this section by showing a general property of the objective function in optimal control problems.

Theorem 3.2.1. *For a quantum system with q qubits, the infidelity $1-\frac{1}{2^q}\left|\operatorname{tr}\left\{X^\dagger_{targ}\,X(t_f)\right\}\right|\in[0,1]$. The objective functions of the optimal control models for CNOT gate* (3.6) *and circuit compilation problem* (3.9) *are bounded between zero and one.*

Proof. By definition, the objective function value cannot be larger than 1. From the definition of our quantum control problem, the quantum operators X_{targ} and $X(t_f)$ are both unitary matrices. Therefore, we can write $X(t_f)$ as $X(t_f)=\sum_{i=1}^{2^q}x_ie_i^T$, where x_i are the orthonormal column vectors of $X(t_f)$ and e_i are $2^q\times 1$ vectors with the ith component as 1 and all the other components as 0. Thus we have

$$\left|\operatorname{tr}\left\{X^\dagger_{\mathrm{targ}}\,X(t_f)\right\}\right|=\left|\sum_{i=1}^{2^q}\operatorname{tr}\left\{X^\dagger_{\mathrm{targ}}\,x_ie_i^T\right\}\right|\le\sum_{i=1}^{2^q}\left\|X^\dagger_{\mathrm{targ}}\,x_i\right\|_\infty\le 2^q. \tag{3.10}$$

The first inequality follows from the fact that the trace of a product is invariant under cyclic permutations of the factors. The second inequality follows from $\left\|X^\dagger_{\mathrm{targ}}\,x_i\right\|_\infty\le 1$. As a result, the objective function is no less than 0, and this completes the proof. □

Remark 3.2.3. *All the problems in this section fall into a class known as right-invariant systems [92] that consists of control problems on Lie groups. Specifically, this class consists of all control problems described by a Hamiltonian of the form $H(t)=H^{(0)}+\sum_j u_j(t)H^{(j)}$.*

Controllability results on this class of control problems are well known [92, 171]. A test of controllability of these problems consists of examining the dynamical Lie algebra generated by the Hamiltonians, $H^{(j)}$, and their nested commutators and comparing that Lie algebra with the Lie algebra of the subgroup defined by the symmetries in a given problem. For the systems considered in this chapter, there are indeed enough control Hamiltonians to reach the target state by using piecewise continuous control functions, $u_j(t)$.

3.3 Binary Quantum Control Algorithms

In this section we propose models and algorithms to obtain binary quantum control results for the discretized model (DQCP). In Section 3.3.1 we review the adjoint approximation method of gradients used in the GRAPE algorithm for solving the continuous relaxation of the optimal control problem (DQCP) and derive the gradient update rule for the aforementioned four specific examples. In Section 3.3.2 we propose an exact penalty function for the SOS1 property that allows us to add constraints to GRAPE. In Section 3.3.3 we introduce rounding techniques to obtain binary controls. These algorithms form the basis of our discrete framework discussed in Sections 3.4–3.5.

The overall framework is outlined in Figure 3.1, which presents the methods in Sections 3.3–3.5 and the corresponding tables and results. We summarize the algorithms of our framework in Table 3.1. In the figure, the rectangles represent the three formulations of our model, and the ellipses represent the algorithms. The parallelograms show the result tables corresponding to algorithms, where the dashed parallelograms refer to the results of the continuous models and the solid parallelograms refer to the results of the discrete models.

Table 3.1: List of acronyms of algorithms in numerical simulations. The pGRAPE and GRAPE algorithms are identical when the SOS1 constraint is absent.

Name	Algorithms
pGRAPE	Solving the continuous relaxation with squared L_2-penalty function by penalized GRAPE.
TR	Solving the continuous relaxation with TV regularizer and SOS1 property by a trust-region method.
ADMM	Solving the continuous relaxation with TV regularizer and SOS1 property by ADMM.
SUR	Rounding the continuous solutions without hard control constraints.
MT	Rounding the continuous solutions with min-up-time constraints.
MS	Rounding the continuous solutions with max-switching constraints.
ALB	Improving the binary solutions by the approximate local-branching method.

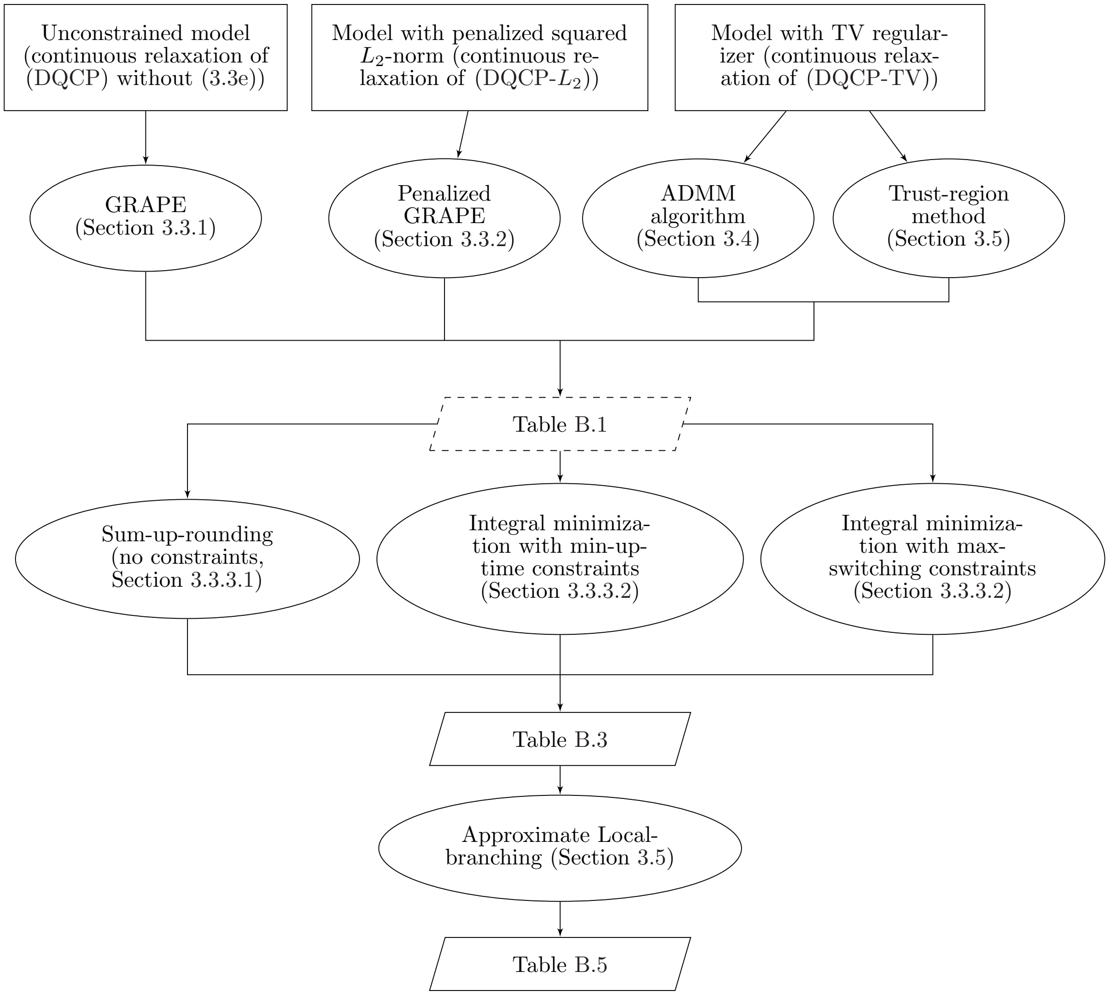

Unconstrained model (continuous relaxation of (DQCP) without (3.3e))
Model with penalized squared L_2-norm (continuous relaxation of (DQCP-L_2))
Model with TV regularizer (continuous relaxation of (DQCP-TV))
GRAPE (Section 3.3.1)
Penalized GRAPE (Section 3.3.2)
ADMM algorithm (Section 3.4)
Trust-region method (Section 3.5)
Table B.1
Sum-up-rounding (no constraints, Section 3.3.3.1)
Integral minimization with min-up-time constraints (Section 3.3.3.2)
Integral minimization with max-switching constraints (Section 3.3.3.2)
Table B.3
Approximate Local-branching (Section 3.5)
Table B.5

Figure 3.1: Overview and results of the algorithmic framework.

3.3.1 GRAPE Approach for Solving Continuous Relaxations

The GRAPE algorithm [95] is a gradient descent algorithm for solving the unconstrained continuous control problem. We derive a gradient descent algorithm based on the adjoint approach used in the GRAPE algorithm to solve the continuous relaxation of our discretized model (DQCP) without the SOS1 property enforced by constraint (3.3e). We eliminate variables X and H by converting them into implicit functions of the control variables u using constraints (3.3b)–(3.3d). Then the minimization problem of $F(X_T)$ on $u,\ X,\ H$ is converted to the minimization problem of $F(X_T(u))$ over the variables u. The goal of the adjoint approach is to approximate the gradient of the objective function F with respect to control variables $u_{jk},\ j = 1, \ldots, N,\ k = 1, \ldots, T$ based on the discretized time horizon. For simplicity, we define propagators for each time step $k = 1, \ldots, T$ as $U_k = \exp\{-iH_k \Delta t\}$.

For small Δt, the gradient of U_k corresponding to u_{jk} is estimated as

$$\frac{\partial U_k}{\partial u_{jk}} = -i\Delta t H^{(j)} U_k,\ j = 1, \dots, N,\ k = 1, \dots, T. \tag{3.11}$$

The gradient of the objective function with respect to the propagators depends on its specific formulation. We discuss gradients of two specific functions for the examples in Section 3.2.

Energy function For generality, we use $\bar{H}$ to represent the Hamiltonian in the energy objective function, which means that the goal is to minimize the energy corresponding to $\bar{H}$. Specifically, we have $\bar{H} = H^{(2)}$ in the example in Section 3.2.1. The general energy function can be expressed as

$$\langle\psi_0| X_T^\dagger \bar{H} X_T |\psi_0\rangle = \langle\psi_0| U_1^\dagger \cdots U_T^\dagger \bar{H} U_T \cdots U_1 |\psi_0\rangle. \tag{3.12}$$

For each time step $k = 1, \dots, T-1$, define variables $|\kappa_k\rangle = U_{k+1}^\dagger \cdots U_T^\dagger \bar{H} X_T |\psi_0\rangle$. For the last time step T, we define a variable $|\kappa_T\rangle = \bar{H} X_T |\psi_0\rangle$. Then the gradient with respect to u_{jk} is computed as

$$\frac{\partial F}{\partial u_{jk}} = \frac{2}{E_{\min}} \mathrm{Re}\left[i\Delta t \langle\kappa_k| H^{(j)} X_k |\psi_0\rangle\right],\ j = 1, \dots, N,\ k = 1, \dots, T. \tag{3.13}$$

Infidelity function The infidelity function can be expressed as

$$1 - \frac{1}{2^q}\left|\mathrm{tr}\left\{X_{\mathrm{targ}}^\dagger X_T\right\}\right| = 1 - \frac{1}{2^q}\left|\mathrm{tr}\left\{X_{\mathrm{targ}}^\dagger U_T \cdots U_1 X_0\right\}\right|. \tag{3.14}$$

For each time step $k = 1, \dots, T-1$, define variables $\lambda_k = U_{k+1}^\dagger \cdots U_T^\dagger X_{\mathrm{targ}}$. For the last time step T, we define a variable $\lambda_T = X_{\mathrm{targ}}$. Then the gradient of the trace with respect to u_{jk} is computed as

$$\frac{\partial\, \mathrm{tr}\left\{X_{\mathrm{targ}}^\dagger X_T\right\}}{\partial u_{jk}} = -i\Delta t \lambda_k^\dagger H^{(j)} X_k,\ j = 1, \dots, N,\ k = 1, \dots, T. \tag{3.15}$$

Using the definition of objective function F, we compute the gradient as

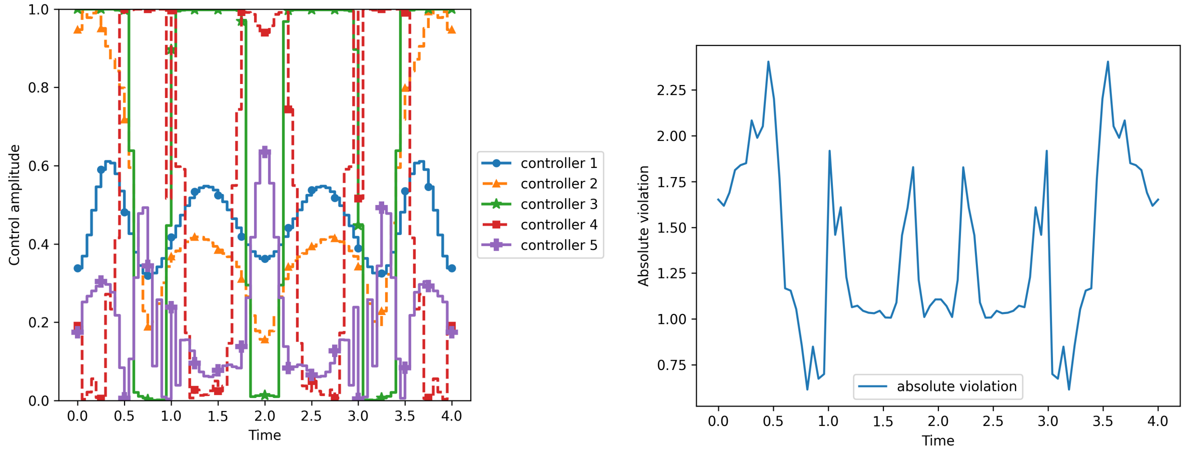
$$\frac{\partial F}{\partial u_{jk}} = \frac{1}{2^q} \mathrm{Re}\left[i\Delta t\, \mathrm{tr}\left\{\lambda_k^\dagger H^{(j)} X_k\right\} e^{-i \arg\left(\mathrm{tr}\left\{X_{\mathrm{targ}}^\dagger X_T\right\}\right)}\right],\ j = 1, \dots, N,\ k = 1, \dots, T, \tag{3.16}$$

where $\arg(\cdot)$ represents the argument of a complex number.

With these computed gradients, we minimize the reduced function $F(X_T(u))$ with respect to control variables $u \in [0,1]^{N \cdot T}$ by L-BFGS-B [34], a well-known quasi-Newton algorithm for solving bound-constrained optimization problems.

Remark 3.3.1. *Our implementation of GRAPE extends the classical gradient-ascend scheme (see, e.g., [95, 109]) in a number of ways: We use a bound-constrained quasi-Newton method to solve problems with bounded control variables, and we introduce a penalized GRAPE (denoted by pGRAPE) that adds a quadratic penalty of the SOS1 property in Section 3.3.2.*

We take the circuit compilation problem (Section 3.2.4) on the molecule H_2 (dihydrogen) as an example to show the control results, and we present the continuous control results without the SOS1 property obtained by the GRAPE algorithm in Figure 3.2a. We define the absolute violation of the SOS1 property at each time step $k = 1, \ldots, T$ as $|\sum_{j=1}^{N} u_{jk} - 1|$ and present the value at each time step in Figure 3.2b. The figure shows that the results violate the SOS1 property at all time steps, with the maximum absolute violation being 2.404. Hence we introduce a penalized function in Section 3.3.2 to impose the SOS1 property.



(a) Control results of 5 controllers

(b) Absolute violation at each step

Figure 3.2: (a) Control results of the continuous relaxation of the discretized model (DQCP) without the SOS1 property constraint (3.3e) of the circuit compilation example on the molecule H_2. (b) Absolute violation at each time step.

3.3.2 Penalty Function for SOS1 Property

Our GRAPE algorithm is defined only for unconstrained or bound-constrained problems [95]. Hence, we convert the SOS1 property (3.3e) into a squared L_2-penalty term and add it to the objective function. Define the squared L_2 function of the constraint violation as

$$l(u,T) = \sum_{k=1}^{T} \left(\sum_{j=1}^{N} u_{jk} - 1 \right)^2 . \tag{3.17}$$

We let a constant $\rho > 0$ be the penalty parameter. Then the model with the penalty function is

$$\min \; F(X_T) + \rho l(u,T) \quad \text{s.t. Constraints (3.3b)–(3.3e).} \tag{DQCP-L_2}$$

We use $l(u^*_\rho, T)$ to denote the optimized value of the squared L_2 function $l(u,T)$ under the penalty parameter ρ. The following theorem discusses the exactness of the penalized objective function.

Theorem 3.3.1. *[Exactness of Squared L_2-Penalty] Let z_1 and $z_2(\rho)$ be the optimal values of* (DQCP) *and* (DQCP-L_2)*, respectively. There exists $\tilde{\rho} < \infty$ such that $z_1 = z_2(\rho)$ for all $\rho > \tilde{\rho}$.*

A more general version of Theorem 3.3.1 is proved in [193]. Next we discuss the value of the optimized squared L_2-penalty term with respect to the penalty parameter ρ for the continuous relaxation of (DQCP-L_2).

Theorem 3.3.2. *For the continuous relaxation of* (DQCP-L_2)*, if the original objective function F is upper bounded by a constant C_F and* (DQCP) *is feasible, then for any T the optimized squared L_2-penalty term uniformly holds that $|l(u^*_\rho, T)| \leq 2C_F/\rho$.*

A detailed proof is provided in Appendix B.1.1.

Remark 3.3.2. *The GRAPE algorithm is still able to solve the unconstrained continuous relaxation with the penalty function* (DQCP-L_2)*. We propose a penalized version (pGRAPE) in Section 3.3.1 by adding a term $2\rho \left(\sum_{j=1}^{N} u_{jk} - 1 \right)$ to the approximated gradient of the objective function.*

Remark 3.3.3. *For the quantum control problem with two controllers, the SOS1 can be enforced directly by substituting*

$$u_{2k} = 1 - u_{1k}, \; k = 1, \ldots, T \tag{3.18}$$

in the model instead of using the penalty function.

We show the control results obtained from the continuous relaxation with the squared L_2-penalty function and penalty parameter $\rho = 1.0$ in Figure 3.3. The value of the squared penalty term is 5.55×10^{-7}, leading to a small error between continuous and binary results with the SOS1 property obtained by rounding techniques. However, we still need to provide rounding techniques to obtain binary results.

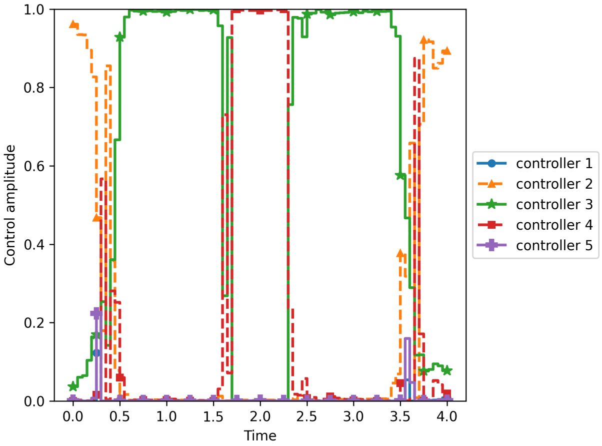


Figure 3.3: Control results of the continuous relaxation of the discretized model with a squared L_2-penalty function (DQCP-L_2) of the circuit compilation example on the molecule H_2 (objective value: 4.37E−07).

3.3.3 Rounding Techniques and Optimality Guarantees

In this section we introduce rounding techniques to obtain binary control results and investigate their optimality guarantees. In Section 3.3.3.1 we review the sum-up rounding technique designed for binary controls without constraints and discuss the difference compared with the continuous results. In Section 3.3.3.2 we introduce the integral minimization problem for rounding to obtain restricted binary controls.

3.3.3.1 Sum-Up Rounding

The sum-Up rounding (SUR) strategy proposed by Sager et al. [181] is a well-known method to obtain integer controls from continuous, relaxed ones in optimal control theory; see, for example, [135, 236]. One concern when applying SUR is that the solution of the relaxation does not satisfy the SOS1 constraints because the exactness of the squared L_2-penalty is only for binary controls. We show that as long as the violation of the SOS1 constraint is small, the solution that is constructed via SUR still satisfies the strong convergence properties of SUR. We define the discretized continuous control function u^c as

$u^c_{jk} \in [0,1],\ j = 1,\ldots,N,\ k = 1,\ldots,T$. We define the discretized binary control function u^b as $u^b_{jk} \in \{0,1\}\,,\ j = 1,\ldots,N,\ k = 1,\ldots,T$. For each time step $k = 1,\ldots,T$, we denote the vector form of control variables as u^c_k and u^b_k. We define the current cumulative deviation between continuous and binary controls as $\hat{p}_{jk},\ j = 1,\ldots,N,\ k = 1,\ldots,T$. The SUR approach to obtain binary controls is described in Algorithm 3.1.

Algorithm 3.1: Sum-Up Rounding with the SOS1 Property.

```
Input: Continuous control $u^c$ on uniform discretization.
/* Iterate over each time step */
for $k = 1,\ldots,T$ do
    /* Iterate over each controller */
    for $j = 1,\ldots,N$ do
        Compute cumulative deviation as $\hat{p}_{jk} = \sum_{\tau=1}^{k} u^c_{j\tau}\Delta t - \sum_{\tau=1}^{k-1} u^b_{j\tau}\Delta t$.
    end
    Choose controller $j^* = \arg\max_{j=1,\ldots,N} \hat{p}_{jk}$. If there is a tie, we break the tie by
      choosing the smallest index.
    Set binary control $u^b_{j^*k} = 1$ and $u^b_{jk} = 0,\ \forall j \neq j^*$.
end
Output: Binary control $u^b$.
```

The construction of u^b_{jk} ensures that the binary control satisfies the SOS1 property. Sager et al. [181] proved that if the continuous control u^c satisfies the SOS1 property, then the rounded control u^b will converge to u^c when the length of a time interval Δt converges to zero. This does not hold for arbitrary continuous controls without the SOS1 property, however, as the next proposition shows.

Proposition 3.3.1. *Let continuous control u^c and binary control u^b defined in Algorithm 3.1 be given for $N \geq 2$. Define*

$$\epsilon(\Delta t) = \max_{k=1,\ldots,T} \left| \sum_{\tau=1}^{k} \left(\sum_{j=1}^{N} u^c_{j\tau} - 1 \right) \Delta t \right|. \tag{3.19}$$

If $t_f < \infty$, then u^b satisfies the SOS1 property $\sum_{j=1}^{N} u^b_{jk} = 1$ at any time step $k = 1,\ldots,T$, and we have

$$\max_{k=1,\ldots,T} \left\| \sum_{\tau=1}^{k} \left(u^c_\tau - u^b_\tau \right) \Delta t \right\|_\infty \geq \frac{1}{N}\epsilon(\Delta t). \tag{3.20}$$

The proof is provided in Appendix B.1.2. We have the following estimate of the difference between the continuous control without the SOS1 property and the rounded control.

Theorem 3.3.3. *With the same definition of $\epsilon(\Delta t)$ in Proposition 3.3.1, it holds that for any time step $k = 1, \ldots, T$,*

$$\left\| \sum_{\tau=1}^{k} \left(u_\tau^c - u_\tau^b \right) \Delta t \right\|_\infty \leq (N-1)\, \Delta t + \frac{2N-1}{N} \epsilon(\Delta t). \tag{3.21}$$

Because u^b and u^c are bounded, we have $\epsilon(\Delta t) < \infty$ if $t_f < \infty$. The proof of the inequality (3.21) extends the proof for Theorem 5 in [181]. We modify the right-hand side of the inequality by adding a term $\frac{2N-1}{N}\epsilon(\Delta t)$. We provide the detailed proof in Appendix B.1.2. In the following corollary we show that for our bounded discretized quantum control problem, $\epsilon(\Delta t)$ converges to zero as Δt converges to zero, ensuring the convergence of the rounded control to the continuous control.

Corollary 3.3.1. *Let $l(u^c, T)$ be the value of the optimized squared L_2 term in the continuous relaxation of the model* (DQCP-L_2)*. Then it holds that*

$$\epsilon(\Delta t) \leq \sqrt{t_f l(u^c, T) \Delta t}, \tag{3.22}$$

where t_f is the evolution time. Furthermore, if the original objective function F is bounded, the rounded control will converge to the continuous control when Δt is small enough.

The proof is provided in Appendix B.1.2. Based on the convergence of binary results to continuous results, we present the following proposition to guarantee the optimality of binary results.

Proposition 3.3.2. *Under the discretized setting of the quantum control problem, let u^c be the continuous control and u^b be the binary control obtained by Algorithm 3.1. Then the state of time evolution with binary control X^b converges to the state with continuous control X^c at each time step, namely,*

$$\lim_{\Delta t \to 0} X_k^b = \lim_{\Delta t \to 0} X_k^c, \;\; k = 1, \ldots, T, \tag{3.23}$$

leading to $\lim_{\Delta t \to 0} F(X_T^b) = \lim_{\Delta t \to 0} F(X_T^c)$ for a continuous objective function.

The proof is provided in Appendix B.1.2.

3.3.3.2 Combinatorial Integral Approximation

To obtain discretized binary controls, Sager [180] proposed a more general rounding technique by minimizing the integral difference between continuous and binary controls with certain additional constraints on the binary controls called combinatorial integral approximation (CIA). Let $\mathcal{U}_B \subseteq \{0,1\}^{N \cdot T}$ be the feasible region for binary controls. For each controller $j = 1, \ldots, N$ and each time step $k = 1, \ldots, T$, define u^c_{jk} and u^b_{jk} as the discretized continuous and binary controls. We use u^c and u^b to represent the corresponding vector forms. The optimization problem for rounding is formulated as a mixed-integer problem

$$\min_{u^b} \quad \max_{j=1,\ldots,N} \max_{k=1,\ldots,T} \left| \sum_{\tau=1}^{k} (u^c_{j\tau} - u^b_{j\tau}) \Delta t \right| \tag{3.24a}$$

$$\text{s.t.} \quad u^b \in \mathcal{U}_B. \tag{3.24b}$$

The rounding optimization problem can be solved by diverse algorithms for solving integer programs. In this chapter we choose a branch-and-cut algorithm [228]. Furthermore, the rounding result obtained from SUR is an optimal solution of model (3.24) with $\mathcal{U}_B = \{0,1\}^{N \cdot T}$ [180]. In practice, researchers and engineers have proposed diverse restrictions on binary controls to avoid frequent switches. We formulate two main types of constraints as linear constraints.

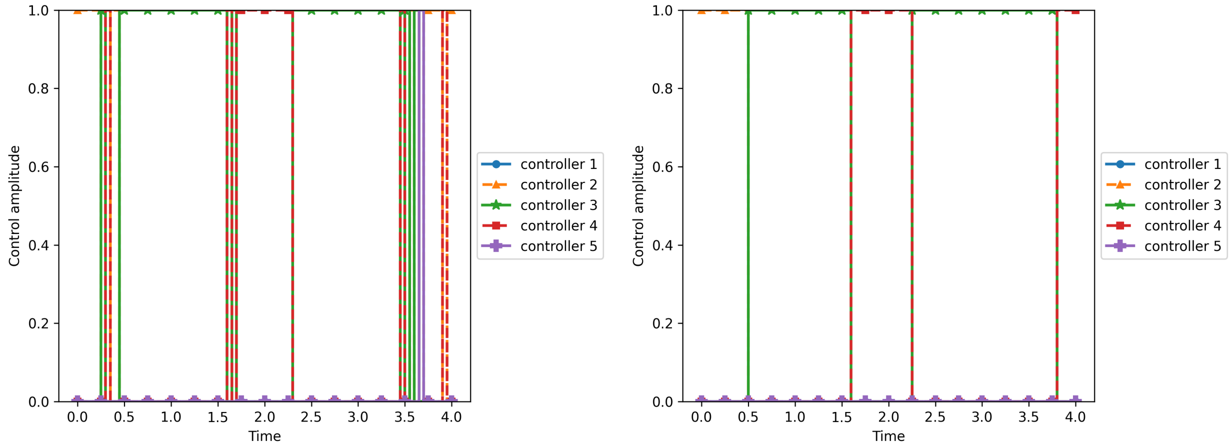
Min-Up-Time Constraints. Let T_{minup} be the minimum number of steps Δt that the controller is active. The min-up-time constraints enforce that each controller is active for at least T_{minup} time steps. The restricted feasible region is formulated by the following linear constraints:

$$\begin{aligned} \mathcal{U}_B = \Bigg\{ (u,v) : & - v_{jk} \le u_{jk} - u_{jk+1} \le v_{jk}, \ j = 1, \ldots, N, \ k = 1, \ldots, T-1 \\ & \sum_{k=t}^{t+T_{\text{minup}}-1} v_{jk} \le 1, \ j = 1, \ldots, N, \ t = 1, \ldots, T - T_{\text{minup}} \Bigg\}. \end{aligned} \tag{3.25}$$

Max-Switching Constraints. Let S be the maximum number of switches during the evolution time horizon $[0, t_f]$. The max-switching constraints enforce an upper bound S on the total number of switches for each controller. We describe the restricted feasible region by linear constraints as

$$\begin{aligned} \mathcal{U}_B = \Bigg\{ (u,v) : & - v_{jk} \le u_{jk} - u_{jk+1} \le v_{jk}, \ j = 1, \ldots, N, \ k = 1, \ldots, T-1 \\ & \sum_{k=1}^{T} v_{jk} \le S, \ j = 1, \ldots, N \Bigg\}. \end{aligned} \tag{3.26}$$

We present the binary results obtained from SUR and CIA with min-up-time constraints in Figure 3.4. We show that SUR leads to chattering on singular arcs. Although CIA can prevent frequent switches by setting hard constraints on the controls, it leads to a serious objective value increase. In Section 3.4 and Section 3.5 we propose models and algorithms to reduce switches when taking the original objective function into account.



(a) SUR results (objective value 0.021)

(b) CIA results with min-up time constraints (objective value 0.600)

Figure 3.4: Binary control results obtained by conducting SUR and CIA on continuous results of the circuit compilation example using the molecule H_2. The min-up time constraints reduce the switches but increase the objective value.

3.4 Model with the TV Regularizer

Binary controls that avoid chattering on singular arcs can be obtained by solving the rounding model (3.24). However, the objective function of the rounding model is not relevant to the original objective function F, leading to a significant increase in the value of F, as illustrated in Figure 3.4. Here we investigate a TV regularizer as an alternative strategy to obtain solutions with fewer switches.

The TV of a function is defined as the integral of the absolute change of the function over the entire space. Since Rudin et al. [178] first introduced the TV regularizer by considering functions describing images, it has become a popular method in image noise reduction; see, for example, [44, 107]. We refer interested readers to [176] for a detailed review of applying a TV regularizer to denoise images. Stella et al. [199] propose an algorithm to solve the continuous control problem with the TV regularizer based on the forward-backward envelope [206]. Sager and Zeile [183] considered the TV of integer control functions to reduce the absolute change of controls. The authors added a constraint imposing that the TV regularizer should be no more than a threshold. However, they considered the TV regularizer only when rounding the control results of continuous relaxation, which leads to an increase in the original objective function. Leyffer and Manns [116] extended the trust-region method to solve the integer optimal control problem with the TV regularizer in the objective function, but it is still highly dependent on the initial values.

We use the TV regularizer to penalize the absolute change between control variables of two consecutive time steps, which is defined as

$$TV(u) = \sum_{j=1}^{N} \sum_{k=1}^{T-1} |u_{jk} - u_{jk+1}| \tag{3.27}$$

for a control variable vector u. Let $\alpha > 0$ be the parameter for the TV regularizer. Then the continuous relaxation model with the TV regularizer is formulated as

$$\min\ F(X_T) + \rho l(u, T) + \alpha TV(u) \quad \text{s.t. Constraints (3.3b)–(3.3d)},\ u \in [0,1]^{N \cdot T}. \tag{DQCP-TV}$$

The first term is the original objective function, the second term is the squared L_2-penalty function for the SOS1 property, and the third term is the TV regularizer function. Because of the L_1 term, we cannot use GRAPE; instead, we propose ADMM to solve the continuous relaxation. For each controller $j = 1, \dots, N$ and time step $k = 1, \dots, T-1$, we introduce auxiliary variables $v_{jk} = u_{jk} - u_{jk+1}$ to describe the change between control variables of two consecutive time steps. We reformulate the model as follows:

$$\begin{aligned}
\min \quad & F(X_T) + \rho l(u, T) + \alpha \sum_{j=1}^{N} \sum_{k=1}^{T-1} |v_{jk}| && \text{(3.28a)} \\
\text{s.t.} \quad & v_{jk} = u_{jk} - u_{jk+1},\ j = 1, \dots, N,\ k = 1, \dots, T-1 && \text{(3.28b)} \\
& \text{Constraints (3.3b)–(3.3d)} && \\
& u \in [0,1]^{N \cdot T}. && \text{(3.28c)}
\end{aligned}$$

Define $\mu_{jk},\ j = 1, \dots, N,\ k = 1, \dots, T-1$ as the dual variables corresponding to constraints (3.28b). We use $u,\ v,\ \mu$ to denote the corresponding vector forms. Let fixed parameters $\beta > 0, \delta > 0$ be the Lagrangian penalty parameter and stopping criterion threshold. The update procedure of ADMM consists of three steps: (i) updating variables u, (ii) updating variables v, and (iii) updating dual variables μ. We solve the minimization problem for updating variables u by the modified GRAPE algorithm with a squared L_2-penalty function and Lagrangian penalty function. We derive an exact form for the update of variables v. We use gradient descent to update the dual variables μ. The specific procedure for updating is presented in Algorithm 3.2.

Algorithm 3.2: ADMM Algorithm for Solving Continuous Relaxation of (DQCP-TV).

```
Input: Initial values of variables $u^0,\ v^0,\ \mu^0$, and number of ADMM iterations $L$.
Initialize iteration $l = 1$.
while $l \leq L$ and the algorithm does not converge do
    Update variables $u$ as
        $$u^l = \underset{u \in \mathcal{U}_C}{\arg\min}\ F(X_T) + \rho l(u, T) + \frac{\beta}{2} \sum_{j=1}^{N} \sum_{k=1}^{T-1} \|u_{jk} - u_{jk+1} - v_{jk}^{l-1} + \mu_{jk}^{l-1}\|^2$$
            s.t. Constraints (3.3b)–(3.3d).
    for $k = 1, \ldots, T$ do
        for $j = 1, \ldots, N$ do
            Update variables $v_{jk}$ as
                $$v_{jk}^l = \begin{cases} u_{jk}^l - u_{jk+1}^l + \mu_{jk}^{l-1} - \alpha/\beta, & \text{if } u_{jk}^l - u_{jk+1}^l + \mu_{jk}^{l-1} > \alpha/\beta, \\ u_{jk}^l - u_{jk+1}^l + \mu_{jk}^{l-1} + \alpha/\beta, & \text{if } u_{jk}^l - u_{jk+1}^l + \mu_{jk}^{l-1} < -\alpha/\beta, \\ 0, & \text{otherwise.} \end{cases}$$
            Update dual variables $\mu_{jk}$ as $\mu_{jk}^l = \mu_{jk}^{l-1} + (u_{jk}^l - u_{jk+1}^l - v_{jk}^l)$.
        end
    end
    if $\sum_{j=1}^{N} \sum_{k=1}^{T-1} \|u_{jk}^l - u_{jk+1}^l - v_{jk}^l\|^2 \leq \delta$ then
        Break the loop.
    end
    Update $l \leftarrow l + 1$.
end
Output: Final solutions of variables $u^l,\ v^l,\ \mu^l$.
```

We present the continuous control results obtained from the model (DQCP-TV) in Figure 3.5 for the circuit compilation example on the molecule H_2. We test the parameter of the TV regularizer $\alpha = 10^{-5},\ 10^{-4},\ 10^{-3},\ 10^{-2},\ 10^{-1}$ and the Lagrangian parameter $\beta = 0.1,\ 0.5,\ 1.0$. We choose $\alpha = 10^{-3}$ and $\beta = 0.5$, which has the smallest objective value after rounding with min-up-time constraints. The number of switches in control results decreases significantly as compared with the results in Figure 3.3, showing the benefits of the TV regularizer.

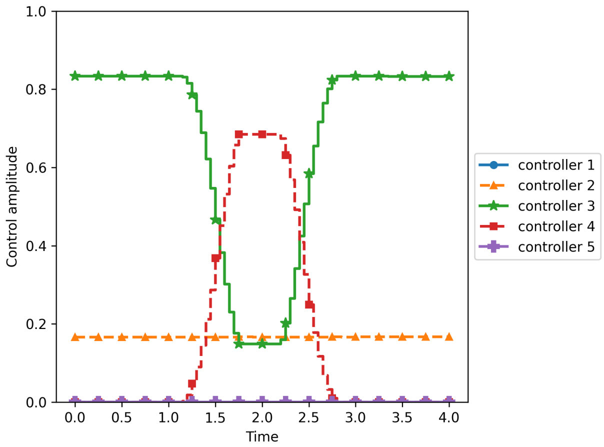


Figure 3.5: Control results of the continuous relaxation of the discretized model with the TV regularizer (DQCP-TV) of the circuit compilation example on the molecule H_2 (objective value: 1.33E−05).

3.5 Improvement Heuristic: Approximate Local-Branching Method

Because the ADMM algorithm does not ensure global optimality for nonlinear objective functions and the objective value increases after rounding, the control results can be improved by applying an approximate local-branching (ALB) heuristic. The trust-region method [153] is a widely used local search method in optimization but highly depends on the initial values. In this section we propose a trust-region subproblem for the quantum control problem and then modify the trust-region method to solve the problem starting from the solutions obtained in Sections 3.3–3.4 to improve the quantum controls.

We introduce our improvement heuristic based on the approach of [116] to solve the binary model with the TV regularizer. Given a feasible point $\hat{u}$, we use the first-order gradient to approximate the objective value around the point $\hat{u}$ and define the trust-region subproblem with radius R as

$$\begin{aligned} \min_{u} \quad & \langle \nabla F(\hat{u}), u - \hat{u} \rangle_{L_2} + \alpha TV(u) - \alpha TV(\hat{u}) && \text{(3.29a)} \\ \text{s.t.} \quad & \|u - \hat{u}\|_1 \leq R && \text{(3.29b)} \\ & \text{Constraints (3.3e)–(3.3f).} \end{aligned}$$

We note that (3.29a) approximates the objective value $F(u)$ but uses an exact form of the TV regularizer. Constraint (3.29b) indicates that we consider only the points with L_1 distance to $\hat{u}$ no more than R. For each controller $j = 1, \ldots, N$ and time step $k = 1, \ldots, T-1$, define variables v_{jk} as the upper bound of the absolute change between the control values of two consecutive time steps. The trust-region subproblem is reformulated as the following mixed-integer linear program:

$$\min_{u} \quad \langle F(\hat{u}), u - \hat{u} \rangle_{L_2} + \alpha \sum_{k=1}^{T-1} \sum_{j=1}^{N} v_{jk} - \alpha TV(\hat{u}) \tag{3.30a}$$

$$\text{s.t.} \quad \sum_{k=1}^{T} \left(\sum_{j:\hat{u}_{jk}=0} u_{jk} + \sum_{j:\hat{u}_{jk}=1} (1 - u_{jk}) \right) \leq R \tag{3.30b}$$

$$- v_{jk} \leq u_{jk} - u_{jk+1} \leq v_{jk}, \; j = 1, \ldots, N, \; k = 1, \ldots, T-1 \tag{3.30c}$$

Constraints (3.3e)–(3.3f).

The objective function (3.30a) is a reformulation of (3.29a) with variables v. Constraint (3.30b) is the trust-region constraint. Constraints (3.30c) ensure that for each controller $j = 1, \ldots, N$, $v_{jk} \geq |u_{jk} - u_{jk+1}|$, $k = 1, \ldots, T-1$. Let $\bar{u}$ be an optimal solution of the model (3.30). Define $\Delta F_p(\hat{u}, \bar{u})$ and $\Delta F_a(\hat{u}, \bar{u})$ respectively as the predictive and actual decrease of the objective function with the following formulations:

$$\Delta F_p(\hat{u}, \bar{u}) = \langle \nabla F(\hat{u}), \hat{u} - \bar{u} \rangle_{L_2} + \alpha TV(\hat{u}_{jk}) - \alpha TV(\bar{u}_{jk}), \tag{3.31a}$$

$$\Delta F_a(\hat{u}, \bar{u}) = F(\hat{u}) - F(\bar{u}) + \alpha TV(\hat{u}_{jk}) - \alpha TV(\bar{u}_{jk}). \tag{3.31b}$$

The trust-region algorithm consists of an inner loop and an outer loop. In the inner loop we solve a sequence of trust-region subproblems with monotonically decreasing radius until the ratio between the actual decrease and predictive decrease is large enough. To obtain a balance between the computational cost and the searched area, we set a threshold $\bar{R}$ for the radius. When the radius is greater than the threshold, we decrease it according to a geometric sequence [116]; otherwise, we decrease it by an arithmetic sequence. In the outer loop we repeat the inner loop for each updated point until the predictive decrease is nonpositive. This procedure is described in Algorithm 3.3.

Remark 3.5.1. *If we relax the feasible region of each controller to* $[0, 1]$*, the trust-region method can solve the continuous relaxation of the model with the TV regularizer. Trust-region methods show convergence to stationary points on continuous relaxation if we allow the radius R to be real and adjusted accordingly (Corollary 3.9 in [9]).*

The trust-region approach can also improve solutions obtained by CIA approaches from Section 3.3.3.2. Based on the restricted feasible region $\mathcal{U}_B$ instead of the TV regularizer, we

```
Algorithm 3.3: Trust-Region Method for Quantum Control.
Input: Starting radius $R^0 > 0$, threshold of radius $\bar{R} > 0$, initial feasible point $u^0$,
       and threshold of decrease $\eta > 0$.
Initialize predictive decrease $\Delta F_p(u^0, \bar{u}) = \infty$ and actual decrease $\Delta F_a(u^0, \bar{u}) = -\infty$.
Set the number of outer iterations $l \leftarrow 0$.
while $\Delta F_p(u^l, \bar{u}) > 0$ do
    Initialize radius $R = R^0$.
    while $\Delta F_a(u^l, \bar{u}) < \eta \Delta F_p(u^l, \bar{u})$ do
        Solve model (3.30) with $\hat{u} = u^l$ to obtain the minimizer $\bar{u}$.
        Compute predictive and actual decrease $\Delta F_p(u^l, \bar{u})$ and $\Delta F_a(u^l, \bar{u})$ by (3.31).
        if $R > \bar{R}$ then
            $R \leftarrow \max\left\{\lfloor R/2 \rfloor, \bar{R}\right\}$.
        else
            $R \leftarrow R - 1$.
        end
    end
    Set $l \leftarrow l + 1$. Update the central point $u^l \leftarrow \bar{u}$.
end
Output: Control results $u^l$.
```

propose the trust-region subproblem with additional linear constraints as follows:

$$\min_{u} \quad \langle \nabla F(\hat{u}), u - \hat{u} \rangle_{L_2} \tag{3.32a}$$

$$\text{s.t.} \quad \text{Constraints (3.3e)–(3.3f), (3.30b)}$$

$$(u, v) \in \mathcal{U}_B. \tag{3.32b}$$

The objective function (3.32a) is a first-order gradient approximation of the original objective function $F(u)$, and constraint (3.32b) restricts feasible regions for controls. The computations of the predictive decrease and actual decrease are modified as

$$\Delta F_p(\hat{u}, \bar{u}) = \langle \nabla F(\hat{u}), \hat{u} - \bar{u} \rangle_{L_2}, \tag{3.33a}$$

$$\Delta F_a(\hat{u}, \bar{u}) = F(\hat{u}) - F(\bar{u}). \tag{3.33b}$$

For the trust-region algorithm (Algorithm 3.3), we replace solving model (3.30) with solving model (3.32) and compute the decrease by (3.33). The other parts remain the same.

In theory, all time-evolution processes in our algorithms can be conducted on quantum computers. The inputs for the quantum computers are the control sequences, and we require the output of the objective function and its (approximate) gradient. The updates of variables are still computed on classical computers.

3.6 Numerical Results

We apply our algorithmic framework proposed in Sections 3.3–3.5 to diverse instances of the four examples introduced in Section 3.2. In Section 3.6.1 we show the results from state-of-the-art optimization solvers as the baseline. In Section 3.6.2 we introduce the design of numerical instances and parameter settings. In Sections 3.6.3–3.6.5 we present the numerical results, including the continuous relaxation results, binary results by combinatorial integral approximation, and binary results after an improvement heuristic.

3.6.1 State-of-the-Art Optimization Solvers

The NEOS server is a frequently used internet-based service containing several state-of-the-art solvers for numerical optimization problems [48, 53, 80]. For nonlinear constrained continuous problems, SNOPT [8] and IPOPT [219] are two widely used solvers. For mixed-integer nonlinear constrained problems, BARON [184] and Couenne [16] are commonly used for obtaining global optimal solutions, while MINLP [17, 115], SCIP [70], and Bonmin [24] are effective solvers with good performance for finding local optima.

As a baseline we first apply the optimization solvers from the NEOS server to solve the energy minimization example problem in Section 3.2.1. All the experiments are conducted on the online server. We set the number of qubits $q = 2$ and $q = 6$. We set the evolution time $t_f = 2$ and the number of time steps $T = 40$. We notice that for all the evolution times in Section 3.6, we use dimensionless units, with $\hbar = 1$. We apply the solver MUSCOD-II (6.0) [98] to solve the continuous formulation with the differential equations (P), and we set the maximum iterations to 100 for $q = 2$ and 700 for $q = 6$. To apply the state-of-the-art nonlinear optimization solvers to a discretized model, we derive a discretized formulation of the ordinary differential equation by the implicit Euler method (see, e.g., [33]) as

$$X_k - X_{k-1} = -iH_k X_k \Delta t, \ k = 1, \dots, T. \tag{3.34}$$

We apply the solvers SNOPT (7.6.1) and IPOPT (3.13.4) to solve the continuous relaxation model and apply the solvers MINLP, Bonmin (1.8.8), Couenne (0.5.8), and SCIP (7.0.3.5) to solve the binary model. We set the time limit to 5 minutes when $q = 2$ and 80 minutes when $q = 6$. In Table 3.2 we present the objective values, TV-norm values, CPU times, and explored nodes (only for binary solvers).

Table 3.2 shows that MUSCOD-II is good for solving the continuous relaxation for the instance with $q = 2$, but it takes a long time to solve the large instance with $q = 6$. SNOPT and IPOPT perform worse than the MUSCOD-II solver especially when the quantum systems include more qubits. For the binary control problem, all the methods reach the time limit. Bonmin obtains the best solution, but the gaps between their results and true energy are all large. For the instance with $q = 6$, some optimization solvers such as MINLP and Couenne run out of memory. The CPU time increases and the number of explored nodes decreases significantly when the size of the quantum system increases.

Table 3.2: Results of solvers on energy minimization example. The results are marked by "OOM" if a solver runs out of memory and "LIMIT" if a solver reaches the time or iteration limit. The explored nodes of continuous solvers are marked by "-" because the node exploration process is conducted only by binary solvers.

Solver	$q = 2$			$q = 6$		
	Obj	Time (s)	Nodes	Obj	Time (s)	Nodes
SNOPT	0.084	0.18	-	0.736	199.64	-
IPOPT	0.084	0.10	-	0.736	46.85	-
MUSCOD-II	5.90E−09	1.67	-	0.154	4399.54	-
MINLP	0.150	LIMIT	9056	OOM	OOM	OOM
Bonmin	0.148	LIMIT	11672	0.793	LIMIT	355
Couenne	0.471	LIMIT	6932	OOM	OOM	OOM
SCIP	0.149	LIMIT	25679	1.000	LIMIT	2480

This experiment shows that existing standard nonlinear programming and mixed-integer nonlinear programming solvers cannot solve discretized optimal control problems. In the following sections we present the numerical results of our proposed algorithms on multiple instances. Our pGRAPE+SUR method obtains binary solutions with an objective value 4.22E−04 in 0.14 seconds for $q = 2$ and binary solutions with an objective value 0.157 in 27.95 seconds for $q = 6$. We show that our methods can obtain better results and with shorter computational time than do the state-of-art solvers.

3.6.2 Experimental Design and Parameter Settings

When the problems have no SOS1 property, the squared L_2 term in the penalized model (DQCP-L_2) is eliminated, so solving (DQCP-L_2) by pGRAPE is equivalent to solving the original model (DQCP) by GRAPE. Because the energy minimization problem has only two controllers, we eliminate the SOS1 property directly by substituting $u_2(t) = 1 - u_1(t)$. We remove the SOS1 property in the CNOT gate estimation problem to show the generality of our models. We solve the model with a squared L_2-penalty function for the circuit compilation problem.

For all the examples, we apply the pGRAPE algorithm to solve the continuous relaxation. We also employ the TR and ADMM algorithms to solve the continuous relaxation with the

TV regularizer. Then we obtain the binary results without constraints by SUR, the binary results with min-up-time constraints, and the binary results with max-switching constraints by integral minimization. Furthermore, we apply the approximate local-branching methods to improve the binary solutions with the TV regularizer and hard control constraints.

For the energy minimization problem, when $q = 2$, the matrix J is two-dimensional and includes only one independent element because it is symmetric. Hence, we set diagonals in J to 0 and other elements to 1. When $q = 4,\ 6$, we generate 5 instances where each instance has a random symmetric matrix J with zero diagonals and elements within a range $[-1, 1]$. We will present averaged objective value results for these problems. The instances are represented by Energy2, Energy4, and Energy6 corresponding to the number of qubits $q = 2,\ 4,\ 6$ in our following results. For the CNOT problem, we conduct experiments with evolution times $t_f = 5,\ 10,\ 15,\ 20$ and represent the corresponding instances as CNOT5, CNOT10, CNOT15, CNOT20, respectively. For the NOT gate estimation problem, we follow the parameter settings in [145] and set $\mu_1 = 0,\ \mu_2 = 2\pi,\ \omega_1 = 1,\ \omega_2 = \sqrt{2}$. We conduct numerical simulations with evolution times $t_f = 2,\ 6,\ 10$ and represent the instances as NOT2, NOT6, and NOT10, respectively. For the circuit compilation problem, we test two molecules, H_2 (dihydrogen) and LiH (lithium hydride), and generate the UCCSD circuits with minimum energy by the VQE algorithm in the Python package Qiskit [5]. The instances are represented by CircuitH2 and CircuitLiH. We set the parameters corresponding to quantum systems as $J_c = 0.2\pi,\ J_f = 3\pi,\ J = 0.1\pi$. We test values of the penalty parameter for the squared L_2-penalty function $\rho = 10^{-6},\ 10^{-5},\ 10^{-4},\ 10^{-3},\ 10^{-2},\ 10^{-1},\ 1,\ 10$, and we choose the one with the smallest objective value among the parameters with a squared penalized term less than 10^{-6}. We set the penalty parameters to be $\rho = 1.0$ and $\rho = 0.1$ for two instances, respectively. We test values for the TV parameter $\alpha = 10^{-5},\ 10^{-4},\ 10^{-3},\ 10^{-2}$ and choose α with the smallest rounding objective value with min-up-time constraints. The settings of α and other parameters are presented in Table 3.3. All the numerical simulations were conducted in Python 3.8 on a MacOS computer with 8 cores, 16 GB of RAM, and a 3.20 GHz processor. All the computational time results are the times on classical computers.

In Sections 3.6.3–3.6.5, for brevity we select four different instances, Energy6, CNOT20, NOT10, and CircuitLiH, to analyze the results of objective values and CPU time. Detailed results of all the methods and instances are presented in Tables B.1, B.2, B.3, B.4, B.5, and B.6 in Appendix B.2. Our full code and results are available on our Github repository [61].

Table 3.3: Parameter settings of examples. The parameters include the number of qubits (q), number of controllers (N), evolution time (t_f), number of time steps (T), TV parameter (α), minimum uptime steps (T_{minup}), and maximum switches (S).

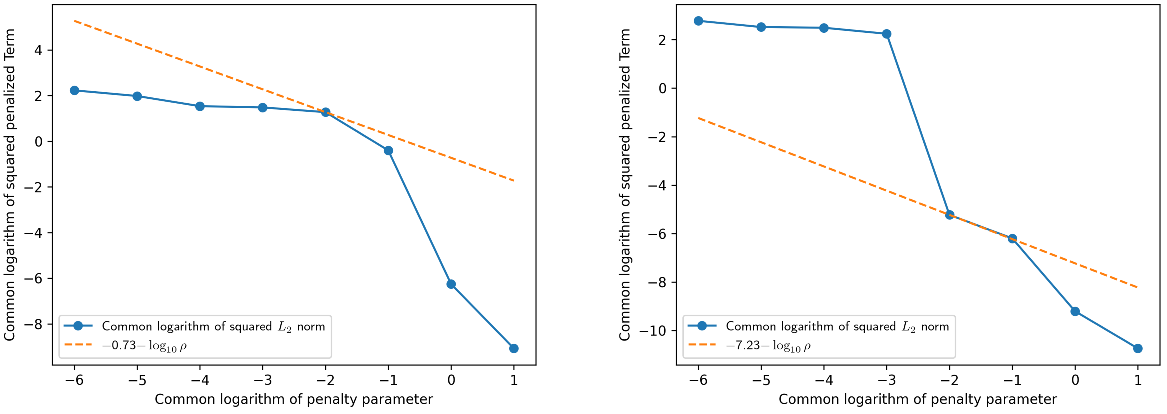
Instance	q	N	t_f	T	α	T_{minup}	S
Energy2	2	2	2	40	0.01	10	5
Energy4	4	2	2	40	0.01	10	5
Energy6	6	2	2	40	0.01	10	5
CNOT5	2	2	5	100	0.01	10	20
CNOT10	2	2	10	200	0.001	10	20
CNOT15	2	2	15	300	0.0001	10	20
CNOT20	2	2	20	400	0.0001	10	20
NOT2	1	2	2	20	0.001	5	4
NOT6	1	2	6	60	0.001	5	12
NOT10	1	2	10	100	0.001	5	20
CircuitH2	2	5	4	80	0.001	10	8
CircuitLiH	4	12	20	200	0.001	5	40

3.6.3 Results of Continuous Relaxations

In Figure 3.6 we show how the common logarithm of the squared L_2-norm value varies with the penalty parameter represented by the lines. We also present an asymptotic bound $\log_{10} c/\rho$ represented by the dashed lines, where c is a constant marking that a selected data point lies on the dashed lines in the simulation results. Notably, the figure confirms that $l(u_\rho^*, T) \sim O(1/\rho)$, as shown in Theorem 3.3.2.

We show that pGRAPE always obtains the lowest objective value but the highest TV regularizer values because it solves the model without the TV regularizer. ADMM has the best performance for reducing the TV regularizer values, showing the benefits of introducing a TV regularizer in Section 3.4. The detailed results are presented in Appendix B.2.

We measure the problem size of each instance by $2^q \times N \times T$, where q, N, and T represent the number of qubits, number of controllers, and number of time steps. For four groups of instances, Energy, CNOT, NOT, and Circuit, we present in Figure 3.7 common logarithm-logarithm figures to show how the CPU time and the number of iterations vary with the problem size. The CPU time increases significantly with an increase in the number of qubits because the dimension of simulation Hamiltonian matrices increases exponentially. The CPU time also increases when the number of controllers and time steps increases. Compared with pGRAPE, TR and ADMM spend more time solving the model because it contains the TV regularizer. The CPU time of ADMM is more than that of TR for Energy instances, while TR spends more time on CNOT, NOT, and Circuit instances than ADMM spends.



(a) Instance CircuitH2

(b) Instance CircuitLiH

Figure 3.6: Common logarithm of squared L_2-norm varying with $\log_{10}\rho$, where ρ is the squared L_2 penalty parameter. Blue lines are the common logarithm of the squared L_2-norm. Orange dashed lines are functions $\log_{10} c - \log_{10}\rho$, where c is a constant. The figures show that the squared L_2-norm $l(u_\rho^*, T)$ decreases as the penalty parameter ρ increases with a rate $O(1/\rho)$.

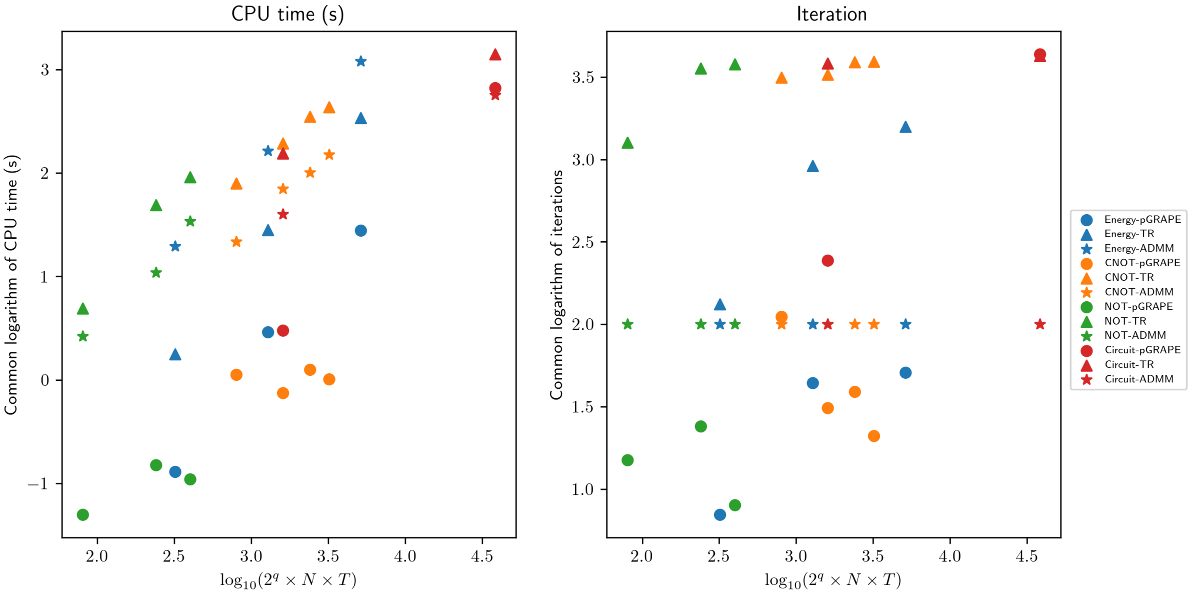


Figure 3.7: Common logarithm-logarithm figure of CPU time and iterations of continuous relaxation for all the instances and number of time steps. Blue, orange, green, and red colors represent instances Energy, CNOT, NOT, and Circuit. Circles, triangles, and stars represent obtaining continuous results by pGRAPE, TR, and ADMM, respectively. Dots with increasing size and transparency represent results by MT and MS, respectively.

3.6.4 Results of Rounding Techniques

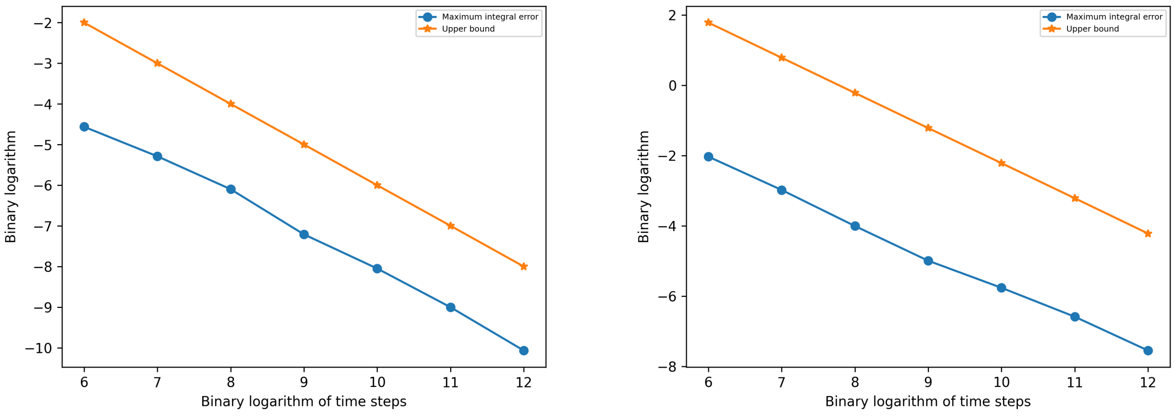
In Figure 3.12 we present the objective values and TV regularizer values of the binary results obtained by SUR as well as CIA with min-up-time constraints and max-switching constraints of selected instances. Compared with pGRAPE+MT/MS, TR+MT/MS or ADMM+MT/MS obtains lower objective values and TV regularizer values for the binary results with min-up-time constraints and max-switching constraints because they solve the model with the TV regularizer.

From the results of pGRAPE we show that the squared L_2-penalty function ensures that the difference between binary controls of SUR and continuous results is small. To further demonstrate the convergence of pGRAPE+SUR in Section 3.3.3.1, we present how the maximum absolute integral error, which is the left-hand side of equation (3.21), varies with the number of time steps in Figure 3.8 represented by the lines. The figures show that the error converges to zero with the increase in the number of time steps, as is claimed in Proposition 3.3.2. We also present the theoretical upper bound of the integral error, which is the right-hand side of (3.21), represented by the dashed lines. We observe that the integral error is always less than the upper bound, demonstrating the conclusion in Theorem 3.3.3. In addition, we present the figures of how $\epsilon(\Delta t)$ and the upper bound $\sqrt{t_f l(u^c, T)\Delta t}$ vary with the number of time steps for instances CircuitH2 and CircuitLiH in Figure 3.9. We show that $\epsilon(\Delta t)$ is always smaller than the upper bound, demonstrating the conclusion (3.22) in Corollary 3.3.1.

We set the time limit to 60 seconds for solving the combinatorial integral approximation with min-up-time and max-switching constraints. In Figure 3.10 we present the CPU time and iterations of all the instances. All the CPU times of SUR and Energy instances are less than 0.01 seconds so we do not show them in the figure. The CPU times increase significantly with the number of variables. For the instance CircuitLiH, all the methods exceed the computational time limit. In most cases, iterations and CPU times of MS are more than MT because the problem with MS has a larger integrality gap.

3.6.5 Results of Improvement Heuristic

In Figure 3.12 we present the objective values and TV regularizer values after the improvement heuristic with the control results of combinatorial integral approximation as initial points for selected instances. Because there are multiple local minima for these quantum control problems, each method may obtain a different performance after the improvement heuristic. For systems with two controllers, all the methods obtain performance at a similar level. The performance of different methods varies more on the circuit compilation problem



(a) Instance CicruitH2

(b) Instance CircuitLiH

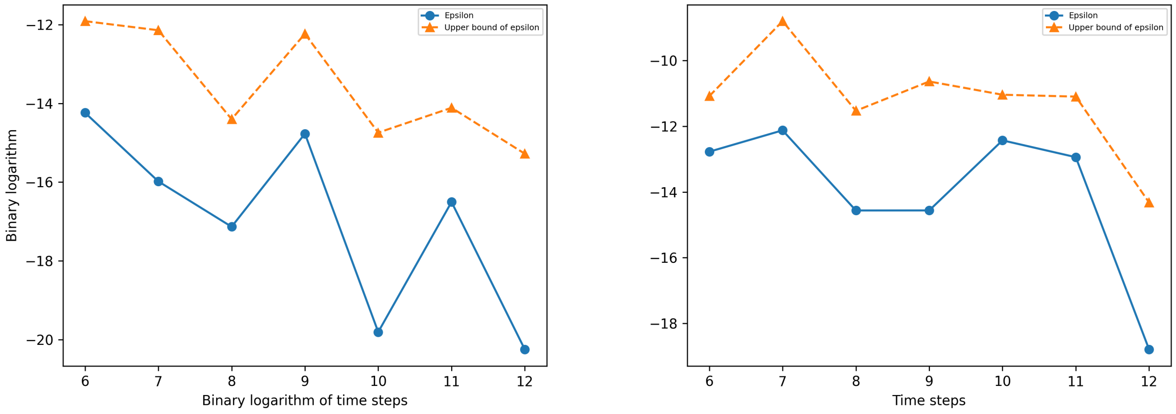
Figure 3.8: Binary logarithm of maximum absolute integral error and upper bound of continuous results and SUR binary results. Blue lines marked by circles represent the maximum absolute integral error, and orange lines marked by stars represent the upper bound. The error is always smaller than the upper bound and converges to zero as the number of time steps increases, demonstrating the conclusion (3.21).

because it contains more local optima.

We show in Figure 3.11 the CPU time and iterations for the improvement heuristic of selected instances. For each group of instances Energy, CNOT, NOT, and Circuit, the CPU time increases with the problem size. The number of iterations highly depends on the initial points.

In addition, we observe that for both continuous relaxation and ALB, the CPU time is exponentially increasing but the number of iterations increases slightly. We note that the main increase in the computational time comes from the time-evolution process for simulating the quantum system because the dimension of the Hamiltonian matrices increases exponentially with the number of qubits q. In our algorithm, we could directly conduct the time evolution on quantum computers with a control pulse, and then only use the final state to compute the objective value and approximate the gradient by the finite difference method. Hence, we do not need to evaluate the state at each time step, indicating the scalability of our approach for handling large-scale instances.

In Figure 3.12 we show the objective values and TV regularizer values of the combinatorial integral approximation and the improvement heuristic. We show that the objective value of SUR keeps the same or slightly increases, demonstrating the theorems and propositions in Section 3.3.3.1. Adding min-up-time and max-switching constraints leads to an increase in objective values because of adding additional constraints stated in Section 3.3.3.2 to



(a) Instance CicruitH2

(b) Instance CircuitLiH

Figure 3.9: Binary logarithm of the maximum SOS1 difference of continuous control $\epsilon(\Delta t)$ and its upper bound in (3.22). Blue lines represent $\epsilon(\Delta t)$ and orange dashed lines represent the upper bound. We show that $\epsilon(\Delta t)$ converges to 0 with the increase in time steps and is always smaller than the upper bound, demonstrating Corollary 3.3.1.

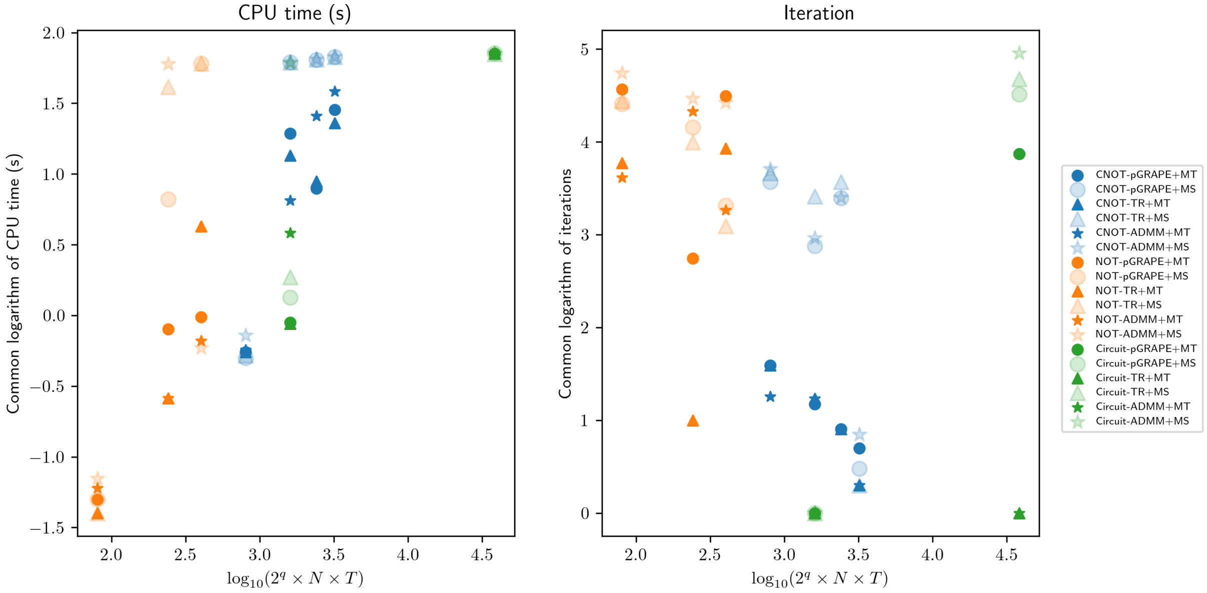


Figure 3.10: Common logarithm-logarithm figure of CPU time and iterations of CIA for all the instances. Blue, orange, and green colors represent instances of CNOT, NOT, and Circuit, respectively. Circles, triangles, and stars represent obtaining continuous results by pGRAPE, TR, and ADMM, respectively. Dots with increasing size and transparency represent results by SUR, MT, and MS, respectively.

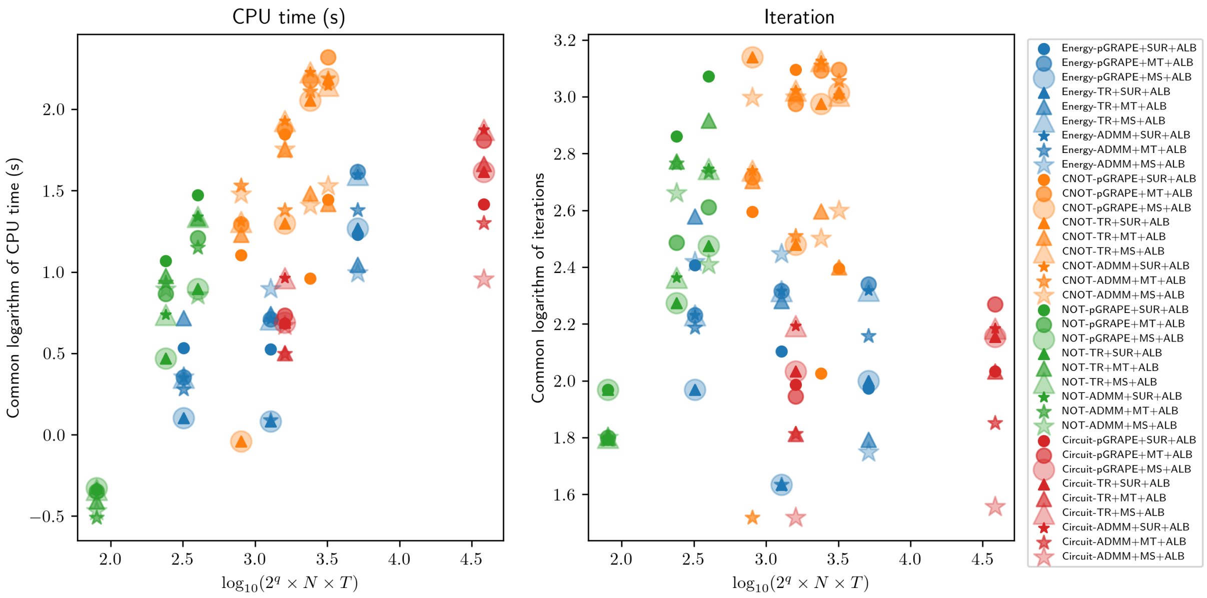


Figure 3.11: Common logarithm-logarithm figure of CPU time and iterations of ALB for all the instances. Blue, orange, green, and red colors represent groups of instances Energy, CNOT, NOT, and Circuit.

Circles, triangles, and stars represent obtaining continuous results by pGRAPE, TR, and ADMM, respectively. Dots with increasing size and transparency represent results by SUR, MT, and MS, respectively.

the feasible set. The increase from adding min-up-time constraints is higher because the restricted feasible region contains fewer feasible solutions. From the aspect of the chattering measured by the TV regularizer value, we show that SUR leads to more chattering compared with continuous solutions because of the rounding process. As discussed in Section 3.3.3.2, imposing min-up-time and max-switching constraints reduces the chattering, and adding min-up-time constraints reduces it more significantly because the constraints are stricter. The improvement heuristic reduces the objective values and chattering remarkably for the results obtained from all the methods with rounding techniques, especially pGRAPE, which shows the benefits of our improvement algorithms in Section 3.5. In Figure 3.12, the dot closest to the lower-left corner indicates that it obtains the best balance between improving objective values and reducing chattering. If the original objective function matters more, one can choose methods that consist of SUR and the improvement heuristic. If one focuses more on reducing the switches, the methods with min-up-time or max-switching constraints and the improvement heuristic are better.

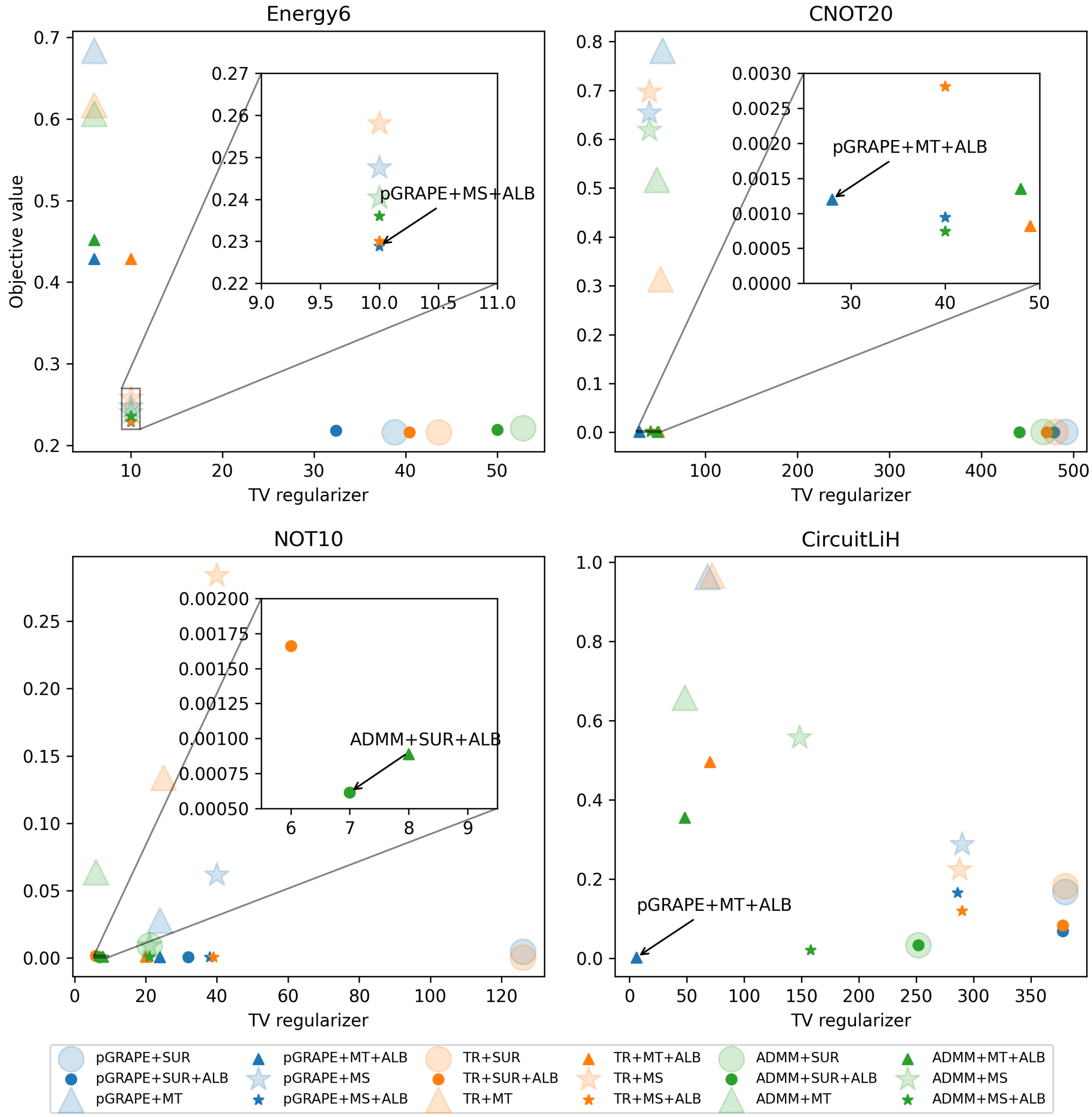


Figure 3.12: Objective values and TV regularizer values of binary results of selected instances. Blue, orange, and green dots represent solving continuous relaxation by pGRAPE, TR, and ADMM, respectively. Circles, triangles, and stars represent obtaining binary results by SUR, MT, and MS, respectively. Big dots represent results before ALB, and opaque small dots represent results after ALB. We annotate the method represented by the closest point to the lower-left corner.

3.7 Concluding Remarks

In this chapter, we built a generic control model with both continuous and discretized formulations for the quantum control problem. We introduced a penalized squared L_2 function into the model to ensure that only one control is active at all times. In addition, we proposed a model with the TV regularizer aiming to reduce the absolute change of controls.

We developed an algorithmic framework combining the GRAPE approach, combinatorial integral approximation, and local-branching improvement heuristic. With numerical simulations on multiple examples with various objective functions and controllers, we demonstrated the feasibility of the discrete optimal quantum control problem and illustrated that our algorithms obtained trade-off controls between high quality and low absolute changes within a short computational time. Specifically, the performance of different relaxation models varies among instances, and therefore, testing all three methods (pGRAPE, TR, and ADMM) is helpful in selecting the best control. If one is more interested in obtaining controls with lower energy or infidelity, we recommend methods using the sum-up-rounding technique. On the other hand, if one aims to reduce switches, it is better to choose methods with min-up-time constraints added. Both cases require the use of the approximate local branching algorithm to improve final solutions. In practice, the performance can be improved further by fine-tuning the corresponding parameters.

All the numerical simulations in our chapter were conducted on classical computers, and the CPU time mainly comes from the time-evolution process of quantum states. Because our algorithm only requires the final state to compute the objective value and approximate the gradient, running the time-evolution process in our algorithms on quantum computers is easy in practice and could help eliminate a significant amount of the exponential slowdown as we observe in our simulations. Furthermore, with continuous control results, we can extract appropriate controller sequences and optimize the location of switching points to obtain high-quality controls. Switching-time optimization using continuous control results is an important future research direction.

CHAPTER 4

Switching Time Optimization for Binary Quantum Optimal Control

4.1 Introductory Remarks

We use methods from classical computer design and engineering to accelerate the development of practical and scalable quantum computing systems. Quantum control theory has a rich history that parallels the development of controllable quantum devices. Notions of quantum control theory [51, 74, 225] first appeared in more analog settings, such as quantum chemistry [100, 162]. As the field evolves, it saw applications in quantum information through gate design [145, 160] and quantum circuit compilation [75], and later has been used in more high-level design of quantum algorithms, especially with the advent of variational quantum circuits [13, 26, 28, 64, 72, 108, 128, 138, 214, 233].

To better implement quantum control, unlike most of the literature that solves quantum control problems using a fixed time discretization, we optimize both control functions and the time between consecutive control switches. Our new algorithmic framework—based on classical computer and optimization models and algorithms—improves the quality of quantum controls and reduces computational time.

Here we generalize the quantum control problem in a continuous time horizon (P) studied in Chapter 3. With the same definition of variables and parameters, and newly defined feasible set $\mathcal{U} \subseteq \{0,1\}^N$ of $u(t)$, we have the following formulation:

$$
\begin{aligned}
(P) \quad \min_{u,X,H} \quad & F(X(t_f)) \\
\text{s.t.} \quad & H(t) = H^{(0)} + \sum_{j=1}^{N} u_j(t) H^{(j)},\ \forall t \in [0, t_f] \\
& \frac{d}{dt} X(t) = -iH(t)X(t),\ \forall t \in [0, t_f]
\end{aligned}
$$

$$\begin{aligned} & X(0) = X_{\text{init}} \\ & u(t) \in \mathcal{U},\ \textit{a.e. } t \in [0, t_f]. \end{aligned} \tag{4.1a}$$

The only different constraint (4.1a) indicates that for $t \in [0, t_f]$ almost everywhere, the feasible region set $\mathcal{U}$ constrains the value of the control function $u(t)$. In this chapter, we consider two cases, one is binary controls $\mathcal{U} = \{0, 1\}^N$, and the other one is binary controls with the additional SOS1 property constraint $\mathcal{U} = \{u \in \{0, 1\}^N : \sum_{j=1}^N u_j = 1\}$ discussed in Chapter 3.

Following the previous chapter, we employ two widely used objective functions in the quantum control field. We use $|\cdot\rangle$ to represent a quantum state vector and $\langle\cdot|$ to represent its conjugate transpose, and $\cdot^\dagger$ to represent the conjugate transpose of a complex matrix. One function is the energy ratio minimization function with the specific form

$$F(X(t_f)) = 1 - \langle\psi_0|X^\dagger(t_f)\tilde{H}X(t_f)|\psi_0\rangle/E_{\min}, \tag{4.2}$$

to minimize the energy corresponding to the Hamiltonian $\tilde{H}$. In the objective function, the initial state of the quantum system is given by $|\psi_0\rangle$ and the constant minimum energy $E_{\min}$ represents the minimum eigenvalue of the Hamiltonian $\tilde{H}$. An alternative objective function is the infidelity function

$$F(X(t_f)) = 1 - \frac{1}{2^q}\left|\mathbf{tr}\left\{X_{\text{targ}}^\dagger X(t_f)\right\}\right|, \tag{4.3}$$

to minimize the difference between $X(t_f)$ and the target operator X_{targ}. Both objective functions are bounded between $[0, 1]$.

In the literature, the most widely used method to solve the original quantum control problem (P) is time discretization, which divides the evolution time horizon into time intervals. We follow the time discretization in Chapter 3 to build the discretized model as follows. The only difference is that we optimize both variables u, X, H and time interval endpoints $t_1, \ldots, t_T$.

$$(P_D) \quad \min_{u,X,H,t_1,\ldots,t_T} \quad F(X_T) \tag{4.4a}$$

$$\begin{aligned} \text{s.t.} \quad & H_k = H^{(0)} + \sum_{j=1}^N u_{jk}H^{(j)},\ k = 1, \ldots, T \\ & X_k = e^{-iH_k(t_k - t_{k-1})}X_{k-1},\ k = 1, \ldots, T \\ & X_0 = X_{\text{init}} \end{aligned}$$

$$u_k \in \mathcal{U},\ k = 1, \ldots, T. \tag{4.4b}$$

The simultaneous optimization of both control functions u, X, H, and time discretization points $t_1, \ldots, t_T$ was proposed by Logsdon and Biegler [133] for differential equation systems. Due to the computational complexity, we divide our problem into two parts: (i) optimizing discretized control variables u_k for each time step $k = 1, \ldots, T$ and (ii) optimizing time points of switching between controllers. Next, we review the main literature focusing on the problem of each part.

Literature of Quantum Optimal Control. Most quantum optimal control algorithms only consider optimizing controls under a *fixed* time discretization. For continuous quantum control problems, Khaneja et al. [95] first proposed the gradient ascent pulse engineering (GRAPE) algorithm to estimate control functions with piece-wise constant functions and apply gradient-based methods to optimize them. Furthermore, Larocca and Wisniacki [109] combined the GRAPE algorithm and Krylov-subspace approach to improve computational efficiency. Brady et al. [26] developed an analytical framework based on time discretization for optimizing bang-bang and smooth annealing controls for a quantum energy minimization problem. Other algorithms based on time discretization include pseudospectral methods [119] and reinforcement learning frameworks [32, 152, 194]. For binary quantum control problems, Vogt and Petersson [217] applied a trust-region method [153] to optimize controls in a single flux quantum system. Fei et al. [64] developed a solution framework combining the GRAPE algorithm, rounding techniques, and local branching heuristics. However, obtaining controls with higher quality by the discretized model requires more precise time discretization, equivalent to a larger number of decision variables associated with each time step, which increases the computational cost severely. Estimating control functions by piece-wise functions also leads to limitations in obtaining better control solutions.

Literature of Switching Time Optimization. The switching time optimization has been a meaningful yet challenging topic in controls of switched-mode dynamic systems [57, 68, 91, 139, 200]. Vossen [218] discussed the switching time optimization for both bang-bang and singular controls. In quantum theory, a class of literature formulates the quantum approximate optimization algorithm (QAOA) as an optimal control problem with two controllers and applies multiple methods to solve it [11, 59, 124], which can be considered as a simplified version of switching time optimization with only two switching controllers. The aforementioned literature assumes known analytical formulations of control functions. Bukov et al. [32] designed simple variational control protocols from discretized control solutions obtained by reinforcement learning which requires numerous training episodes and lacks generality for various control problems.

Main Contributions. The main contributions of this chapter are as follows. First, we develop a new algorithmic framework that not only optimize control functions but also switching time points. Specifically, we develop two heuristic methods to obtain binary discretized controls from continuous discretized controls balancing the quality of controls and the number of switches for general quantum control problems. Second, we build and solve a generic switching time optimization model for quantum control problems with given Hamiltonian controllers as parameters. In addition, we accelerate the time-evolution simulations by pre-computing the eigenvalues of a small number of Hamiltonian matrices. Third, we conduct numerical simulations on multiple quantum control examples and show that our method obtains controls with higher quality and a similar number of switches within significantly less computational time compared to the discretized model.

The remainder of the chapter is organized as follows. In Section 4.2, we construct our algorithmic framework and develop heuristic methods to obtain controller sequences. In Section 4.3, we formulate and solve the switching time optimization model with acceleration techniques for time evolution. In Section 4.4, we conduct numerical simulations and discuss the results. In Section 4.5, we conclude our work and propose future research directions.

4.2 Algorithm Framework and Controller Sequence Extraction

Our algorithmic framework consists of four steps, which we describe in detail in Algorithm 4.1. First, we solve the continuous relaxation of the discretized model (DQCP) with a fixed (equal) time discretization and time interval length Δt by the penalized GRAPE algorithm proposed in the paper by Fei et al. [64]. Second, we obtain binary controls from solutions of continuous relaxation by rounding algorithms, such as algorithms in [64, Table 1], Algorithm 4.3, and Algorithm 4.4. Third, we merge consecutive time intervals with the same control values as a time interval to derive a sequence of controllers by Algorithm 4.2. Fourth, we optimize the switching time points of merged time intervals with the sequence of controllers as parameters, which is discussed in detail in Section 4.3. We present the flow chart for our overall algorithm in Figure 4.1.

We present control results of a simple example in Figure 4.2. In the left figure, we present the piece-wise continuous control function u^{con} obtained after solving the discretized

Algorithm 4.1: Switching time optimization method for binary quantum control problem.

```
  Input: Discretized binary quantum control model (DQCP).
1 Solve the continuous relaxation of (DQCP) with a fixed time discretization and
    obtain continuous solutions $u^{\text{con}}$.
2 Obtain heuristic binary solutions $u^{\text{bin}}$ from continuous solutions $u^{\text{con}}$ by a given
    rounding algorithm, such as algorithms in [64, Table 1], Algorithms 4.3, and 4.4.
3 Merge time intervals with the same control value and derive the controller sequence
    $\bar{\mathcal{H}}$ with length $S$ by Algorithm 4.2.
4 Solve switching time optimization model (STO) in Section 4.3 to optimize switching
    time points with the given controller sequence $\bar{\mathcal{H}}$.
  Output: Control time interval lengths for each controller in the controller sequence
          $\bar{\mathcal{H}}$.
```

Algorithm 4.2: Controller sequence extraction from binary controls.

```
Input: Discretized binary controls $u^{\text{bin}}$.
Initialize controller sequence $\bar{\mathcal{H}}_1 = \left[H^{(0)} + \sum_{j=1}^{N} u_{j1} H^{(j)}\right]$ and sequence length
  $S \leftarrow 1$.
for $k = 2, \dots, T$ do
    if $u_k \neq u_{k-1}$ then
        $S \leftarrow S + 1,\ \bar{H}_S = H^{(0)} + \sum_{j=1}^{N} u_{jk} H^{(j)}$.
        $\bar{\mathcal{H}}_S \leftarrow \left[\bar{\mathcal{H}}_{S-1}, \bar{H}_S\right]$.
    end
end
Output: Controller sequence $\bar{\mathcal{H}}_S$ and length $S$.
```

continuous relaxation (step 1). In the middle figure, we present the binary control obtained after rounding methods (step 2). We convert the control functions from fractional values to binary values. In the right figure, we present the optimized control after optimizing the time of switches obtained by step 4.

Various methods to obtain binary controls have been proposed but they either require high computational costs or frequent switches [64, 135, 181, 182, 236]. In this section, we present two heuristic rounding methods to obtain controller sequences from given continuous discretized controls u^{con}. Our key idea is to balance the difference between continuous and binary controls and the number of switches. For any control u, we evaluate the frequency of switching by the total variational (TV) norm defined as cumulative absolute differences

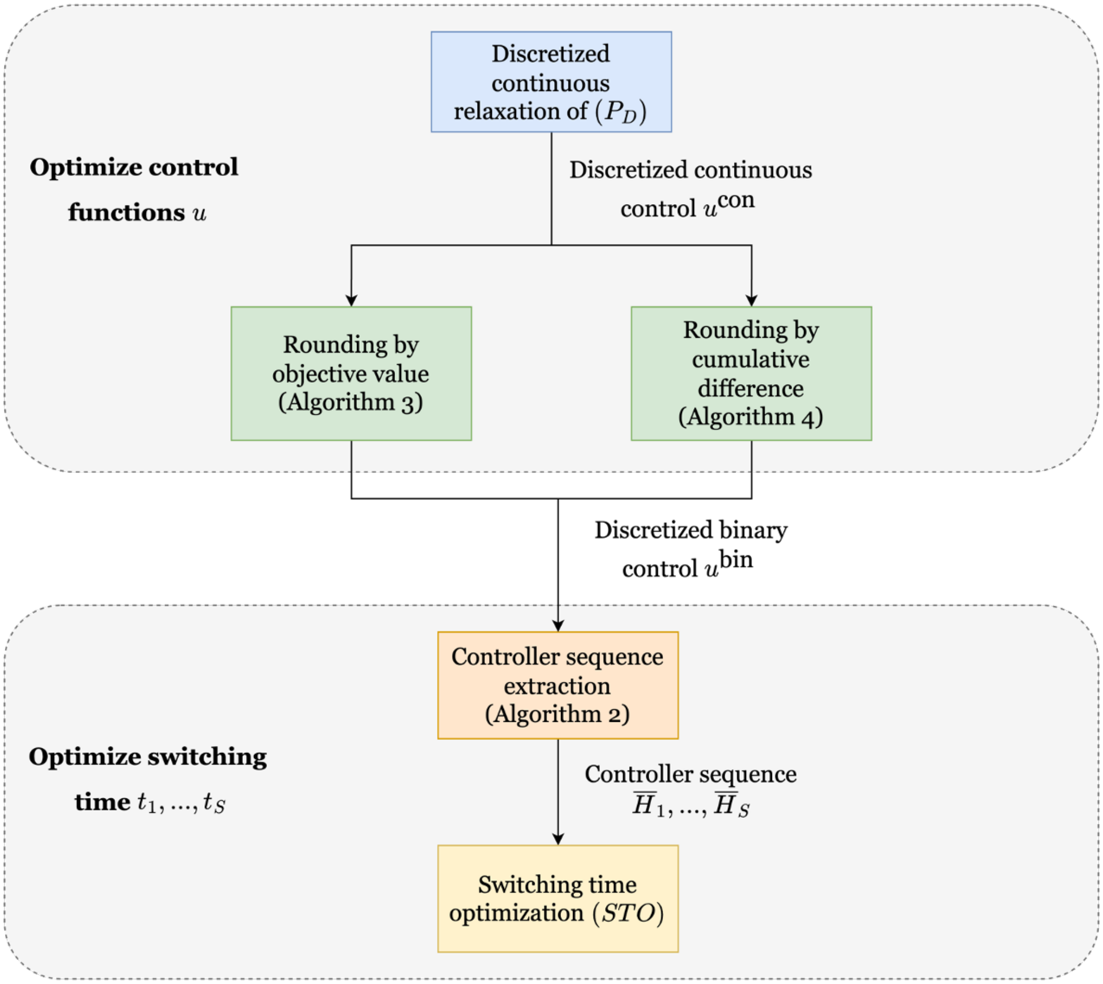

Discretized continuous relaxation of (P_D)
Optimize control functions u
Discretized continuous control u^{con}
Rounding by objective value (Algorithm 3)
Rounding by cumulative difference (Algorithm 4)
Discretized binary control u^{bin}
Controller sequence extraction (Algorithm 2)
Optimize switching time $t_1, ..., t_S$
Controller sequence $\overline{H}_1, ..., \overline{H}_S$
Switching time optimization (STO)

Figure 4.1: Flow chart for the overall algorithmic framework (Algorithm 4.1). We first optimize control functions u, including solving the discretized continuous relaxation of model (DQCP) and applying rounding algorithms (Algorithm 4.3–4.4). Then we optimize the switching times by solving the switching time optimization model (STO) with controller sequences extracted from binary controls (Algorithm 4.2) as input.

between consecutive steps with the formulation:

$$TV(u) = \sum_{k=1}^{T-1} \sum_{j=1}^{N} |u_{jk} - u_{jk+1}|. \tag{4.5}$$

We evaluate the difference based on two metrics and propose the corresponding methods in Sections 4.2.1 and 4.2.2, respectively. In each section, we also prove that for both methods, if we only consider minimizing the difference between continuous and binary controls, when the length of time intervals goes to zero, the limit of the objective value of binary controls is

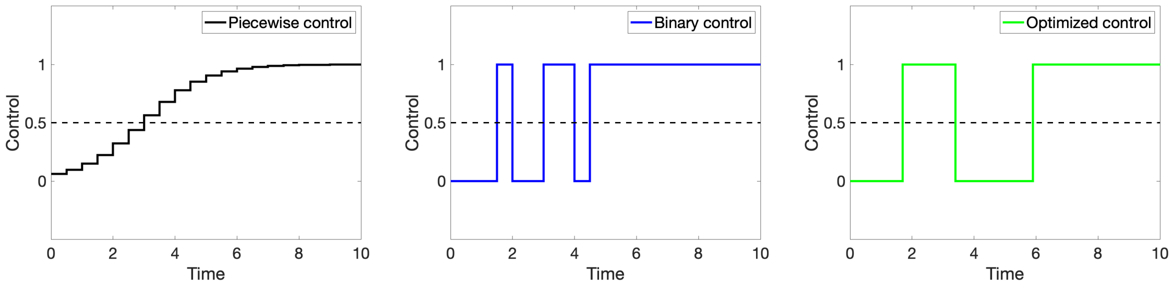


Figure 4.2: Simple example for the overall algorithm framework (Algorithm 4.1). Left: Piece-wise continuous control u^{con} after step 1. Middle: Binary control u^{bin} after step 2. Right: Optimized control after step 4.

no more than the limit of the objective value of the continuous control under the objective functions (4.2) and (4.3) considered in this chapter.

4.2.1 Method Based on Objective Value

We convert unitary operator variables X and Hamiltonian variables H to implicit functions of control variables u by constraints (3.3b)–(3.3d). The final operator X_T can also be computed as a function of u as

$$X_T(u) = \prod_{k=1}^{T} e^{-i\left(H^{(0)}+\sum_{j=1}^{N} u_{jk} H^{(j)}\right)\Delta t} X_{\text{init}}, \tag{4.6}$$

which is equivalent to solving the Schrödinger equation (3.3b)–(3.3d). Substituting $X_T(u)$ into the objective function F, we derive an objective function only dependent on u, denoted as $\bar{F}(u)$ by eliminating intermediate variables H_k and X_k, $k = 1, \dots, T$.

In this section, we propose Algorithm 4.3 based on evaluating the difference between binary and continuous solutions by the difference of objective values $\bar{F}(u^{\text{bin}})$ and $\bar{F}(u^{\text{con}})$. We define α as the TV norm penalty parameter. We present the explanation as follows. At each time step k, we examine the current control $[u_1^{\text{bin}}, \dots, u_{k-1}^{\text{bin}}, \hat{u}, u_{k+1}^{\text{con}}, \dots, u_T^{\text{con}}] \in [0,1]^{N\times T}$ for all $\hat{u} \in \mathcal{U}$ and choose the control with smallest objective value $\hat{u}^*$ (see step 5–6). We compare the choices of keeping the control values of the previous time step, i.e. $\hat{u} = u_{k-1}^{\text{bin}}$, and updating control values, i.e. $\hat{u} = \hat{u}^*$ (see step 7). If the objective value of keeping the control is no larger than the TV-norm value of updating the control weighted by α, we keep the control (see step 8). Otherwise, we choose the control with the smallest objective value and update the TV-norm value (see step 10). If $\alpha = 0$, we always choose the control with the smallest objective value. Furthermore, we update the binary control u_k^{bin} by the selected control and move to the next time step $k+1$ (see step 12). This approach requires $T-1$ simulations at a receding time horizon.

The computational advantages of conducting time-evolution processes on quantum computers allow computing objective values quickly. For simulations on classical computers, we propose an acceleration technique to avoid conducting time evolution for every examined control. With continuous solutions u^{con}, we define back propagators μ_k as

$$\mu_k = \mu_{k+1} e^{-i\left(H^{(0)}+\sum_{j=1}^{N} u_{jk}^{\text{con}} H^{(j)}\right)\Delta t},\ \ k=1,\ldots,T, \tag{4.7}$$

where μ_{T+1} is an identity matrix. At each time step, after determining the binary control, the operator corresponding to binary controls is updated as

$$X_k^{\text{bin}} = e^{-i\left(H^{(0)}+\sum_{j=1}^{N} u_{jk}^{\text{bin}} H^{(j)}\right)\Delta t} X_{k-1}^{\text{bin}}, \tag{4.8}$$

where $X_0^{\text{bin}} = X_{\text{init}}$. We have the following proposition to evaluate the objective value.

Proposition 4.2.1. *At time step $k=1,\ldots,T$, the objective value of a current examined control is computed by*

$$\bar{F}\left([u_1^{bin},\ldots,u_{k-1}^{bin},\hat{u},u_{k+1}^{con},\ldots,u_T^{con}]\right) = F\left(\mu_{k+1} e^{-i\left(H^{(0)}+\sum_{j=1}^{N}\hat{u}_j H^{(j)}\right)\Delta t} X_{k-1}^{bin}\right),\ \forall \hat{u}\in\mathcal{U}. \tag{4.9}$$

Proof. For any examined control, the final operator $\hat{X}_T$ is computed by

$$\hat{X}_T = \prod_{t=k+1}^{T} e^{-i\left(H^{(0)}+\sum_{j=1}^{N} u_{jt}^{\text{con}} H^{(j)}\right)\Delta t} X_k = \mu_{k+1} X_k = \mu_{k+1} e^{-i\left(H^{(0)}+\sum_{j=1}^{N}\hat{u}_j H^{(j)}\right)\Delta t} X_{k-1}^{\text{bin}}, \tag{4.10}$$

where the last two equalities directly follow the definition μ_k and X_k^{bin} in (4.7)–(4.8). From the definition of $\bar{F}$, we have

$$\bar{F}\left([u_1^{\text{bin}},\ldots,u_{k-1}^{\text{bin}},\hat{u},u_{k+1}^{\text{con}},\ldots,u_T^{\text{con}}]\right) = F(\hat{X}_T) = F\left(\mu_{k+1} e^{-i\left(H^{(0)}+\sum_{j=1}^{N}\hat{u}_j H^{(j)}\right)\Delta t} X_{k-1}^{\text{bin}}\right). \tag{4.11}$$

□

From Proposition 4.2.1, we show that the pre-computing and the reformulation of evaluating the objective value significantly reduce the matrix operation times, which we discuss in detail in the following proposition.

Proposition 4.2.2. *Let $N'=\mathcal{U}$ be the number of feasible examined controls at each time* *step. With Algorithm 4.3, we reduce the computation of matrix exponentials from $O(N'T^2)$ to $O(N'+T)$ and the computation of matrix multiplications from $O(N'T^2)$ to $O(N'T)$.*

Proof. Without our reformulation of the objective value and pre-computation, we require computing $O(T)$ matrix exponentials and $O(T)$ matrix multiplications during each time evolution process. We need to evaluate all the examined controls at each time step, which means that we need to do $N'T$ time evolution processes, hence the total number is $O(N'T^2)$ for both matrix exponentials and multiplications. With our acceleration, we only need to compute T unitary propagators of the continuous control and the matrix exponential of each possible control in $\mathcal{U}$. Hence, the order of computing matrix exponentials is $O(N'+T)$. We need to perform T matrix multiplications for computing $\mu_k,\ k=1,\dots,T$. At each time step, we need to compute the objective value of setting each controller active, which requires total $O(N')$ matrix multiplications, and then update the current operator X_k^{bin}, which only requires 1 multiplication. Therefore, taking a summation of all the time steps, the order of matrix multiplications is $O(N'T)$. □

We present the details of the algorithm in Algorithm 4.3. For any bounded objective function with upper bound F_{ub}, the TV-norm value with penalty parameter α is upper-bounded by F_{ub}/α. In the following theorems, we discuss the difference between the objective values of binary controls u^{bin} and input continuous controls u^{con} when setting the penalty

Algorithm 4.3: Heuristic Rounding Methods based on Objective Values (Obj)

```
Input: Continuous control $u^{\text{con}}$ and constant TV norm parameter $\alpha$.
Initialize binary control $u^{\text{bin}} = u^{\text{con}}$ and current objective value $F_{\text{cur}} = \bar{F}(u^{\text{con}})$.
Pre-compute back propagators $\mu_k,\ k=1,\dots,T$ by Eqn (4.7).
Pre-compute matrix exponentials $e^{-i\left(H^{(0)}+\sum_{j=1}^{N}\hat{u}_j H^{(j)}\right)}$ for all $\hat{u}\in\mathcal{U}$.
Let $u_1^{\text{bin}} = \arg\min_{\hat{u}\in\mathcal{U}} \bar{F}\left([\hat{u}, u_2^{\text{con}},\dots,u_T^{\text{con}}]\right)$.
for $k=2,\dots,T$ do
    Let $\hat{u}^* = \arg\min_{\hat{u}\in\mathcal{U}} \bar{F}\left([u_1^{\text{bin}},\dots,u_{k-1}^{\text{bin}},\hat{u},u_{k+1}^{\text{con}},\dots,u_T^{\text{con}}]\right)$.
    if $\bar{F}\left([u_1^{bin},\dots,u_{k-1}^{bin},u_{k-1}^{bin},u_{k+1}^{con},\dots,u_T^{con}]\right) \le$
      $\alpha TV([u_1^{bin},\dots,u_{k-1}^{bin},\hat{u}^*,u_{k+1}^{con},\dots,u_T^{con}])$ then
        Update binary control $u_k^{\text{bin}} = u_{k-1}^{\text{bin}}$.
    else
        Update binary control $u_k^{\text{bin}} = \hat{u}^*$, break ties with smaller TV value.
    end
    Update $X_k^{\text{bin}}$ by Eqn. (4.8) and objective value
      $F_{\text{cur}} = \bar{F}\left([u_1^{\text{bin}},\dots,u_k^{\text{bin}},u_{k+1}^{\text{con}},\dots,u_T^{\text{con}}]\right)$.
end
Output: Binary control $u^{\text{bin}}$.
```

parameter $\alpha = 0$. All the conclusions in this section are proved for both the energy objective function (4.2) and the infidelity objective function (4.3). Before presenting our main results, we review a series of lemmas corresponding to unitary matrices and singular values used in our proof. The proof of Lemma 4.2.1–4.2.3 is presented by Horn and Johnson [87].

Lemma 4.2.1. *([87, Section 3.1]) The singular values are invariant under the multiplication of unitary matrices.*

Lemma 4.2.2. *([87, Theorem 3.3.4, 3.3.16]) Let $A,\ B \in \mathbb{C}^{m\times m}$ be two complex matrices and let $\sigma_1(\cdot)$ represent the maximum singular value of a matrix, then $\sigma_1(AB) \leq \sigma_1(A)\sigma_1(B),\ \sigma_1(A+B) \leq \sigma_1(A) + \sigma_1(B)$.*

Lemma 4.2.3. *([87, Theorem 3.1.2]) Let A be an arbitrary complex matrix, then for any unit vector $|\psi\rangle$ such that $\| |\psi\rangle \|_2 = 1$, it holds that $|\langle\psi|A|\psi\rangle| \leq \sigma_1(A)$.*

Lemma 4.2.4. *Let $U \in \mathbb{C}^{m\times m}$ be a unitary matrix and $A \in \mathbb{C}^{m\times m}$ be an arbitrary complex matrix, then $|\mathbf{tr}\{UA\}| \leq \sum_{i=1}^{m} \sigma_i(A) \leq m\sigma_1(A)$, where $\sigma_1(A) \geq \ldots \geq \sigma_m(A) \geq 0$ are the singular values of A.*

The proof of Lemma 4.2.4 is presented in Appendix C.1. In the following theorem, we prove that if u^{con} has the SOS1 property, for the energy objective function and infidelity objective function discussed in Section 4.1, the difference of objective values $\bar{F}(u^{\text{bin}}) - \bar{F}(u^{\text{con}})$ is bounded by $O(\Delta t)$.

Theorem 4.2.1. *Let u^{con} be the solution of the continuous relaxation of* (DQCP) *with a fixed time discretization and u^{bin} be the binary controls obtained by Algorithm 4.3 when $\alpha = 0$. If the SOS1 property holds for continuous controls, which means that $\sum_{j=1}^{N} u^{con}_{jk} = 1,\ k = 1, \ldots, T$, and is required for binary controls, then there exists constants $C_1,\ C_2 > 0$ such that for energy objective function* (4.2) *and infidelity objective function* (4.3)*, the difference between the objective value of binary and continuous control satisfies*

$$\bar{F}(u^{bin}) - \bar{F}(u^{con}) \leq 2C_1 e^{C_2\Delta t}\Delta t. \tag{4.12}$$

Furthermore, we have

$$\limsup_{\Delta t\to 0} \bar{F}(u^{bin}) - \bar{F}(u^{con}) \leq 0. \tag{4.13}$$

The detailed proof is presented in Appendix C.1. Next, we discuss a more general case when the SOS1 property of the continuous control u^{con} does not exactly hold but the SOS1 property of the binary control u^{bin} is still required. We define the cumulative error between binary and continuous control as

$$\epsilon^c(\Delta t) = \sum_{k=1}^{T} \left| \sum_{j=1}^{N} u_{jk}^{\mathrm{con}} - 1 \right| \Delta t = \sum_{k=1}^{t_f/\Delta t} \left| \sum_{j=1}^{N} u_{jk}^{\mathrm{con}} - 1 \right| \Delta t. \tag{4.14}$$

In the following theorem, we present the estimation of the difference in objective values between continuous and binary controls.

Theorem 4.2.2. *Let u^{con} be the solution of the continuous relaxation of model* (DQCP) *without the SOS1 property under fixed time discretization. Let u^{bin} be the binary control with the enforced SOS1 property obtained by Algorithm 4.3 with $\alpha = 0$. There exist constants $C_0,\ C_1,\ C_2$ such that the difference between objective values of continuous and binary controls satisfies*

$$\bar{F}(u^{bin}) - \bar{F}(u^{con}) \leq 2C_1 e^{C_2 \Delta t} \Delta t + C_0 \epsilon^c(\Delta t). \tag{4.15}$$

We provide detailed proof in Appendix C.1. We consider a uniform discretized model with L_2 penalty function for the SOS1 property with the following formulation [64]:

$$\min_{u \in [0,1]^{N \times T}, X, H} \quad \bar{F}(u) + \rho \sum_{k=1}^{T} \left(\sum_{j=1}^{N} u_{jk} - 1 \right)^2, \tag{P-L_2}$$

where ρ is the penalty parameter. We assume that the original problem (P) has a feasible solution with the SOS1 property. In the following corollary, we show that if we solve the model with L_2 penalty function (P-L_2), then the convergence result in Theorem 4.2.1 still holds.

Corollary 4.2.1. *Assume that the original problem* (P) *has a feasible solution with the SOS1 property. For any Δt, let the discretized continuous controls u^{con} be the solution of model* (P-L_2) *and u^{bin} be the binary solutions obtained by Algorithm 4.3 with $\alpha = 0$, then we have*

$$\limsup_{\Delta t \to 0} \bar{F}(u^{bin}) - \bar{F}(u^{con}) \leq 0. \tag{4.16}$$

Proof. It is obvious that all the constants $C_0,\ C_1,\ C_2$ are bounded. We only need to prove that $\limsup\limits_{\Delta t \to 0} \epsilon^c(\Delta t) = 0$. With the assumption that the original problem (P) has a feasible solution with the SOS1 property, we have that there exist continuous functions $u_j(t),\ j =$

$1, \ldots, N$ such that $\sum_{j=1}^{N} u_j(t) = 1$ for $t \in [0, t_f]$ almost everywhere. For any time discretization with T time intervals and length $\Delta t = t_f/T$, we construct a discretized solution $u^{(T)}$ such that

$$u_{jk}^{(T)} = \int_{t_{k-1}}^{t_k} u_j(t) dt/\Delta t, \ k = 1, \ldots, T. \tag{4.17}$$

With the definition, we have $\sum_{j=1}^{N} u_{jk}^{(T)} = 1, \ k = 1, \ldots, T$. Let $u^{(T)*}$ be the optimal solution of problem (P-L_2) with time steps T, then we have

$$\bar{F}(u^{(T)*}) + \rho T \sum_{k=1}^{T} \left(\sum_{j=1}^{N} u_{jk}^{(T)*} - 1 \right)^2 \frac{t_f}{T} \leq \bar{F}(u^{(T)}) \tag{4.18}$$

from the definition of the optimal solution. Taking the limit when $T \to \infty$, i.e. $\Delta t \to 0$, we have

$$\lim_{T\to\infty} \bar{F}(u^{(T)*}) + \lim_{T\to\infty} \rho T \sum_{k=1}^{T} \left(\sum_{j=1}^{N} u_{jk}^{(T)*} - 1 \right)^2 \frac{t_f}{T} \leq \lim_{T\to\infty} \bar{F}(u^{(T)}) \leq F_{UB}, \tag{4.19}$$

where F_{UB} is the upper bound of the objective function F and the last inequality follows $\lim_{T\to\infty} u_j^{(T)} = u_j(t)$.

With the definition that $u^{\text{con}} = u^{(T)*}$, we have

$$\sum_{k=1}^{T} \left| \sum_{j=1}^{N} u_{jk}^{\text{con}} - 1 \right| \leq \sqrt{T} \left(\sum_{k=1}^{T} \left(\sum_{j=1}^{N} u_{jk}^{(T)*} - 1 \right)^2 \right)^{1/2} \tag{4.20}$$

for any T following the norm inequality between L_1 and L_2 norm. Consider two cases that the L_2 norm in equation (4.20) is smaller or not smaller than 1, then we have

$$\sum_{k=1}^{T} \left| \sum_{j=1}^{N} u_{jk}^{\text{con}} - 1 \right| \leq \max\{\sqrt{T}, \ \sqrt{T} \sum_{k=1}^{T} \left(\sum_{j=1}^{N} u_{jk}^{(T)*} - 1 \right)^2 \} \tag{4.21}$$

for any T, respectively. Next we prove $\limsup_{\Delta t \to 0} \epsilon^c(\Delta t) = 0$. Because $\epsilon^c(\Delta t)$ can be written as

$$\epsilon^c(\Delta t) = \sum_{k=1}^{T} \left| \sum_{j=1}^{N} u_{jk}^{\text{con}} - 1 \right| \frac{t_f}{T}, \tag{4.22}$$

we only need to prove $\lim_{T \to \infty} \sqrt{T} \frac{t_f}{T} = 0$ and $\lim_{T \to \infty} \sqrt{T} \sum_{k=1}^{T} \left(\sum_{j=1}^{N} u_{jk}^{(T)*} - 1 \right)^2 \frac{t_f}{T} = 0$.

With t_f as a constant, we directly have $\lim_{T \to \infty} \sqrt{T} \frac{t_f}{T} = 0$. We prove the second limit by contradiction. Assume there exists a constant $\epsilon_0 > 0$ such that $\sqrt{T} \sum_{k=1}^{T} \left(\sum_{j=1}^{N} u_{jk}^{(T)*} - 1 \right)^2 \frac{t_f}{T} \geq \epsilon_0 > 0$ for any T, then the quadratic penalty term in the objective function of (P-L_2)

$$\begin{aligned} \lim_{T \to \infty} \rho T \sum_{k=1}^{T} \left(\sum_{j=1}^{N} u_{jk}^{(T)*} - 1 \right)^2 \frac{t_f}{T} &= \lim_{T \to \infty} \rho \sqrt{T} \sqrt{T} \sum_{k=1}^{T} \left(\sum_{j=1}^{N} u_{jk}^{(T)*} - 1 \right)^2 \frac{t_f}{T} \\ &\geq \lim_{T \to \infty} \rho \sqrt{T} \epsilon_0 = \infty, \end{aligned} \tag{4.23}$$

which contradicts the upper bound F_{UB} in (4.19). Therefore, we have

$$\limsup_{\Delta t \to 0} \epsilon^c(\Delta t) \leq \limsup_{T \to \infty} \max\{\sqrt{T} \frac{t_f}{T},\ \sqrt{T} \sum_{k=1}^{T} \left(\sum_{j=1}^{N} u_{jk}^{(T)*} - 1 \right)^2 \frac{t_f}{T}\} = 0. \tag{4.24}$$

□

In the following corollary, we prove that for quantum control problems without the SOS1 property, the convergence results still hold.

Corollary 4.2.2. *Let u^{con} be the solution of the continuous relaxation of model* (DQCP) *without the SOS1 property. Let u^{bin} be the binary control obtained by Algorithm 4.3 without the SOS1 property constraint (i.e. $\mathcal{U} = \{0, 1\}^N$) setting $\alpha = 0$. Then we have*

$$\limsup_{\Delta t \to 0} \bar{F}(u^{bin}) - \bar{F}(u^{con}) \leq 0. \tag{4.25}$$

Proof. We complete the proof by showing that we can convert a continuous solution u without the SOS1 property to a solution u' with the SOS1 property with re-defined Hamiltonian controllers.

Specifically, for a control system with N Hamiltonian controllers $H^{(1)}, \ldots, H^{(N)}$, we introduce a new control system with 2^N controllers, including Hamiltonian controllers

$\hat{H}^{(l_1,\dots,l_i)} = \sum_{j=1}^{i} H^{(l_j)},\ \forall \{l_1,\dots,l_i\} \subseteq \{1,\dots,N\},\ \forall i = 1,\dots,N,$ Especially $\hat{H}^{(\emptyset)}$ is an idle controller with represented as an all-zero Hamiltonian matrix. For any given controls $u_k \in [0,1]^N$ of the continuous relaxation of the discretized model (DQCP) without the SOS1 property, we can convert them to controls $u'_k \in [0,1]^{2^N}$ such that $\sum_{j=1}^{2^N} u'_{jk} = 1$ as the following rule for every time step $k = 1,\dots,T$. At time step k, given control variable u_k for N controllers, we sort the control values by descending order as $u_{l_1(k)k} \geq \dots \geq u_{l_N(k)k}$ where $l_i(k)$ is the controller index with the ith high control value. For simplicity, we represent the sorted controller index as $l_1,\dots,l_N$ by eliminating k, then we reconstruct controls u' as

$$u'_{(l_1,\dots,l_i)k} = u_{l_ik} - u_{l_{i+1}k},\ i = 1,\dots,N-1 \tag{4.26a}$$

$$u'_{(l_1,\dots,l_N)k} = u_{l_Nk} \tag{4.26b}$$

$$u'_0 = 1 - u_{l_1k}, \tag{4.26c}$$

where $u'_{(l_1,\dots,l_i)}$ is the control value corresponding to Hamiltonian controller $H^{(l_1,\dots,l_i)}$, $i = 1,\dots,N$ and u'_0 is the control value corresponding to all-zeros Hamiltonian controller $\hat{H}^{(\emptyset)}$. We have that

$$\begin{aligned}\sum_{i=1}^{N} u'_{(l_1\dots l_i)k}\hat{H}^{(l_1\dots l_i)} + u'_0\hat{H}^{(\emptyset)} &= \sum_{i=1}^{N} u'_{(l_1\dots l_i)k}\sum_{j=1}^{i} H^{(l_j)} = \sum_{j=1}^{N}\left(\sum_{i=j}^{N} u'_{(l_1\dots l_i)k}\right) H^{(l_j)}\\ &= \sum_{j=1}^{N} u_{l_jk}H^{(l_j)} = \sum_{l=1}^{N} u_{lk}H^{(l)}.\end{aligned} \tag{4.27}$$

Therefore the two control variables u and u' with corresponding control systems have the same impact on the states. Based on the definition of u' and bound constraints that $u \in [0,1]^{N\times T}$, we have $\sum_{i=1}^{N} u'_{(l_1\dots l_i)k} + u'_0 = 1$ and $u' \in [0,1]^{2^N\times T}$, which means that the SOS1 property holds for the new control system. From Corollary 4.2.1, the convergence results of Algorithm 4.3 hold for u' with the SOS1 property, hence, hold for u. □

4.2.2 Method Based on Cumulative Difference

Evaluating the difference between binary controls and continuous controls by cumulative difference was proposed by Sager et al. [182] and is widely used in optimal control theory [135, 181, 236]. In this section, we propose another method (Algorithm 4.4) to balance the cumulative difference between binary controls and continuous controls and the penalized number of switches. We define β as the TV norm penalty parameter. The explanation of the algorithm is as follows. At each time step k, we compute the cumulative difference of current controls $[u_1^{\mathrm{bin}}, \ldots, u_{k-1}^{\mathrm{bin}}, \hat{u}] \in [0,1]^{N \times T}$ for all $\hat{u} \in \mathcal{U}$ and select $\hat{u}^*$ as the one with the smallest cumulative difference (see step 4–5). If the cumulative difference of keeping control of the current time step the same as the previous time step, i.e. $\hat{u} = u_{k-1}$, is no larger than the weighted TV-norm value of choosing the control with the smallest cumulative difference, i.e. $\hat{u} = \hat{u}^*$, we keep the previous time step controllers (see step 6–7). Otherwise, we choose the controller with the smallest cumulative difference. Then update the binary control u^{bin} and move to the next time step $k+1$ (see step 9).

According to the paper by Sager et al. [182], the cumulative difference at each time step k for any feasible control $\hat{u} \in \mathcal{U}$ is defined by

$$\mathrm{Diff}\left(u^{\mathrm{con}}, [u_1^{\mathrm{bin}}, \ldots, u_{k-1}^{\mathrm{bin}}, \hat{u}]\right) = \max_{j=1,\ldots,N} \left| \sum_{l=1}^{k-1} \left(u_{jl}^{\mathrm{con}} - u_{jl}^{bin}\right) \Delta t + \left(u_{jk}^{\mathrm{con}} - \hat{u}_j\right) \Delta t \right|. \tag{4.28}$$

Following the paper by Sager et al. [181], we introduce auxiliary deviation variables $\hat{p}_{jk}$ for each controller $j = 1, \ldots, N$ and each time step $k = 1, \ldots, T$, such that:

$$\hat{p}_{jk} = \sum_{l=1}^{k} u_{jl}^{\mathrm{con}} \Delta t - \sum_{l=1}^{k-1} u_{jl}^{\mathrm{bin}} \Delta t, \ j = 1, \ldots, N, \ k = 1, \ldots, T. \tag{4.29}$$

At time step $k = 1, \ldots, T$, we use $\hat{p}_k$ to represent the column vector of deviation variables. We update deviation variables by a recursive formula as:

$$\hat{p}_k = \hat{p}_{k-1} + u_k^{\mathrm{con}} - u_{k-1}^{\mathrm{bin}}, \ k = 2, \ldots, T. \tag{4.30}$$

For any binary control $\hat{u}$, the cumulative difference at time step k can be computed by the deviation variables as

$$\mathrm{Diff}\left(u^{\mathrm{con}}, [u_1^{\mathrm{bin}}, \ldots, u_{k-1}^{\mathrm{bin}}, \hat{u}]\right) = \max\{\max_{j:\hat{u}_j=1} \left|\hat{p}_{jk} - \Delta t\right|, \max_{j:\hat{u}_j=0} \left|\hat{p}_{jk}\right|\}. \tag{4.31}$$

Therefore we only require to compute $\hat{p}_{jk}$ and $\hat{p}_{jk} - \Delta t$ for each time step. With the definition of deviation variables, we conclude that setting controller j active does not increase the cumulative difference if and only if $|\hat{p}_{jk}| > |\hat{p}_{jk} - \Delta t|$, which means that $\hat{p}_{jk} \geq 0.5\Delta t$. Furthermore, for any two controllers, setting the one with a higher deviation leads to a higher decrease or lower increase in the cumulative difference.

Remark 4.2.1. *If binary control u^{bin} requires the SOS1 property, at each time step k, let $j^* = \arg\max_{j=1,\ldots,N} \hat{p}_{jk}$, we choose $\hat{u}_k^*$ as $\hat{u}_{j^*k}^* = 1, \ \hat{u}_{jk}^* = 0, \ \forall j \neq j^*$. If binary control does* *not require any additional constraints, at each time step k, let $J^* = \{j = 1, \dots, N : \hat{p}_{jk} \geq 0.5\Delta t\}$, we choose $\hat{u}^*_k$ as $\hat{u}^*_{jk} = 1,\ \forall j \in J^*,\ \hat{u}^*_{jk} = 0,\ \forall j \notin J^*$.*

We describe the detailed algorithm in Algorithm 4.4. Notice that the maximum number

Algorithm 4.4: Heuristic Rounding Method based on Cumulative Difference (Cd-iff)

```
Input: Continuous control $u^{\mathrm{con}}$ and TV penalty parameter $\beta$.
Initialize binary control $\hat{p}_1 = u^{\mathrm{con}}_1$.
Let $u^{\mathrm{bin}}_1 = \arg\min_{\hat{u}\in\mathcal{U}} \mathrm{Diff}\,(u^{\mathrm{con}}, [\hat{u}])$.
for $k = 2, \dots, T$ do
    $\hat{p}_k = \hat{p}_{k-1} + u^{\mathrm{con}}_k - u^{\mathrm{bin}}_{k-1}$.
    Let $\hat{u}^* = \arg\min_{\hat{u}\in\mathcal{U}} \mathrm{Diff}\left(u^{\mathrm{con}}, [u^{\mathrm{bin}}_1, \dots, u^{\mathrm{bin}}_{k-1}, \hat{u}]\right)$.
    if $\mathit{Diff}\left(u^{con}, [u^{bin}_1, \dots, u^{bin}_{k-1}, u^{bin}_{k-1}]\right) \leq \beta TV\left([u^{bin}_1, \dots, u^{bin}_{k-1}, \hat{u}^*, u^{con}_{k+1}, \dots, u^{con}_T]\right)$
      then
        Update binary control $u^{\mathrm{bin}}_k = u^{\mathrm{bin}}_{k-1}$.
    else
        Update control $u^{\mathrm{bin}}_k = \hat{u}^*$, breaking ties choosing control with the smallest
          TV value.
    end
end
Output: Binary control $u^{\mathrm{bin}}$
```

of switches with penalty parameter β is t_f/β following from the definition of cumulative differences. When the weight parameter $\beta = 0$, the convergence results of SUR [64, 181] ensure that the cumulative difference between u^{bin} and u^{con} converges to zero, no matter whether the SOS1 property holds or not, leading to the objective value of the binary control $\bar{F}(u^{\mathrm{bin}})$ converges to the objective value of the continuous control $\bar{F}(u^{\mathrm{con}})$ when Δt goes to zero.

4.3 Switching Time Optimization

With derived controller sequences in Section 4.2, we next propose our generic switching time optimization model based on the continuous time horizon. Then we introduce the solution method and an acceleration technique for time-evolution simulations.

Formulation Let S be the total number of controllers in the controller sequence $\bar{\mathcal{H}}$, we divide the whole time horizon into S time intervals as $0 = t_0 < t_1 < \ldots < t_S = t_f$ where each time interval has a fixed Hamiltonian controller. For each time interval $s = 1, \dots, S$, we define variables $\tau_s \in [0, t_f]$ as the length and $X_s \in \mathbb{C}^{2^q \times 2^q}$ as the unitary operator at the end of the time interval. Let τ be the corresponding vector form of τ_s. By definition, the final unitary operator $X(t_f) = X_S$. The start and end time of time interval s is computed by $t_{s-1} = \sum_{l=1}^{s-1} \tau_l$ and $t_s = \sum_{l=1}^{s} \tau_l$. The unitary operators follow the differential equation

$$\frac{d}{dt}X(t) = -i\bar{H}_s X(t),\ \forall t \in (t_{s-1}, t_s],\ s = 1, \dots, S. \tag{4.32}$$

We obtain the explicit solution as $X(t) = \exp\{-i\bar{H}_s(t - t_{s-1})\}X(t_{s-1}),\ \forall t \in (t_{s-1}, t_s],\ s = 1, \dots, S$. As a result, the final operator $X(t_f)$ can be computed as an implicit function of τ:

$$X(t_f, \tau) = X_S(\tau) = \prod_{s=1}^{S} e^{-i\bar{H}_s \tau_s} X_{\text{init}}. \tag{4.33}$$

Substituting the final operator $X(t_f, \tau)$ into a general objective function F, we obtain an objective function $\hat{F}(\tau)$ with respect to variables τ and formulate the switching time optimization problem as

$$(STO) \quad \min_{\tau} \quad \hat{F}(\tau) = F(X(t_f, \tau)) \tag{4.34a}$$

$$\text{s.t.} \quad \sum_{s=1}^{S} \tau_s = t_f \tag{4.34b}$$

$$\tau_s \geq 0,\ s = 1, \dots, S. \tag{4.34c}$$

The objective function (4.34a) is a general function only corresponding to τ. Constraint (4.34b) enforces that the sum of the time intervals equals the evolution time. Constraint (4.34c) ensures that the length of each time interval is in $[0, t_f]$. Note that time interval s with length as zero means that we do not apply Hamiltonian controller $\bar{H}_s$ to the control system, leading to a reduction in the number of switches.

Solution method Because the switching optimization model (STO) includes the additional summation constraint (4.34b), we derive the gradient of $\hat{F}(\tau)$ as follows and solve the model by the sequential least-squares programming (SLSQP) algorithm [215], which is a widely used iterative algorithm for solving bounded optimization problems with equality constraints. For each time interval $s = 1, \dots, S$, we define propagators $U_s = \exp\{-i\bar{H}_s\tau_s\}$. The gradient of U_s corresponding to τ_s is computed as

$$\frac{\partial U_s}{\partial \tau_s} = -i\bar{H}_s U_s,\ s = 1, \dots, S. \tag{4.35}$$

The gradient of the objective function with respect to the propagators varies with its specific formulation. For the energy function, we define back propagation variables $\kappa_s = U_{s+1}^\dagger \cdots U_S^\dagger \tilde{H} X_S|\psi_0\rangle$ for each interval $s = 1, \ldots, S-1$ and $\kappa_S = \tilde{H} X_S|\psi_0\rangle$. The gradient with respect to τ_s is computed as

$$\frac{\partial \hat{F}}{\partial \tau_s} = \frac{2}{E_{\min}} \mathrm{Re}\left[i\langle \kappa_s|\bar{H}_s X_s|\psi_0\rangle\right], \; s = 1, \ldots, S. \tag{4.36}$$

For the infidelity function, we define back propagation variables $\lambda_s = U_{s+1}^\dagger \cdots U_S^\dagger X_{\mathrm{targ}}$ for each interval $s = 1, \ldots, S-1$ and $\lambda_S = X_{\mathrm{targ}}$. Then the gradient of the trace with respect to τ_s is computed as

$$\frac{\partial \operatorname{tr}\left\{X_{\mathrm{targ}}^\dagger X_S\right\}}{\partial \tau_s} = -i\lambda_s^\dagger \bar{H}_s X_s, \; s = 1, \ldots, S. \tag{4.37}$$

Using the definition of the infidelity objective function (4.3), we compute the gradient as

$$\frac{\partial \hat{F}}{\partial \tau_s} = \frac{1}{2^q} \mathrm{Re}\left[i\mathbf{tr}\left\{\lambda_s^\dagger \bar{H}_s X_s\right\} e^{-i\mathrm{arg}\left(\mathrm{tr}\left\{X_{\mathrm{targ}}^\dagger X_S\right\}\right)}\right], \; s = 1, \ldots, S, \tag{4.38}$$

where $\arg(\cdot)$ represents the argument of a complex number.

Time-evolution simulation For each set of switching time points, we can input controllers into quantum systems and conduct the time-evolution process to obtain final objectives and gradients by the finite difference method. Conducting time-evolution simulations on classical computers is still a widely used approach in quantum control research. Directly computing matrix exponentials during time evolution requires high computational costs. Therefore, we introduce an acceleration technique of time-evolution simulations with pre-computed eigenvalue decompositions. Because all the Hamiltonian controllers in quantum systems are Hermitian matrices, they can be written as

$$\bar{H}_s = Q_s \Lambda_s Q_s^\dagger, \; s = 1, \ldots, S \tag{4.39}$$

where $\Lambda_s = \mathbf{diag}\{\lambda_{1s}, \; \ldots, \; \lambda_{2^q s}\}$ are diagonal matrices with eigenvalues $\lambda_{1s}, \; \ldots, \; \lambda_{2^q s}$ of $\bar{H}_s$ as diagonals and Q_s are unitary matrices. According to the paper by Hall [83], the corresponding propagators are computed as

$$U_s = e^{-i\bar{H}_s \tau_s} = Q_s e^{-i\Lambda_s \tau_s} Q_s^\dagger = Q_s \mathbf{diag}\{e^{\lambda_{1s}}, \; \ldots, \; e^{\lambda_{2^q s}}\} Q_s^\dagger, \; s = 1, \ldots, S. \tag{4.40}$$

Therefore, with pre-decomposed $\bar{H}_s,\ s = 1, \ldots, S$, for any interval length variables τ, we only need to compute 2^q real number exponentials for all the eigenvalues and 2 matrix exponentials for computing each propagator during the time-evolution process. The number of eigenvalue decompositions we require to pre-compute is no more than the number of distinctive controllers in the sequence $\bar{\mathcal{H}}$, which is at most N for problems with the SOS1 property and 2^N for problems without the SOS1 property.

4.4 Numerical Studies

We apply our solution frameworks from Sections 4.2 and 4.3 with switching time optimization to solve multiple specific quantum control instances, to demonstrate their computational efficacy. In Section 4.4.1, we introduce our experimental design and four specific problems. In Section 4.4.2, we select a specific instance to conduct the sensitivity analysis of the switching penalty parameter for obtaining binary controls. In Section 4.4.3, we first present the objective value results of different methods and then show the computational time results and the acceleration of our pre-computing matrix decomposition technique. We demonstrate that our switching time optimization framework eliminates the requirement for precise time discretization. In Section 4.4.4, we discuss the optimal control figures of selected instances, to provide intuitive insights for our algorithms.

4.4.1 Simulation Design

For all the instances, we first solve the continuous relaxation of the discretized model (DQCP) to obtain continuous solutions. Then we apply three methods to obtain discretized binary controls and their corresponding controller sequences. The first one is Algorithm TR+MT+ALB in Chapter 3 which solves the continuous relaxation with TV regularizer by a trust-region method (TR), then rounds continuous solutions with min-up-time constraints (MT), and improves binary solutions by an approximate local-branching method (ALB). We select TR+MT+ALB as the benchmark algorithm because it obtains the best trade-off between objective values and TV regularizer values among all the instances. The second and third methods are heuristic methods based on objective value (Algorithm 4.3) and cumulative difference (Algorithm 4.4). With obtained controller sequences, we solve the switching time optimization model (STO) and obtain final solutions. We test our new algorithm framework on the four quantum control problems, including (i) energy minimization problem, (ii) NOT gate estimation problem, (iii) CNOT gate estimation problem, and (iv) circuit compilation problem introduced in Section 3.2.1–3.2.4. The parameter settings in these problems follow the setting previous chapter (see Section 3.6.2). We additionally test the circuit compilation problem with a new molecule BeH_2 (Beryllium dihydride).

The settings of all the parameters are presented in Table 4.1. We test parameters for penalizing the switching when rounding continuous controls α and β both in the interval $[10^{-n},\ 10^{-n+1}]$ with step size $5 \times 10^{-n-1}$ for $n = 1,\ 2,\ 3,\ 4$. We choose the parameter with the best trade-off between objective values and the number of switches to present the results. All the numerical simulations were conducted on a macOS computer with 8 cores, 16GB RAM, and a 3.20GHz processor in Python 3.8. All the computational time results are the times on classical computers. Our full code and results are available on our GitHub repository [62].

Table 4.1: Parameter settings of examples. The parameters include the number of qubits (q), number of controllers (N), evolution time t_f, number of time steps (T), L_2 penalty parameter (ρ), and switching penalty parameter ($\alpha,\ \beta$). The L_2 penalty parameter ρ is only for circuit compilation examples and it is marked by "-" for other examples.

Instance	q	N	t_f	T	ρ	α	β
Energy2	2	2	2	40	-	0.1	0.075
Energy4	4	2	2	40	-	0.15	0.015
Energy6	6	2	5	100	-	0.015	0.01
CNOT5	2	2	5	100	-	0.02	0.02
CNOT10	2	2	10	200	-	0.003	0.008
CNOT20	2	2	20	400	-	0.01	0.015
NOT2	1	2	2	20	-	0.01	0.03
NOT6	1	2	6	60	-	0.0015	0.015
NOT10	1	2	10	100	-	0.009	0.035
CircuitH2	2	5	10	100	1.0	0.045	0.01
CircuitLiH	4	12	20	200	0.1	0.03	0.06
CircuitBeH2	6	19	20	200	0.01	0.03	0.2

4.4.2 Sensitivity Analysis of Switching Penalty

We take an instance Energy6 to show the performance of controls of Algorithm 4.1 with different switching penalty parameters α when obtaining binary controls based on objective values (Algorithm 4.3). The performance of changing β when obtaining binary controls based on the cumulative differences (Algorithm 4.4) is similar.

In the quantum energy minimization problem, the first excited state is the state having energy as the second smallest eigenvalue of the corresponding matrix $\tilde{H}$. We use E_{fe} to represent the energy of the first excited state. In most applications, we are interested in comparing obtained energy and E_{fe}. If the obtained energy is smaller, we say that the control is good enough to distinguish the states. In Table 4.2, we present the difference between obtained energy and minimum energy $E_{\min}$ for different penalty parameters as well as the difference between the first excited state energy and the minimum energy for the 5 randomly generated instances.

Table 4.2: Difference between obtained energy and minimum energy of Energy6 example for Algorithm 4.1 using Algorithm 4.3 at step 2 with different switching penalty parameters α. Column "First-excited" represents the difference between the first-excited energy and the ground energy. We bold the maximum parameter with obtained energy less than the first-excited energy for all 5 instances.

α	First-excited	0	0.001	0.003	0.005	0.01	**0.015**	0.02
Instance 1	0.8966	0.1503	0.1507	0.1518	0.1549	0.1825	**0.2772**	0.5358
Instance 2	0.9013	0.1235	0.1243	0.1289	0.1310	0.1993	**0.2800**	0.2800
Instance 3	0.8587	0.2156	0.2163	0.2194	0.2236	0.2313	**0.2458**	0.3283
Instance 4	1.4315	0.0356	0.0367	0.0409	0.0434	0.1639	**0.1639**	0.3532
Instance 5	0.2772	0.2059	0.2064	0.2082	0.2094	0.2200	**0.2719**	0.3969
Average	0.8731	0.1462	0.1469	0.1499	0.1525	0.1994	**0.2478**	0.3788
α	0.03	0.04	0.05	0.07	0.1	0.2	0.3	0.6
Instance 1	0.5358	0.5358	0.9666	0.9666	1.4327	1.5090	3.9625	3.9625
Instance 2	0.4363	0.7910	0.7910	0.7910	1.2370	3.2501	5.3743	6.2999
Instance 3	0.4504	0.6540	0.6540	0.6540	1.3813	2.4889	4.7132	4.7573
Instance 4	0.7195	0.7195	0.7195	1.6350	1.6350	2.4832	5.3054	5.3054
Instance 5	0.3969	0.6277	0.6277	0.9823	0.9823	1.5300	1.5300	4.1083
Average	0.5078	0.6656	0.7518	1.0058	1.3337	2.2522	4.1771	4.8867

We show that with the increase of switching penalty parameter α, the differences of energy for all the instances increase, until reaching the maximum difference $-E_{\min}$. We notice that when the difference between first-excited and minimum energy is small (Instance 5), we require a smaller energy ratio difference to distinguish the states, therefore requiring a smaller penalty parameter α. We bold the maximum penalty parameter $\alpha = 0.015$ with obtained energy smaller than the first-excited energy in all the instances.

In our model, obtaining energy less than the first-excited energy E_{fe} is equivalent to obtaining an objective value smaller than $1 - E_{\text{fe}}/E_{\min}$. We present average objective values $1-E/E_{\min}$ and TV-norm values among 5 instances for different switching penalty parameters in Table 4.3. The detailed objective values and TV-norm value results for each instance can be found in Appendix C.2. We show that with the increase of the switching penalty parameter α, the average objective value increases monotonically and the average TV-norm value decreases monotonically. Furthermore, we present the trade-off relationship between objective values computed by $1 - E/E_{\min}$ and TV-norm values in Figure 4.3. Blue circles,

Table 4.3: Average objective value and TV-norm results of Energy6 example for Algorithm 4.1 using Algorithm 4.3 at step 2 with different switching penalty parameters α. TV-norm value of the first-excited state is not applicable and marked by "-".

α	First-excited	0	0.001	0.003	0.005	0.01	0.015	0.02
Objective	0.1787	0.0319	0.0321	0.0327	0.0332	0.0422	0.0526	0.0824
TV-norm	-	116.4	74.0	41.2	32.4	23.2	20.4	17.6
α	0.03	0.04	0.05	0.07	0.1	0.2	0.3	0.6
Objective	0.1063	0.1373	0.1591	0.2109	0.2791	0.4521	0.8432	1.0000
TV-norm	15.2	12.8	12.0	10.4	8.4	4.8	2.4	0.0

orange triangles, green stars, red squares, and purple plus signs represent the results of various α of instances 1–5, respectively. We use larger transparent dots to represent the results with higher energy than first-excited state energy and use smaller opaque dots to represent the results with lower energy. We choose the point with energy lower than the first-excited state energy for both 5 instances and the smallest TV-norm as our best switching penalty parameter, which is $\alpha = 0.015$.

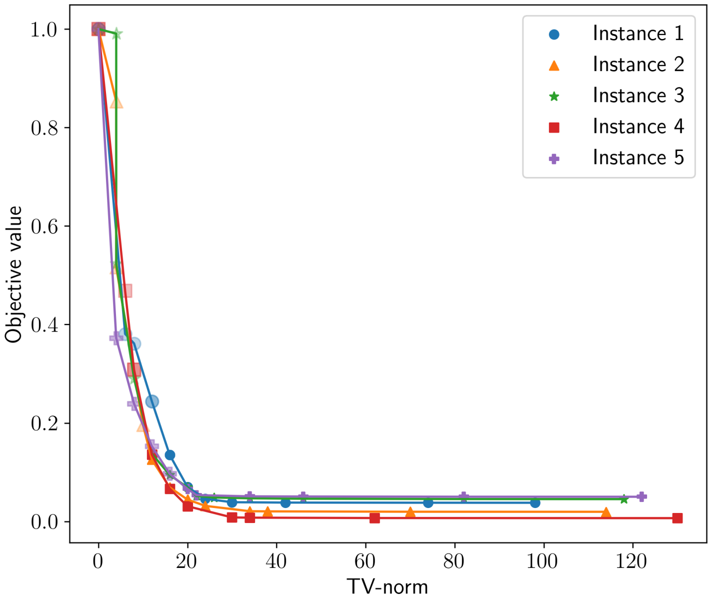


Figure 4.3: Objective value and TV-norm results of Energy6 example for Algorithm 4.1 using Algorithm 4.3 at step 2 with different switching penalty parameters α. Blue circles, orange triangles, green stars, red squares, and purple plus signs represent the results of instances 1–5, respectively. Larger transparent markers represent the results with higher energy than first-excited state energy while smaller opaque markers represent the results with lower energy.

4.4.3 Numerical Results

In this section, we discuss the numerical results for four methods introduced in Section 4.4.1. In all the following tables, figures, and discussion, we label results of discretized controls obtained by Algorithm TR+MT+ALB in Chapter 3 as “Chpt. 3”. We label the results obtained by Algorithm 4.1 using the method in Chapterr 3, heuristic methods based on objective values (Algorithm 4.3), and heuristic methods based on cumulative difference (Algorithm 4.4) to attain binary controls u^{bin} at step 2 as “Alg. 4.1w/Chpt. 3”, “Alg. 4.1w/4.3”, and “Alg. 4.1w/4.4”, respectively.

Objective results We present the histograms of objective values with log-scale and TV-norm values in Figure 4.4 for all the instances and methods. Blue and orange bars represent objective and TV-norm value results. Bars marked by slashes, backslashes, stars, and dots represent the results of methods TR+MT+ALB in Chapter 3, Algorithm 4.1 with binary controls from TR+MT+ALB in Chapter 3, Algorithm 4.1 with binary controls from Algorithm 4.3, and Algorithm 4.1 with binary controls from Algorithm 4.4, respectively.

Comparing the discretzied results in Chapter 3 and the results after Algorithm 4.1 starting from the discretized solutions, we demonstrate that switching time optimization significantly improves the objective value because we can eliminate a fixed time discretization without increasing the number of switches. Compared to the results of Algorithm 4.1 with binary controls from Chapter 3, we show that Algorithm 4.1 with binary controls obtained from new heuristic methods (Algorithm 4.3–4.4) is not only more concise and convenient for adjusting parameters but also gains a better trade-off results for most instances. For the results of Algorithm 4.1 with our two new methods to obtain binary control, we illustrate that the objective values are in the same order but the TV-norm values vary among instances. In practice, we recommend testing two methods and choosing the best one.

We select an instance NOT10 and present the control results with objective values (represented by “Obj”) and TV-norm values in Figure 4.5. We show that switching optimization improves the objective value significantly by slightly modifying the time points of switches by comparing the upper-left and upper-right figures in Figure 4.5. We demonstrate that switching time optimization with our new rounding algorithms reduces both objective values and TV-norm values in Figure 4.5.

CPU time results We present the CPU time of solving continuous relaxations (Step 1), extracting binary controls (Step 2), and conducting switching time optimization (Step 4) for all the methods in Table 4.4. We eliminate the time of merging intervals because it is not closely related to various instances and methods. We show our new methods of obtaining bi-

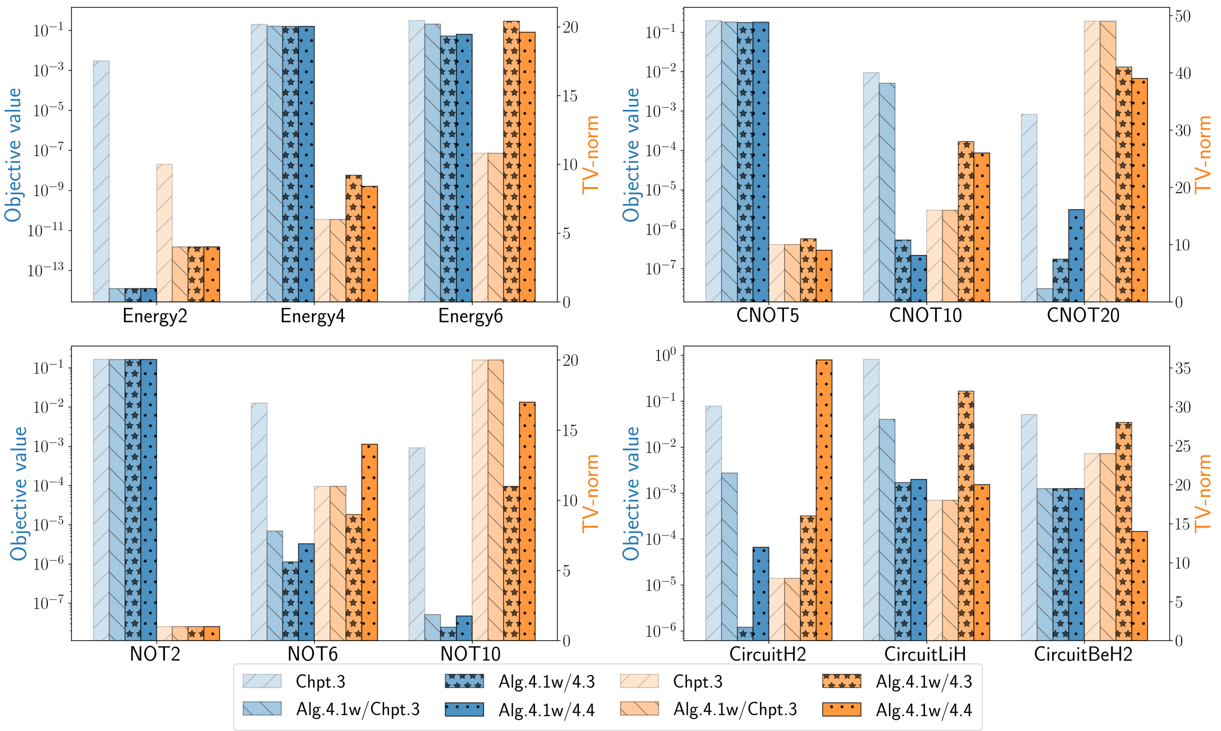


Figure 4.4: Histograms of objective values with log-scale and TV-norm values for all the instances and methods. Blue and orange bars represent objective and TV-norm value results. Bars marked by slashes, backslashes, stars, and dots represent methods TR+MT+ALB in Chapter 3, Alg. 4.1 with binary controls from TR+MT+ALB in Chapter 3, Alg. 4.1 with binary controls from Alg. 4.3, and Alg. 4.1 with binary controls from Alg. 4.4.

nary controls dramatically reduce the computational time compared to the previous method by eliminating the local search process. With our proposed acceleration technique in the heuristic method based on objective values, we avoid conducting time-evolution for $N \cdot T$ times and complete the process in seconds. The time of solving the switching optimization model (STO) is all less than one second.

We present how the common logarithm of CPU time and iterations vary among problem sizes computed by $2^q \cdot T \cdot N$ when solving (STO) in Figure 4.6. We show that CPU times grow exponentially with q because the dimensions of Hamiltonian matrices grow exponentially, while the number of iterations is stable. The main CPU time of large-scale examples such as Energy6 and CircuitBeH2 comes from solving continuous relaxations. In practice, using quantum computers to conduct the time-evolution process can help save computational times with at most exponential reduction. We only need to evaluate the final state for computing the objective value and the gradient by the finite difference method.

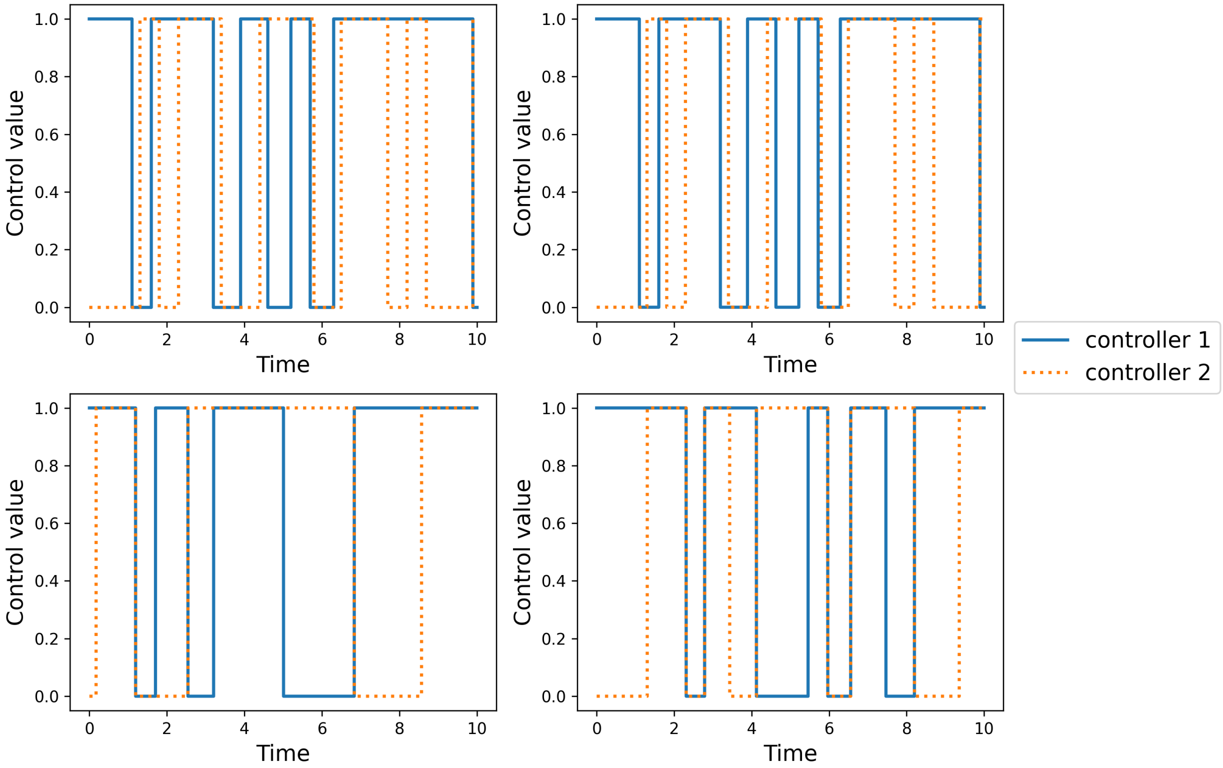


Figure 4.5: Control results of instance NOT10 for all the methods. Upper-left: discretized controls obtained by TR+MT+ALB in Chapter 3 (objective 9.087E−04, TV-norm 20). Upper-right: controls of Algorithm 4.1 with binary controls obtained by TR+MT+ALB in Chapter 3 (objective: 5.132E−08, TV-norm: 20). Lower-left: controls of Algorithm 4.1 with binary controls obtained by Algorithm 4.3 (objective: 2.439E−08, TV-norm: 11). Lower-right: controls of Algorithm 4.1 with binary controls obtained by Algorithm 4.4 (objective: 4.692E−08, TV-norm: 17). Blue lines and orange dashed lines represent controllers 1 and 2, respectively.

Acceleration of time-evolution simulation We select two large-scale instances, Energy6 and CircuitBeH2 to indicate the benefits of our time-evolution simulation with pre-decomposed Hamiltonian controllers proposed in Section 4.3. We present the CPU time of conducting time-evolution processes and the total CPU time when solving the model (STO) in Table 4.5, where rows “Accelerated” represent the results of our accelerated simulation and rows “Baseline” represent the results of the baseline that computes matrix exponentials at each iteration. We demonstrate our accelerated simulation obtains remarkable improvement in CPU times with at most 16.3x speed-up.

Results of fewer time steps Because the switching time optimization model (STO) only requires the controller sequence and optimizes the time of all the switching points, we no

Table 4.4: CPU time (s) results of all the methods for main steps in Algorithm 4.1. We eliminate the time of merging intervals. Column "Continuous" represents the results of solving continuous relaxations (Step 1). Columns "Alg. 4.1w/Chpt. 3", "Alg. 4.1w/4.3", and "Alg. 4.1w/4.4" represent the time of rounding continuous controls (Step 2) and solving the switching time optimization model (STO) (Step 4) in Algorithm 4.1 using TR+MT+ALB in Chapter 3, Algorithm 4.3, and Algorithm 4.4 in Step 2, respectively.

	Continuous	Binary control extraction (Step 2)			Switching time optimization (Step 4)		
	(Step 1)	Alg. 4.1w/Chpt. 3	Alg. 4.1w/4.3	Alg. 4.1w/4.4	Alg. 4.1w/Chpt. 3	Alg. 4.1w/4.3	Alg. 4.1w/4.4
Energy2	0.130	4.038	0.011	0.013	0.002	0.002	0.002
Energy4	2.892	33.173	0.030	0.027	0.003	0.009	0.007
Energy6	105.788	3174.384	0.678	0.484	0.063	0.463	0.167
CNOT5	1.125	100.506	0.036	0.023	0.015	0.020	0.009
CNOT10	0.725	291.447	0.071	0.043	0.035	0.036	0.044
CNOT20	1.025	595.570	0.135	0.080	0.036	0.038	0.030
NOT2	0.046	0.560	0.009	0.006	0.002	0.003	0.003
NOT6	0.147	12.240	0.022	0.015	0.013	0.007	0.016
NOT10	0.105	82.500	0.033	0.021	0.008	0.017	0.009
CircuitH2	1.754	353.006	0.045	0.022	0.008	0.018	0.032
CircuitLiH	238.685	2347.433	0.392	0.131	0.015	0.042	0.025
CircuitBeH2	3060.900	37368.162	2.026	1.758	0.124	0.555	0.311

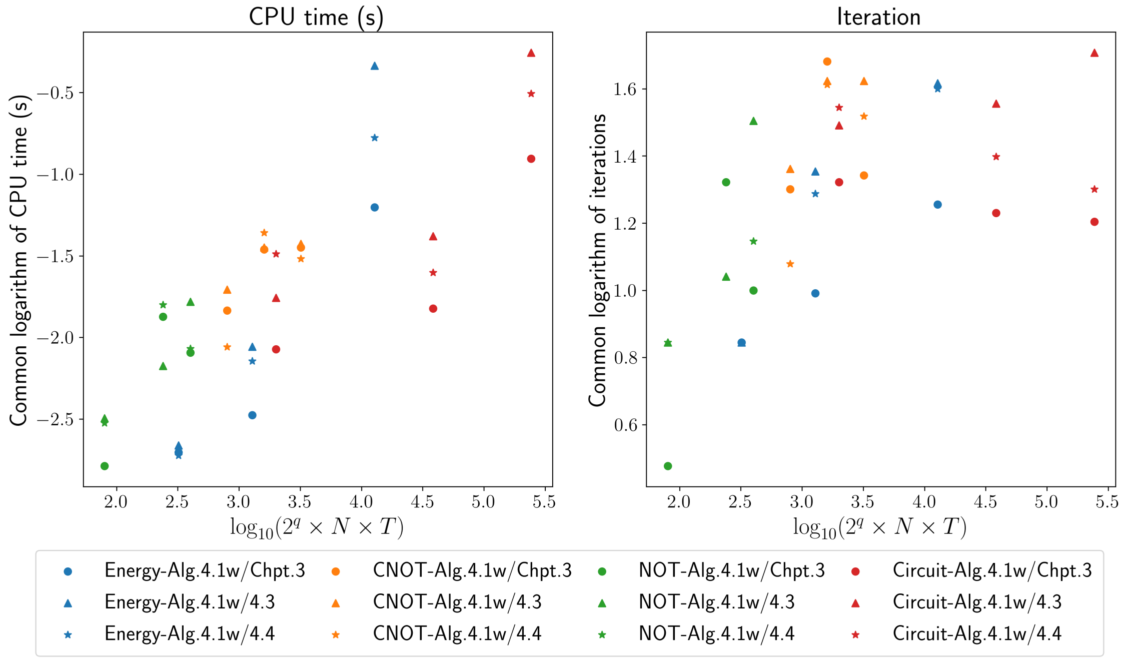


Figure 4.6: CPU times and numbers of iterations varying among problem sizes computed by $2^q \cdot T \cdot N$. We take the common logarithm of all the values. Blue, orange, green, and red dots represent instances of Energy, CNOT, NOT, and Circuit examples. Circles, triangles, and stars represent time results of Alg. 4.1 with methods in Chpater 3, Alg. 4.3, Alg. 4.4 to extract binary controls at Step 2.

Table 4.5: CPU time (s) results of solving the switching time optimization model (Step 4) with extracted controllers by various methods, including the benchmark TR+MT+ALB [64] in Column "Alg. 4.1w/Chpt. 3", Algorithm 4.3 in Column "Alg. 4.1w/4.3", and Algorithm 4.4 in Column "Alg. 4.1w/4.4". We present CPU times of evolution simulations and total CPU times. Row "# Evolution" presents the number of evolution times. Rows "Acceleration" and "Baseline" represent results after and before our acceleration by pre-decomposing matrices. Row "Speed-up" represents the speed-up results.

		Energy 6			CircuitBeH2		
		Alg. 4.1w/Chpt. 3	Alg. 4.1w/4.3	Alg. 4.1w/4.4	Alg. 4.1w/Chpt. 3	Alg. 4.1w/4.3	Alg. 4.1w/4.4
# Evolution		18.0	41.4	39.8	16	51	20
Evolution (s)	Acceleration	0.035	0.255	0.095	0.052	0.235	0.163
	Baseline	0.577	1.742	1.454	0.648	2.858	0.594
	Speed-up	16.3x	6.8x	15.3x	12.6x	12.1x	12.5x
Total (s)	Acceleration	0.063	0.463	0.167	0.124	0.555	0.311
	Baseline	0.902	2.711	2.266	1.063	5.829	1.038
	Speed-up	14.4x	5.9x	13.6x	8.6x	10.5x	3.3x

longer require precise time discretization. Therefore, we can solve continuous relaxation with longer discretized time intervals, thus reducing the number of time steps and computational costs. We take Energy6 and CircuitBeH2 as examples and solve instances with $T = 20$ and $T = 40$ by Algorithm 4.1 using our new methods, Algorithm 4.3–4.4 to round controls. In Table 4.6, we compare objective values, TV-norm values, and CPU times between different numbers of time steps. We show that both methods obtain similar objective values and TV-norm values after switching time optimization with a significant decrease in CPU times.

Table 4.6: Comparison of objective value, TV-norm value, and CPU time results with different time steps on example Energy6 ($T = 20,\ 100$) and CircuitBeH2 ($T = 40,\ 200$) for the switching time optimization model. Columns "Alg. 4.1w/4.3" and "Alg. 4.1w/4.4" represent results of Algorithm 4.1 with extracted binary controls obtained by Algorithm 4.3 and Algorithm 4.4.

	Energy6-T20		Energy6-T100		CircuitBeH2-T40		CircuitBeH2-T200	
	Alg. 4.1w/4.3	Alg. 4.1w/4.4	Alg. 4.1w/4.3	Alg. 4.1w/4.4	Alg. 4.1w/4.3	Alg. 4.1w/4.4	Alg. 4.1w/4.3	Alg. 4.1w/4.4
Objective	0.0723	0.0453	0.0526	0.0632	1.250E−03	1.249E−03	1.250E−03	1.250E−03
TV-norm	18.8	22	20.4	19.6	22	14	28	14
CPU time	38.579	38.609	106.929	106.439	329.643	329.383	3063.481	3062.387

4.4.4 Discussion of Controls

In this section, we choose Algorithm 4.1 with the method of obtaining binary controls based on objective values (Algorithm 4.3) as an example to show the figures of controls and provide intuitive explanations of our overall framework. We select examples CircuitLiH, CircuitH2, and the fourth randomly generated instance of Energy6 to present the figures. For each example, we present the continuous controls u^{con} after solving the continuous relaxation (Step 1), extracted binary controls by Algorithm 4.3 (Step 2), and controls after switching time optimization (Step 4) with their objective values.

In Figure 4.7, we show the results of the example CircuitLiH. Comparing the objective value of binary controls and optimized control, we show that solving the switching time optimization model significantly improves the objective value. We notice that the binary controls obtained based on objective values have the most switches at the beginning of the evolution time interval because our heuristic algorithm aims to balance the objective value and the number of switches. Specifically, at the beginning of the time evolution interval, keeping the same control leads to a high objective value increase, so the system switches frequently among different controllers. At the second half of the time interval, the objective value becomes stable because it is upper-bounded by one, hence the algorithm chooses to keep the current controller to avoid an increase in the number of switches.

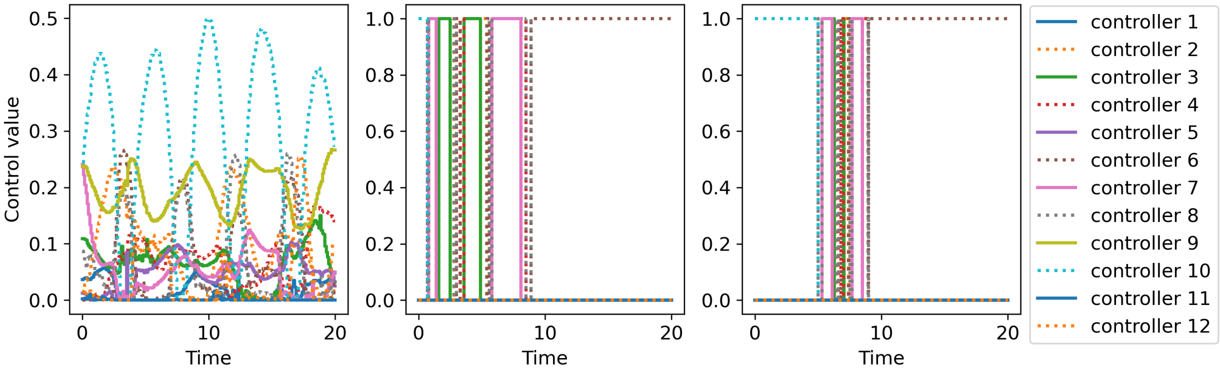


Figure 4.7: Control results for CircuitLiH of Algorithm 4.1 with binary controls obtained based on objective values (Algorithm 4.3). Left: Continuous controls with objective 1.310E−03. Middle: Binary controls with objective 0.9993. Right: Optimized controls with objective 1.702E−03.

In Figure 4.8, we present the control results for the example CircuitH2. We demonstrate that the optimized control results obtain a smaller objective value even compared to the continuous controls, showing the advantages of the switching time optimization model by eliminating time discretization. Furthermore, we show that the switching time optimization model can obtain an optimal solution in that some time intervals have zero length, leading to a reduction in the number of switches.

In Figures 4.9 and 4.10, we present the control results with larger time steps $T = 100$ and fewer time steps $T = 20$. We show that reducing the time steps leads to a higher objective value of continuous results and rounded binary controls, which is a disadvantage

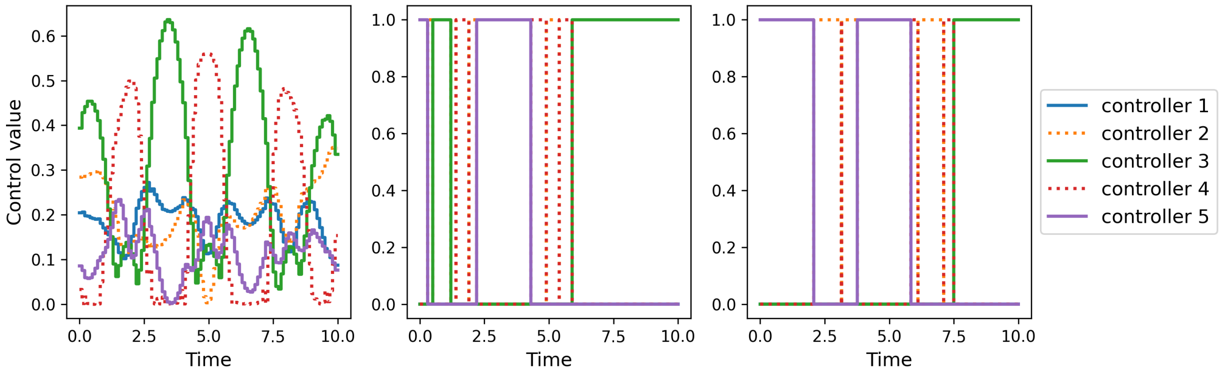


Figure 4.8: Control results for CircuitH2 of Algorithm 4.1 with binary controls obtained based on objective values (Algorithm 4.3). Left: Continuous controls with objective 2.150E−04. Middle: Binary controls with objective 0.9619. Right: Optimized controls with objective 1.208E−06.

of the discretized model. However, our control results after the switching time optimization obtain the same objective for $T = 100$ and $T = 20$, indicating the capability of our model for reducing the computation of continuous relaxation with no degradation in the final optimized results.

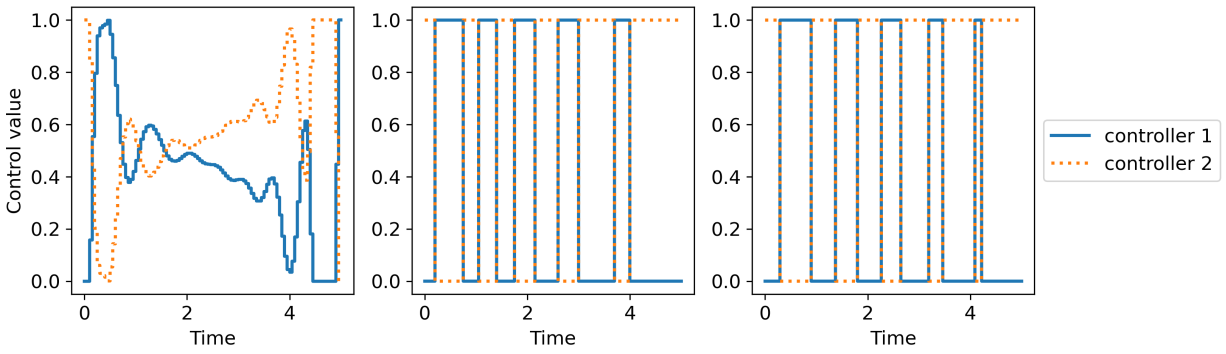


Figure 4.9: Control results for instance 4 of Energy6 of Algorithm 4.1 with binary controls obtained based on objective values (Algorithm 4.3). The number of time steps $T = 100$. Left: Continuous controls with objective 0.0067. Middle: Binary controls with objective 0.1808. Right: Optimized controls with objective 0.0309.

4.5 Concluding Remarks

In this chapter, we studied a generic binary quantum control problem. We proposed an algorithmic framework optimizing control functions as well as time points of switching con-

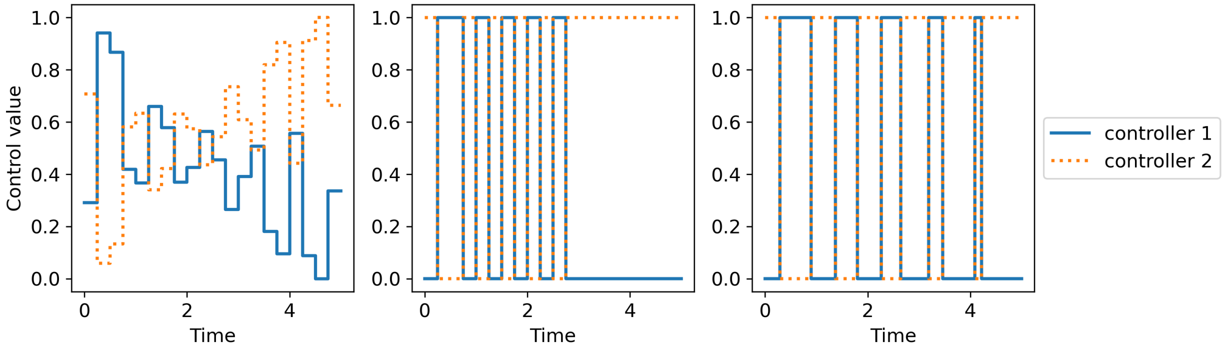


Figure 4.10: Control results for instance 4 of Energy6 of Algorithm 4.1 with binary controls obtained based on objective values (Algorithm 4.3). The number of time steps $T = 20$. Left: Continuous controls with objective 0.0078. Middle: Binary controls with objective 0.2231. Right: Optimized controls with objective 0.0309.

trollers. We proposed two heuristic methods to obtain controller sequences from discretized continuous controls with a penalty on the number of switches. With obtained controller sequences, we developed a switching time optimization model to optimize the switching points of controllers and applied a sequential least-squares programming algorithm to solve it. Furthermore, we proposed an acceleration technique to compute matrix exponentials for the time-evolution simulation on classical computers.

With numerical experiments on multiple instances in different quantum systems, we demonstrated that our new framework significantly outperforms binary solutions of the time discretization model in balancing objective values and switching frequency. Although approaches to obtaining binary controls are various, we showed that our new heuristic methods are concise and effective, and obtain the best quality control after switching time optimization. In practice, one can test new heuristic methods and adjust penalty parameters to achieve the best control. Furthermore, we indicated that the switching time optimization model requires less precise time discretization, leading to a reduction of the computational burden.

Because our algorithm only requires the final state of quantum systems to evaluate the objective value and the gradient, using quantum computers to conduct the time-evolution process is one of our future research directions. Noise and uncertainty in quantum systems have been a long-standing problem, making it a valuable topic to take account into uncertainties in quantum control problems and design robust controls.

CHAPTER 5

Binary Quantum Control Optimization with Uncertain Hamiltonians

5.1 Introductory Remarks

Quantum control theory [51, 74, 225] is an area of quantum research that focuses on designing efficient and accurate controls to manipulate quantum systems toward desired quantum states and operations. The early applications of quantum control theory include nuclear magnetic resonance experiments [94, 95, 136, 195] and quantum chemistry [100, 162]. With the development of quantum technologies, quantum control theories have played an important role in quantum information [75, 111, 145, 157, 158, 160, 172] and the high-level design of quantum algorithms [12, 26, 27, 64, 65, 138, 214].

Various methods have been proposed for tackling quantum optimal control problems. Khaneja et al. [95] first develop the gradient ascent pulse engineering (GRAPE) algorithm which estimates controls by piece-wise constant functions. Larocca and Wisniacki [109] later improve the computational efficiency of the GRAPE algorithm using Krylov subspace. Another popular algorithm is the chopped random basis algorithm, which describes the control space by basis functions [36, 56, 198]. Some other studies solve the optimal control problem by gradient-free methods, including the evolution algorithm [239] and reinforcement learning [32, 38, 152, 194, 244]. For binary control problems, Vogt and Petersson [216] propose a trust-region method for optimal control in a single flux quantum system. Recently, Fei et al. [64, 65] develop a solution framework for general quantum systems and improve it by switching time optimization.

All the aforementioned papers only study the deterministic quantum optimal control problem. However, the imprecise estimation of Hamiltonian controllers and time-varying noises in quantum systems has recently raised the need for robust quantum control [50, 78, 79, 102, 167]. Moreover, designing a robust uniform control is an important topic in inhomogeneous quantum ensembles, involving a large number of quantum systems with variations in system parameters [39, 118, 141, 155, 168]. Fourier decomposition methods are applied to design a uniform control for inhomogeneous quantum fields [118, 168]. Barr et al. [14] extend quantum noise spectroscopy to design optimized amplitude control waveforms suppressing low-frequency dephasing noise and detuning errors. Ruths and Li [179] propose a multidimensional pseudospectral method with uncertainty sampling for optimal control of quantum ensembles. Chen et al. [39] apply a sample approximation algorithm and Wu et al. [231] extend it to a batch-based sampling algorithm to minimize the expectation of the error between final and target operations. To hedge against risk, some studies focus on optimizing the worst-case performance under uncertainties. Wesenberg [226] solves a robust quantum optimal control problem using a general minmax algorithm based on a series of constrained quasi-Newton sequential quadratic programs. Kosut et al. [103] develop a sample-based sequential convex programming scheme to obtain an optimal control for the worst-case robust optimization problem.

The main contributions of this chapter are threefold. First, we develop a stochastic optimization model and a sample-based reformulation for the general quantum optimal control problem under uncertain Hamiltonians to balance risk-neutral and risk-averse objectives. Second, we apply multiple gradient-based methods and rounding techniques to solve the reformulated mixed-integer stochastic programming model. We provide theoretical discussions for the derivative of the objective function and the gap between solutions before and after rounding. Third, we analyze the performance of our approaches under various variance settings and demonstrate the benefits of considering uncertainties when conducting binary controls of quantum systems.

The remainder of this chapter is organized as follows. In Section 5.2, we present a general mixed-integer stochastic optimization model and its reformulation based on finite samples of the uncertain Hamiltonians. In Section 5.3, we derive the reformulation of the original stochastic optimization model and propose our gradient-based algorithm to solve the continuous relaxation. We apply rounding techniques to obtain binary controls and analyze the gap between binary and continuous control solutions. In Section 5.4, we introduce two specific quantum control instances and discuss the results of our numerical tests and simulation. In Section 5.5, we conclude our work and state future research directions.

5.2 Model with Uncertain Hamiltonians

Following the discretized binary quantum control model (DQCP) introduced in Chapter 3, we extend the deterministic model to a general stochastic optimization model in Section 5.2.1 and propose the sample-based reformulation. Next, we describe the risk measure function used in this chapter in Section 5.2.2.

5.2.1 Stochastic Optimization Model

In practice, there exist time-varying noises in the intrinsic and control Hamiltonians due to various reasons such as decoherence, hardware limitations, and environmental noise. On the other hand, multiple applications, such as inhomogeneous quantum ensembles, require applying a uniform control to manipulate quantum systems with different Hamiltonians values. These properties and applications lead to studies that take the uncertainty of Hamiltonians into consideration. In this chapter, we denote the uncertainty parameters as $\xi = [\xi_{jk}] \in \Xi \subseteq \mathbb{R}^{(N+1)\cdot T}$ where $\xi_{jk},\ j = 0, \ldots, N,\ k = 1, \ldots, T$ represents the uncertainty of the intrinsic Hamiltonian ($j = 0$) and jth control Hamiltonian ($j \geq 1$) at time step k. We assume that the uncertainty parameters have a known distribution $\mathbb{P}$ and use finite samples to approximate the distribution. We define set $\mathcal{S}$ as a finite set of uncertainty realizations $\mathcal{S} = \{\xi^1, \ldots, \xi^S\}$ according to the distribution $\mathbb{P}$ such that ξ^s, for each $s = 1, \ldots, S$ is associated with probability p_s with $\sum_{s=1}^{S} p_s = 1$. The time-dependent Hamiltonians and unitary operators are the functions of uncertain parameters ξ. For each sample $\xi^s,\ s = 1, \ldots, S$, at each time step $k = 1, \ldots, T$, we denote the corresponding time-dependent Hamiltonian H_k and unitary operator X_k as H_k^s and X_k^s, respectively. We define $\rho\left[F_X(X_T^1), \ldots, F_X(X_T^S)\right]$ as a risk measure function based on the uncertainty sample set $\mathcal{S}$. The generic stochastic optimization model variant of Model (DQCP) is given by

$$
\begin{aligned}
(\mathrm{SP}(\mathcal{S}))\quad \min_{u,X,H}\quad & \rho\left[F_X(X_T^1), \ldots, F_X(X_T^S)\right] && (5.1\text{a})\\
\text{s.t.}\quad & H_k^s = (1 + \xi_{0k}^s) H^{(0)} + \sum_{j=1}^{N} (1 + \xi_{jk}^s) u_{jk} H^{(j)},\ k = 1, \ldots, T,\ s = 1, \ldots, S && (5.1\text{b})\\
& X_k^s = e^{-iH_k^s \Delta t} X_{k-1}^s,\ k = 1, \ldots, T,\ s = 1, \ldots, S && (5.1\text{c})\\
& X_0^s = X_{\mathrm{init}},\ s = 1, \ldots, S && (5.1\text{d})\\
& \sum_{j=1}^{N} u_{jk} = 1,\ k = 1, \ldots, T\\
& u_{jk} \in \{0, 1\},\ j = 1, \ldots, N,\ k = 1, \ldots, T.
\end{aligned}
$$

The objective function (5.1a) uses a risk measure ρ to evaluate the risk of having huge deviations from the desired cost of controling quantum operators under uncertainty, which we will describe the details in Section 5.2.2. (In Section 5.4 for the numerical results, we introduce specific quantum control objective functions for different examples.) Constraints (5.1b) illustrate how to compute time-dependent Hamiltonians given sample $\xi^s,\ s = 1, \ldots, S$. Constraints (5.1c)–(5.1d) are the copies of constraints (3.3c)–(3.3d) for $s = 1, \ldots, S$.

5.2.2 Risk Measure and Objective Function

The risk measure ρ in the objective function (5.1a) in the stochastic optimization model can take different forms depending on the decision-maker's risk attitudes and uncertainty levels. One of the most widely-used measures is the expectation of a random variable, which measures the average performance, also known as the risk-neutral measure [187]. For any stochastic function $f(\xi)$ where ξ has support Ξ and distribution $\mathbb{P}$, the expectation is defined as $\mathbb{E}_\xi[f(\xi)] = \int_\Xi f(\xi) d\mathbb{P}$. With a sample set $\mathcal{S}$, the approximation expectation formulation for the stochastic function $f(\xi)$ is $\sum_{s=1}^{S} p_s f(\xi^s)$.

However, a particular realization of ξ can be significantly different from its expectation. In quantum control applications, avoiding extremely bad performance in some quantum systems is important and necessary. Here we consider a risk-averse measure, to control the risk in the solutions given by the stochastic optimization model (SP($\mathcal{S}$)) [164, 174, 175, 185]. The Conditional Value-at-Risk (CVaR) function [175] is considered in this paper given that it is a coherent risk measure with nice properties such as convexity. For any stochastic function $f(\xi)$, the CVaR function with risk level η is defined as the expected value of $f(\xi)$ subject to that the value of $f(\xi)$ is no less than the lower $1 - \eta$ percentile [185]. We show an illustration figure for $\eta = 0.05$ in Figure 5.1, where the blue line at 2 represents the 95 percentile of $f(\xi)$ and the red dashed line represents the CVaR value as the average of all values of $f(\xi)$ that is larger than 2. We introduce the following equivalent formulation for the CVaR function [175]:

$$\mathrm{CVaR}_\eta\left(f(\xi)\right) = \inf\left\{\zeta + \frac{1}{\eta}\int_\Xi \max\{0, f(\xi) - \zeta\} d\mathbb{P} : \zeta \in \mathbb{R}\right\}. \tag{5.2}$$

With a sample set $\mathcal{S}$, the CVaR function has the following sample approximation formulation:

$$\min_{\zeta \in \mathbb{R}} \left(\zeta + \frac{1}{\eta}\sum_{s=1}^{S} p_s \max\{0, F_X(X_T^s) - \zeta\}\right). \tag{5.3}$$

One can consider a linear combination of expectation and CVaR function as the specific objective function in (SP($\mathcal{S}$)) to balance between risk-neutral and risk-averse attitudes. With

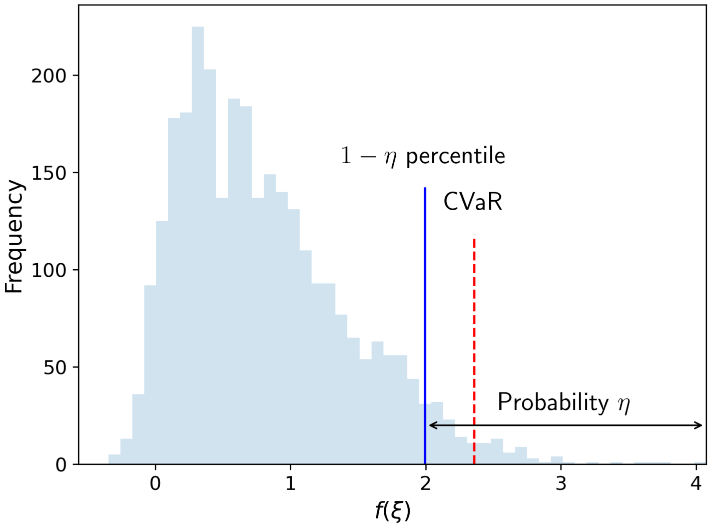


Figure 5.1: Illustration for CVaR function with risk level $\eta = 0.05$. The histograms are the distribution of $f(\xi)$. The blue line represents the 95 percentile of $f(\xi)$. The red dashed line represents the CVaR value.

a sample set $\mathcal{S}$, the risk measure (5.1a) has the following specific formulation:

$$\begin{aligned}\rho\left[F_X(X_T^1),\dots,F_X(X_T^S)\right] =& \alpha\sum_{s=1}^{S} p_s F_X(X_T^s)\\ &+(1-\alpha)\min_{\zeta\in\mathbb{R}}\left(\zeta+\frac{1}{\eta}\sum_{s=1}^{S} p_s\max\{0,F_X(X_T^s)-\zeta\}\right), \qquad (5.4)\end{aligned}$$

where $\alpha \in [0,1]$ is the weight parameter and η is the risk level parameter. When $\alpha = 0$, the problem is equivalent to minimizing the CVaR function to obtain a risk-averse control. When $\alpha = 1$, the goal is to optimize the expected performance of the control.

5.3 Gradient-based Algorithm

Our algorithm for solving the stochastic optimization model Our algorithm for solving the stochastic optimization model consists of two parts, (i) solving continuous relaxation and (ii) rounding solutions. In Section 5.3.1, we first convert the stochastic optimization model with sample approximation (SP($\mathcal{S}$)) to an unconstrained optimization model. Then, we discuss the derivative for the objective function and introduce two gradient-based algorithms to solve the continuous relaxation of the model. In Section 5.3.2, we apply the sum-up-rounding algorithm to obtain binary solutions with an optimality guarantee.

5.3.1 Solution Methods for Continuous Relaxation

Following constraints (5.1b)–(5.1d), we convert the final operator X_T^s to an implicit function of control variables u as

$$X_T^s(u) = \prod_{k=1}^{T} \exp\left\{-i\left((1+\xi_{0k}^s)H^{(0)} + \sum_{j=1}^{N}(1+\xi_{jk}^s)u_{jk}H^{(j)}\right)\Delta t\right\} X_{\text{init}},\ s = 1,\dots,S. \quad (5.5)$$

We use $F^s(u)$ to denote the objective function of u given uncertainty realization ξ^s by substituting $X_T^s(u)$ into the objective function (3.3a), i.e. $F^s(u) = F_X(X_T^s(u))$. We penalize the SOS1 property (3.3e) by an L_2 penalty function in the form of:

$$F_L(u) = \sum_{k=1}^{T}\left(\sum_{j=1}^{N} u_{jk} - 1\right)^2. \quad (5.6)$$

By relaxing the binary constraints (3.3f), the stochastic optimization model (SP($\mathcal{S}$)) is converted to an unconstrained optimization problem over a bounded feasible region as

$$\min_{u\in[0,1]^{N\times T},\zeta} \quad \alpha\sum_{s=1}^{S} p_s F^s(u) + (1-\alpha)\left(\zeta + \frac{1}{\eta}\sum_{s=1}^{S} p_s \max\{0, F^s(u)-\zeta\}\right) + \theta F_L(u), \quad (5.7)$$

where θ is the penalty weight parameter for the SOS1 property. The maximization function in the second term leads to the indifferentiability of variables u and ζ and we will discuss the derivative of it in the following theorems. For simplicity, we denote the second term without weight $(1-\alpha)$ by $F_{\text{CVaR}}(u,\zeta)$, i.e.,

$$F_{\text{CVaR}}(u,\zeta) = \zeta + \frac{1}{\eta}\sum_{s=1}^{S} p_s \max\{0, F^s(u)-\zeta\}. \quad (5.8)$$

We derive a closed-form expression of $\min_\zeta F_{\text{CVaR}}(u,\zeta)$ as follows.

Theorem 5.3.1. *For a given control variable u, define $s^*(u)$ as the scenario number with the largest original objective value $F^{s^*(u)}(u)$ such that*

$$\sum_{s=1}^{S} p_s \mathbb{1}_{\{F^s(u) > F^{s^*(u)}(u)\}} \geq \eta. \quad (5.9)$$

Then the closed-form expression of $\min_\zeta F_{CVaR}(u,\zeta)$ *at point* u *is given by*

$$F_C(u) = F^{s^*(u)}(u) + \frac{1}{\eta}\sum_{s:F^s(u)>F^{s^*(u)}(u)} p_s(F^s(u) - F^{s^*(u)}(u)) \tag{5.10a}$$

$$= F^{s^*(u)}(u) + \frac{1}{\eta}\sum_{s:F^s(u)\geq F^{s^*(u)}(u)} p_s(F^s(u) - F^{s^*(u)}(u)). \tag{5.10b}$$

The main idea of the proof is showing that given a control variable u, the optimal solution for the minimization problem $\min_\zeta F_{\text{CVaR}}(u,\zeta)$ is $\zeta^*(u) = F^{s^*(u)}(u)$. We provide the details in Appendix D.1.

Remark 5.3.1. *For a special case where we sample the scenario with equal probability, i.e.,* $p_s = \frac{1}{S}$*, the optimal solution* $\zeta^* = F^{s^*(u)}(u)$ *is the* $\lceil \eta S \rceil$ *largest original objective function value* $F^s(u)$ *among all the scenarios* $s = 1, \ldots, S$*.*

Using the closed-form expression, we convert the original problem to minimizing an unconstrained continuous relaxation with uncertainty sample set $\mathcal{S} = \{\xi^1, \ldots, \xi^S\}$. For simplicity, in the remaining discussion, we define the summation of terms coming from the original objective function as

$$\tilde{F}(u) = \alpha\sum_{s=1}^{S} p_s F^s(u) + (1-\alpha)F_C(u) \tag{5.11}$$

Taking the L_2 penalty function into consideration, the unconstrained continuous relaxation is formulated as

$$\min_{u\in[0,1]^{N\times T}} \quad F_R(u) = \alpha\sum_{s=1}^{S} p_s F^s(u) + (1-\alpha)F_C(u) + \theta F_L(u), \tag{SP-R($\mathcal{S}$)}$$

where the differentiability of $F_C(u)$ depends on the objective values of all the scenarios. For a given control variable point $\hat{u}$, we present the following theorem about the derivative.

Theorem 5.3.2. *For any given control variable point* $\hat{u}$*, if* $F^s(\hat{u}) \neq F^{s^*(\hat{u})}(\hat{u}),\ \forall s \neq s^*(\hat{u})$*, then the closed form* $F_C(u)$ *is differentiable at point* $\hat{u}$*, with the derivative formulation as*

$$\frac{\partial F_C(\hat{u})}{\partial \hat{u}} = \left(1 - \frac{1}{\eta}\sum_{s:F^s(\hat{u})>F^{s^*(\hat{u})}(\hat{u})} p_s\right)\frac{\partial F^{s^*(\hat{u})}(\hat{u})}{\partial \hat{u}} + \frac{1}{\eta}\sum_{s:F^s(\hat{u})>F^{s^*(\hat{u})}(\hat{u})} p_s\frac{\partial F^s(\hat{u})}{\partial \hat{u}}. \tag{5.12}$$

The detailed proof of differentiability and derivative is presented in Appendix D.1. With the derivative of $F_C(u)$, we compute the derivative of the objective function $F_R(u)$

in (SP-R($\mathcal{S}$)) by the chain rule as

$$\frac{\partial F_R(u)}{\partial u_{jk}} = \alpha \sum_{s=1}^{S} p_s \frac{\partial F^s(u)}{\partial u_{jk}} + (1-\alpha)\frac{\partial F_C(u)}{\partial u_{jk}} + 2\theta \left(\sum_{j=1}^{N} u_{jk} - 1 \right),\ j=1,\dots,N,\ k=1,\dots,T, \tag{5.13}$$

where the gradient of the original objective functions $F^s(u)$ for every scenario $s = 1, \dots, S$ depends on specific quantum problems and can be computed by the popular GRAPE algorithm [95]. We apply two optimization methods, L-BFGS-B algorithm [34] and Adam method [97] with our derived gradient for $F_R(u)$ in (5.13) to solve the continuous relaxation of the stochastic model. In numerical studies, we empirically show that the L-BFGS-B algorithm is better for quantum problems aiming to minimize the energy of a quantum state, while the Adam method performs better at problems minimizing the infidelity compared to a target quantum operator.

L-BFGS-B algorithm L-BFGS-B algorithm is a widely-used quasi-Newton method for optimizing unconstrained models with a deterministic objective function. We first generate S samples for the uncertain parameters ξ, then apply the L-BFGS-B algorithm to solve the model (SP-R($\mathcal{S}$)). Specifically, during each iteration of the L-BFGS-B algorithm, we compute the derivative by Equation (5.13) and the search direction, then conduct a line search to update control variables, following the details in Byrd et al. [34], Zhu et al. [248].

Adam method Adam is a well-known first-order gradient-based optimization method for optimizing unconstrained models with a stochastic objective function [97]. We modify the Adam method to solve our problem with a bounded feasible region by adding a projection step. The details of the algorithm are presented in Algorithm 5.1 where $\otimes$ represents element-wise multiplication between two vectors. Specifically, during each iteration, we first generate S samples to formulate the corresponding continuous relaxation (SP-R($\mathcal{S}$)) (see Step 3). Then we compute the derivative using (5.13) (see Step 4), update the control variables, and project the updated variables to the feasible region $[0,1]^{N\times T}$ (see Steps 5–8).

5.3.2 Sum-Up-Rounding Technique

With continuous solutions $u^{\text{con}} \in [0,1]^{N\times T}$, we apply the sum-up-rounding (SUR) technique to obtain binary solutions u^{bin}. The SUR technique is proposed by the work of Sager et al. [181] and is widely used in integer control optimization problems. To the best of our knowledge, most work using SUR rounds either a continuous-time control function [182, 183] or

Algorithm 5.1: Adam for solving the continuous relaxation of the stochastic model.

```
Input: Initial control values $u^{(0)}$. Maximum iteration number $K$.
Input: Step size $\gamma_1 > \gamma_2 > 0$. Objective value threshold to change step size $\bar{F}$.
Input: Exponential decay rates for the moment estimates $\beta_1,\ \beta_2 \in [0,1)$. Constant
       for numerical computation $\epsilon$.
Initialize the first and second moment vectors $m^{(0)},\ v^{(0)}$ as $N \times T$-dimensional zero
  vectors.
Initialize step size $\hat{\gamma} = \gamma_1$.
for Iteration $i = 1, \ldots, K$ do
    Generate a sample set $\mathcal{S} = \{\xi^1, \ldots, \xi^S\}$ with probability $p_s$ for $\xi^s,\ s = 1, \ldots, S$
      and formulate the unconstrained sample approximation model (SP-R($\mathcal{S}$)).
    Compute the corresponding derivative $g^{(i)} = \frac{\partial F_R(u)}{\partial u^{(i-1)}}$ by Equation (5.13).
    Compute the first moment vector $m^{(i)} = \big(\beta_1 m^{(i-1)} + (1-\beta_1) g^{(i)}\big)$.
    Compute the second moment vector $v^{(i)} = \big(\beta_2 v^{(i-1)} + (1-\beta_2) g^{(i)} \otimes g^{(i)}\big)$.
    Compute the bias-corrected moment vectors
      $\hat{m}^{(i)} = m^{(i)}/(1-\beta_1^i),\ \hat{v}^{(i)} = v^{(i)}/(1-\beta_2^i)$.
    Update control variables and project to the feasible bounded region $[0,1]^{N \times T}$ as
      $u^{(i)} = \Pi_{[0,1]^{N\times T}}\left(u^{(i-1)} - \hat{\gamma}\hat{m}^{(i)}/(\sqrt{\hat{v}^{(i)}} + \epsilon)\right)$.
    if $F_R(u^{(i)}) < \bar{F}$ then
        $\hat{\gamma} = \gamma_2$.
    end
end
Output: Continuous control solutions $u^{(K)}$.
```

controls of the continuous relaxation with the same time discretization [64, 65, 117, 135].

In our problem, the time of solving the continuous relaxation is the major part of the overall computational time and significantly increases when the number of time steps T and scenarios S is high. Therefore, we solve the continuous relaxation using fewer time steps T and round the solutions using more time steps T_R to achieve a balance between computational time and the difference between continuous and binary solutions. For simplicity, we assume that $T_R = C_{\text{SUR}} T$ in our chapter where $C_{\text{SUR}} > 1$ is a pre-determined integer constant. We present the rounding algorithm procedure in Algorithm 5.2.

We discuss how the difference between continuous and binary controls varies with time steps T in the rest of this section. In the remaining discussion, we use u to represent all the discretized controls and $u(t)$ to represent all the control functions on a continuous time horizon (i.e., $T = \infty$). We first propose two assumptions for the original problem, which are satisfied in most quantum control problems.

Assumption 5.3.1. *We assume that the original objective function for each quantum system*

Algorithm 5.2: Sum-up-rounding algorithm for continuous and binary solutions with different time steps.

```
Input: Time steps of continuous solution $T$. The multiplier factor between time
       steps of continuous and binary solutions $C_{\text{SUR}}$. Continuous control
       $u^{\text{con}} \in [0,1]^{N \times T}$.
for $k = 1, \ldots, C_{SUR}T$ do
    for $j = 1, \ldots, N$ do
        Compute cumulative deviation $\delta_{jk} = \sum_{\tau=1}^{k} u^{\text{con}}_{j\lfloor \tau/C_{\text{SUR}} \rfloor} \frac{\Delta t}{C_{\text{SUR}}} - \sum_{\tau=1}^{k-1} u^{\text{bin}}_{j\tau} \frac{\Delta t}{C_{\text{SUR}}}$.
    end
    if Binary control is required to have SOS1 property then
        Let $j^* = \arg\max_{j=1,\ldots,N} \delta_{jk}$. If there is a tie, we break the tie by choosing the
          smallest index.
        Update binary control $u^{\text{bin}}_{j^*k} = 1$ and $u^{\text{bin}}_{jk} = 0,\ \forall j \neq j^*$.
    else
        Let $J^* = \{j : \ \delta_{jk} \geq 0.5\Delta t / C_{\text{SUR}}\}$.
        Update binary control $u^{\text{bin}}_{jk} = 1,\ \forall j \in J^*$ and $u^{\text{bin}}_{jk} = 0,\ \forall j \notin J^*$.
    end
end
Output: Binary control $u^{\text{bin}} \in \{0,1\}^{N \times C_{\text{SUR}}T}$.
```

F_X is continuous, non-negative, and upper-bounded.

Assumption 5.3.2. *We assume that the stochastic optimization model* (SP($\mathcal{S}$)) *is feasible.*

We define piece-wise constant control functions $u^{\text{con}}(t)$ and $u^{\text{bin}}(t)$ as equivalent formulations to discretized controls u^{con} and u^{bin}:

$$u^{\text{con}}_j(t) = u^{\text{con}}_{jk},\ \forall t \in [(k-1)\Delta t, k\Delta t),\ j = 1, \ldots, N,\ k = 1, \ldots, T. \tag{5.14a}$$

$$u^{\text{bin}}_j(t) = u^{\text{bin}}_{jk},\ \forall t \in [(k-1)\frac{\Delta t}{C_{\text{SUR}}}, k\frac{\Delta t}{C_{\text{SUR}}}),\ j = 1, \ldots, N,\ k = 1, \ldots, C_{\text{SUR}}T. \tag{5.14b}$$

In the following theorem, we discuss the cumulative difference between continuous and binary control functions:

Theorem 5.3.3. *With Assumptions 5.3.1–5.3.2, let F_{UB} be the upper bound of F_X, then the cumulative difference between continuous and binary controls at any time t satisfies*

$$\left\| \int_0^t \left(u^{con}(\tau) - u^{bin}(\tau) \right) d\tau \right\|_\infty \leq \frac{(N-1)}{C_{\text{SUR}}}\Delta t + \frac{2N-1}{N}\sqrt{t_f F_L(u^{con})\Delta t},\ \forall t \in [0, t_f]. \tag{5.15}$$

Furthermore, we have the following convergence results for the objective values defined *in* (5.11) *of continuous and binary solutions:*

$$\lim_{\Delta t \to 0} \tilde{F}(u^{con}) = \lim_{\Delta t \to 0} \tilde{F}(u^{bin}). \tag{5.16}$$

Proof. For any time interval length Δt, let $\hat{k}$ be the index of the time step in the SUR algorithm that t falls in, then the integral can be written as follows based on the definition of piece-wise constant functions.

$$\int_0^t \left(u^{\text{con}}(\tau) - u^{\text{bin}}(\tau)\right) d\tau = \sum_{\tau=0}^{\hat{k}} \left(u^{\text{con}}_{j\lfloor \tau/C_{\text{SUR}} \rfloor} - u^{\text{bin}}_{j\tau}\right) \frac{\Delta t}{C_{\text{SUR}}}. \tag{5.17}$$

The remaining proof of the upper bound in (4.28) directly follows the proof of Theorem 4 and Corollary 1 in Fei et al. [64] with details being omitted here. For any control value u, we have

$$F_C(u) = \min_{\zeta} F_{\text{CVaR}}(u, \zeta) \leq F_{\text{CVaR}}(u, F_{UB}) = F_{UB}. \tag{5.18}$$

Hence, the original risk measure function $\tilde{F}(u)$ defined in (5.11) is upper bounded by F_{UB}. Because u^{con} is the optimal solution of penalized continuous relaxation, for a feasible solution u^f of model (SP($\mathcal{S}$)), we have

$$\tilde{F}(u^{\text{con}}) + F_L(u^{\text{con}}) \leq \tilde{F}(u^f) + F_L(u^f) = \tilde{F}(u^f) \leq F_{UB}, \tag{5.19}$$

where the last equality follows by the fact that u^f is a feasible solution of the model with the SOS1 property, so $F_L(u^f) = 0$. Therefore, we have $F_L(u^{\text{con}}) \leq F_{UB}$ and the convergence of objective values directly follows Corollary 8 in Sager et al. [181]. □

We prove that $F_L(u^{\text{con}}) \leq F_{UB}$ and the cumulative difference is upper bounded by $O(\sqrt{\Delta t})$ in Theorem 5.3.3. In the following propositions, we show that with additional assumptions, the upper bound can be tightened. We first introduce the infinite dimension formulation (i.e., $T = \infty$) for the original objective function of a single scenario s:

$$F^s(u(t)) = F_X(X^s(t_f; u)),\ s = 1, \ldots, S. \tag{5.20}$$

The operator $X^s(t_f; u)$ is the value of $X^s(t; u)$ at time t_f and $X^s(t; u)$ is the solution for the following differential equation with given control functions $u(t)$:

$$\frac{d}{dt}X^s(t) = -i\left((1+\xi_0^s(t))H^{(0)} + \sum_{j=1}^{N}(1+\xi_j^s(t))u_j(t)H^{(j)}\right)X^s(t), \tag{5.21}$$

where $\xi_0^s(t), \ldots, \xi_N^s(t)$ are time-dependent noises. We use the infinite dimension objective function $F^s(u(t))$ in (5.20) to replace $F^s(u)$ in the stochastic objective function $\tilde{F}(u)$ defined in (5.10a) and (5.11). The infinite dimension stochastic objective function $\tilde{F}(u(t))$ has the following formulation:

$$\begin{aligned}\tilde{F}(u(t)) = {} & \alpha\sum_{s=1}^{S} p_s F^s(u(t)) \\ & + (1-\alpha)\left[F^{s^*(u(t))}(u(t)) + \frac{1}{\eta}\sum_{s:F^s(u(t))>F^{s^*(u(t))}(u(t))} p_s\left(F^s(u(t)) - F^{s^*(u(t))}(u(t))\right)\right],\end{aligned} \tag{5.22}$$

where $s^*(u(t))$ is the scenario number with the largest original objective value $F^{s^*(u(t))}$ such that $\sum_{s=1}^{S} p_s \mathbb{1}_{F^s(u(t))>F^{s^*(u(t))}(u(t))} \geq \eta$ (see discretized version in Theorem 5.3.1). With the definition of infinite dimension stochastic objective function $\tilde{F}(u(t))$ in (5.11), we define the infinite dimension formulation with the SOS1 property for the stochastic optimization model (SP($\mathcal{S}$)) as

$$\begin{aligned}
\text{(SP-C)} \quad & \min_{u} \quad \tilde{F}(u(t)) && \text{(5.23a)}\\
& \text{s.t.} \quad \sum_{j=1}^{N} u_j(t) = 1, \ \textit{a.e. } t \in [0, t_f] && \text{(5.23b)}\\
& \qquad u_j(t) \in \{0,1\}, \ j = 1, \ldots, N, \ \textit{a.e. } t \in [0, t_f]. && \text{(5.23c)}
\end{aligned}$$

The objective function (5.23a) is the stochastic objective function defined in (5.22). Constraint (5.23b) enforces that the control function holds the SOS1 property for $t \in [0, t_f]$ almost everywhere. Constraint (5.23c) indicates that the control function value is binary for $t \in [0, t_f]$ almost everywhere. Every feasible solution of the discretized model (SP($\mathcal{S}$)) can be considered as a piece-wise constant control function, and thus is a feasible solution for the infinite dimension formulation (SP-C). For the above model, we impose the following assumption and derive an $O(\Delta t)$ upper bound for the cumulative difference based on it.

Assumption 5.3.3. *We assume that there exists an optimal solution for the continuous* *relaxation of the infinite dimension model with the SOS1 property* (SP-C)*, represented by* $u^{*,\mathrm{SOS1}}(t)$ *such that the original objective value* $\tilde{F}(u^{*,\mathrm{SOS1}}(t)) = 0$*.*

Proposition 5.3.1. *Recall that θ is the weight parameter of the SOS1 L_2 penalty function, with Assumptions 5.3.1–5.3.3, then we have the following bound for the cumulative difference:*

$$\left\| \int_0^t \left(u^{con}(\tau) - u^{bin}(\tau) \right) d\tau \right\|_\infty \le \left(\frac{(N-1)}{C_{\mathrm{SUR}}} + \frac{2N-1}{N\sqrt{\theta}} C_{\mathrm{diff}} \right) \Delta t, \ \forall t \in [0, t_f], \tag{5.24}$$

where C_{diff} is a constant determined by control Hamiltonians and evolution time t_f.

The goal of the proof is to find a constant C_{diff} such that $\sqrt{t_f F_L(u^{\mathrm{con}})} \le C_{\mathrm{diff}} \sqrt{\Delta t}/\sqrt{\theta}$. Then the proposition follows the conclusion in Theorem 5.3.3. The details of the proof is presented in Appendix D.1.

Furthermore, we prove in Proposition 5.3.2, with an additional Assumption 5.3.4 for the infinite dimension model (SP-C) that the second term of the cumulative difference is upper bounded by $o(\Delta t)\Delta t$ where $\lim_{\Delta t \to 0} o(\Delta t) = 0$.

Assumption 5.3.4. *We assume that there exists a constant time interval Δt_0 such that for any time discretization with $\Delta t \le \Delta t_0$, the optimal solution $u^{*,\mathrm{SOS1}}(t)$ for the continuous relaxation of infinite dimension model with the SOS1 property* (SP-C) *is continuous in each time subinterval.*

Proposition 5.3.2. *With Assumptions 5.3.1–5.3.4, for any $\Delta t \le \Delta t_0$, we have the following bound for the cumulative difference:*

$$\left\| \int_0^t \left(u^{con}(\tau) - u^{bin}(\tau) \right) d\tau \right\|_\infty \le \frac{(N-1)}{C_{\mathrm{SUR}}} \Delta t + \frac{2N-1}{N\sqrt{\theta}} o(\Delta t) \Delta t, \ \forall t \in [0, t_f], \tag{5.25}$$

where $\lim_{\Delta t \to 0} o(\Delta t) = 0$*.*

The detailed proof is similar to Proposition 5.3.1 and is presented in Appendix D.1. The upper bound in Proposition 5.3.2 indicates that the first term dominates the second term, and therefore increasing the multiplier factor for time steps in the SUR algorithm (C_{SUR}) significantly reduces the cumulative difference between binary and continuous controls if Assumptions 5.3.1–5.3.4 hold.

For a fixed number of time steps T, the optimal value of the continuous relaxation ($\tilde{F}(u^{\mathrm{con}})$) provides a lower bound for the binary model with the same time steps, but not necessarily for the binary model with time steps $C_{\mathrm{SUR}}T$. We provide a counter example in the following remark.

Remark 5.3.2. *We provide an example showing that the objective value of the continuous relaxation optimal solution $\tilde{F}(u^{con})$ is larger than the objective value of the binary solution $\tilde{F}(u^{bin})$ obtained by SUR with rounding time steps $C_{SUR}T$. We consider a quantum control problem with zero noises as follows. The objective function is defined as*

$$1 - \frac{1}{4}\left|\mathrm{tr}\left\{X_{targ}^{\dagger} X_T\right\}\right|, \tag{5.26}$$

where X_{targ} is

$$X_{targ} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{5.27}$$

We set intrinsic and control Hamiltonians as

$$H^{(0)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 \\ 0 & 2 & -1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}, \; H^{(1)} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \; H^{(2)} = \begin{pmatrix} 0 & 0 & -i & 0 \\ 0 & 0 & 0 & -i \\ i & 0 & 0 & 0 \\ 0 & i & 0 & 0 \end{pmatrix}. \tag{5.28}$$

The initial operator X_{init} is a 4-dimensional identical matrix. The evolution time $t_f = 8$. The number of time steps for continuous relaxation $T = 1$ and the number of time steps for SUR $T_R = C_{\mathrm{SUR}}T = 100$. We consider only one scenario $S = 1$ and uncertainty $\xi_{jk} = 0$ for all $j = 0, 1, 2$ and $k = 1$. The time-dependent Hamiltonians H_k^s are computed by (5.1b) *and the time-dependent operators X_k^s are computed by* (5.1c)*. By solving the model, we show that $\tilde{F}(u^{bin}) = 0.673 < 0.678 = \tilde{F}(u^{con})$.*

5.4 Numerical Studies

We apply the algorithms discussed in Section 5.3 to solve two quantum control examples with uncertain Hamiltonians, as an energy minimization problem and a circuit compilation problem. In Section 5.4.1, we introduce our simulation design for the uncertainty of Hamiltonians in quantum systems. In Section 5.4.2, we introduce the settings of the energy minimization problem and present the numerical results. In Section 5.4.3, we introduce the circuit compilation problem and describe the numerical results. All the numerical simulations were conducted on a macOS computer with 8 cores, 16GB RAM, and a 3.20GHz processor. The implementation was in Python with version 3.8. Our full code and results are available on our GitHub repository [63].

5.4.1 Uncertainty Design

Our proposed stochastic optimization model with uncertain Hamiltonians only focuses on control noises in quantum systems. In realistic experiments, the control noises of Hamiltonians vary among each simulation, each time step, and have different distribution parameters for different Hamiltonians. The variances of the uncertain parameters are larger across simulations and smaller within each time step during a single simulation. Specifically, we assume that for each controller j and each time step k, the uncertain parameter ξ_{jk} follows a normal distribution $\mathcal{N}(\mu, \sigma_j^{\text{time}})$ where μ is a random variable following a normal distribution $\mathcal{N}(0, \sigma_j^{\text{offset}})$, with constant variances $\sigma_j^{\text{time}}, \sigma_j^{\text{offset}}$ determined by Hamiltonians.

For each scenario $s = 1, \ldots, S$, we generate the corresponding samples as follows. We first sample a parameter μ_j^s from a normal distribution with mean value 0 and variance σ_j^{offset} for each Hamiltonian $H^{(j)},\ j = 0, \ldots, N$, representing the mean value of uncertainties for Hamiltonian $H^{(j)}$ among all the time steps under scenario s, defined as an offset. Then, we sample ξ_{jk}^s for each Hamiltonian $H^{(j)}$, for all $j = 0, \ldots, N$ and time step $k = 1, \ldots, T$ from a normal distribution with mean value μ_j^s and variance σ_j^{time}.

We show the values of 10 sampled scenarios of $\xi_{0k},\ \forall k = 1, \ldots, T$ in Figure 5.2. The different intercepts of lines reflect the variances of each simulation described by σ_0^{offset}, and the fluctuation of each line indicates the variances among time steps for each simulation, described by σ_0^{time}.

5.4.2 Energy Minimization Problem

We apply the algorithm L-BFGS-B to solve the stochastic optimization model of an energy minimization quantum control problem. The deterministic mathematical formulation of this problem is introduced in Section 3.2.1. We follow the parameter setting of the quantum system in Section 3.6.2.

We assume that $\sigma_j^{\text{time}} = 0.1\sigma_j^{\text{offset}}$ for control Hamiltonians $j = 1,\ 2$. We set the CVaR risk-level parameter $\eta = 0.05$, the number of qubits $q = 6$, the evolution time $t_f = 5$, the number of time steps for solving the continuous relaxation $T = 50$, and the number of time steps for rounding $T_R = 200$. We conduct out-of-sample tests for all controls under the same distribution as in-sample tests across 10 groups, each with 500 scenarios. We present the results of various numbers given by different scenario numbers, weight choices, and variance settings in Sections 5.4.2.1–5.4.2.3, respectively. We discuss the CPU time of solving the

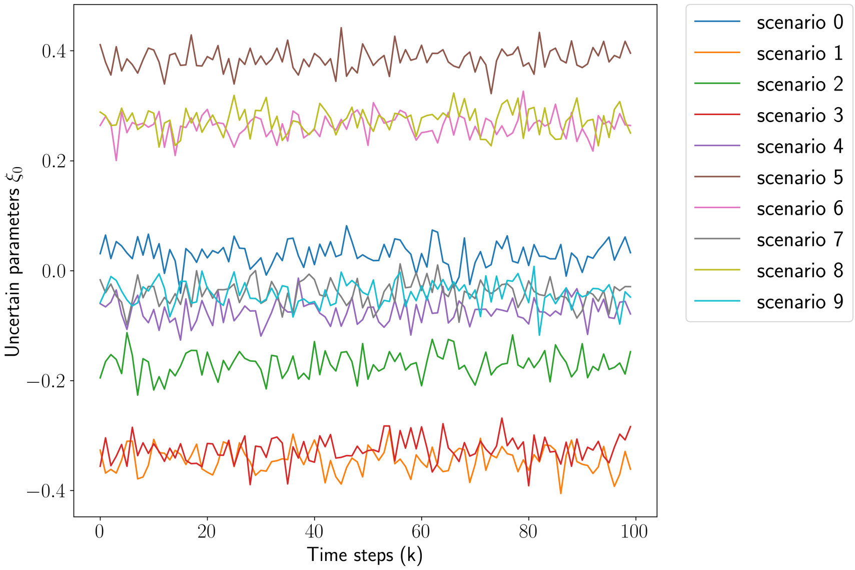


Figure 5.2: Sampled values of ξ_0 with 10 scenarios. The x-axis is time step $k = 1, \ldots, T$ and the y-axis is the value of ξ_0. The lines represent values of corresponding samples for each scenario $s = 1, \ldots, 10$.

problem with different sizes in Section 5.4.2.4.

5.4.2.1 Results of Scenarios

We set the weight parameter $\alpha = 0.5$, variance $\sigma_j^{\text{offset}} = 0.05$ for both controllers $j = 1,\ 2$, and solve the stochastic optimization model with in-sample scenarios $S = 1,\ 20,\ 100,\ 300,\ 500$. We present mean values, CVaR function values defined in (5.3), and weighted summation values of the mean and CVaR with weight $\alpha = 0.5$ in Table 5.1. The columns under "In-sample objective" represent the results among in-sample scenarios generated to solve the model. The columns under "Out-of-sample objective" represent the results across 5000 independently generated samples for evaluating different control solutions. The columns under "Gap" represent the gaps between in-sample and out-of-sample tests for all the function values. We show that generally, the gap of all the objective values decreases when the number of scenarios increases and the gap of the CVaR function value is higher than the mean value. For the remaining tests, we fix the in-sample size as $S = 300$ and keep the out-of-sample size as 5000 scenarios.

5.4.2.2 Results of Weight Parameter

In this section, we set the number of scenarios $S = 300$, variance $\sigma_j^{\text{offset}} = 0.05$, for $j = 1, 2$, and solve the stochastic optimization model with different weight parameters $\alpha =$

Table 5.1: Objective values result for in-sample and out-of-sample tests and their gaps with a different number of scenarios, including mean values ("Mean"), CVaR function values ("CVaR"), and weighted summation ("Total").

S	In-sample objective			Out-of-sample objective			Gap		
	Mean	CVaR	Total	Mean	CVaR	Total	Mean	CVaR	Total
1	0.038	0.038	0.038	0.107	0.357	0.232	64.16%	89.27%	83.48%
20	0.100	0.206	0.153	0.141	0.410	0.276	29.07%	49.82%	44.50%
100	0.105	0.290	0.198	0.104	0.326	0.215	−0.84%	10.96%	8.10%
300	0.102	0.286	0.194	0.108	0.309	0.208	5.67%	7.29%	6.87%
500	0.100	0.300	0.200	0.099	0.318	0.208	−0.90%	5.64%	4.08%

0, 0.25, 0.5, 0.75, 1. When $\alpha = 0$, the model only optimizes the CVaR function, as compared to when $\alpha = 1$, the model only optimizes the expected value of the random objective. We present how the values of mean and CVaR in out-of-sample tests vary depending on the weight parameter in Figure 5.3. The blue line marked by dots represents the mean and the orange line marked by triangles represents the CVaR. Furthermore, we present box plots describing the objective values of 5000 out-of-sample test scenarios for each weight parameter α. The red lines, the box edges, and the caps represent the medians, the first to the third quartiles, and the whiskers based on the interquartile range, respectively (see Wickham and Stryjewski [227] for details).

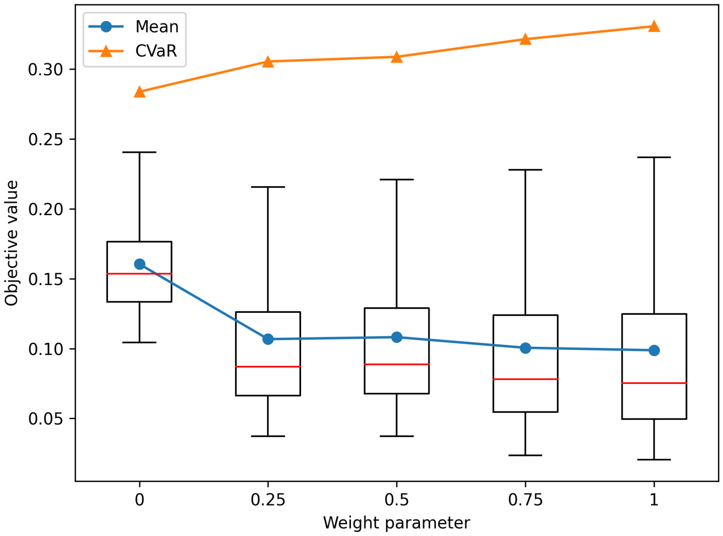
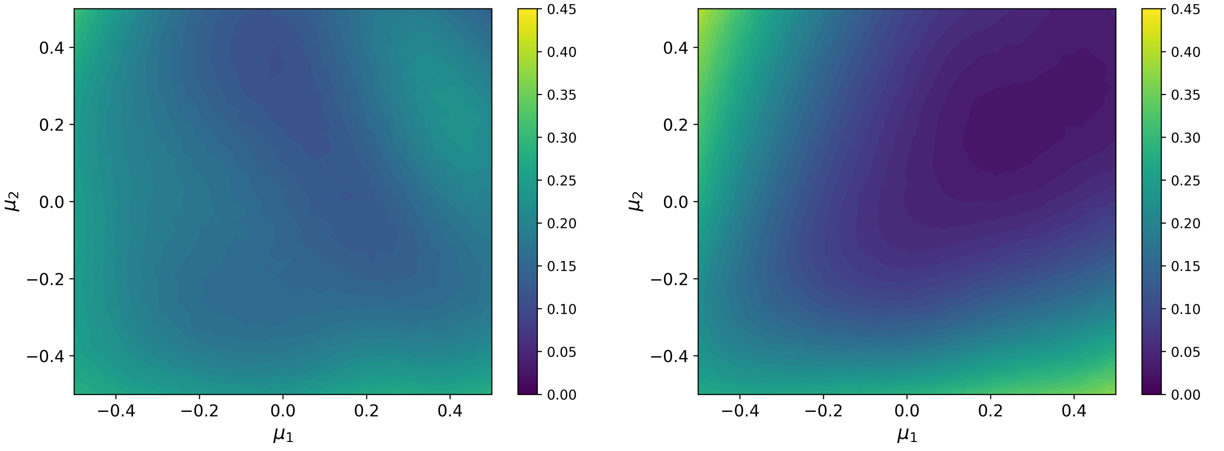


Figure 5.3: Figure of objective values in out-of-sample tests with multiple weight parameters α. The blue line marked by dots represents the mean value. The orange line marked by triangles represents the CVaR function value. Red lines, box edges, and caps represent medians, first and third quartiles, and whiskers [227]

We show that when α increases, the out-of-sample mean values decrease while the CVaR function values increase because the objective function assigns more weight to the expectation. Moreover, the box plots illustrate that decreasing α results in a reduced deviation, showing the advantages of incorporating risk aversion into the objective function.

In our numerical out-of-sample tests, we find that with the same offset μ_j^s for $s = 1, \ldots, S$, the standard deviations of the out-of-sample objective value with different ξ^s are always smaller than 0.005. Therefore, we focus on comparing the objective values with various offsets in our following discussion. Using a derived control u from a given weight parameter α, and offsets $\mu_1,\ \mu_2 \in [-0.5, 0.5]$, we generate 20 different scenarios for ξ following the normal distribution $\mathcal{N}(\mu_j, \sigma_j^{\text{time}}),\ j = 1, 2$ and compute the objective value $F_X(X_T(u;\xi))$. The average objective value is considered as the performance of control u under a specific simulation uncertainty offset (μ_1, μ_2). In Figure 5.4, we select the risk-averse case ($\alpha = 0$) and the risk-neutral case ($\alpha = 1$) to present the figures of average objective value among 20 scenarios for offsets $\mu_1,\ \mu_2 \in [-0.5, 0.5]$.



(a) Risk-averse ($\alpha = 0$)

(b) Risk-neutral ($\alpha = 1$)

Figure 5.4: Average objective values among samples of uncertainty ξ as a function of uncertainty offsets $\mu_1,\ \mu_2 \in [-0.5, 0.5]$. The control solutions are obtained from the stochastic optimization model with $\alpha = 0,\ 1$ and variance as 0.05.

We show that the control of the risk-neutral case attains a lower objective value when the uncertainty offsets are small, leading to better average performance. On the other hand, the control of the risk-neutral case has a significantly higher objective value when the uncertainty offsets are large, as shown in the upper-left and lower-right corners of the figure, while the control of the risk-averse case is more robust among all scenarios.

5.4.2.3 Results of Variance

Again, we fix the number of scenarios $S = 300$ and solve the stochastic optimization model with weight $\alpha = 0,\ 0.5,\ 1$ for different values of variances as $\sigma_1^{\text{offset}},\ \sigma_2^{\text{offset}} \in \{0.01,\ 0.05,\ 0.1\}$. We evaluate the derived control solution using three metrics: the mean value, the CVaR function value, and the success rate in distinguishing states in the energy minimization problem. In this instance, a control successfully distinguishes the first-excited state from the minimum energy state if its objective value is smaller than the energy difference ratio of these two states, which is one of our control design goals. The third metric is thus the percentage of scenarios in the out-of-sample test that achieve this distinction.

To compare the performance of stochastic optimization models with the deterministic model more straightforwardly, we compute the percentage of change in metrics as $(me_{SP} - me_D)/me_D$ where me_D and me_{SP} represent the evaluation metric value of the deterministic model and stochastic optimization model, respectively. For mean and CVaR, a negative percentage of change means lower energy consumption and better performance, while for the distinguished percentage, a positive percentage of change means better performance. We present the percentage of change in Table 5.2 and bold the best results of each variance setting. Columns with "$\alpha = 1$", "$\alpha = 0$", and "$\alpha = 0.5$" represent the results of the model optimizing the expectation, optimizing the CVaR function, and optimizing the weighted summation function.

For the mean value, we show that the results with $\alpha = 1$ always have better performance compared to the deterministic model, while the results with $\alpha = 0$ are worse because the model only focuses on the tail distribution. The balanced model with $\alpha = 0.5$ performs worse with low variance but better with high variance. For the CVaR function value, we show that models with all the weights have better results and the model with $\alpha = 0$ is the best, demonstrating an improvement in robustness when considering parameter uncertainty. For the distinguished percentage, we show that the model with $\alpha = 1$ and $\alpha = 0.5$ are both better than the deterministic model for all the variance settings, showing the benefits of our stochastic optimization model. The model with $\alpha = 0$ performs worse with high variance because it optimizes for scenarios with a high error ξ and sacrifices the performance in other scenarios.

Furthermore, we observe that with a fixed uncertainty offset variance for one controller, increasing the variance of the other controller leads to higher mean values, higher CVaR function values, and lower distinguished percentages, because the uncertainty in the quantum system increases. Increasing the uncertainty offset variance of the first controller $H^{(1)}$ has a larger negative impact on objective values compared to increasing the uncertainty variance of $H^{(2)}$, which means this quantum control system is more sensitive to the uncertainty of

Table 5.2: Percentage change compared to the deterministic model in mean values ("Mean"), CVaR function values ("CVaR"), and distinguished first-excited state percentage of different offset variances. The results include the model optimizing mean ("$\alpha = 0$"), CVaR function ("$\alpha = 1$"), and weighted summation of two functions ("$\alpha = 0.5$"). The in-sample and out-of-sample tests have the same distribution. The best results are bolded.

σ_1^{offset}	σ_2^{offset}	Mean			CVaR			Distinguished percentage		
		$\alpha = 1$	$\alpha = 0$	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 0$	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 0$	$\alpha = 0.5$
0.01	0.01	**−3.43%**	5.29%	−1.15%	−4.10%	**−5.24%**	−1.92%	**0.00%**	**0.00%**	**0.00%**
0.01	0.05	**−5.30%**	38.85%	0.22%	−4.90%	**−17.55%**	−14.95%	0.25%	**1.44%**	1.03%
0.01	0.1	**−8.85%**	80.32%	4.43%	−7.21%	**−26.08%**	−17.66%	1.68%	−9.27%	**4.20%**
0.05	0.01	**−8.65%**	25.35%	4.28%	−9.30%	**−29.15%**	−16.33%	0.95%	**2.57%**	1.53%
0.05	0.05	**−9.06%**	47.67%	−0.44%	−7.50%	**−20.61%**	−13.62%	2.04%	**3.08%**	2.89%
0.05	0.1	**−11.45%**	64.49%	−7.61%	−7.29%	**−19.86%**	−12.18%	3.71%	−24.40%	**4.20%**
0.1	0.01	**−11.66%**	38.76%	−2.02%	−8.96%	**−31.38%**	−18.00%	3.87%	7.97%	**6.00%**
0.1	0.05	**−12.12%**	33.77%	−5.13%	−7.87%	**−23.16%**	−12.37%	5.20%	0.74%	**6.38%**
0.1	0.1	**−14.26%**	55.23%	−5.45%	−6.87%	**−17.99%**	−10.52%	6.79%	−72.22%	**7.08%**

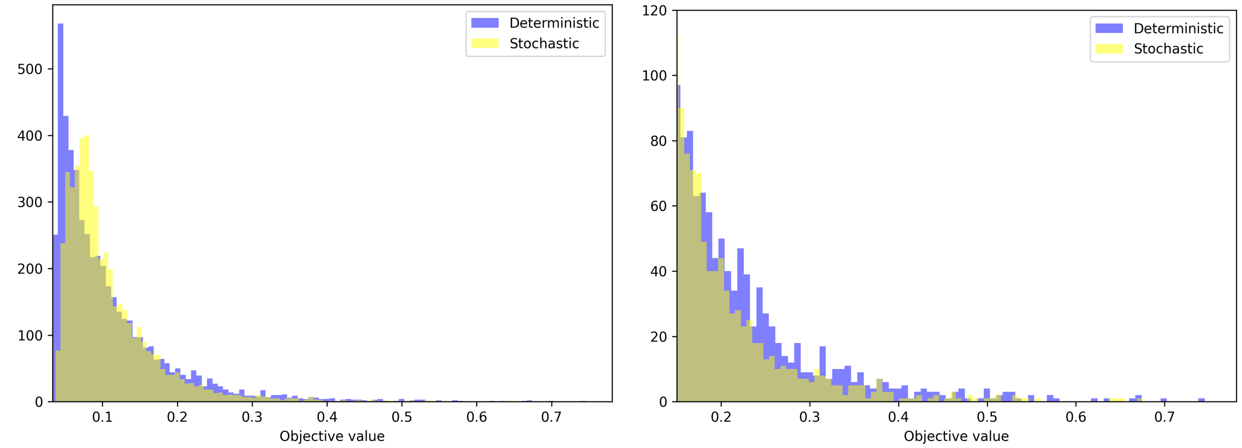
controller $H^{(1)}$.

We show the histogram of out-of-sample tests for both deterministic and stochastic optimization models and the zoomed-in tail distribution with variance 0.05 in Figure 5.5. The blue and yellow histograms represent the results of the deterministic and the stochastic optimization model, respectively. The figures show that our stochastic optimization model obtains a lighter tail distribution.

Similar to Figure 5.4, for each obtained control u, and for every combination of offsets value $\mu_1,\ \mu_2 \in [-1, 1]$, we generate 20 different scenarios for ξ with a normal distribution $\mathcal{N}(\mu_j, \sigma_j^{\text{time}}),\ j = 1, 2$ and compute the average objective value $F_X(X_T(u;\xi))$. The average objective value represents the performance of control u under a specific simulation uncertainty offset (μ_1, μ_2). In Figure 5.6, we present the average objective values for different offset values $\mu_1,\ \mu_2 \in [-1, 1]$ for u obtained from the deterministic and stochastic optimization models with both offset variances set as 0.1. We show that although both controls have high objective value when $|\mu|$ goes to 1, the control of the stochastic optimization model is more robust, especially for $(\mu_1,\ \mu_2) \in [-1, -0.75] \times [0.5, 1]$ and $(\mu_1,\ \mu_2) \in [0.5, 1] \times [-1, -0.75]$.

5.4.2.4 Results of CPU Time

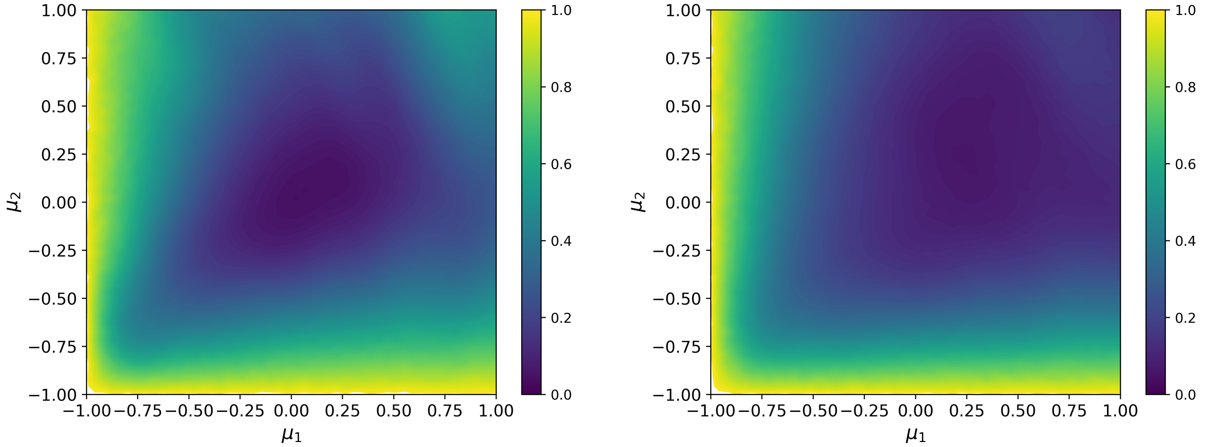
We fix the offset variances $\sigma_j^{\text{offset}} = 0.05,\ j = 1,\ 2$, the weight parameter $\alpha = 0.5$ and solve the stochastic optimization model with different numbers of qubits $q = 2,\ 6$, different numbers of time steps $T = 20,\ 50$, and different numbers of scenarios $S = 1,\ 20,\ 100,\ 200,\ 300$. We present the CPU time and the number of iterations for the algorithm L-BFGS-B in Table 5.3. We show that the number of qubits q has the most important impact on the



(a) All scenarios

(b) Tail distribution

Figure 5.5: Histograms of out-of-sample test for the deterministic and stochastic model with offset variances 0.05 for both controllers. Blue and yellow histograms represent the results of the deterministic and the stochastic optimization model. (a) The figure for all the scenarios. (b) Zoomed-in tail distribution.



(a) Deterministic

(b) Stochastic

Figure 5.6: Average objective values among samples of uncertainty ξ as a function of $\mu_1,\ \mu_2 \in [-1, 1]$. The control solutions are obtained from the deterministic and stochastic optimization model with $\alpha = 0.5$ and offset variances as 0.1.

Table 5.3: CPU time and iteration results of different problem sizes, including the number of qubits q, the number of time steps T, and the number of scenarios S.

q	T	S	CPU time (s)	Iteration
2	20	300	32.90	15
2	50	300	99.74	27
6	50	300	2814.33	26
6	50	200	971.06	14
6	50	100	401.20	19
6	50	20	121.18	15
6	50	1	21.66	34

CPU time because the dimension of Hamiltonian matrices grows exponentially with q. This issue can be potentially resolved by using quantum computers to conduct time evolution. An increasing number of scenarios S leads to an increase in CPU time, which can be reduced by parallel computing on multiple CPU cores of classical computers or multiple quantum computers. The CPU time also increases with the increase in the number of time steps T. Moreover, we notice that the number of iterations is robust regardless of the problem size.

5.4.3 Circuit Compilation Problem

Quantum circuit compilation aims to represent a circuit by specific controllers and constraints, to build a foundation for general quantum algorithms. In this section, we apply the modified Adam method (Algorithm 5.1) to study a compilation problem for the quantum circuit that has the ground state energy of molecules generated by the unitary coupled-cluster single-double method [15, 177]. The detailed mathematical formulation of the deterministic circuit compilation problem is introduced in Section 3.2.4. We follow the parameter setting of the quantum system in Section 3.6.2.

We assume that the variance of uncertainty among time steps $\sigma_j^{\text{time}} = 0.1\sigma_j^{\text{offset}}$ for all the control Hamiltonians. All the single-qubit control Hamiltonians have the same uncertainty offset variance, represented by σ_s^{offset}, and all two-qubit control Hamiltonians have the same uncertainty offset variance, represented by σ_t^{offset}. In Sections 5.4.3.1–5.4.3.2, we discuss the performance of the stochastic optimization model on an instance of the molecule H_2 (Dihydrogen). The system includes $q = 2$ qubits, 4 single-qubit controllers, and a two-qubit controller. We set the evolution time $t_f = 20$, number of time steps $T = 50$, and number of rounding time steps $T_R = 4000$, the risk level $\eta = 0.05$. We generate 10 groups, each with 500 scenarios sampled from the same distribution under in-sample tests to conduct out-of-sample tests for evaluating the obtained controls. In Section 5.4.3.3, we present the

CPU time of solving the circuit compilation problem with different molecules and problem sizes.

5.4.3.1 Results of Scenarios

In this section, we set the weight parameter $\alpha = 0.5$ and offset variances $\sigma_s^{\text{offset}} = \sigma_t^{\text{offset}} = 0.01$. We test our algorithm with a different number of scenarios $S = 20,\ 40,\ 80,$ and 160 with adjusted learning rates $0.05,\ 0.06,\ 0.08,$ and 0.15. To compare the performance under the same computational costs which is represented by the product of the number of scenarios and iterations ($S \times K$), we set the number of iterations to $2000,\ 1000,\ 500,$ and 250 accordingly. We show how the objective value varies with the computational costs during the algorithm procedure by a log-log scale in Figure 5.7. We show that with a larger number of scenarios, the objective value is more stable because the method learns more about the distribution at each iteration. However, the convergence is slower because the algorithm runs for fewer iterations.

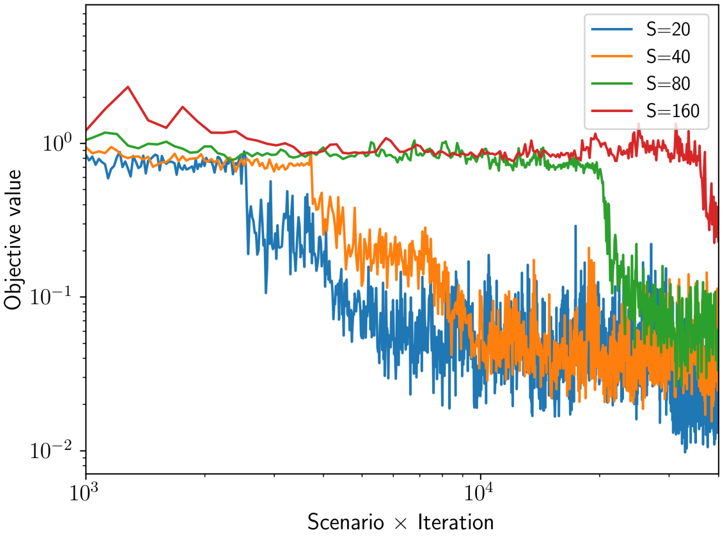


Figure 5.7: Log-log scale figure for the objective values during the in-sample test iterations. The x-axis represents the multiplication of the number of scenarios and iterations. Blue, orange, green, and red lines represent $S = 20$, 40, 80, and 160.

We present the out-of-sample test results for the controls obtained by a different number of scenarios, including the mean value, the CVaR function value, and the total objective value as weighted summation with $\alpha = 0.5$ in Table 5.4. We show that the control with $S = 20$ achieves the lowest objective value primarily because of its higher number of iterations within the same computational cost.

Table 5.4: Mean, CVaR function value, and total objective values ($\alpha = 0.5$) in out-of-sample tests for a different number of scenarios. The offset variances for both in-sample and out-of-sample tests and all the controllers are 0.01.

Scenario	Mean	CVaR	Total
20	8.19E−03	3.21E−02	2.02E−02
40	1.21E−02	4.30E−02	2.76E−02
80	1.35E−02	6.76E−02	4.06E−02
160	2.38E−02	9.51E−02	5.94E−02

5.4.3.2 Results of Variance

We compare the performance of the deterministic and the stochastic optimization model with sample size $S = 20$, weight parameter $\alpha = 0.5$ under different offset variances $\sigma_s^{\text{offset}} \in \{0.01, 0.05\}$, $\sigma_t^{\text{offset}} \in \{0.01, 0.05\}$. We present the mean value and the CVaR function value of the deterministic model (represented by "Deter") and the stochastic program (represented by "SP") for different variances of the uncertainty offsets in Table 5.5.

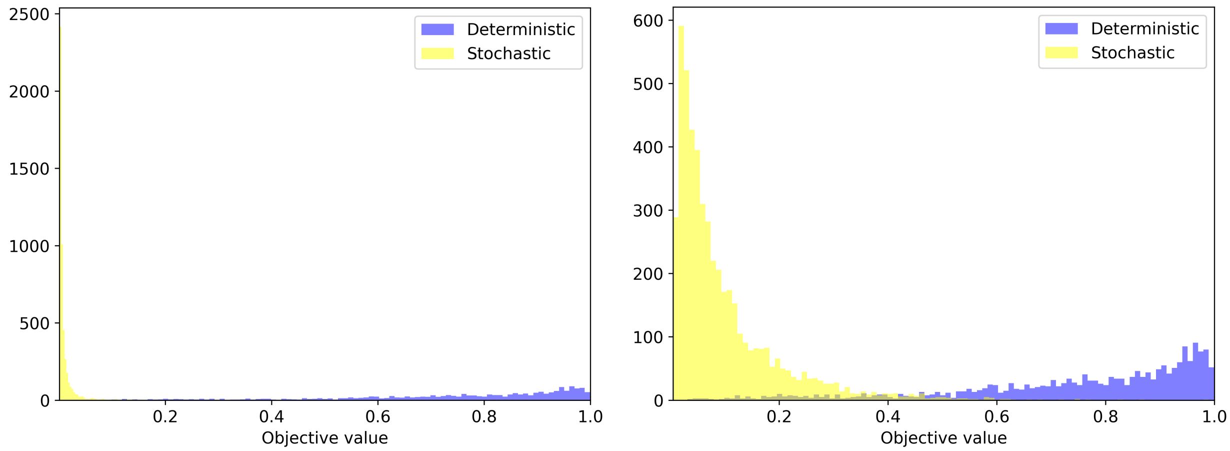
Table 5.5: Mean values and CVaR function values of different offset variances of single-qubit controllers (σ_s^{offset}) and two-qubit controllers (σ_t^{offset}) for the deterministic model ("Deter") and the stochastic program ("SP"). The in-sample and out-of-sample tests are under the same distribution. We bold the better results for each variance setting.

σ_s^{offset}	σ_t^{offset}	Mean		CVaR	
		Deter	SP	Deter	SP
0.01	0.01	0.639	**8.19E−03**	0.986	**3.21E−02**
0.01	0.05	0.639	**8.74E−03**	0.988	**4.32E−02**
0.05	0.01	0.748	**5.44E−02**	0.990	**0.353**
0.05	0.05	0.748	**9.84E−02**	0.990	**0.419**

Comparing the results of different variances, we show that the uncertainty in single-qubit controllers significantly affects the objective values more than the two-qubit controllers do, mainly because single-qubit controllers are expected to have more impact on unitary operators and they are the majority of controllers in the quantum system. For example, the instance of H_2 includes 4 single-qubit controllers but only 1 two-qubit controller. Moreover, increasing variance leads to a larger increase in the CVaR function value, indicating a larger negative impact on scenarios with large deviations.

We demonstrate that the control of the deterministic model performs badly even under a small variance, with all the mean values larger than 0.6 and all the CVaR function values larger than 0.9. On the other hand, the control of our stochastic optimization model performs dramatically better on the mean values and CVaR function values for all the settings, illustrating the advantages of our model considering the uncertainty in quantum control systems.

We present the histogram of out-of-sample tests for both deterministic and stochastic optimization models with variances for all the controllers as 0.01 and 0.05 in Figure 5.8. The blue and yellow histograms represent the results of the deterministic and the stochastic optimization model, respectively. We show that with the increase of variance, both models have heavier tail distribution, but the stochastic optimization model always has a much lighter tail distribution compared to the deterministic model.



(a) $\sigma_s^{\text{offset}} = \sigma_t^{\text{offset}} = 0.01$

(b) $\sigma_s^{\text{offset}} = \sigma_t^{\text{offset}} = 0.05$

Figure 5.8: Histograms of out-of-sample tests with offset variances 0.01 and 0.05 for all the controllers. Blue and yellow histograms represent the results of the deterministic and the stochastic optimization model.

5.4.3.3 Results of CPU Time

We set the iteration number for the modified Adam method at 2000, weight parameter $\alpha = 0.5$, and offset variances $\sigma_s^{\text{offset}} = \sigma_t^{\text{offset}} = 0.01$. We solve the stochastic optimization model for molecules H_2 and LiH, with time steps $T = 50,\ 100$, and scenario numbers $S = 20,\ 40$. In Table 5.6, we present the CPU time of the algorithm for different problem sizes and molecules, with the respective number of qubits q and controllers N. We show that with the same number of time steps and scenarios, changing molecules leads to a significant CPU time increase, mainly because the dimension of Hamiltonian matrices increases exponentially with q and the number of controllers also increases. The CPU time increases with time steps T and scenario numbers S approximately linearly. In practice, the CPU time can

Table 5.6: CPU time results of different molecules, different numbers of times steps T, and different numbers of scenarios S. We present the number of qubits q and the number of controllers N for molecules.

Molecule	q	N	T	S	CPU time (s)
H_2	2	5	50	20	1383.62
	2	5	50	40	2713.80
	2	5	100	20	2888.77
LiH	4	12	50	20	4168.16
	4	12	50	40	7918.20
	4	12	100	20	8482.01

be potentially reduced by conducting time evolution on quantum computers and parallel computing among different scenarios on large amounts of CPU cores.

5.5 Concluding Remarks

In this chapter, we built a stochastic mixed-integer program with the sample-based reformulation for the quantum optimal control problem with uncertain Hamiltonians. We introduced an objective function aiming to balance risk-neutral and risk-averse measurements, which are evaluated by expectation and CVaR function, respectively. We derived a closed-form expression and discussed the derivative for the objective function. We modified and applied two gradient-based methods to solve the continuous relaxation and obtained binary solutions by the sum-up-rounding technique with a discussion of the rounding errors.

We conducted numerical simulations on multiple quantum control instances. Based on the results, we recommend the L-BFGS-B method for quantum control problems minimizing system energy and the modified Adam method for problems minimizing infidelity. The results show that our stochastic optimization model outperforms the deterministic model in terms of both average and robust performance for different variance levels.

With all the simulations completed on classical computers, we find that the number of qubits in quantum systems has a significant impact on the computational time. Conducting time-evolution processes on quantum computers to reduce computational time is an interesting direction for future research. Furthermore, model-free optimization methods, including reinforcement learning, provide chances to capture more complex uncertainties in quantum systems.

CHAPTER 6

Conclusion

In this dissertation, we focus on developing optimization methods for mixed-integer control problems in systems with complex physical constraints and parameter uncertainties. Specifically, we develop new deterministic and stochastic optimization models to formulate control problems in traffic signal systems and quantum systems and design innovative algorithms based on widely recognized optimization methods to solve these models.

In Chapter 2, we develop a Cell Transmission Model (CTM) based mixed-integer program for a network-level traffic signal control problem, which is further extended to a two-stage stochastic optimization model to account for uncertain traffic conditions. We design a distributed algorithm that overcomes scalability difficulties and ensures optimality in a non-convex model with mixed-integer variables. Our numerical results from synthetic and real-world scenarios demonstrate an improvement in computational time and highlight the benefits of considering stochastic traffic demand and coordination among intersections.

In Chapter 3, we build a generic model for the binary quantum pulse control problem, introducing a penalized squared L_2 function and a Total Variation (TV) regularizer to ensure a single active control and minimize control changes, respectively. We propose an algorithmic framework combining the gradient ascent pulse engineering (GRAPE) approach, combinatorial integral approximation, and local-branching improvement heuristic. Through multiple numerical simulations, we indicate that our algorithm successfully obtains high-quality controls with fewer switches. In Chapter 4, we continue the research on the binary quantum control problem and construct an innovative framework optimizing both control functions and switching times of controllers. We derive new heuristic methods for obtaining controller sequences with a time-evolution simulation acceleration technique, and then build and solve a switching time optimization model. Our numerical studies show a notable enhancement in balancing control quality and switching frequency compared to solutions of the time discretization model within significantly reduced computational time.

In Chapter 5, we develop a stochastic mixed-integer program with a sample-based reformulation for the binary quantum control problem with time-varying noise in Hamiltonians.

The model aims to balance between risk-neutral and risk-averse measurements, evaluated by expectation and CVaR function, respectively. We modify and apply gradient-based methods and sum-up-rounding techniques to solve the model with theoretical discussions on differentiability and rounding errors. By numerical simulations, we demonstrate that our stochastic optimization model outperforms the deterministic model in both average and robust performance under different noise levels.

There are several possible directions for future research. We anticipate exploring the potential of combining optimization models and learning techniques to solve mixed-integer control problems with physical constraints effectively. Additionally, we are interested in applying our binary quantum pulse control optimization methods to improve variational quantum algorithms. Moreover, we plan to extend the optimization methods to different application areas such as resource planning and route recommendation, offering practical solutions to more real-world challenges.

APPENDIX A

Appendix For Chapter 2

A.1 Benders Decomposition Algorithm

We first create new variables $\theta^k,\ k = 1, \ldots, K$ and define a relaxed master problem as follows.

$$
\begin{aligned}
(\mathbf{RMP}) \quad \min \quad & \sum_{k=1}^{K} p^k \theta^k && \text{(A.1a)} \\
\text{s.t.} \quad & (l, o, g, b, e, z_1, z_2) \in X, \\
& (z_1, z_2, \theta) \in \Sigma(z_1, z_2, \theta). && \text{(A.1b)}
\end{aligned}
$$

Here, $\Sigma(z_1, z_2, \theta)$ is the set of Benders cuts as linear functions of z_1, z_2 generated up to the current iteration from sample-based subproblems:

$$
\begin{aligned}
(\mathbf{SP}^k) \quad \mathcal{Q}^k(z_1, z_2) = \min \quad & -\sum_{c\in\mathcal{D}} \sum_{t=1}^{T} n^k_{ct} + \alpha \sum_{c\in\mathcal{C}} \sum_{t=1}^{T} (T-t) y^k_{ct} && \text{(A.2a)} \\
\text{s.t.} \quad & (y^k, n^k) \in \text{Constraints (2.2b)–(2.2m) for each } k. && \text{(A.2b)}
\end{aligned}
$$

Given first-stage integer solutions $\hat{z}_1, \hat{z}_2$, one can solve $(\mathbf{SP}^k)$ with $z_1 = \hat{z}_1,\ z_2 = \hat{z}_2$ and obtain $\mathcal{Q}^k(\hat{z}_1, \hat{z}_2)$, for each $k = 1, \ldots, K$. The signal constraints (2.1a)–(2.1j) in the first stage ensure that for each $i \in \mathcal{R},\ j \in \mathcal{F}_i,\ m = 1, \ldots, N_{\text{cy}}$ and $t = 1, \ldots, T$, we have $\hat{z}_{1ijmt} + \hat{z}_{2ijmt} \geq 1$. Given non-negative parameters $Q, N, D, n^{\text{init}}, W$ and positive parameter ϵ, each subproblem $(\mathbf{SP}^k)$ always has at least one feasible solution $y^k = n^k = \mathbf{0}$ for any given $\hat{z}_1, \hat{z}_2$. Therefore, we only need to add optimality cuts to the set Σ.

We next derive optimality cuts from the dual formulations of the second-stage linear programs $(\mathbf{SP}^k)$. Associate dual variables $\rho^k_{ct},\ \sigma^k_{ct},\ \pi^k_{cc't},\ \gamma^k_{cc't},\ \delta^k_{ct},\ \tau^k_c$ to constraints (2.2b), (2.2c)–(2.2d), (2.2e)–(2.2f), (2.2g)–(2.2h), (2.2i)–(2.2k), and (2.2l), respectively. Following weak duality, for each $k = 1, \dots, K$, the optimality cut is of the form:

$$\begin{aligned}\theta^k \geq & \sum_{c \in \mathcal{C}/\mathcal{I}} \sum_{t=1}^{T} Q_{ct}\sigma_{ct}^k + \sum_{i \in \mathcal{R}} \sum_{j \in \mathcal{F}_i} \sum_{c \in \mathcal{I}_{ij}} \sum_{m=1}^{N_{cy}} \sum_{t=1}^{T} (z_{1ijmt} + z_{2ijmt} - 1) Q_{ct}\sigma_{ct}^k \\ & + \sum_{c \in \mathcal{C}} \sum_{c' \in d(c)} \sum_{t=1}^{T} Q_{c't}\pi_{cc't}^k + \sum_{c \in \mathcal{C}} \sum_{c' \in d(c)} \sum_{t=1}^{T} W_{c't} N_{c't} \gamma_{cc't}^k \\ & + \sum_{c \in \mathcal{O}} \sum_{t=1}^{T} D_{ct}^k \delta_{ct}^k + \sum_{c \in \mathcal{C}} n_c^{\text{init}} \tau_c^k. \end{aligned} \tag{A.3}$$

The generic Benders procedures are presented in Algorithm A.1. In every iteration, we solve (**RMP**) with current Benders' cuts and obtain feasible solutions $\hat{z}_1, \hat{z}_2$. The optimal value of (**RMP**) provides a lower bound to the original stochastic MIP (2.2). We then compute optimal dual solutions to the second-stage subproblems (**SP**^{k}) for each $k = 1, \dots, K$, and the expectation of their optimal values provides an upper bound of the overall optimal objective value. We terminate the algorithm when the gap between the upper bound and the lower bound is sufficiently small.

Algorithm A.1: A generic Benders Decomposition approach for solving Model (2.2).

```
Initialize (RMP) with with $\Sigma = \emptyset$.
while the termination criteria is not satisfied do
    Solve (RMP) to obtain optimal solutions $(\hat{z}_1, \hat{z}_2, \hat{\theta})$.
    for $k = 1, \dots, K$ do
        Solve (SP$^k$) to obtain optimal dual solutions. if $\hat{\theta}^k < \mathcal{Q}^k(\hat{z}_1, \hat{z}_2)$ then
            Add an optimality cut (A.3) to the set $\Sigma(z_1, z_2, \theta)$.
        end
    end
end
Return the objective value as $\sum_{k=1}^{K} p^k \hat{\theta}^k$ and the solutions of (RMP).
```

A.2 Additional Numerical Results

We present additional numerical results in this appendix section.

A.2.1 Sensitivity Analysis

Cycle length We solve the deterministic model with various cycle lengths $\hat{l}$ =19, 24, 29, 34, 39, where for all the intersections $i \in \mathcal{R}$, $\hat{l}_i = \hat{l}$. We present the average delay and throughput under the out-of-sample test with 500 scenarios in Table A.1.

Table A.1: Delay and total throughput of different cycle lengths.

Cycle length ($\hat{l}$)	19	**24**	29	34	39
In-sample Obj	-1507598.54	-1498923.98	-1490859.43	-1479932.66	-1427223.55
Average Delay	286.34	297.40	293.04	303.72	302.76
Total Throughput	3881.38	3789.91	3848.80	3773.03	3820.57

We show that for all the metrics, the gap among different cycle lengths is no larger than 5%, which demonstrates that the performance of signal timing plans is not sensitive to the cycle length. In our following numerical experiments, we follow the rule in Remark 2.4.3 and choose cycle length as $\hat{l}_i = 24,\ \forall i \in \mathcal{R}$. In real-world applications, one can test the model with multiple selected cycle lengths and choose the one with the best performance.

In-sample scenarios We choose in-sample scenarios $K = 5,\ 10,\ 15,\ 20$ and solve the stochastic MIP. We present the in-sample and out-of-sample objective values as well as the gaps between them. We test the signal timing plans on the same out-of-sample test set with 500 scenarios and present the results in Table A.2.

Table A.2: Objective value and evaluation metrics of different in-sample scenarios k.

In-sample Scenarios	5	10	15	20
In-sample Obj	-1493833.50	-1540021.95	-1462882.47	-1535876.09
Out-of-sample Obj	-1925990.42	-2023592.97	-1992383.23	-1929821.75
Gap	22.44%	23.90%	26.58%	20.41%
Average Delay	298.96	276.33	283.97	297.95
Total Throughput	3811.87	4009.33	3931.91	3817.81

We show that when the total number of scenarios is 20, the gaps between in-sample and out-of-sample tests decrease slightly. However, the signal timing plan given by the sample with $K = 10$ scenarios has the best performance in out-of-sample tests. The main reason is that we terminate early in solving second-stage problems by ADMM allowing at most 20% gap between the obtained objective value and the optimal one. Also, adding in-sample scenarios increases the difficulty of solving the problem, making solutions of $K = 15,\ 20$ not as good as $K = 10$ with the same number of Benders iterations. To obtain the balance between computation and results, we use $K = 10$ as our in-sample scenario size in our later numerical studies.

Out-of-sample scenarios We vary the number of out-of-sample test scenarios among 500, 1000, 1500, 2000 and evaluate the performance of the signal timing plan obtained by the solution of $K = 10$ instance. We present the out-of-sample objective values, gaps between in-sample and out-of-sample objectives, average delay, total arrival, and the total throughput results in Table A.3.

Table A.3: Objective value and evaluation metrics of different out-of-sample scenarios.

Out-of-sample Scenarios	500	1000	1500	2000
Out-of-sample Obj	−2023592.97	−2021604.03	−2024107.83	−2024080.06
Gap	23.90%	23.82%	23.92%	23.91%
Average Delay	276.33	274.40	275.09	275.61
Total Arrival	7956.79	7922.89	7932.95	7942.05
Total Throughput	4009.33	4016.07	4020.70	4020.16

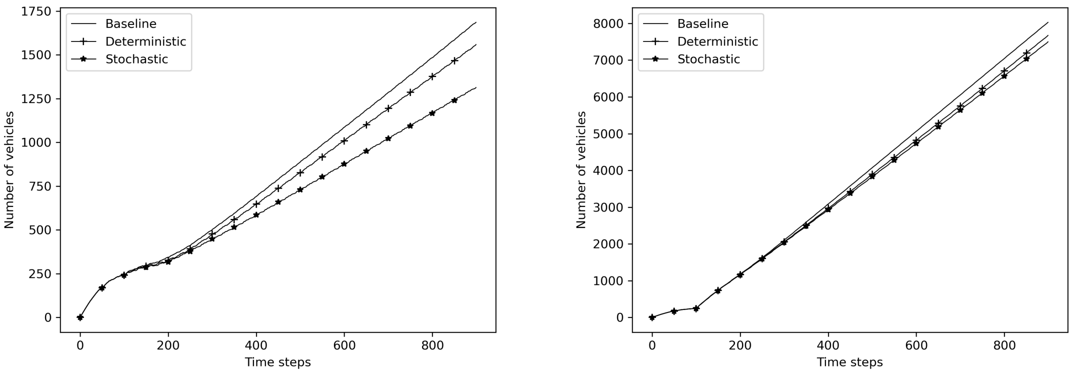
Note that all the results are similar regardless of the choice of out-of-sample scenario size. We use 500 scenarios for our later computation in all out-of-sample tests.

A.2.2 Number of Vehicles in Real-world Traffic Networks

Results of Morning Peak Hours To evaluate the traffic signal plans, in figure A.1, we visualize the number of vehicles in the network of the scenarios with the lowest and highest arrivals. The figure shows that in both scenarios, the stochastic model outperforms the deterministic model and the baseline, and it is more significant when the arrival is lower.

In Figure A.2, we visualize the number of vehicles in the traffic network under the scenarios with minimum delay and maximum delay. For both scenarios, the number of vehicles keeps increasing. The increasing rate of the stochastic model is lower under the scenario with minimum delay while the increase rate of the deterministic model is lower under the scenario with maximum delay. The increase rate of the stochastic model is higher than the deterministic model because the total arrival of the stochastic model is larger.

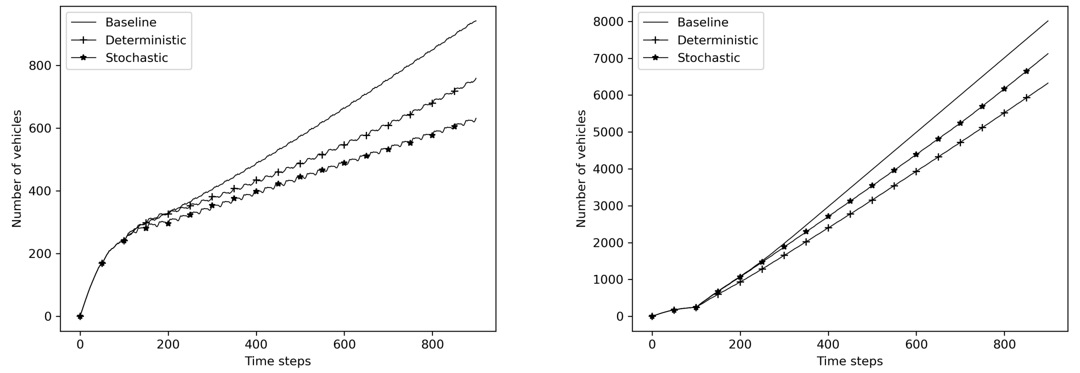
In Figure A.3, we present the spatial distribution of intersections compared between the stochastic model and the baseline in the morning peak hours, where the delay of intersections marked by green color is reduced or unchanged and the delay of others is increased. We show that for every corridor, the delay of the majority of intersections is reduced or unchanged. The intersections with increased delay are mostly in the upper-left corner and around a parking lot.



(a) Scenario with lowest arrival

(b) Scenario with highest arrival

Figure A.1: Number of vehicles in the traffic network during morning peak hours



(a) Scenario with minimum delay

(b) Scenario with maximum delay

Figure A.2: Number of vehicles in the traffic network during morning peak hours

Results of Off-peak Hours In Figure A.4, we visualize the number of vehicles in the network of the scenarios with the lowest and highest arrivals. Figure A.4a shows that when the arrival is low, the number of vehicles in the traffic work keeps stable for both deterministic and stochastic models. Figure A.4b shows that when the arrival is high, the increase rates of the number of vehicles of the stochastic model is slower than the other two settings.

In Figure A.5, we present the spatial distribution of intersections compared between the stochastic model and the baseline in the traffic network of off-peak hours. The intersections with reduced delay are marked by green color. We show that for every corridor, the delay of the majority of intersections is reduced or unchanged. The intersections with increased delay are mostly in the lower-left corner and around a parking lot.

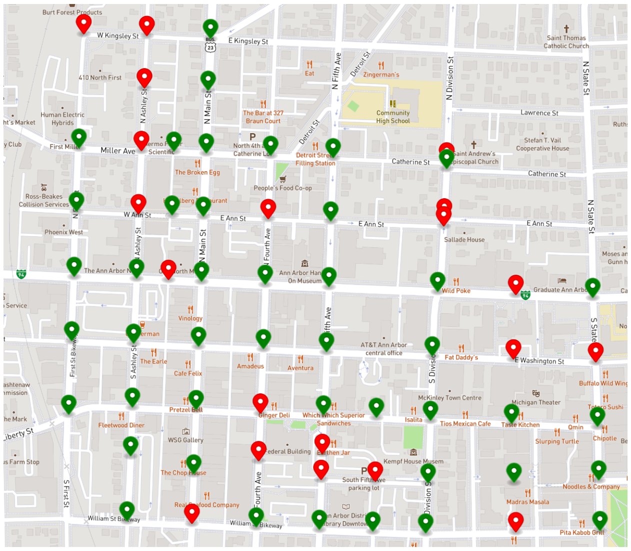
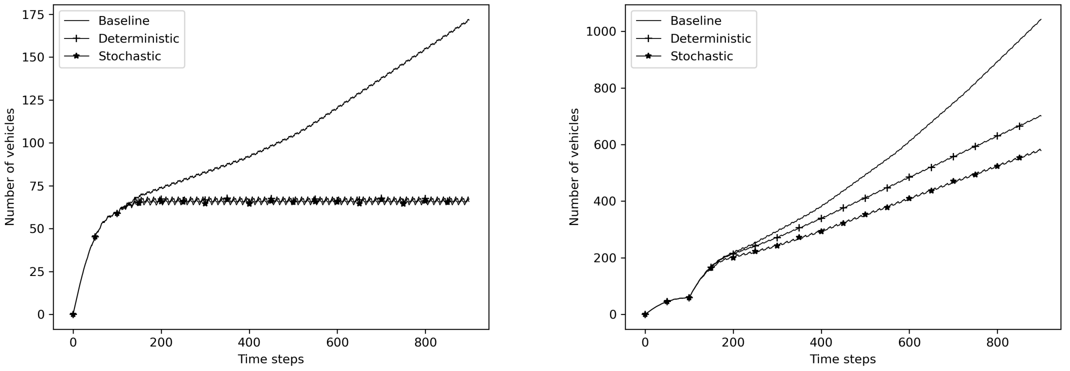


Figure A.3: Spatial distribution of intersections under the stochastic model solution with delay improved or unchanged compared to the baseline in the traffic network during morning peak hours, marked by green color in the figure.



(a) Scenario with lowest arrival

(b) Scenario with highest arrival

Figure A.4: Number of vehicles in the traffic network during off-peak hours

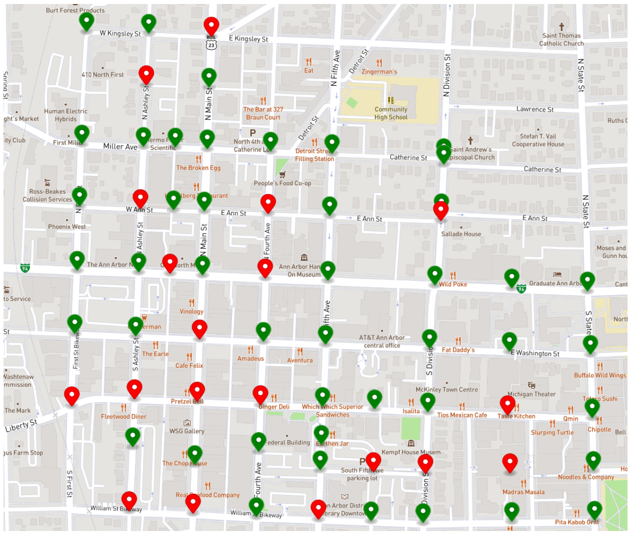


Figure A.5: Spatial distribution of intersections under the stochastic model solution with delay improved or unchanged compared to the baseline in the traffic network during off-peak hours, marked by green color in the figure.

A.2.3 Detailed Results of Afternoon Peak Hours

We present the computational time of the deterministic and stochastic models in seconds in Table A.4. We show that our distributed algorithm solves the problem within an acceptable time limit. The computational time of master problems is slightly higher but the time of subproblems is lower compared to the morning peak hours instances.

Table A.4: CPU time results of the traffic network of Downtown Ann Arbor during afternoon peak hours

	MP-min (s)	MP-max (s)	MP-A (s)	SP-min(s)	SP-max (s)	SP-A (s)
Deterministic	0.23	0.94	0.50	37.50	37.81	37.67
Stochastic	1.18	3.17	2.08	378.51	393.16	386.89

We present the in-sample and out-of-sample objective values as well as their gaps of the deterministic and stochastic model in Table A.5. We show that the gap of the stochastic model is slightly smaller than the deterministic model because it has a better realization of the uncertainties. Both gaps are similar to the results of morning peak hours because they have the same deviation. Furthermore, we present the average delay and total throughput to evaluate the signal controls. In Row "Deterministic", we also present the improvement of the deterministic model compared to the baseline. Similarly, in Row "Stochastic", we present the improvement of the stochastic model compared to the deterministic model. We demonstrate that our deterministic and stochastic models both outperform dramatically compared to

Table A.5: Out-of-sample evaluation results of Downtown Ann Arbor during afternoon peak hours

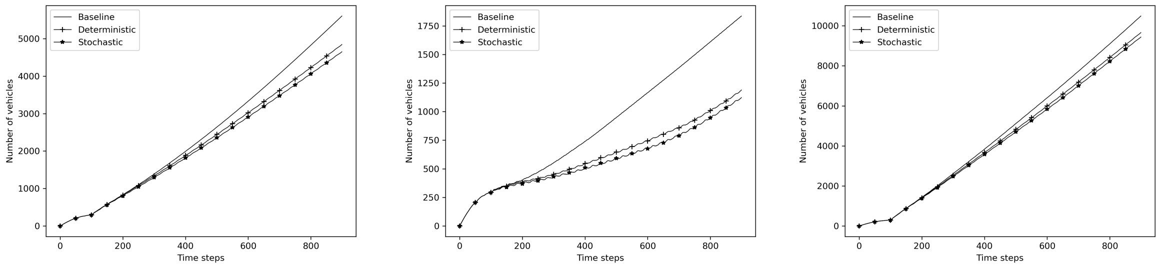
	In-sample Obj (veh·s)	Out-of-sample Obj (veh·s)	Gap
Deterministic	−1510461.85	−2082966.89	27.49%
Stochastic	−1602873.23	−2177611.47	26.39%
	Average Delay (s)	Total Arrival (veh)	Total Throughput (veh)
Baseline	425.93	9447.04	3029.47
Deterministic	344.58 (23.61%)	9447.04	3961.59 (30.77%)
Stochastic	323.25 (6.60%)	9447.04	4179.38 (5.50%)

the baseline, indicating that our centralized model obtains better network coordination of intersections. The stochastic model obtains a smaller average travel delay and larger total throughput, showing the benefits of considering uncertainties.

Compared to the results of morning peak hours, the impact of network coordination is more significant, demonstrated by the higher improvement when being compared with the baseline case. In particular, the improvement of total throughput is larger than morning peak hours and off-peak hours, mainly because the traffic flow is highest during afternoon peak hours. On the other hand, the improvement of the stochastic model compared to the deterministic model is smaller. The main reason is that under extremely high traffic flow, the optimization space for traffic signals is small so the difference between the results of the two models decreases.

We visualize the average number of vehicles in the traffic network for all the scenarios in Figure A.6a. We show that although the number of vehicles increases with time for all the methods because of high traffic arrival, the stochastic model has the slowest increase while the baseline has the fastest increase. The increase rate of afternoon peak hours is higher than the two other TODs because the traffic arrival is higher. Furthermore, we visualize the number of vehicles in the network under the scenario with the lowest and highest traffic arrival in Figure A.6b–A.6c. We show that for both scenarios, the stochastic model outperforms the deterministic model and the improvement is more significant under the scenario with the lowest arrival because, with higher traffic arrival, the intersections are more likely to be saturated under all the signal timing plans, causing an increase of the number of vehicles in the network.

We provide the snapshots of the spatial distribution of the number of vehicles of the deterministic and stochastic models under the best and worst scenario at time step $t = 800$ in Figure A.7. We show that even under the best scenario, traffic congestion exists on two main corridors, Main Street and Huron Street. The stochastic model obtains significantly better performance on main corridors but still causes traffic congestion on a few local roads with the direction of north and south. For the worst scenario, traffic congestion in the stochastic



(a) Average number

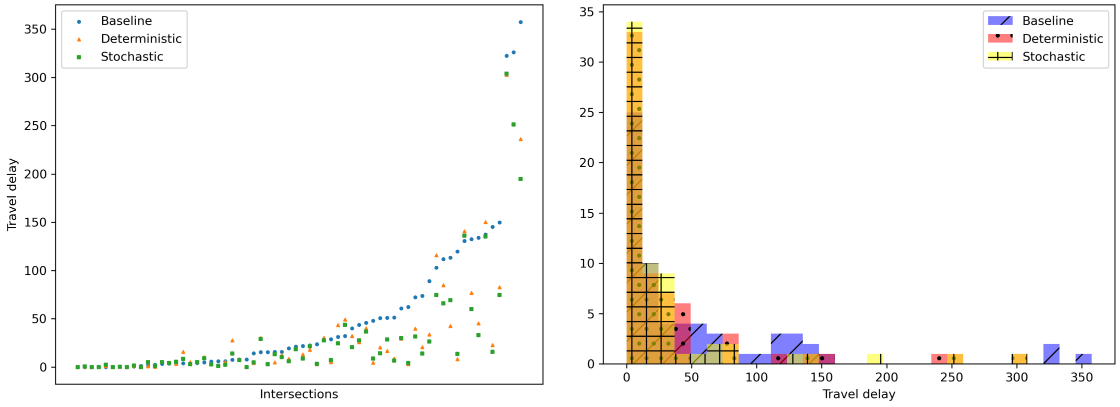
(b) Lowest arrival

(c) Highest arrival

Figure A.6: Number of vehicles in the Downtown Ann Arbor traffic network during afternoon peak hours. Left: Average number of all the scenarios. Middle: Number under best scenario. Right: Number under worst scenariosenne

model is mostly in the direction of north and south direction while traffic congestion in the deterministic model is in both directions.

The average traffic delay of each intersection is presented in Figure A.8a and the histograms of all intersections are presented in Figure A.8b.

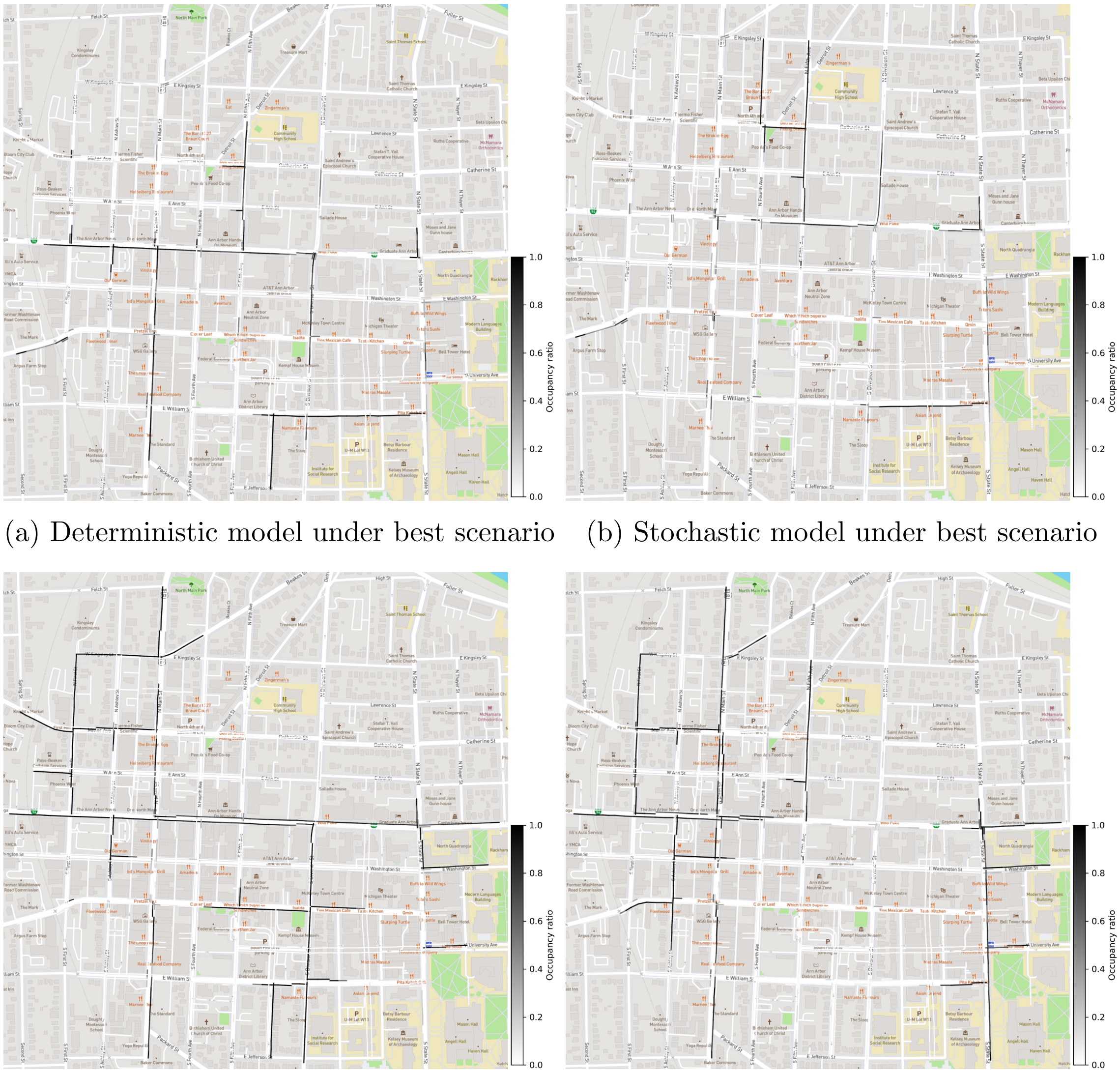


(a) Delay of all the methods.

(b) Histogram statistics of the delay.

Figure A.8: Average travel delay for all the intersections during off-peak hours.

Compared to the baseline, the delays at 69% of the intersections are reduced or unchanged by the deterministic model, and at 75% of all the intersections are reduced or unchanged by the stochastic model. All the percentages are higher than the other two instances, showing that our model is able to reduce the congestion in more intersections with higher source demand. Compared to the baseline, our centralized model obtains a lower maximum delay. The stochastic model obtains the highest percentage of intersections in the minimum delay



(a) Deterministic model under best scenario (b) Stochastic model under best scenario

(c) Deterministic model under worst scenario (d) Stochastic model under worst scenario

Figure A.7: Spatial distribution of vehicles at time step 800 during afternoon peak hours.

group, which is less than 12 seconds during afternoon off-peak hours, showing the advantages of considering network coordination and traffic uncertainties.

In Figure A.9, we present the spatial distribution of intersections compared between the stochastic model and the baseline in the traffic network of afternoon peak hours. The intersections with reduced delay are marked by green color and the others are marked by red color. We show that for every corridor, the delay of the majority of intersections is reduced or unchanged.

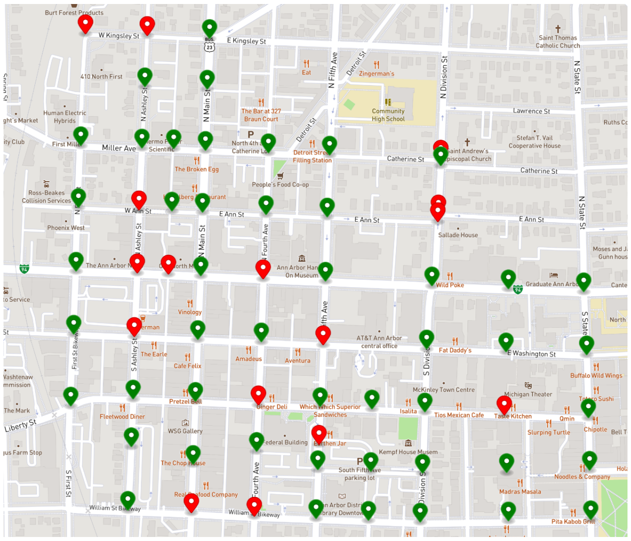


Figure A.9: Spatial distribution of intersections under the stochastic model solution with delay improved or unchanged compared to the baseline in the traffic network during afternoon peak hours, marked by green color in the figure.

APPENDIX B

Appendix For Chapter 3

B.1 Proofs of Results in Section 3.3

In this section, we provide proof of the results in Section 3.3.

B.1.1 Proofs of Results in Section 3.3.2

Proof of Theorem 3.3.2. Because F is upper bounded by constant C_F, we have $|F(u)| \leq C_F$ for all the u. We prove that for any ρ and T, $|\rho l(u^*_\rho, T)| \leq 2C_F$. Assume that there exists an optimal solution $u^{(1)}$ of (DQCP-L_2) such that $|\rho l(u^{(1)}, T)| > 2C_F$, then we have $F(u^{(1)}) + \rho l(u^{(1)}, T) > C_F$. This is a contradiction to the definition of $u^{(1)}$ because we can find a feasible solution $u^{(2)}$ of (DQCP) such that $F(u^{(2)}) + \rho l_(u^{(2)}, T) = F(u^{(2)}) \leq C_F$. Therefore we drive that $|l(u^*_\rho, T)| \leq 2C_F/\rho$, which means that $l(u^*_\rho, T) \sim O(1/\rho)$. □

B.1.2 Proofs of Results in Section 3.3.3

Proof of Proposition 3.3.1. Because the SOS1 property holds for binary control u^b, we have

$$\begin{aligned}
\max_{k=1,\dots,T} \left\| \sum_{\tau=1}^{k} (u^c_\tau - u^b_\tau)\Delta t \right\|_\infty &= \max_{k=1,\dots,T} \max_{j=1,\dots,N} \left| \sum_{\tau=1}^{k} (u^c_{j\tau} - u^b_{j\tau})\Delta t \right| \\
&\geq \max_{k=1,\dots,T} \frac{1}{N} \left| \sum_{j=1}^{N} \sum_{\tau=1}^{k} (u^c_{j\tau} - u^b_{j\tau})\Delta t \right| \\
&= \max_{k=1,\dots,T} \frac{1}{N} \left| \sum_{\tau=1}^{k} \left(\sum_{j=1}^{N} u^c_{j\tau} - 1 \right) \Delta t \right| = \frac{1}{N}\epsilon(\Delta t). \qquad \text{(B.1)}
\end{aligned}$$

□

Proof of Theorem 3.3.3. For simplicity, we use ϵ to denote $\epsilon(\Delta t)$. It is obvious that for any time step $k = 1, \dots, T$, exactly one control can be set to 1 by u^b by the construction. Hence

u^b satisfies the SOS1 property. For a specific time step $k = 1, \ldots, T$, define

$$i_k = \operatorname*{arg\,max}_{j=1,\ldots,N} \left| \sum_{\tau=1}^{k} u^c_{j\tau} \Delta t - \sum_{\tau=1}^{k} u^b_{j\tau} \Delta t \right|. \tag{B.2}$$

We assume that there exists a time step r such that the claim does not hold. For simplicity, we use i to represent the corresponding i_r. We consider the assumption in two cases.

Case 1: We assume that,

$$\sum_{\tau=1}^{r} u^c_{i\tau} \Delta t - \sum_{\tau=1}^{r} u^b_{i\tau} \Delta t < -\left(N-1\right)\Delta t - \frac{2N-1}{N}\epsilon. \tag{B.3}$$

Let $\hat{k}$ be the highest index of the time step in which control i is rounded up,

$$\hat{k} = \operatorname*{arg\,max}_{1 \le k \le r} \left\{ k \mid u^b_{ik} = 1 \right\}. \tag{B.4}$$

Then by (B.3), we have,

$$\sum_{\tau=1}^{\hat{k}} u^c_{i\tau} \Delta t + \left(N-1\right)\Delta t + \frac{2N-1}{N}\epsilon \le \sum_{\tau=1}^{r} u^c_{i\tau} \Delta t + \left(N-1\right)\Delta t + \frac{2N-1}{N}\epsilon \tag{B.5a}$$

$$< \sum_{\tau=1}^{r} u^b_{i\tau} \Delta t = \sum_{\tau=1}^{\hat{k}} u^b_{i\tau} \Delta t. \tag{B.5b}$$

Because $u^b_{i\hat{k}} = 1$, we know that i has the maximum value of (B.5a) among $j = 1, \ldots, N$. Hence it follows from (B.5),

$$\sum_{\tau=1}^{\hat{k}} u^c_{j\tau} \Delta t - \sum_{\tau=1}^{\hat{k}} u^b_{j\tau} \Delta t < -\left(N-1\right)\Delta t - \frac{2N-1}{N}\epsilon, \; j = 1, \ldots, N. \tag{B.6}$$

Summing up over all the controls j,

$$-\epsilon \le \sum_{j=1}^{N} \left(\sum_{\tau=1}^{\hat{k}} u^c_{j\tau} \Delta t - \sum_{\tau=1}^{\hat{k}} u^b_{j\tau} \Delta t \right) < -N\left(N-1\right)\Delta t - (2N-2)\epsilon. \tag{B.7}$$

The first inequality comes from the definition of ϵ. This leads to $0 < -N(N-1)\Delta t - (N-1)\epsilon$, which is a contradiction because $\epsilon \ge 0$ and $N \ge 2$.

Case 2: We assume that,

$$\sum_{\tau=1}^{r} u^{c}_{i\tau}\Delta t - \sum_{\tau=1}^{r} u^{b}_{i\tau}\Delta t > (N-1)\,\Delta t + \frac{2N-1}{N}\epsilon. \tag{B.8}$$

From the definition of ϵ, we have

$$\sum_{1=j\neq i}^{N}\left(\sum_{\tau=1}^{r} u^{c}_{j\tau}\Delta t - \sum_{\tau=1}^{r} u^{b}_{j\tau}\Delta t\right) + \sum_{\tau=1}^{r} u^{c}_{i\tau}\Delta t - \sum_{\tau=1}^{r} u^{b}_{i\tau}\Delta t \leq \epsilon \tag{B.9}$$

and by substituting (B.8) into (B.9), it holds that

$$\sum_{1=j\neq i}^{N}\left(\sum_{\tau=1}^{r} u^{c}_{j\tau}\Delta t - \sum_{\tau=1}^{r} u^{b}_{j\tau}\Delta t\right) + (N-1)\Delta t + \frac{N-1}{N}\epsilon < 0. \tag{B.10}$$

Because the left-hand side can be written as the sum of $N-1$ terms as

$$\Delta t + \frac{1}{N}\epsilon + \sum_{\tau=1}^{r} u^{c}_{j\tau}\Delta t - \sum_{\tau=1}^{r} u^{b}_{j\tau}\Delta t, \tag{B.11}$$

at least one of them must be negative, therefore there exists $\hat{j}$ such that

$$\Delta t + \frac{1}{N}\epsilon + \sum_{\tau=1}^{r} u^{c}_{\hat{j}\tau}\Delta t - \sum_{\tau=1}^{r} u^{b}_{\hat{j}\tau}\Delta t < 0. \tag{B.12}$$

Let $\hat{k}$ be the highest index of the time step in which control $\hat{j}$ is rounded up,

$$\hat{k} = \underset{1\leq k\leq r}{\arg\max}\left\{k \mid u^{b}_{\hat{j}k} = 1\right\}. \tag{B.13}$$

Then we have

$$\sum_{\tau=1}^{\hat{k}} u^{c}_{\hat{j}\tau}\Delta t - \sum_{\tau=1}^{\hat{k}-1} u^{b}_{\hat{j}\tau}\Delta t \leq \Delta t + \sum_{\tau=1}^{r} u^{c}_{\hat{j}\tau}\Delta t - \sum_{\tau=1}^{r} u^{b}_{\hat{j}\tau}\Delta t < -\frac{1}{N}\epsilon. \tag{B.14}$$

The first inequality follows from $u^{b}_{\hat{j}\hat{k}} = 1$ and $u^{b}_{\hat{j}k} = 0$, $k \geq \hat{k}$. The second inequality follows from (B.12). Because $\hat{j}$ is the control which is rounded up at time step $\hat{k}$, for any

$j = 1, \dots, N$, it holds that,

$$\sum_{\tau=1}^{\hat{k}} u_{j\tau}^c \Delta t - \sum_{\tau=1}^{\hat{k}} u_{j\tau}^b \Delta t \leq \sum_{\tau=1}^{\hat{k}} u_{j\tau}^c \Delta t - \sum_{\tau=1}^{\hat{k}-1} u_{j\tau}^b \Delta t < -\frac{1}{N}\epsilon. \tag{B.15}$$

Summing over all the controls, we have

$$\sum_{j=1}^{N} \sum_{\tau=1}^{\hat{k}} \left(u_{j\tau}^c - u_{j\tau}^b \right) \Delta t < -\epsilon. \tag{B.16}$$

This contradicts the definition of the parameter ϵ. □

Proof of Corollary 3.3.1. Based on the formulation of the discretized control in the continuous relaxation of the model (DQCP-L_2), we have

$$\begin{aligned} \epsilon(\Delta t) &= \max_{k=1,\dots,T} \left| \sum_{\tau=1}^{k} \left(\sum_{j=1}^{N} u_{j\tau}^c - 1 \right) \Delta t \right| \leq \sum_{k=1}^{T} \left| \sum_{j=1}^{N} u_{jk}^c - 1 \right| \Delta t \\ &\leq \sqrt{T} \sqrt{\sum_{k=1}^{T} \left(\sum_{j=1}^{N} u_{jk}^c - 1 \right)^2 \Delta t} \, \Delta t = \sqrt{T l(u^c, T)} \Delta t = \sqrt{t_f l(u^c, T) \Delta t}. \end{aligned} \tag{B.17}$$

From Theorem 3.3.3, we directly obtain the statement (3.22). From Theorem 3.3.2, if the original function F is bounded, the optimized squared L_2 term $l(u^c, T)$ is uniformly bounded over time steps T, then the absolute integral error between continuous and discretized control converges with $O(\sqrt{\Delta t})$. □

Proof of Proposition 3.3.2. For any time step $k = 1, \dots, T$, we have

$$\begin{aligned} \lim_{\Delta t \to 0} X_k^c &= \lim_{\Delta t \to 0} \prod_{\tau=1}^{k} e^{-i\Delta t H_\tau} X_0 = \lim_{\Delta t \to 0} \prod_{\tau=1}^{k} e^{-i\Delta t \left(H^{(0)} + \sum_{j=1}^{N} u_{j\tau}^c H^{(j)} \right)} X_0 \\ &= \lim_{\Delta t \to 0} e^{-i\Delta t H^{(0)} - i\Delta t \sum_{j=1}^{N} \sum_{\tau=1}^{k} u_{j\tau}^c H^{(j)}} X_0. \end{aligned} \tag{B.18}$$

The last equality follows by Trotter expansion that $e^{\Delta t(A+B)} = e^{\Delta t A} e^{\Delta t B} + O(\Delta t^2)$ and taking the limit as Δt goes to zero. Similarly, for binary control, we have

$$\lim_{\Delta t \to 0} X_k^b = \lim_{\Delta t \to 0} e^{-i\Delta t H^{(0)} - i\Delta t \sum_{j=1}^{N} \sum_{\tau=1}^{k} u_{j\tau}^b H^{(j)}} X_0. \tag{B.19}$$

From Theorem 3.3.3, we have for any time step $k = 1, \dots, T$ and controller $j = 1, \dots, N$,

$$\left|\sum_{\tau=1}^{k} u_{j\tau}^{c} \Delta t - \sum_{\tau=1}^{k} u_{j\tau}^{b} \Delta t\right| \leq \left\|\sum_{\tau=1}^{k} \left(u_{\tau}^{c} - u_{\tau}^{b}\right) \Delta t\right\|_{\infty} \leq (N-1)\,\Delta t + \frac{2N-1}{N}\epsilon(\Delta t). \tag{B.20}$$

Combining with Corollary 3.3.1 and taking the limit as Δt goes to zero, we have

$$\lim_{\Delta t \to 0} \sum_{\tau=1}^{k} u_{j\tau}^{c} \Delta t = \lim_{\Delta t \to 0} \sum_{\tau=1}^{k} u_{j\tau}^{b} \Delta t. \tag{B.21}$$

We substitute (B.21) into the formulation of states (B.18) and (B.19), then we obtain the conclusion that

$$\lim_{\Delta t \to 0} X_k^b = \lim_{\Delta t \to 0} X_k^c,\ k = 1, \dots, T. \tag{B.22}$$

Substituting states into the objective function, we prove that

$$\lim_{\Delta t \to 0} F(X_T^b) = \lim_{\Delta t \to 0} F(X_T^c). \tag{B.23}$$

□

B.2 Detailed Numerical Results

In this section, we present the results of continuous relaxation, combinatorial integral approximation, and the improvement heuristic for all the methods and instances in Table B.1–B.6. We also present the objective values and TV regularizer values for all the instances and annotate the best method in Figure B.1.

Table B.1: Objective value results of continuous relaxation.

Instance	Objective value			TV regularizer		
	pGRAPE	TR	ADMM	pGRAPE	TR	ADMM
Energy2	1.10E−12	9.11E−05	8.94E−05	0.999	0.567	0.523
Energy4	0.154	0.155	0.168	5.064	4.114	2.752
Energy6	0.213	0.213	0.219	5.999	4.508	3.237
CNOT5	0.169	0.124	0.191	16.000	9.419	6.094
CNOT10	1.16E−09	2.65E−04	3.21E−04	20.979	15.194	11.056
CNOT15	1.00E−10	8.43E−06	3.65E−06	29.846	24.348	16.795
CNOT20	5.93E−10	4.06E−06	8.07E−07	26.162	23.481	15.099
NOT2	0.163	0.163	0.163	2.226	1.188	0.908
NOT6	4.28E−10	6.55E−05	1.07E−04	3.251	0.652	0.578
NOT10	6.55E−11	5.14E−05	4.79E−05	2.740	1.530	1.005
CircuitH_2	4.37E−07	1.24E−04	1.33E−05	18.720	8.421	2.744
CircuitLiH	1.14E−03	1.45E−03	1.43E−03	53.976	48.720	0.677

Table B.2: CPU time and iterations of continuous relaxation.

Instance	CPU time (s)			Iterations		
	pGRAPE	TR	ADMM	pGRAPE	TR	ADMM
Energy2	0.13	1.77	19.56	7	132	100
Energy4	2.89	27.99	163.08	44	914	100
Energy6	27.94	341.43	1195.41	51	1578	100
CNOT5	1.12	79.55	21.75	111	3146	100
CNOT10	0.75	192.78	70.21	31	3279	100
CNOT15	1.26	348.96	100.59	39	3900	100
CNOT20	1.02	432.73	150.25	21	3929	100
NOT2	0.05	4.93	2.63	15	1268	100
NOT6	0.15	48.81	10.93	24	3563	100
NOT10	0.10	91.39	34.05	6	3762	100
CircuitH_2	3.01	154.05	39.86	244	3832	100
CircuitLiH	663.08	1403.03	564.95	4345	4247	100

Table B.3: Objective value results of combinatorial integral approximation.

Instance	Objective value/TV regularizer								
	pGRAPE+SUR	TR+SUR	ADMM+SUR	pGRAPE+MT	TR+MT	ADMM+MT	pGRAPE+MS	TR+MS	ADMM+MS
Energy2	4.22E−04/54	4.91E−03/54	4.01E−04/48	0.159/4	0.159/6	0.154/6	0.029/10	0.040/10	0.028/10
Energy4	0.158/26.8	0.158/32.4	0.170/44.8	0.367/6	0.317/6	0.363/6	0.163/10	0.162/10	0.195/10
Energy6	0.216/38.8	0.216/43.6	0.221/52.8	0.684/6	0.616/6	0.606/6	0.247/10	0.258/10	0.240/10
CNOT5	0.170/16	0.332/24	0.190/41	0.243/10	0.525/6	0.285/7	0.170/16	0.593/15	0.191/32
CNOT10	6.01E−04/116	1.78E−03/116	1.68E−03/86	0.158/22	0.323/21	0.084/15	0.011/39	0.019/38	0.006/32
CNOT15	1.12E−03/266	2.30E−03/276	2.90E−03/279	0.539/37	0.290/36	0.176/27	0.325/38	0.284/40	0.214/39
CNOT20	1.45E−03/491	8.30E−04/480	1.46E−03/467	0.782/53	0.314/51	0.517/47	0.654/39	0.697/39	0.619/39
NOT2	0.164/3	0.164/1	0.164/1	0.164/1	0.164/1	0.164/1	0.164/3	0.164/1	0.164/1
NOT6	2.38E−03/40	3.10E−03/46	1.55E−03/52	4.05E−02/8	3.65E−02/9	7.56E−02/10	1.38E−02/19	1.58E−01/14	1.54E−01/14
NOT10	4.73E−03/126	3.91E−04/126	9.22E−03/21	2.78E−02/24	1.34E−01/25	6.38E−02/6	6.16E−02/40	2.84E−01/40	9.22E−03/21
CircuitH_2	0.027/32	0.027/36	0.006/76	0.600/8	0.591/8	0.063/8	0.026/22	0.038/24	0.008/22
CircuitLiH	0.168/380	0.182/380	0.033/252	0.963/68	0.966/72	0.658/48	0.287/290	0.224/288	0.557/148

Table B.4: CPU time and iterations of combinatorial integral approximation.

Instance	CPU time (s)/Iterations					
	pGRAPE+MT	TR+MT	ADMM+MT	pGRAPE+MS	TR+MS	ADMM+MS
Energy2	< 0.01/39	< 0.01/39	< 0.01/18	< 0.01/3715	< 0.01/4627	< 0.01/5106
Energy4	< 0.01/15	< 0.01/17	< 0.01/17	< 0.01/759	< 0.01/2578	< 0.01/925
Energy6	< 0.01/8	< 0.01/8	< 0.01/8	< 0.01/2473	< 0.01/3730	< 0.01/2508
CNOT5	0.55/5	0.55/3	0.57/2	0.50/2	0.52/2	0.72/7
CNOT10	19.29/36786	13.50/5916	6.49/4136	61.83/25685	61.83/27432	61.83/54921
CNOT15	7.92/555	8.89/10	25.79/21326	64.36/14349	64.62/9878	64.70/29421
CNOT20	28.46/31236	22.86/8482	38.42/1837	67.41/2069	67.22/1242	67.30/26378
NOT2	0.05/1	0.05/1	0.04/1	0.04/1	0.06/1	0.07/1
NOT6	0.80/7403	6.62/32640	0.26/1	41.37/47882	0.26/1	60.19/90179
NOT10	0.97/1255	60.51/1815	4.26/4136	60.50/29462	0.66/1	0.59/1
CircuitH_2	0.89/1	0.87/1	3.83/1514	1.34/1	1.87/1	60.76/24639
CircuitLiH	70.94/349	70.87/889	71.01/1570	71.45/1545	70.83/1595	71.23/2137

Table B.5: Objective value results of improvement heuristic.

Instance	Objective value/TV regularizer Improvement of objective value/Improvement of TV regularizer								
	pGRAPE+ SUR+ALB	TR+ SUR+ALB	ADMM+ SUR+ALB	pGRAPE+ MT+ALB	TR+ MT+ALB	ADMM+ MT+ALB	pGRAPE+ MS+ALB	TR+ MS+ALB	ADMM+ MS+ALB
Energy2	0.003/10 −0.20%/81.48%	0.005/8 0.00%/85.19%	0.003/4 −0.20%/91.67%	0.003/4 18.55%/0.00%	0.003/4 18.55%/0.00%	0.041/6 13.36%/0.00%	0.001/10 2.88%/0.00%	0.003/10 3.85%/0.00%	0.002/8 2.67%/20.00%
Energy4	0.160/17.2 −0.14%/35.82%	0.162/14 −0.42%/56.79%	0.166/14.4 0.53%/67.86%	0.199/6 26.61%/0.00%	0.198/6 17.34%/0.00%	0.218/5.6 22.84%/6.67%	0.160/9.2 0.38%/8.00%	0.160/10 0.32%/16.00%	0.184/8.4 1.31%/0.00%
Energy6	0.218/32.4 −0.25%/16.49%	0.216/40.4 −0.01%/7.34%	0.219/50.0 0.25%/5.30%	0.429/6 80.74%/0.00%	0.429/6 49.09%/0.00%	0.451/6 39.28%/0.00%	0.229/10 2.47%/0.00%	0.230/10 3.77%/0.00%	0.216/10 0.57%/0.00%
CNOT5	0.176/9 −3.53%/43.75%	0.206/7 37.95%/70.83%	0.176/9 7.37%/78.05%	0.195/9 19.75%/10.00%	0.196/10 62.67%/−66.67%	0.195/9 31.58%/−28.57%	0.170/16 0.00%/0.00%	0.173/20 70.83%/−33.33%	0.172/16 9.95%/50.00%
CNOT10	1.58E−03/30 −66.42%/74.14%	2.46E−03/24 −38.20%/70.69%	1.15E−03/20 0.00%/76.74%	4.06E−03/23 97.47%/−4.55%	9.43E−03/16 97.21%/23.81%	6.04E−03/15 92.86%/0.00%	9.80E−04/39 90.91%/0.00%	1.31E−03/38 94.74%/0.00%	1.18E−03/36 83.33%/−12.50%
CNOT15	5.59E−04/262 44.10%/1.50%	4.67E−04/256 76.65%/7.25%	8.51E−04/263 71.64%/5.73%	6.31E−03/33 98.89%/10.81%	6.25E−03/34 97.93%/5.56%	1.63E−03/30 99.43%/−11.11%	1.30E−03/39 99.69%/−2.63%	3.72E−03/39 98.94%/2.50%	1.91E−03/40 99.07%/−2.56%
CNOT20	4.56E−04/479 54.44%/2.44%	6.13E−04/471 38.68%/1.88%	5.07E−04/441 49.27%/5.57%	1.20E−03/38 99.87%/28.30%	8.22E−04/49 99.74%/3.92%	1.35E−03/48 99.81%/−2.13%	9.47E−04/40 99.86%/−2.56%	2.81E−03/40 99.57%/−2.56%	7.45E−04/40 99.88%/−2.56%
NOT2	0.164/1 0.46%/66.67%	0.164/1 0.36%/0.00%	0.164/1 0.36%/0.00%	0.164/1 0.36%/0.00%	0.164/1 0.36%/0.00%	0.164/1 0.36%/0.00%	0.163/3 0.62%/0.00%	0.163/3 0.52%/−200.00%	0.163/3 0.52%/−200.00%
NOT6	1.42E−03/6 40.24%/85.00%	7.33E−04/10 76.34%/86.96%	7.24E−04/13 53.25% /75.00%	1.45E−03/9 96.41%/−12.50%	1.27E−02/11 65.21%/−22.22%	3.39E−03/12 95.52%/−20.00%	1.06E−03/22 92.33%/−15.79%	8.90E−04/18 99.44%/−28.57%	1.74E−03/16 98.87% /−14.29%
NOT10	5.50E−04/32 88.37%/74.60%	1.66E−03/6 −76.45%/92.06%	6.16E−04/7 93.32%/66.67%	7.37E−04/24 97.35%/0.00%	9.09E−04/20 99.32%/20.00%	8.90E−04/8 98.60%/−33.33%	5.44E−04/38 99.12%/5.00%	7.57E−04/39 99.73%/2.50%	8.90E−04/21 90.35%/0.00%
CircuitH_2	0.003/24 88.89%/25.00%	0.013/32 51.85%/11.11%	0.006/76 0.00%/0.00%	0.245/12 59.17%/−50.00%	0.007/2 98.82%/75.00%	0.054/10 14.29%/−25.00%	0.014/18 46.15%/18.18%	0.003/22 92.11%/8.33%	0.008/22 0.00%/0.00%
CircuitLiH	0.069/378 58.93%/0.53%	0.083/378 54.40%/0.53%	0.033/252 0.00%/0.00%	0.002/6 99.79%/91.18%	0.496/70 48.65%/2.78%	0.355/48 46.05%/0.00%	0.165/286 42.51%/1.38%	0.120/290 46.43%/−0.69%	0.021/158 96.23%/−6.76%

Table B.6: CPU time and iterations of improvement heuristic.

Instance	CPU time (s)/Iterations								
	pGRAPE+ SUR+ALB	TR+ SUR+ALB	ADMM+ SUR+ALB	pGRAPE+ MT+ALB	TR+ MT+ALB	ADMM+ MT+ALB	pGRAPE+ MS+ALB	TR+ MS+ALB	ADMM+ MS+ALB
Energy2	3.41/255	5.21/379	2.26/262	2.26/170	2.25/170	1.59/150	1.27/93	1.90/154	1.24/91
Energy4	3.36/127	5.53/191	7.90/280	5.12/207	5.13/207	4.94/194	1.21/43	1.23/43	2.14/68
Energy6	16.95/94	11.11/62	9.88/56	41.42/219	39.53/208	33.36/174	18.51/100	24.02/144	10.84/56
CNOT5	12.69/394	16.96/506	30.09/993	19.67/521	20.34/550	21.46/575	0.91/1377	33.99/33	12.58/464
CNOT10	70.36/1245	56.97/1031	56.90/998	74.83/945	84.95/1050	51.37/574	19.89/301	24.04/323	16.65/270
CNOT15	9.12/106	30.48/395	25.57/317	150.25/1238	169.47/1335	107.17/895	113.02/944	129.21/1288	82.44/935
CNOT20	27.82/249	26.22/251	33.77/397	208.79/1246	139.69/1010	157.57/1009	153.63/1035	155.07/1138	157.87/1182
NOT2	0.46/93	0.40/63	0.52/93	0.39/63	0.46/62	0.31/63	0.34/63	0.47/63	0.31/62
NOT6	11.71/725	7.36/306	2.49/188	9.46/592	5.47/231	8.63/582	7.82/458	5.82/236	6.90/470
NOT10	29.69/1181	16.16/409	7.87/298	21.46/824	21.89/558	14.11/541	7.22/256	17.05/465	4.06/161
CircuitH_2	4.84/97	3.18/65	4.70/33	5.40/88	9.21/156	2.73/39	4.88/108	3.14/65	1.68/33
CircuitLiH	26.08/108	46.26/108	9.05/36	64.29/186	74.88/153	59.21/135	41.43/143	20.02/71	39.78/158

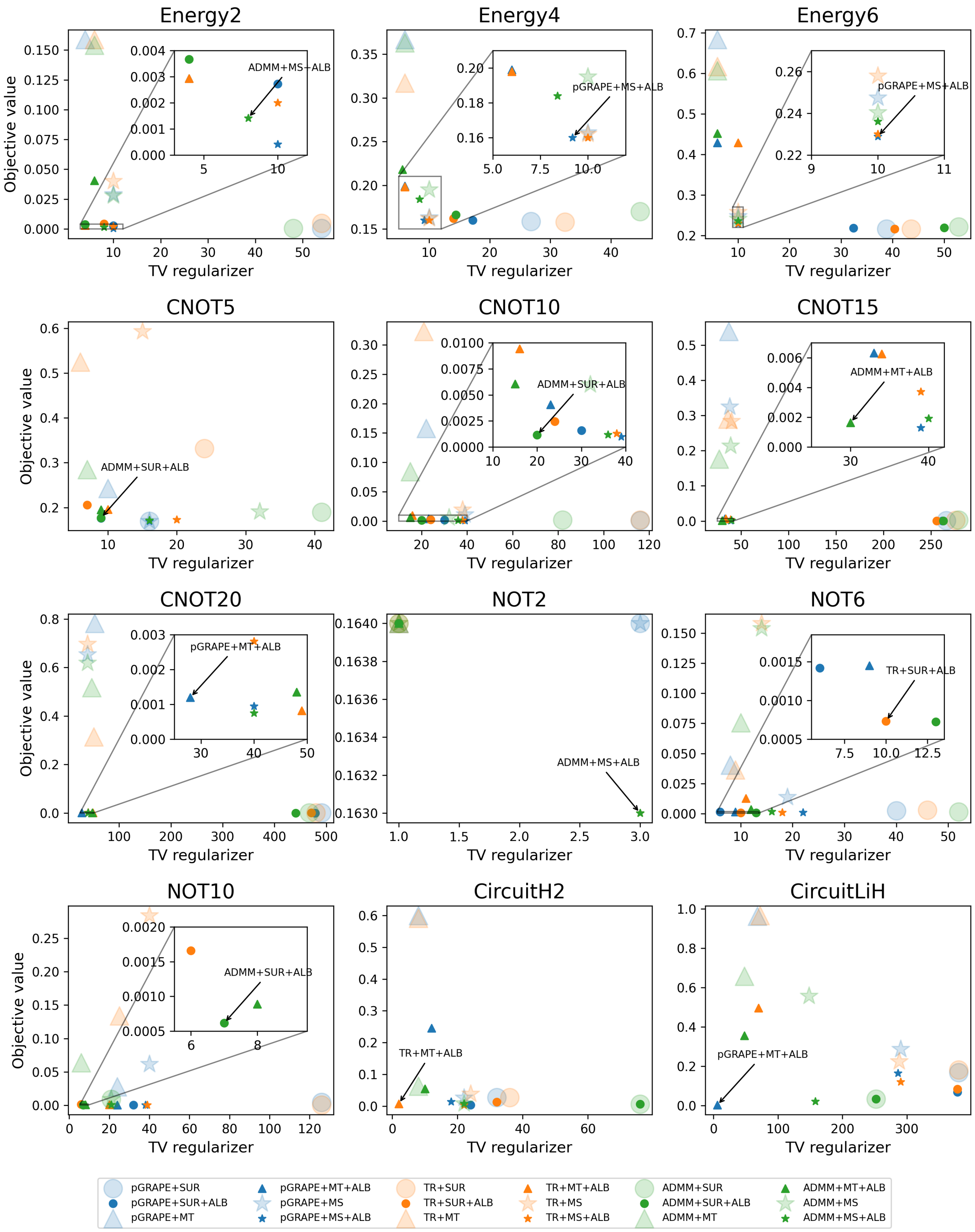
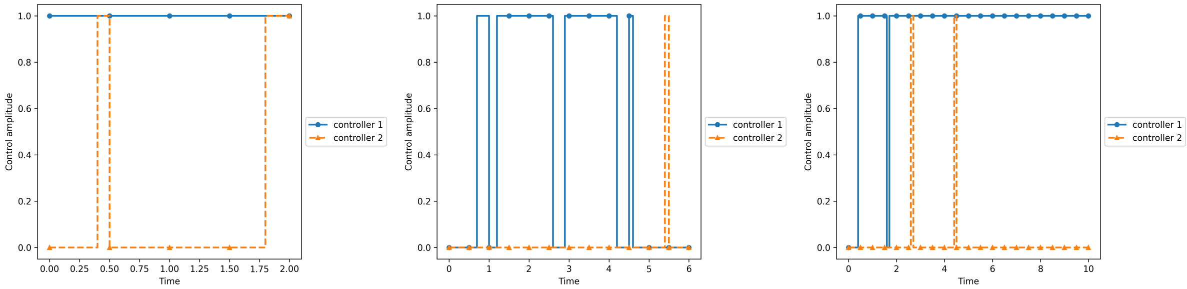


Figure B.1: Objective values and TV regularizer values of binary results of all the instances.

B.3 Control Results of NOT Estimation Problem

For the NOT gate estimation problem, we present the control obtaining the best trade-off between objective values and TV regularizer values which are annotated in Figure B.2. All three instances show that the controller 1 has more significant impact on the final infidelity. When the evolution time is short ($t_f = 2$), the optimal control sets controller 1 active all the time. When the evolution time is longer ($t_f = 6,\ 10$), the active time of controller 2 is shorter. The main reason is that only considering controller 1 can also obtain a low infidelity with a sufficiently long evolution time but the performance of a single controller is worse than two controllers with the same evolution time [145, 172].



(a) $t_f = 2$

(b) $t_f = 6$

(c) $t_f = 10$

Figure B.2: Control results of NOT gate estimation example with different evolution times. Blue lines represent the controller 1 and orange dashed lines represent the controller 2.

APPENDIX C

Appendix For Chapter 4

C.1 Proofs of All Theorems in Section 4.2

Proof of Lemma 4.2.4. Let $A = \sum_{i=1}^{m} w_i \sigma_i(A) v_i^T$ be the singular value decomposition of A where $w_1, \ldots, w_m$ and $v_1, \ldots, v_m$ are orthonormal vectors. Then the norm of the trace

$$\begin{aligned} |\mathbf{tr}\{UA\}| = \left|\mathbf{tr}\left\{\sum_{i=1}^{m} U w_i \sigma_i(A) v_i^\dagger\right\}\right| &\leq \sum_{i=1}^{m} \left|\mathbf{tr}\left\{U w_i \sigma_i(A) v_i^\dagger\right\}\right| \\ \leq \sum_{i=1}^{m} \|U w_i\|_2 \sigma_i(A) \|v_i\|_2 &\leq \sum_{i=1}^{m} \sigma_i(A) \leq m\sigma_1(A). \end{aligned} \tag{C.1}$$

The first and second inequalities follow from the norm inequality. The third inequality holds because $Uw_1, \ldots, Uw_m$ and $v_1, \ldots, v_m$ are orthonormal vectors. The last inequality comes from the definition of singular values. □

Proof of Theorem 4.2.1. At time step k, for each examined control $u^{(j)}$ with only controller j active, we define the change of objective value as $R_{jk} = \bar{F}(u^{(j)}) - F_{\text{cur}}$. We prove the theorem by showing that at each time step, there exists a controller such that the change of objective value $R_{jk} \leq 2C_1 e^{C_2 \Delta t} \Delta t^2$. Our proof mainly consists of two parts. Firstly we prove that the sum of all the terms in each R_{jk} corresponding to Δt^l, $l \geq 2$ is upper bounded by $2C_1 e^{C_2 \Delta t} \Delta t^2$. Then we prove that there exists a controller $\hat{j}$ such that the sum of the constant terms and terms corresponding to Δt in $R_{\hat{j}k}$ is upper bounded by 0. Before we present our detailed proof, we define a constant parameter

$$\sigma_{\max} = \max_{j=1,\ldots,N} \sigma_1(H^{(j)}) + \sigma_1(H^{(0)}), \tag{C.2}$$

where $\sigma_1(\cdot)$ is the maximum singular value of a given constant Hamiltonian controller. We discuss the change of objective value for the aforementioned two specific objective function formulations (4.2)–(4.3). We start from the infidelity function because the formulation is more concise to prove.

Infidelity function For each time step k, the change of objective value R_{jk} is computed as

$$\begin{aligned} R_{jk} =& \frac{1}{2^q}\left|\mathbf{tr}\left\{X_{\text{targ}}^\dagger \prod_{l=k+1}^{T} U_l e^{-iH^{(0)}\Delta t - i\sum_{j'=1}^{N} u^{\text{con}}_{j'k} H^{(j')}\Delta t} \prod_{l=1}^{k-1} U_l X_{\text{init}}\right\}\right| \\ & - \frac{1}{2^q}\left|\mathbf{tr}\left\{X_{\text{targ}}^\dagger \prod_{l=k+1}^{T} U_l e^{-iH^{(0)}\Delta t - iH^{(j)}\Delta t} \prod_{l=1}^{k-1} U_l X_{\text{init}}\right\}\right|, \end{aligned} \tag{C.3}$$

where

$$U_l = \begin{cases} e^{-iH^{(0)}\Delta t - i\sum_{j'=1}^{N} u^{\text{bin}}_{j'l} H^{(j)}\Delta t} & 1 \le l \le k-1 \\ e^{-iH^{(0)}\Delta t - i\sum_{j'=1}^{N} u^{\text{con}}_{j'l} H^{(j)}\Delta t} & k+1 \le l \le T \end{cases}. \tag{C.4}$$

For simplicity, we denote

$$\mathbf{U}^{(k)} = \prod_{l=1}^{k-1} U_l X_{\text{init}} X_{\text{targ}}^\dagger \prod_{l=k+1}^{T} U_l, \tag{C.5}$$

which is still a unitary matrix because it is the product of a series of unitary matrices. Following the definition of trace, we know that the trace of a product of matrices is invariant under cyclic permutations. Hence,

$$R_{jk} = \frac{1}{2^q}\left|\mathbf{tr}\left\{\mathbf{U}^{(k)} e^{-iH^{(0)}\Delta t - i\sum_{j'=1}^{N} u^{\text{con}}_{j'k} H^{(j')}\Delta t}\right\}\right| - \frac{1}{2^q}\left|\mathbf{tr}\left\{\mathbf{U}^{(k)} e^{-iH^{(0)}\Delta t - iH^{(j)}\Delta t}\right\}\right|. \tag{C.6}$$

Without loss of generality, we consider a general matrix $A \in \mathbb{C}^{m\times m}$, then the exponential matrix $e^{A\Delta t} = \sum\limits_{l=0}^{\infty} \frac{1}{l!} A^l \Delta t^l$. Therefore we have

$$\mathbf{tr}\left\{\mathbf{U}^{(k)} e^{A\Delta t}\right\} = \mathbf{tr}\left\{\mathbf{U}^{(k)}(I + A\Delta t)\right\} + \sum_{l=2}^{\infty} \frac{1}{l!}\mathbf{tr}\left\{\mathbf{U}^{(k)} A^l\right\}\Delta t^l. \tag{C.7}$$

The norm of the second term of the trace in (C.7) is upper bounded as

$$\left|\sum_{l=2}^{\infty}\frac{1}{l!}\mathbf{tr}\left\{\mathbf{U}^{(k)}A^l\right\}\Delta t^l\right| \leq \sum_{l=2}^{\infty}\frac{m}{l!}\sigma_1(A^l)\Delta t^l \leq \sum_{l=2}^{\infty}\frac{m}{l!}(\sigma_1(A)\Delta t)^l \leq m\sigma_1^2(A)e^{\sigma_1(A)\Delta t}\Delta t^2. \tag{C.8}$$

The first inequality follows from Lemma 4.2.4 and the second inequality follows from Lemma 4.2.2. The last inequality directly comes from the Taylor expansion. We define constants $C_1(A),\ C_2(A)$ corresponding to matrix A as

$$C_1(A) = m\sigma_1^2(A), \quad C_2(A) = \sigma_1(A), \tag{C.9}$$

then the second term is upper bounded by $C_1(A)e^{C_2(A)\Delta t}\Delta t^2$. Therefore we have the following inequality for the norm of trace (C.7):

$$\begin{aligned}\left|\mathbf{tr}\left\{\mathbf{U}^{(k)}(I+A\Delta t)\right\}\right| - C_1(A)e^{C_2(A)\Delta t}\Delta t^2 &\leq \left|\mathbf{tr}\left\{\mathbf{U}^{(k)}e^{A\Delta t}\right\}\right| \\ &\leq \left|\mathbf{tr}\left\{\mathbf{U}^{(k)}(I+A\Delta t)\right\}\right| + C_1(A)e^{C_2(A)\Delta t}\Delta t^2,\end{aligned} \tag{C.10}$$

In our examples, all the Hamiltonian control matrices have a formulation $A = -i\left(^{(0)} + \sum_{j=1}^{N} u_j H^{(j)}\right)$ with given controls $\sum_{j=1}^{N} u_j = 1$ and given constant controllers $H^{(0)},\ H^{(j)} \in \mathbb{C}^{2^q\times 2^q}$. Then the maximum singular value of any control matrix

$$\sigma_1(A) \leq \sigma_1(H^{(0)}) + \sum_{j=1}^{N} u_j\sigma_1(H^{(j)}) \leq \sigma_1(H^{(0)}) + \max_{j=1,\dots,N}\sigma_1(H^{(j)}) = \sigma_{\max}. \tag{C.11}$$

We define constants $C_1,\ C_2$ as

$$C_1 = \sigma_{\max}^2, \quad C_2 = \sigma_{\max}. \tag{C.12}$$

For any time-dependent Hamiltonian matrix A, the constants $C_1(A) \leq 2^q C_1,\ C_2(A) \leq C_2$. Substituting $A = -iH^{(0)} - i\sum_{j'=1}^{N} u_{j'k}^{\text{con}} H^{(j')}$ and $A = -iH^{(0)} - iH^{(j)}$ into the inequality (C.10) and combining (C.12), the change of objective value satisfies

$$\begin{aligned}R_{jk} \leq &\frac{1}{2^q}\left|\mathbf{tr}\left\{\mathbf{U}^{(k)}\left(I - iH^{(0)}\Delta t - i\sum_{j'=1}^{N} u_{j'k}^{\text{con}} H^{(j')}\Delta t\right)\right\}\right| \\ &- \frac{1}{2^q}\left|\mathbf{tr}\left\{\mathbf{U}^{(k)}\left(I - iH^{(0)}\Delta t - iH^{(j)}\Delta t\right)\right\}\right| + 2C_1e^{C_2\Delta t}\Delta t^2.\end{aligned} \tag{C.13}$$

We prove that there exists a controller j such that the value of the first two terms is no more than 0 by contradiction. Assume that for any j,

$$\left|\mathbf{tr}\left\{\mathbf{U}^{(k)}\left(I - iH^{(0)}\Delta t - i\sum_{j'=1}^{N} u^{\text{con}}_{j'k} H^{(j')}\Delta t\right)\right\}\right| - \left|\mathbf{tr}\left\{\mathbf{U}^{(k)}\left(I - iH^{(0)}\Delta t - iH^{(j)}\Delta t\right)\right\}\right| > 0. \tag{C.14}$$

Then we have

$$\begin{aligned}\left|\mathbf{tr}\left\{\mathbf{U}^{(k)}\left(I - iH^{(0)}\Delta t - i\sum_{j'=1}^{N} u^{\text{con}}_{j'k} H^{(j')}\Delta t\right)\right\}\right| &\leq \sum_{j'=1}^{N} u^{\text{con}}_{j'k}\left|\mathbf{tr}\left\{\mathbf{U}^{(k)}\left(I - iH^{(0)}\Delta t - iH^{(j')}\Delta t\right)\right\}\right| \\ &< \left|\mathbf{tr}\left\{\mathbf{U}^{(k)}\left(I - iH^{(0)}\Delta t - i\sum_{j'=1}^{N} u^{\text{con}}_{j'k} H^{(j')}\Delta t\right)\right\}\right|.\end{aligned} \tag{C.15}$$

The first inequality follows from $\sum_{j'=1}^{N} u^{\text{con}}_{j'k} = 1$ and $0 \leq u^{\text{con}}_{j'k} \leq 1$. The second inequality follows from (C.14). It obviously leads to a contradiction. Therefore we prove that there exists at least one controller j such that

$$R_{jk} \leq 2C_1 e^{C_2\Delta t}\Delta t^2. \tag{C.16}$$

In the algorithm, at each time step, we choose the $\hat{j}$ with minimum change of objective value. Taking the summation over all time steps, we have

$$\bar{F}(u^{\text{bin}}) - \bar{F}(u^{\text{con}}) = \sum_{k=1}^{T} \min_{j=1,\dots,N} R_{jk} \leq 2C_1 e^{C_2\Delta t}\Delta t. \tag{C.17}$$

Energy function We have a similar proof for the energy function. For each time step k, the change of objective value R_{jk} is computed as

$$\begin{aligned}R_{jk} =& \langle\psi_0|\left(\mathbf{U}^{(2k)} e^{-iH^{(0)}\Delta t - i\sum_{j'=1}^{N} u^{\text{con}}_{j'k} H^{(j')}\Delta t}\mathbf{U}^{(1k)}\right)^{\dagger} \tilde{H}\mathbf{U}^{(2k)} e^{-iH^{(0)}\Delta t - i\sum_{j'=1}^{N} u^{\text{con}}_{j'k} H^{(j')}\Delta t}\mathbf{U}^{(1k)}|\psi_0\rangle / E_{\min} \\ &- \langle\psi_0|\left(\mathbf{U}^{(2k)} e^{-iH^{(0)}\Delta t - iH^{(j)}\Delta t}\mathbf{U}^{(1k)}\right)^{\dagger} \tilde{H}\mathbf{U}^{(2k)} e^{-iH^{(0)}\Delta t - iH^{(j)}\Delta t}\mathbf{U}^{(1k)}|\psi_0\rangle / E_{\min},\end{aligned} \tag{C.18}$$

where

$$\mathbf{U}^{(1k)} = \prod_{l=1}^{k-1} e^{-iH^{(0)}\Delta t - i\sum_{j'=1}^{N} u^{\text{bin}}_{j'l} H^{(j')}\Delta t} X_{\text{init}}, \quad \mathbf{U}^{(2k)} = \prod_{l=k+1}^{T} e^{-iH^{(0)}\Delta t - i\sum_{j'=1}^{N} u^{\text{con}}_{j'l} H^{(j')}\Delta t}. \tag{C.19}$$

It is obvious that $\mathbf{U}^{(1k)}, \mathbf{U}^{(2k)}$ are both unitary matrices. We consider a general Hamiltonian matrix $A \in \mathbb{C}^{m\times m}$, with the matrix exponential expansion, we have

$$\begin{aligned}
&\langle\psi_0|\left(\mathbf{U}^{(2k)}e^{A\Delta t}\mathbf{U}^{(1k)}\right)^\dagger \tilde{H}\mathbf{U}^{(2k)}e^{A\Delta t}\mathbf{U}^{(1k)}|\psi_0\rangle \\
=&\langle\psi_0|\left(\mathbf{U}^{(2k)}\mathbf{U}^{(1k)}\right)^\dagger \tilde{H}\mathbf{U}^{(2k)}\mathbf{U}^{(1k)}|\psi_0\rangle \\
&+\langle\psi_0|\left(\mathbf{U}^{(2k)}A\mathbf{U}^{(1k)}\right)^\dagger \tilde{H}\mathbf{U}^{(2k)}\mathbf{U}^{(1k)}|\psi_0\rangle\Delta t+\langle\psi_0|\left(\mathbf{U}^{(2k)}\mathbf{U}^{(1k)}\right)^\dagger \tilde{H}\mathbf{U}^{(2k)}A\mathbf{U}^{(1k)}|\psi_0\rangle\Delta t \\
&+\langle\psi_0|\left(\mathbf{U}^{(2k)}A\mathbf{U}^{(1k)}\right)^\dagger \tilde{H}\mathbf{U}^{(2k)}A\mathbf{U}^{(1k)}|\psi_0\rangle\Delta t^2 \qquad \text{(C.20a)} \\
&+\langle\psi_0|\left(\mathbf{U}^{(2k)}\sum_{l=2}^{\infty}\frac{1}{l!}A^l\Delta t^l\mathbf{U}^{(1k)}\right)^\dagger \tilde{H}\mathbf{U}^{(2k)}e^{A\Delta t}\mathbf{U}^{(1k)}|\psi_0\rangle \qquad \text{(C.20b)} \\
&+\langle\psi_0|\left(\mathbf{U}^{(2k)}\mathbf{U}^{(1k)}\right)^\dagger \tilde{H}\mathbf{U}^{(2k)}\sum_{l=2}^{\infty}\frac{1}{l!}A^l\Delta t^l\mathbf{U}^{(1k)}|\psi_0\rangle \qquad \text{(C.20c)} \\
&+\langle\psi_0|\left(\mathbf{U}^{(2k)}A\Delta t\mathbf{U}^{(1k)}\right)^\dagger \tilde{H}\mathbf{U}^{(2k)}\sum_{l=2}^{\infty}\frac{1}{l!}A^l\Delta t^l\mathbf{U}^{(1k)}|\psi_0\rangle. \qquad \text{(C.20d)}
\end{aligned}$$

We first prove that there exists constants $C_1(A)$, $C_2(A)$ such that the summation of terms (C.20a)–(C.20d) is bounded by $C_1(A)e^{C_2(A)\Delta t}\Delta t^2$. Because $\mathbf{U}^{(1k)}$ and $\mathbf{U}^{(2k)}$ are unitary matrices and $\||\psi_0\rangle\|_2 = 1$, following from Lemma 4.2.1 and Lemma 4.2.3, the norm value of term (C.20a) is no more than

$$\sigma_1(A^\dagger\tilde{H}A)\Delta t^2 \le \sigma_1^2(A)\sigma_1(\tilde{H})\Delta t^2 \le \sigma_1^2(A)\sigma_1(\tilde{H})e^{\sigma_1(A)\Delta t}\Delta t^2. \qquad \text{(C.21)}$$

The first inequality follows from Lemma 4.2.2. The second inequality holds because Δt and $\sigma_1(A)$ are non-negative. Recall that $e^{A\Delta t}$ is a unitary matrix because A is a Hamiltonian matrix, hence by Lemma 4.2.1 and Lemma 4.2.3, the norm value of term (C.20b) is no more than

$$\sum_{l=2}^{\infty}\frac{1}{l!}\sigma_1(A^l\tilde{H})\Delta t^l \le \sigma_1(\tilde{H})\sum_{l=2}^{\infty}\frac{1}{l!}\sigma_1^l(A)\Delta t^l \le \sigma_1(\tilde{H})e^{\sigma_1(A)\Delta t}\Delta t^2. \qquad \text{(C.22)}$$

The first inequality follows from Lemma 4.2.2 and the second inequality follows from Taylor expansion. Similarly, the norm value of term (C.20c) is bounded by

$$\sigma_1(\tilde{H})e^{\sigma_1(A)\Delta t}\Delta t^2, \qquad \text{(C.23)}$$

and the norm value of term (C.20d) is bounded by

$$\sigma_1(A^\dagger\tilde{H})e^{\sigma_1(A)\Delta t}\Delta t^3 \le t_f\sigma_1(A)\sigma_1(\tilde{H})e^{\sigma_1(A)\Delta t}\Delta t^2, \qquad \text{(C.24)}$$

where the inequality holds because of the fact that $\Delta t \leq t_f$ and Lemma 4.2.2. We define the constants $C_1(A),\ C_2(A)$ as

$$C_1(A) = \sigma_1(\tilde{H})\left(\sigma_1^2(A) + 2 + t_f\sigma_1(A)\right) \tag{C.25a}$$

$$C_2(A) = \sigma_1(A). \tag{C.25b}$$

The values of the summation of terms (C.20a)–(C.20d) is upper bounded by the summation of their norms, which is bounded by $C_1(A)e^{C_2(A)\Delta t}\Delta t^2$. We define the constants $C_1,\ C_2$ as

$$C_1 = \sigma_1(\tilde{H})\left(\sigma_{\max}^2 + 2 + t_f\sigma_{\max}\right)/|E_{\min}| \tag{C.26a}$$

$$C_2 = \sigma_{\max}. \tag{C.26b}$$

From (C.11), we know that for any time-dependent Hamiltonian controller A, we have $C_1(A) \leq |E_{\min}|C_1,\ C_2(A) \leq C_2$. Substituting $A = -iH^{(0)} - i\sum_{j'=1}^{N} u_{j'k}^{\text{con}} H^{(j')}$ and $A = -iH^{(0)} - iH^{(j)}$ into the expanded trace equation (C.20) and combining with the constants $C_1,\ C_2$, the change of objective value satisfies

$$\begin{aligned} R_{jk} \leq & \frac{1}{E_{\min}}\langle\psi_0|\left(\mathbf{U}^{(2k)} i\left(H^{(j)} - \sum_{j'=1}^{N} u_{j'k}^{\text{con}} H^{(j')}\right)\mathbf{U}^{(1k)}\right)^\dagger \tilde{H}\mathbf{U}^{(2k)}\mathbf{U}^{(1k)}|\psi_0\rangle\Delta t \\ & + \frac{1}{E_{\min}}\langle\psi_0|\left(\mathbf{U}^{(2k)}\mathbf{U}^{(1k)}\right)^\dagger \tilde{H}\mathbf{U}^{(2k)} i\left(H^{(j)} - \sum_{j'=1}^{N} u_{j'k}^{\text{con}} H^{(j')}\right)\mathbf{U}^{(1k)}|\psi_0\rangle\Delta t \\ & + 2C_1 e^{C_2\Delta t}\Delta t^2. \end{aligned} \tag{C.27}$$

We prove that there exists a controller j such that the value of terms of Δt is upper bounded by zero. Assume that there does not exist such a controller, then by taking summation weighted by u_{jk}^{con} over all the controllers, we have

$$\begin{aligned} 0 < & \frac{1}{E_{\min}}\langle\psi_0|\left(\mathbf{U}^{(2k)} i\sum_{j=1}^{N} u_{jk}^{\text{con}}\left(H^{(j)} - \sum_{j'=1}^{N} u_{j'k}^{\text{con}} H^{(j')}\right)\mathbf{U}^{(1k)}\right)^\dagger \tilde{H}\mathbf{U}^{(2k)}\mathbf{U}^{(1k)}|\psi_0\rangle\Delta t \\ & + \frac{1}{E_{\min}}\langle\psi_0|\left(\mathbf{U}^{(2k)}\mathbf{U}^{(1k)}\right)^\dagger \tilde{H}\mathbf{U}^{(2k)} i\sum_{j=1}^{N} u_{jk}^{\text{con}}\left(H^{(j)} - \sum_{j'=1}^{N} u_{j'k}^{\text{con}} H^{(j')}\right)\mathbf{U}^{(1k)}|\psi_0\rangle\Delta t. \end{aligned} \tag{C.28}$$

Because $\sum_{j=1}^{N} u_{jk}^{\text{con}} = 1$, we have

$$\sum_{j=1}^{N} u_{jk}^{\text{con}} \left(H^{(j)} - \sum_{j'=1}^{N} u_{j'k}^{\text{con}} H^{(j')} \right) = \sum_{j=1}^{N} u_{jk}^{\text{con}} H^{(j)} - \sum_{j'=1}^{N} u_{j'k}^{\text{con}} H^{(j')} = 0. \tag{C.29}$$

Therefore the right-hand side of the inequality (C.28) is 0, which leads to a contradiction. At each time step k, by our update rule in the algorithm, the change of objective value

$$R_{\hat{j}k} = \min_{j=1,\ldots,N} R_{jk} \leq 2C_1 e^{C_2 \Delta t} \Delta t^2. \tag{C.30}$$

Summing over all time steps, we have

$$\bar{F}(u^{\text{bin}}) - \bar{F}(u^{\text{con}}) = \sum_{k=1}^{T} \min_{j=1,\ldots,N} R_{jk} \leq 2C_1 e^{C_2 \Delta t} \Delta t. \tag{C.31}$$

□

Proof of Theorem 4.2.2. We first define the constants $C_1,\ C_2$. Similar to (C.11), for any time-dependent Hamiltonian matrix $A = -i \left(H^{(0)} + \sum_{j=1}^{N} u_j H^{(j)} \right)$, we now have $\sum_{j=1}^{N} u_j \leq N$ because $u_j \in [0,1],\ j = 1, \ldots, N$. Hence the maximum singular value

$$\sigma_1(A) \leq \sigma_1(H^{(0)}) + \sum_{j=1}^{N} u_j \sigma_1(H^{(j)}) \leq N \sigma_{\max}. \tag{C.32}$$

For energy objective function (4.2), we define constants $C_1,\ C_2$ are specified as

$$C_1 = \sigma_1(\tilde{H}) \left(N^2 \sigma_{\max}^2 + 2 + t_f N \sigma_{\max} \right) / |E_{\min}| \tag{C.33a}$$

$$C_2 = N \sigma_{\max}. \tag{C.33b}$$

For infidelity objective function (4.3), $C_1,\ C_2$ are specified as

$$C_1 = N^2 \sigma_{\max}^2, \quad C_2 = N \sigma_{\max}. \tag{C.33c}$$

With the same proof in Theorem 4.2.1, we show that the summation of terms with order $O(\Delta t^n),\ n \geq 2$ in R_{jk} is upper bounded by $C_1 e^{C_2 \Delta t} \Delta t^2$. We modify the proof of Theorem 4.2.1 to prove that the terms with respect to $O(\Delta t)$ in R_{jk} is upper bounded by

$C_0\epsilon^c(\Delta t)$ where C_0 is a constant parameter. For simplicity of the proof, we define ϵ_k as

$$\epsilon_k = \left|\sum_{j=1}^N u_{jk}^{\text{con}} - 1\right|, \ k = 1, \ldots, T. \tag{C.34}$$

We discuss the proof specifically for two objective functions following the sequence in the proof of Theorem 4.2.1.

Infidelity function We define the constant $C_0 = \sigma_{\max}$ and prove the conclusion by contradiction. Assume that for any controller j, it holds that

$$\begin{aligned} &\frac{1}{2^q}\left|\mathbf{tr}\left\{\mathbf{U}^{(k)}\left(I - iH^{(0)}\Delta t - i\sum_{j'=1}^N u_{j'k}^{\text{con}} H^{(j')}\Delta t\right)\right\}\right| \\ &-\frac{1}{2^q}\left|\mathbf{tr}\left\{\mathbf{U}^{(k)}\left(I - iH^{(0)}\Delta t - iH^{(j)}\Delta t\right)\right\}\right| > C_0\epsilon_k\Delta t. \end{aligned} \tag{C.35}$$

Now we have

$$\begin{aligned} 2^q C_0\epsilon_k\Delta t &= 2^q\sum_{j=1}^N \frac{u_{jk}^{\text{con}}}{\sum_{j'=1}^N u_{j'k}^{\text{con}}} C_0\epsilon_k\Delta t && \text{(C.36a)} \\ &< \left|\mathbf{tr}\left\{\mathbf{U}^{(k)}\left(I - iH^{(0)}\Delta t - i\sum_{j=1}^N u_{jk}^{\text{con}} H^{(j)}\Delta t\right)\right\}\right| \\ &\quad -\sum_{j=1}^N \frac{u_{jk}^{\text{con}}}{\sum_{j'=1}^N u_{j'k}^{\text{con}}}\left|\mathbf{tr}\left\{\mathbf{U}^{(k)}\left(I - iH^{(0)}\Delta t - iH^{(j)}\Delta t\right)\right\}\right| && \text{(C.36b)} \\ &\leq \left|\mathbf{tr}\left\{\mathbf{U}^{(k)}\left(-i\sum_{j=1}^N u_{jk}^{\text{con}} H^{(j)}\Delta t + i\sum_{j=1}^N \frac{u_{jk}^{\text{con}}}{\sum_{j'=1}^N u_{j'k}^{\text{con}}} H^{(j)}\Delta t\right)\right\}\right| && \text{(C.36c)} \\ &\leq \frac{\left|\sum_{j'=1}^N u_{j'k}^{\text{con}} - 1\right|}{\sum_{j'=1}^N u_{j'k}^{\text{con}}}\left|\mathbf{tr}\left\{\mathbf{U}^{(k)}\left(-i\sum_{j=1}^N u_{jk}^{\text{con}} H^{(j)}\Delta t\right)\right\}\right| && \text{(C.36d)} \\ &\leq \frac{\epsilon_k}{\sum_{j'=1}^N u_{j'k}^{\text{con}}} 2^q\sum_{j=1}^N u_{jk}^{\text{con}}\sigma_{\max}\Delta t = 2^q\sigma_{\max}\epsilon_k\Delta t = 2^q C_0\epsilon_k\Delta. && \text{(C.36e)} \end{aligned}$$

The inequality (C.36b) comes from the weighted summation of (C.35) over all the controllers with weight $u_{jk}/\sum_{j'=1}^N u_{j'k}$ for controller $j = 1, \ldots, N$. The inequalities (C.36c)–(C.36d) follow from the norm inequality and the fact that the summation of all the weights is one. The last inequality follows from the definition of ϵ_k and Lemma 4.2.4. By substituting C_0, the inequalities lead to a contradiction, which means that at each step, the minimum change of objective value

$$\min_{j=1,\dots,N} R_{jk} \leq 2C_1 e^{C_2\Delta t}\Delta t^2 + C_0\epsilon_k\Delta t. \tag{C.37}$$

Taking summation over all the time steps, we have

$$\bar{F}(u^{\text{bin}}) - \bar{F}(u^{\text{con}}) = \sum_{k=1}^{T} \min_{j=1,\dots,N} R_{jk} \leq 2C_1 e^{C_2\Delta t}\Delta t + C_0\epsilon^c(\Delta t). \tag{C.38}$$

Energy function We define $C_0 = 2\sigma_1(\tilde{H})\sigma_{\max}/|E_{\min}|$ and prove the statement by contradiction. Assume that for any controller j, it holds that

$$\begin{aligned}&\frac{1}{E_{\min}}\langle\psi_0|\left(\mathbf{U}^{(2k)}i\left(H^{(j)} - \sum_{j'=1}^{N} u_{j'k}^{\text{con}} H^{(j')}\right)\mathbf{U}^{(1k)}\right)^{\dagger}\tilde{H}\mathbf{U}^{(2k)}\mathbf{U}^{(1k)}|\psi_0\rangle\Delta t\\ &+\frac{1}{E_{\min}}\langle\psi_0|\left(\mathbf{U}^{(2k)}\mathbf{U}^{(1k)}\right)^{\dagger}\tilde{H}\mathbf{U}^{(2k)}i\left(H^{(j)} - \sum_{j'=1}^{N} u_{j'k}^{\text{con}} H^{(j')}\right)\mathbf{U}^{(1k)}|\psi_0\rangle\Delta t > C_0\epsilon_k\Delta t.\end{aligned} \tag{C.39}$$

Then we have

$$C_0\epsilon_k\Delta t = \sum_{j=1}^{N}\frac{u_{jk}^{\text{con}}}{\sum_{j'=1}^{N} u_{j'k}^{\text{con}}} C_0\epsilon_k\Delta t \tag{C.40a}$$

$$\begin{aligned}<&\frac{\sum_{j=1}^{N} u_{jk}^{\text{con}}}{E_{\min}}\langle\psi_0|\left(\mathbf{U}^{(2k)}i\frac{1-\sum_{j'=1}^{N} u_{j'k}^{\text{con}}}{\sum_{j'=1}^{N} u_{j'k}^{\text{con}}}H^{(j)}\mathbf{U}^{(1k)}\right)^{\dagger}\tilde{H}\mathbf{U}^{(2k)}\mathbf{U}^{(1k)}|\psi_0\rangle\Delta t\\ &+\frac{\sum_{j=1}^{N} u_{jk}^{\text{con}}}{E_{\min}}\langle\psi_0|\left(\mathbf{U}^{(2k)}\mathbf{U}^{(1k)}\right)^{\dagger}\tilde{H}\mathbf{U}^{(2k)}i\frac{1-\sum_{j'=1}^{N} u_{j'k}^{\text{con}}}{\sum_{j'=1}^{N} u_{j'k}^{\text{con}}}H^{(j)}\mathbf{U}^{(1k)}|\psi_0\rangle\Delta t\end{aligned} \tag{C.40b}$$

$$\leq\frac{2}{|E_{\min}|\sum_{j'=1}^{N} u_{j'k}^{\text{con}}}\epsilon_k\sum_{j=1}^{N} u_{jk}^{\text{con}}\sigma_{\max}\sigma_1(\tilde{H})\Delta t = \frac{2}{|E_{\min}|}\sigma_{\max}\sigma_1(\tilde{H})\epsilon_k\Delta t = C_0\epsilon_k\Delta t. \tag{C.40c}$$

The inequality (C.40b) comes from the weighted summation of (C.39) over all the controllers with weight $u_{jk}/\sum_{j=1}^{N} u_{jk}$ for controller $j = 1,\dots,N$. Notice that we use the fact that the summation of all the weights is one. The last inequality (C.40c) follows from the definition of ϵ_k and Lemma 4.2.4. By substituting C_0, the inequalities lead to a contradiction, which means that at each step, the minimum change of objective value

$$\min_{j=1,\dots,N} R_{jk} \leq 2C_1 e^{C_2\Delta t}\Delta t^2 + C_0\epsilon_k\Delta t. \tag{C.41}$$

Taking summation over all the time steps, we have

$$\bar{F}(u^{\mathrm{bin}}) - \bar{F}(u^{\mathrm{con}}) = \sum_{k=1}^{T} \min_{j=1,\dots,N} R_{jk} \leq 2C_1 e^{C_2 \Delta t} \Delta t + C_0 \epsilon^c(\Delta t). \tag{C.42}$$

□

C.2 Detailed Numerical Results

Detailed Results for Sensitivity Analysis For the sensitivity analysis in Section 4.4.2, we present the detailed objective values for 5 instances in Table C.1. We present the detailed TV-norm values for 5 instances in Table C.2.

Table C.1: Objective values for Energy6 example solved by Algorithm 4.1 using Algorithm 4.3 to round continuous controls with various switching penalty parameter α.

	First-excited	0	0.001	0.003	0.005	0.01	0.015	0.02
Instance 1	0.2263	0.0379	0.0380	0.0383	0.0391	0.0461	0.0700	0.1352
Instance 2	0.1431	0.0196	0.0197	0.0205	0.0208	0.0316	0.0444	0.0444
Instance 3	0.1805	0.0453	0.0455	0.0461	0.0470	0.0486	0.0517	0.0690
Instance 4	0.2698	0.0067	0.0069	0.0077	0.0082	0.0309	0.0309	0.0666
Instance 5	0.0675	0.0501	0.0502	0.0507	0.0510	0.0536	0.0662	0.0966
Average	0.1787	0.0319	0.0321	0.0327	0.0332	0.0422	0.0526	0.0824
	0.03	0.04	0.05	0.07	0.1	0.2	0.3	0.6
Instance 1	0.1352	0.1352	0.2439	0.2439	0.3616	0.3808	1.0000	1.0000
Instance 2	0.0693	0.1256	0.1256	0.1256	0.1963	0.5159	0.8531	1.0000
Instance 3	0.0947	0.1375	0.1375	0.1375	0.2904	0.5232	0.9907	1.0000
Instance 4	0.1356	0.1356	0.1356	0.3082	0.3082	0.4680	1.0000	1.0000
Instance 5	0.0966	0.1528	0.1528	0.2391	0.2391	0.3724	0.3724	1.0000
Average	0.1063	0.1373	0.1591	0.2109	0.2791	0.4521	0.8432	1.0000

Detailed Results for Objective We present the detailed numerical results for objective values and TV-norm values for all the methods and instances in Table C.3. We present the best discretized control obtained by Algorithm TR+MT+ALB in Chapter 3 in column "Chpt. 3" as a baseline. Columns "Alg. 4.1w/Chpt. 3", "Alg. 4.1w/4.3", and "Alg. 4.1w/4.4" represent results obtained by Algorithm 4.1 with binary control obtained by Algorithm TR+MT+ALB in Chapter 3, the heuristic method based on objective value (Algorithm 4.3), and the heuristic method based on the cumulative difference (Algorithm 4.4), respectively.

Table C.2: TV-norm results for Energy6 example solved by Algorithm 4.1 using Algorithm 4.3 to round continuous controls with various switching penalty parameter α. TV-norm values of the first-excited state are marked by ”-”.

	First-excited	0	0.001	0.003	0.005	0.01	0.015	0.02
Instance 1	-	98	74	42	30	24	20	16
Instance 2	-	114	70	38	34	24	20	20
Instance 3	-	118	82	46	34	26	22	20
Instance 4	-	130	62	34	30	20	20	16
Instance 5	-	122	82	46	34	22	20	16
Average	-	116.4	74.0	41.2	32.4	23.2	20.4	17.6
	0.03	0.04	0.05	0.07	0.1	0.2	0.3	0.6
Instance 1	16	16	12	12	8	6	0	0
Instance 2	16	12	12	12	10	4	4	0
Instance 3	16	12	12	12	8	4	4	0
Instance 4	12	12	12	8	8	6	0	0
Instance 5	16	12	12	8	8	4	4	0
Average	15.2	12.8	12.0	10.4	8.4	4.8	2.4	0.0

Table C.3: Objective and TV-norm value results of various approaches. Column “Chpt. 3” represent the results of the benchmark discretized controls without optimizing switching points. Columns “Alg. 4.1w/Chpt. 3”, “Alg. 4.1w/4.3”, and “Alg. 4.1w/4.4” represent the results of Algorithm 4.1 with extracting binary controls by various approaches, including TR+MT+ALB in Chapter 3, Algorithm 4.3 and Algorithm 4.4.

	(Objective value, TV-norm value)			
	Chpt. 3	Alg. 4.1w/Chpt. 3	Alg. 4.1w/4.3	Alg. 4.1w/4.4
Energy2	(2.930E−03, 10)	(1.255E−14, 4)	(1.255E−14, 4)	(1.255E−14, 4)
Energy4	(0.1986, 6)	(0.1611, 6)	(0.1569, 9.2)	(0.1568, 8.4)
Energy6	(0.3046, 10.8)	(0.2049, 10.8)	(0.0526, 20.4)	(0.0632, 19.6)
CNOT5	(0.1968, 10)	(0.1807, 10)	(0.1763, 11)	(0.1792, 9)
CNOT10	(9.431E−03, 16)	(5.071E−03, 16)	(5.326E−07, 28)	(2.163E−07, 26)
CNOT20	(8.224E−04, 49)	(3.124E−08, 49)	(1.728E−07, 41)	(3.136E−07, 39)
NOT2	(0.1636, 1)	(0.1632, 1)	(0.1632, 1)	(0.1632, 1)
NOT6	(1.270E−02, 11)	(6.885E−06, 11)	(1.122E−06, 0)	(3.267E−06, 14)
NOT10	(9.087E−04, 20)	(5.132E−08, 20)	(2.439E−08, 11)	(4.692E−08, 17)
CircuitH2	(7.777E−02, 8)	(2.747E−03, 8)	(1.208E−06, 16)	(6.612E−05, 36)
CircuitLiH	(8.110E−01, 18)	(4.054E−02, 18)	(1.702E−03, 32)	(1.986E−03, 20)
CircuitBeH2	(5.115E−02, 24)	(1.250E−03, 24)	(1.250E−03, 28)	(1.250E−03, 14)

APPENDIX D

Appendix For Chapter 5

D.1 Proofs of All Theorems in Section 5.3

We present the proofs for all the theorems and propositions in Section 5.3.

D.1.1 Proofs of Theorems in Section 5.3.1

Proof of Theorem 5.3.1. First, note that the two forms (5.10a) and (5.10b) are equivalent because scenarios s with $F^s(u) - F^{s^*(u)}(u) = 0$ contribute nothing to the summation, and thus we only need to prove (5.10a). To prove the closed-form expression, it is equivalent to prove that given a control variable u, the optimal solution for the minimization problem $\min_\zeta F_{\text{CVaR}}(u,\zeta)$ is $\zeta^*(u) = F^{s^*(u)}(u)$. When $\zeta < \zeta^*(u) = F^{s^*(u)}(u)$, we have

$$\begin{aligned}
F_{\text{CVaR}}(u,\zeta) =&\zeta + \frac{1}{\eta}\sum_{s=1}^{S} p_s \max\{0, F^s(u) - \zeta\} = \zeta + \frac{1}{\eta}\sum_{s:\zeta<F^s(u)} p_s(F^s(u) - \zeta)\\
=&F_{\text{CVaR}}(u,\zeta^*(u)) + \zeta - \zeta^*(u) + \frac{1}{\eta}\sum_{s:\zeta<F^s(u)\le\zeta^*(u)} p_s(F^s(u)-\zeta)\\
&+ (\zeta^*(u)-\zeta)\frac{1}{\eta}\sum_{s:F^s(u)>\zeta^*(u)} p_s\\
\ge&F_{\text{CVaR}}\left(u,\zeta^*(u)\right) + \frac{1}{\eta}\sum_{s:\zeta<F^s(u)\le\zeta^*(u)} p_s(F^s(u)-\zeta) \ge F_{\text{CVaR}}(u,\zeta^*(u)).
\end{aligned} \tag{D.1}$$

The equalities directly follow the definition of $F_{\text{CVaR}}(u,\zeta)$. The first inequality holds because of the definition of $s^*(u)$ such that $\sum_{s:F^s(u)>\zeta^*(u)} p_s \ge \eta$. The last inequality holds because all the terms in the summation have $F^s(u) > \zeta$. Similarly, we can show that when $\zeta >$

$\zeta^*(u) = F^{s^*(u)}(u)$, we have

$$\begin{aligned}
F_{\text{CVaR}}(u,\zeta) =& \zeta + \frac{1}{\eta}\sum_{s=1}^{S} p_s \max\{0, F^s(u)-\zeta\} = \zeta + \frac{1}{\eta}\sum_{s:\zeta<F^s(u)} p_s(F^s(u)-\zeta) \\
=& F_{\text{CVaR}}(u,\zeta^*(u)) + \zeta - \zeta^*(u) + \frac{1}{\eta}\sum_{s:\zeta^*(u)<F^s(u)\leq\zeta} p_s(\zeta - F^s(u)) \\
&+ (\zeta^*(u)-\zeta)\frac{1}{\eta}\sum_{s:F^s(u)>\zeta^*(u)} p_s \\
\geq& F_{\text{CVaR}}(u,\zeta^*(u)) + \frac{1}{\eta}\sum_{s:\zeta^*(u)<F^s(u)\leq\zeta} p_s(\zeta - F^s(u)) \geq F_{\text{CVaR}}(u,\zeta^*(u)). \qquad \text{(D.2)}
\end{aligned}$$

The only difference is that for the last inequality, it holds because all the terms in the summation have $F^s(u) \leq \zeta$. This completes the proof. □

Proof of Theorem 5.3.2. Because the functions $F_X(X_T)$ and $X_T^s(u)$ are differentiable for each scenario $s = 1, \ldots, S$, the function $F^s(u)$ is differentiable, and thus continuous. Because the objective function $F^s(u)$ is continuous, given a control variable point $\hat{u}$, for any $\omega > 0$, there exists a distance $r_s(\hat{u},\omega)$ for each scenario s such that for any $\|u-\hat{u}\| \leq r_s(\hat{u},\omega)$, we have $|F^s(u) - F^s(\hat{u})| < \omega$. Choosing $\omega = \min_{s\neq s^*(\hat{u})} |F^s(\hat{u}) - F^{s^*(\hat{u})}(\hat{u})|$, from the assumption that $\forall s \neq s^*(\hat{u})$, $F^s(\hat{u}) \neq F^{s^*(\hat{u})}(\hat{u})$, we have $\omega > 0$. Define $r = \min_{s=1,\ldots,S} r_s(\hat{u}, \omega/2)$. Then, for any u such that $|u - \hat{u}| \leq r$, $s^*(u) = s^*(\hat{u})$. We prove this claim by the following statements that

$$\begin{aligned}
F^s(u) > F^s(\hat{u}) - \frac{\omega}{2} &\geq F^{s^*(\hat{u})}(\hat{u}) + \omega - \frac{\omega}{2} \\
&= F^{s^*(\hat{u})}(\hat{u}) + \frac{\omega}{2} > F^{s^*(\hat{u})}(u),\ \forall s: F^s(\hat{u}) > F^{s^*(\hat{u})}(\hat{u}) \qquad \text{(D.3)} \\
F^s(u) < F^s(\hat{u}) + \frac{\omega}{2} &\leq F^{s^*(\hat{u})}(\hat{u}) - \omega + \frac{\omega}{2} \\
&= F^{s^*(\hat{u})}(\hat{u}) - \frac{\omega}{2} < F^{s^*(\hat{u})}(u),\ \forall s: F^s(\hat{u}) < F^{s^*(\hat{u})}(\hat{u}). \qquad \text{(D.4)}
\end{aligned}$$

For both formulas, the first and last inequalities follow the continuity of $F^s(u)$ and the other inequalities follow from the definition of ω. Now we show that $s^*(\hat{u})$ is still the scenario number with the largest original objective value such that $\sum_{s=1}^{S} p_s \nVdash_{\{F^s(u) > F^{s^*(u)}(u)\}} \geq \eta$, which means that $s^*(u) = s^*(\hat{u})$. Furthermore, we also show that $\{s : F^s(u) > F^{s^*(u)}(u)\} = \{s : F^s(\hat{u}) > F^{s^*(\hat{u})}(\hat{u})\}$. Therefore, the derivative of $F_C(u)$ at point $\hat{u}$ is

$$\frac{\partial F_C(\hat{u})}{\partial \hat{u}} = \lim_{u\to\hat{u}} \frac{F_C(u) - F_C(\hat{u})}{u - \hat{u}}$$

$$= \lim_{u \to \hat{u}} \left(\frac{F^{s^*(\hat{u})}(u) - F^{s^*(\hat{u})}(\hat{u})}{u - \hat{u}} + \frac{1}{\eta} \sum_{s: F^s(u) > F^{s^*(\hat{u})}(u)} p_s \frac{F^s(u) - F^s(\hat{u}) - F^{s^*(\hat{u})}(u) + F^{s^*(\hat{u})}(\hat{u})}{u - \hat{u}} \right)$$

$$= \left(1 - \frac{1}{\eta} \sum_{s: F^s(\hat{u}) > F^{s^*(\hat{u})}(\hat{u})} p_s \right) \frac{\partial F^{s^*(\hat{u})}(\hat{u})}{\partial \hat{u}} + \frac{1}{\eta} \sum_{s: F^s(\hat{u}) > F^{s^*(\hat{u})}(\hat{u})} p_s \frac{\partial F^s(\hat{u})}{\partial \hat{u}}. \tag{D.5}$$

This completes the proof. □

D.1.2 Proofs of Theorems in Section 5.3.2

Proof of Proposition 5.3.1. We only need to prove that there exists a constant C_{diff} such that $\sqrt{t_f F_L(u^{\text{con}})} \leq C_{\text{diff}} \sqrt{\Delta t} / \sqrt{\theta}$. We define $u^{\text{con,SOS1}}$ as the optimal solution for the continuous relaxation of the discretized model with the SOS1 property (SP($\mathcal{S}$)). From the optimality of u^{con}, we have

$$\tilde{F}(u^{\text{con}}) + \theta F_L(u^{\text{con}}) \leq \tilde{F}(u^{\text{con,SOS1}}) + \theta F_L(u^{\text{con,SOS1}}) = \tilde{F}(u^{\text{con,SOS1}}), \tag{D.6}$$

where the last inequality follows the fact that $u^{\text{con,SOS1}}$ holds the SOS1 property, so the penalty term $F_L(u^{\text{con,SOS1}}) = 0$. Combining with the Assumption 5.3.1 that $\tilde{F}(u^{con}) \geq 0$, we have

$$F_L(u^{\text{con}}) \leq \frac{1}{\theta} \tilde{F}(u^{\text{con,SOS1}}). \tag{D.7}$$

We then consider the difference in objective values between the optimal infinite dimension relaxation solution $u^{*,\text{SOS1}}(t)$ (defined in Assumption 5.3.3) and the optimal discretized relaxation solution $u^{\text{con,SOS1}}$. We first construct a piece-wise constant control function $u^{d,\text{SOS1}}(t)$ satisfying the inequality as:

$$u^{d,\text{SOS1}}(t) = \frac{1}{\Delta t} \int_{(k-1)\Delta t}^{k\Delta t} u^{*,\text{SOS1}}(\tau) d\tau, \ \forall t \in [(k-1)\Delta t, k\Delta t), \ k = 1, \ldots, T. \tag{D.8}$$

It is obvious that during each time interval, we have

$$\int_{(k-1)\Delta t}^{k\Delta t} u^{d,\text{SOS1}}(\tau) d\tau = \int_{(k-1)\Delta t}^{k\Delta t} u^{*,\text{SOS1}}(\tau) d\tau, \ k = 1, \ldots, T. \tag{D.9}$$

For any time $t \in [0, t_f]$, let $\hat{k}$ be the index of time interval that t falls in, we have

$$\left\| \int_0^t u^{d,\mathrm{SOS1}}(\tau) d\tau - \int_0^t u^{*,\mathrm{SOS1}}(\tau) d\tau \right\|_\infty \leq \left\| \int_{(\hat{k}-1)\Delta t}^t u^{d,\mathrm{SOS1}}(\tau) d\tau - \int_{(\hat{k}-1)\Delta t}^t u^{*,\mathrm{SOS1}}(\tau) d\tau \right\|_\infty \tag{D.10}$$

In the time subinterval $[(\hat{k}-1)\Delta t, \hat{k}\Delta t]$, the two integrals hold

$$\int_{(\hat{k}-1)\Delta t}^t u^{d,\mathrm{SOS1}}(\tau) d\tau \leq \max_{\tau \in [(\hat{k}-1)\Delta t, \hat{k}\Delta t]} u^{d,\mathrm{SOS1}}(\tau)\Delta t \tag{D.11a}$$

$$\int_{(\hat{k}-1)\Delta t}^t u^{*,\mathrm{SOS1}}(\tau) d\tau \geq \min_{\tau \in [(\hat{k}-1)\Delta t, \hat{k}\Delta t]} u^{*,\mathrm{SOS1}}(\tau)\Delta t \tag{D.11b}$$

From the definition of $u^{d,\mathrm{SOS1}}$ in (D.8), we know that

$$\max_{\tau \in [(\hat{k}-1)\Delta t, \hat{k}\Delta t]} u^{d,\mathrm{SOS1}}(\tau) \leq \max_{\tau \in [(\hat{k}-1)\Delta t, \hat{k}\Delta t]} u^{*,\mathrm{SOS1}}(\tau). \tag{D.12}$$

Therefore, for any $t \in [0, t_f]$,

$$\begin{aligned} &\left\| \int_0^t u^{d,\mathrm{SOS1}}(\tau) d\tau - \int_0^t u^{*,\mathrm{SOS1}}(\tau) d\tau \right\|_\infty \\ \leq &\left\| \max_{\tau \in [(\hat{k}-1)\Delta t, \hat{k}\Delta t]} u^{*,\mathrm{SOS1}}(\tau) - \min_{\tau \in [(\hat{k}-1)\Delta t, \hat{k}\Delta t]} u^{*,\mathrm{SOS1}}(\tau) \right\|_\infty \Delta t. \end{aligned} \tag{D.13}$$

We notice that the values of control functions $u^{*,\mathrm{SOS1}}(t)$ and $u^{d,\mathrm{SOS1}}(t)$ are both bounded by $[0, 1]$. Hence the difference of integral is upper bounded by Δt. From Theorem 2 in Sager and Zeile [183], we have

$$\left\| X(t_f; u^{*,\mathrm{SOS1}}) - X(t_f; u^{d,\mathrm{SOS1}}) \right\| \leq C' \Delta t, \tag{D.14}$$

where C' is a constant determined by control Hamiltonians and evolution time. Combining with the continuity of the objective function $\tilde{F}(u(t))$, we have

$$\tilde{F}(u^{d,\mathrm{SOS1}}(t)) - \tilde{F}(u^{*,\mathrm{SOS1}}(t)) \leq C'' \Delta t, \tag{D.15}$$

where C'' is a constant determined by objective function $\tilde{F}(u)$. From the definition of the piece-wise constant control function $u^{\mathrm{d,SOS1}}(t)$, we can construct an equivalent discretized solution $u_k^{\mathrm{d,SOS1}} = u^{\mathrm{d,SOS1}}(t)$ where k is the index of time interval that t falls in. Because

$u^{con,\mathrm{SOS1}}$ is the optimal solution of the discretized formulation, it holds that

$$\begin{aligned}\tilde{F}(u^{con,\mathrm{SOS1}}) - \tilde{F}(u^{*,\mathrm{SOS1}}(t)) &\leq \tilde{F}(u^{d,\mathrm{SOS1}}) - \tilde{F}(u^{*,\mathrm{SOS1}}(t)) \\ &= \tilde{F}(u^{d,\mathrm{SOS1}}(t)) - \tilde{F}(u^{*,\mathrm{SOS1}}(t)) \leq C''\Delta t. \end{aligned} \tag{D.16}$$

With Assumption 5.3.3 that $\tilde{F}(u^{*,\mathrm{SOS1}}(t)) = 0$ and the upper bound for $F_L(u^{\mathrm{con}})$ in (D.7), we prove that

$$\sqrt{t_f F_L(u^{\mathrm{con}})} \leq C_{\mathrm{diff}}\sqrt{\Delta t}/\sqrt{\theta}, \tag{D.17}$$

where $C_{\mathrm{diff}} = \sqrt{t_f C''}$. □

Proof of Proposition 5.3.2. We consider the upper bound for the cumulative difference between $u^{*,\mathrm{SOS1}}(t)$ and $u^{d,\mathrm{SOS1}}(t)$ in (D.13). Because we assume that $u^{*,\mathrm{SOS1}}(t)$ is continuous in each subinterval in Assumption 5.3.4, the right-hand side is upper bounded by $o(\Delta t)\Delta t$. The other parts of the proof are the same as the proof of Proposition 5.3.1. □

BIBLIOGRAPHY

[1] Baher Abdulhai, Rob Pringle, and Grigoris J Karakoulas. 2003. Reinforcement learning for true adaptive traffic signal control. *Journal of Transportation Engineering* 129, 3 (2003), 278–285.

[2] Konstantinos Aboudolas, Markos Papageorgiou, and E Kosmatopoulos. 2009. Store-and-forward based methods for the signal control problem in large-scale congested urban road networks. *Transportation Research Part C: Emerging Technologies* 17, 2 (2009), 163–174.

[3] Ludovica Adacher and Marco Tiriolo. 2018. A macroscopic model with the advantages of microscopic model: A review of Cell Transmission Model's extensions for urban traffic networks. *Simulation Modelling Practice and Theory* 86 (2018), 102–119.

[4] SMA Bin Al Islam and Ali Hajbabaie. 2017. Distributed coordinated signal timing optimization in connected transportation networks. *Transportation Research Part C: Emerging Technologies* 80 (2017), 272–285.

[5] Gadi Aleksandrowicz, Thomas Alexander, Panagiotis Barkoutsos, Luciano Bello, Yael Ben-Haim, David Bucher, Francisco Jose Cabrera-Hernández, Jorge Carballo-Franquis, Adrian Chen, Chun-Fu Chen, et al. 2019. Qiskit: An open-source framework for quantum computing. *Accessed on: Mar* 16 (2019). https://doi.org/10.5281/ZENODO.2562110

[6] Richard E Allsop. 1968. Selection of offsets to minimize delay to traffic in a network controlled by fixed-time signals. *Transportation Science* 2, 1 (1968), 1–13.

[7] Richard E Allsop. 1971. Delay-minimizing settings for fixed-time traffic signals at a single road junction. *IMA Journal of Applied Mathematics* 8, 2 (1971), 164–185.

[8] Neculai Andrei. 2017. A SQP algorithm for large-scale constrained optimization: SNOPT. In *Continuous nonlinear optimization for engineering applications in GAMS technology*. Springer, 317–330.

[9] Aleksandr Y. Aravkin, Robert Baraldi, and Dominique Orban. 2022. A Proximal Quasi-Newton Trust-Region Method for Nonsmooth Regularized Optimization. *SIAM Journal on Optimization* 32, 2 (2022), 900–929. https://doi.org/10.1137/21m1409536

[10] Necdet Serhat Aybat and Erfan Yazdandoost Hamedani. 2019. A distributed ADMM-like method for resource sharing over time-varying networks. *SIAM Journal on Optimization* 29, 4 (2019), 3036–3068.

[11] Seraph Bao, Silken Kleer, Ruoyu Wang, and Armin Rahmani. 2018. Optimal control of superconducting gmon qubits using Pontryagin's minimum principle: Preparing a maximally entangled state with singular bang-bang protocols. *Physical Review A* 97, 6 (2018), 062343. https://doi.org/10.1103/physreva.97.062343

[12] Aniruddha Bapat and Stephen Jordan. 2018. Bang-bang control as a design principle for classical and quantum optimization algorithms. (2018). https://doi.org/10.48550/ARXIV.1812.02746

[13] Aniruddha Bapat and Stephen Jordan. 2019. Bang-bang control as a design principle for classical and quantum optimization algorithms. *Quantum Information & Computation* 19 (2019), 424–446.

[14] Robert Barr, Yasuo Oda, Gregory Quiroz, B David Clader, and Leigh M Norris. 2022. Qubit control noise spectroscopy with optimal suppression of dephasing. *Physical Review A* 106, 2 (2022), 022425.

[15] Rodney J Bartlett and Monika Musiał. 2007. Coupled-cluster theory in quantum chemistry. *Reviews of Modern Physics* 79, 1 (2007), 291. https://doi.org/10.1103/RevModPhys.79.291

[16] Pietro Belotti. 2009. *Couenne: A user's manual.* Technical Report. Technical report, Lehigh University.

[17] Pietro Belotti, Christian Kirches, Sven Leyffer, Jeff Linderoth, James Luedtke, and Ashutosh Mahajan. 2013. Mixed-integer nonlinear optimization. *Acta Numerica* 22 (2013), 1–131.

[18] Jacques F Benders. 1962. Partitioning procedures for solving mixed-variables programming problems. *Numer. Math* 4, 1 (1962), 238–252.

[19] Michel Bergmann and Laurent Cordier. 2008. Optimal control of the cylinder wake in the laminar regime by trust-region methods and POD reduced-order models. *J. Comput. Phys.* 227, 16 (2008), 7813–7840.

[20] Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermanni Heimonen, Jakob S. Kottmann, Tim Menke, Wai-Keong Mok, Sukin Sim, Leong-Chuan Kwek, and Alán Aspuru-Guzik. 2022. Noisy intermediate-scale quantum algorithms. *Reviews of Modern Physics* 94, 1 (2022). https://doi.org/10.1103/revmodphys.94.015004

[21] Silvio Binato, Mário Veiga F Pereira, and Sérgio Granville. 2001. A new Benders decomposition approach to solve power transmission network design problems. *IEEE Transactions on Power Systems* 16, 2 (2001), 235–240.

[22] Brian Bing and Alan Carter. 1995. SCOOT: The world's foremost adaptive TRAFFIC control system. *TRAFFIC TECHNOLOGY INTERNATIONAL'95* (1995).

[23] John R Birge and Francois Louveaux. 2011. *Introduction to stochastic programming.* Springer Science & Business Media.

[24] Pierre Bonami, Lorenz T. Biegler, Andrew R. Conn, Gérard Cornuéjols, Ignacio E. Grossmann, Carl D. Laird, Jon Lee, Andrea Lodi, François Margot, Nicolas Sawaya, and Andreas Wächter. 2008. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete optimization* 5, 2 (2008), 186–204. https://doi.org/10.1016/j.disopt.2006.10.011

[25] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning* 3, 1 (2011), 1–122.

[26] Lucas T Brady, Christopher L Baldwin, Aniruddha Bapat, Yaroslav Kharkov, and Alexey V Gorshkov. 2021. Optimal Protocols in Quantum Annealing and QAOA Problems. *Physical Review Letters* 126 (2021), 070505.

[27] Lucas T. Brady, Lucas Kocia, Przemyslaw Bienias, Yaroslav Kharkov Aniruddha Bapat, and Alexey V. Gorshkov. 2021. Behavior of Analog Quantum Algorithms. *arXiv preprint arXiv:2107.01218* (2021).

[28] Lucas T. Brady, Lucas Kocia, Przemyslaw Bienias, Aniruddha Bapat, Yaroslav Kharkov, and Alexey V. Gorshkov. 2021. Behavior of Analog Quantum Algorithms. https://doi.org/10.48550/ARXIV.2107.01218

[29] Constantin Brif, Raj Chakrabarti, and Herschel Rabitz. 2010. Control of quantum phenomena: Past, present and future. *New Journal of Physics* 12 (2010), 075008.

[30] Arthur Earl Bryson. 2018. *Applied optimal control: optimization, estimation and control.* Routledge.

[31] Zdzislaw Bubnicki. 2005. *Modern control theory.* Springer.

[32] Marin Bukov, Alexandre GR Day, Dries Sels, Phillip Weinberg, Anatoli Polkovnikov, and Pankaj Mehta. 2018. Reinforcement learning in different phases of quantum control. *Physical Review X* 8, 3 (2018), 031086.

[33] John Charles Butcher. 2016. *Numerical Methods for Ordinary Differential Equations.* John Wiley & Sons.

[34] Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. 1995. A limited memory algorithm for bound constrained optimization. *SIAM Journal on scientific computing* 16, 5 (1995), 1190–1208.

[35] Richard H Byrd, Robert B Schnabel, and Gerald A Shultz. 1987. A trust region algorithm for nonlinearly constrained optimization. *SIAM J. Numer. Anal.* 24, 5 (1987), 1152–1170.

[36] Tommaso Caneva, Tommaso Calarco, and Simone Montangero. 2011. Chopped random-basis quantum optimization. *Physical Review A* 84, 2 (2011), 022326. https://doi.org/10.1103/physreva.84.022326

[37] M. Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles. 2021. Variational quantum algorithms. *Nature Reviews Physics* 3, 9 (2021), 625–644. https://doi.org/10.1038/s42254-021-00348-9

[38] Chunlin Chen, Daoyi Dong, Han-Xiong Li, Jian Chu, and Tzyh-Jong Tarn. 2013. Fidelity-based probabilistic Q-learning for control of quantum systems. *IEEE transactions on neural networks and learning systems* 25, 5 (2013), 920–933.

[39] Chunlin Chen, Daoyi Dong, Ruixing Long, Ian R Petersen, and Herschel A Rabitz. 2014. Sampling-based learning control of inhomogeneous quantum ensembles. *Physical Review A* 89, 2 (2014), 023402.

[40] Yu Chen, C Neill, P Roushan, N Leung, M Fang, R Barends, J Kelly, B Campbell, Z Chen, B Chiaro, et al. 2014. Qubit architecture with high coherence and fast tunable coupling. *Physical review letters* 113, 22 (2014), 220502. https://doi.org/10.1103/PhysRevLett.113.220502

[41] Suh-Wen Chiou. 2019. A two-stage model for period-dependent traffic signal control in a road networked system with stochastic travel demand. *Information Sciences* 476 (2019), 256–273.

[42] Thomas F Coleman and Aiping Liao. 1995. An efficient trust region method for unconstrained discrete-time optimal control problems. *Computational optimization and applications* 4, 1 (1995), 47–66.

[43] Gurcan Comert and Mecit Cetin. 2011. Analytical evaluation of the error in queue length estimation at traffic signals from probe vehicle data. *IEEE Transactions on Intelligent Transportation Systems* 12, 2 (2011), 563–573.

[44] Laurent Condat. 2013. A direct algorithm for 1-D total variation denoising. *IEEE Signal Processing Letters* 20, 11 (2013), 1054–1057.

[45] Shuang Cong, Fangfang Meng, et al. 2013. A survey of quantum lyapunov control methods. *The Scientific World Journal* 2013 (2013).

[46] Shuang Cong, Lei Zhou, and Fangfang Meng. 2020. Lyapunov-based unified control method for closed quantum systems. *Journal of the Franklin Institute* 357, 14 (2020), 9220–9247.

[47] Andrew R Conn, Nicholas IM Gould, and Philippe L Toint. 2000. *Trust region methods.* SIAM.

[48] Joseph Czyzyk, Michael P. Mesnier, and Jorge J. Moré. 1998. The NEOS Server. *IEEE Journal on Computational Science and Engineering* 5, 3 (1998), 68–75.

[49] Carlos Daganzo. 1992. The cell transmission model. Part I: A simple dynamic representation of highway traffic. (1992).

[50] M Dahleh, AP Peirce, and H Rabitz. 1990. Optimal control of uncertain quantum systems. *Physical Review A* 42, 3 (1990), 1065.

[51] Domenico d'Alessandro. 2021. *Introduction to quantum control and dynamics*. Chapman and hall/CRC.

[52] Pierre de Fouquieres, Sophie G Schirmer, Steffen J Glaser, and Ilya Kuprov. 2011. Second order gradient ascent pulse engineering. *Journal of Magnetic Resonance* 212, 2 (2011), 412–417.

[53] Elizabeth D. Dolan. 2001. *The NEOS Server 4.0 Administrative Guide*. Technical Memorandum ANL/MCS-TM-250. Mathematics and Computer Science Division, Argonne National Laboratory.

[54] Florian Dolde, Ville Bergholm, Ya Wang, Ingmar Jakobi, Boris Naydenov, Sébastien Pezzagna, Jan Meijer, Fedor Jelezko, Philipp Neumann, Thomas Schulte-Herbrüggen, Jacob Biamonte, and Jörg Wrachtrup. 2014. High-fidelity spin entanglement using optimal control. *Nature Communications* 5, 3371 (2014).

[55] Daoyi Dong. 2021. Learning control of quantum systems. *arXiv preprint arXiv:2101.07461* (2021).

[56] Patrick Doria, Tommaso Calarco, and Simone Montangero. 2011. Optimal Control Technique for Many-Body Quantum Dynamics. *Phys. Rev. Lett.* 106 (May 2011), 190501. Issue 19. https://doi.org/10.1103/PhysRevLett.106.190501

[57] Magnus Egerstedt, Yorai Wardi, and Henrik Axelsson. 2006. Transition-time optimization for switched-mode dynamical systems. *IEEE Trans. Automat. Control* 51, 1 (2006), 110–115.

[58] Tomaso Erseghe. 2014. Distributed optimal power flow using ADMM. *IEEE transactions on power systems* 29, 5 (2014), 2370–2380.

[59] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. 2014. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028* (2014).

[60] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. 2000. Quantum Computation by Adiabatic Evolution. *arXiv preprint arXiv:quant-ph/0001106* (2000).

[61] Xinyu Fei. 2022. Code and Results: Binary Control Pulse Optimization for Quantum Systems. https://github.com/xinyufei/Quantum-Control-qutip.

[62] Xinyu Fei. 2022. Code and Results: Switching Time Optimization for Binary Control Pulse in Quantum Systems. https://github.com/xinyufei/Switching-time-optimization.

[63] Xinyu Fei. 2023. Code and Results: Quantum Control Optimization with Uncertain Hamiltonians. https://https://github.com/xinyufei/Stochastic-quantum-control.

[64] Xinyu Fei, Lucas T. Brady, Jeffrey Larson, Sven Leyffer, and Siqian Shen. 2023. Binary Control Pulse Optimization for Quantum Systems. *Quantum* 7 (Jan. 2023), 892. https://doi.org/10.22331/q-2023-01-04-892

[65] Xinyu Fei, Lucas T Brady, Jeffrey Larson, Sven Leyffer, and Siqian Shen. 2023. Switching Time Optimization for Binary Quantum Optimal Control. *arXiv preprint arXiv:2308.03132* (2023).

[66] Xinyu Fei, Xingmin Wang, Xian Yu, Yiheng Feng, Henry Liu, Siqian Shen, and Yafeng Yin. 2023. Traffic signal control under stochastic traffic demand and vehicle turning via decentralized decomposition approaches. *European Journal of Operational Research* 310, 2 (2023), 712–736.

[67] Yiheng Feng, K Larry Head, Shayan Khoshmagham, and Mehdi Zamanipour. 2015. A real-time adaptive signal control in a connected vehicle environment. *Transportation Research Part C: Emerging Technologies* 55 (2015), 460–473.

[68] Kathrin Flaßkamp, Todd Murphey, and Sina Ober-Blöbaum. 2013. Discretized switching time optimization problems. In *2013 European Control Conference (ECC)*. IEEE, 3179–3184.

[69] Pirmin Fontaine and Stefan Minner. 2014. Benders decomposition for discrete–continuous linear bilevel problems with application to traffic network design. *Transportation Research Part B: Methodological* 70 (2014), 163–172.

[70] Gerald Gamrath, Daniel Anderson, Ksenia Bestuzheva, Wei-Kun Chen, Leon Eifler, Maxime Gasse, Patrick Gemander, Ambros Gleixner, Leona Gottwald, Katrin Halbig, et al. 2020. The scip optimization suite 7.0. (2020).

[71] Denos C Gazis, Robert Herman, and Richard W Rothery. 1961. Nonlinear follow-the-leader models of traffic flow. *Operations research* 9, 4 (1961), 545–567.

[72] Xiaozhen Ge, Re-Bing Wu, and Herschel Rabitz. 2022. The Optimization Landscape of Hybrid Quantum-Classical Algorithms: from Quantum Control to NISQ Applications. https://doi.org/10.48550/ARXIV.2201.07448

[73] Arthur M Geoffrion and Glenn W Graves. 1974. Multicommodity distribution system design by Benders decomposition. *Management science* 20, 5 (1974), 822–844.

[74] Steffen J. Glaser, Ugo Boscain, Tommaso Calarco, Christiane P. Koch, Walter Köckenberger, Ronnie Kosloff, Ilya Kuprov, Burkhard Luy, Sophie Schirmer, Thomas Schulte-Herbrüggen, Dominique Sugny, and Frank K. Wilhelm. 2015. Training Schrödinger's cat: quantum optimal control. *Eur. Phys. J. D* 69 (2015). Issue 12. https://doi.org/10.1140/epjd/e2015-60464-1

[75] Pranav Gokhale, Yongshan Ding, Thomas Propson, Christopher Winkler, Nelson Leung, Yunong Shi, David I Schuster, Henry Hoffmann, and Frederic T Chong. 2019. Partial compilation of variational algorithms for noisy intermediate-scale quantum machines. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*. 266–278.

[76] Alexey V. Gorshkov, Tommaso Calarco, Mikhail D. Lukin, and Anders S. Sørensen. 2008. Photon storage in Λ-type optically dense atomic media. IV. Optimal control using gradient ascent. *Physical Review A* 77 (2008), 043806. https://doi.org/10.1103/physreva.77.043806

[77] Matthew Grace, Constantin Brif, Herschel Rabitz, Ian A. Walmsley, Robert L. Kosut, and Daniel A. Lidar. 2007. Optimal control of quantum gates and suppression of decoherence in a system of interacting two-level particles. *Journal of Physics B* 40 (2007), S103.

[78] Matthew D Grace, Jason M Dominy, Wayne M Witzel, and Malcolm S Carroll. 2012. Optimized pulses for the control of uncertain qubits. *Physical Review A* 85, 5 (2012), 052313.

[79] Todd J Green, Jarrah Sastrawan, Hermann Uys, and Michael J Biercuk. 2013. Arbitrary quantum control of qubits in the presence of universal noise. *New Journal of Physics* 15, 9 (2013), 095004.

[80] William Gropp and Jorge J. Moré. 1997. Optimization Environments and the NEOS Server. In *Approximation Theory and Optimization*, Martin D. Buhman and Arieh Iserles (Eds.). Cambridge University Press, 167–182.

[81] Mirko Hahn, Sven Leyffer, and Sebastian Sager. 2023. Binary optimal control by trust-region steepest descent. *Mathematical Programming* 197, 1 (2023), 147–190.

[82] Ali Hajbabaie and Rahim F Benekohal. 2015. A program for simultaneous network signal timing optimization and traffic assignment. *IEEE Transactions on Intelligent Transportation Systems* 16, 5 (2015), 2573–2586.

[83] Brian C Hall. 2013. Lie groups, Lie algebras, and representations. In *Quantum Theory for Mathematicians*. Springer, 333–366.

[84] Matthew P. Harrigan, Kevin J. Sung, Matthew Neeley, Kevin J. Satzinger, Frank Arute, Kunal Arya, Juan Atalaya, Joseph C. Bardin, Rami Barends, Sergio Boixo, Michael Broughton, Bob B. Buckley, David A. Buell, Brian Burkett, Nicholas Bushnell, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Sean Demura, Andrew Dunsworth, Daniel Eppens, Austin Fowler, Brooks Foxen, Craig Gidney, Marissa Giustina, Rob Graff, Steve Habegger, Alan Ho, Sabrina Hong, Trent Huang, L. B. Ioffe, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Cody Jones, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Seon Kim, Paul V. Klimov, Alexander N. Korotkov, Fedor Kostritsa, David Landhuis, Pavel Laptev, Mike Lindmark, Martin Leib, Orion Martin, John M. Martinis, Jarrod R. McClean, Matt McEwen, Anthony

Megrant, Xiao Mi, Masoud Mohseni, Wojciech Mruczkiewicz, Josh Mutus, Ofer Naaman, Charles Neill, Florian Neukart, Murphy Yuezhen Niu, Thomas E. O'Brien, Bryan O'Gorman, Eric Ostby, Andre Petukhov, Harald Putterman, Chris Quintana, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Andrea Skolik, Vadim Smelyanskiy, Doug Strain, Michael Streif, Marco Szalay, Amit Vainsencher, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Leo Zhou, Hartmut Neven, Dave Bacon, Erik Lucero, Edward Farhi, and Ryan Babbush. 2021. Quantum Approximate Optimization of Non-Planar Graph Problems on a Planar Superconducting Processor. *Nature Physics* 17 (2021), 332–336.

[85] Benjamin Heydecker. 1987. Uncertainty and variability in traffic signal calculations. *Transportation Research Part B: Methodological* 21, 1 (1987), 79–85.

[86] Serge P Hoogendoorn and Piet HL Bovy. 2001. State-of-the-art of vehicular traffic flow modelling. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* 215, 4 (2001), 283–303.

[87] Roger A. Horn and Charles R. Johnson. 1991. *Topics in Matrix Analysis*. Cambridge University Press. https://doi.org/10.1017/CBO9780511840371

[88] Norulhidayah Isa, Marina Yusoff, and Azlinah Mohamed. 2014. A review on recent traffic congestion relief approaches. In *2014 4th international conference on artificial intelligence with applications in engineering and technology*. IEEE, 121–126.

[89] W. Jakubetz, J. Manz, and H. J. Schreier. 1990. Theory of optimal laser pulses for selective transitions between molecular eigenstates. *Chemical Physics* 165 (1990), 100–106. Issue 1.

[90] Huiwen Jia and Siqian Shen. 2021. Benders cut classification via support vector machines for solving two-stage stochastic programs. *INFORMS Journal on Optimization* 3, 3 (2021), 278–297.

[91] Elliot R Johnson and Todd D Murphey. 2011. Second-order switching time optimization for nonlinear time-varying dynamic systems. *IEEE Trans. Automat. Control* 56, 8 (2011), 1953–1957.

[92] Velimir Jurdjevic and Héctor J Sussmann. 1972. Control systems on Lie groups. *Journal of Differential equations* 12, 2 (1972), 313–329.

[93] Tadashi Kadowaki and Hidetoshi Nishimori. 1998. Quantum Annealing in the Transverse Ising Model. *Physical Review E* 58 (1998), 5355.

[94] Cindie T Kehlet, Astrid C Sivertsen, Morten Bjerring, Timo O Reiss, Navin Khaneja, Steffen J Glaser, and Niels Chr Nielsen. 2004. Improving solid-state NMR dipolar recoupling by optimal control. *Journal of the American Chemical Society* 126, 33 (2004), 10202–10203.

[95] Navin Khaneja, Timo Reiss, Cindie Kehlet, Thomas Schulte-Herbrüggen, and Steffen J Glaser. 2005. Optimal control of coupled spin dynamics: Design of NMR pulse sequences by gradient ascent algorithms. *Journal of magnetic resonance* 172, 2 (2005), 296–305. https://doi.org/10.1016/j.jmr.2004.11.004

[96] Sumeet Khatri, Ryan LaRose, Alexander Poremba, Lukasz Cincio, Andrew T. Sornborger, and Patrick J. Coles. 2019. Quantum-assisted quantum compiling. *Quantum* 3 (2019), 140.

[97] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[98] Christian Kirches and Sven Leyffer. 2013. TACO: A toolkit for AMPL control optimization. *Mathematical Programming Computation* 5, 3 (2013), 227–265.

[99] Peter Koonce and Lee Rodegerdts. 2008. *Traffic signal timing manual.* Technical Report. United States. Federal Highway Administration.

[100] R. Kosloff, S.A. Rice, P. Gaspard, S. Tersigni, and D.J. Tannor. 1989. Wavepacket dancing: Achieving chemical selectivity by shaping light pulses. *Chemical Physics* 139, 1 (1989), 201–220. https://doi.org/10.1016/0301-0104(89)90012-8

[101] R. Kosloff, S. A. Rice, P. Gaspard, S. Tersigni, and D. J. Tannor. 1989. Wavepacket dancing: Achieving chemical selectivity by shaping light pulses. *Chemical Physics* 139 (1989), 201–220. Issue 1.

[102] Robert L Kosut, Gaurav Bhole, and Herschel Rabitz. 2022. Robust Quantum Control: Analysis & Synthesis via Averaging. *arXiv preprint arXiv:2208.14193* (2022).

[103] Robert L Kosut, Matthew D Grace, and Constantin Brif. 2013. Robust control of quantum gates via sequential convex programming. *Physical Review A* 88, 5 (2013), 052326.

[104] VF Krotov. 1989. Global methods to improve control and optimal control of resonance interaction of light and matter. In *Modeling and Control of Systems: in Engineering, Quantum Mechanics, Economics and Biosciences Proceedings of the Bellman Continuum Workshop 1988, June 13–14, Sophia Antipolis, France.* Springer, 267–298.

[105] Sen Kuang and Shuang Cong. 2008. Lyapunov control methods of closed quantum systems. *Automatica* 44, 1 (2008), 98–108.

[106] Sen Kuang, Daoyi Dong, and Ian R Petersen. 2017. Rapid Lyapunov control of finite-dimensional quantum systems. *Automatica* 81 (2017), 164–175.

[107] Karl Kunisch and Michael Hintermüller. 2004. Total bounded variation regularization as a bilaterally constrained optimization problem. *SIAM J. Appl. Math.* 64, 4 (2004), 1311–1333.

[108] Martin Larocca, Piotr Czarnik, Kunal Sharma, Gopikrishnan Muraleedharan, Patrick J. Coles, and M. Cerezo. 2022. Diagnosing Barren Plateaus with Tools from Quantum Optimal Control. *Quantum* 6 (sep 2022), 824. https://doi.org/10.22331/q-2022-09-29-824

[109] Martín Larocca and Diego Wisniacki. 2021. Krylov-subspace approach for the efficient control of quantum many-body dynamics. *Physical Review A* 103, 2 (Feb 2021). https://doi.org/10.1103/physreva.103.023107

[110] Eugene L Lawler and David E Wood. 1966. Branch-and-bound methods: A survey. *Operations research* 14, 4 (1966), 699–719.

[111] Nelson Leung, Mohamed Abdelhafez, Jens Koch, and David Schuster. 2017. Speedup for quantum optimal control from automatic differentiation based on graphics processing units. *Physical Review A* 95, 4 (2017), 042318. https://doi.org/10.1103/PhysRevA.95.042318

[112] Michael W Levin, Jeffrey Hu, and Michael Odell. 2020. Max-pressure signal control with cyclical phase structure. *Transportation Research Part C: Emerging Technologies* 120 (2020), 102828.

[113] Time Levin. 2020. The 31 US cities that had the worst traffic in 2019 according to a study. https://www.businessinsider.com/us-cities-most-traffic-2019-2020-3.

[114] Sven Leyffer. 2001. Integrating SQP and branch-and-bound for mixed integer nonlinear programming. *Computational optimization and applications* 18, 3 (2001), 295–309.

[115] Sven Leyffer, Ashutosh Mahajan, JJ Cochran, LA Cox, P Keskinocak, JP Kharoufeh, and JC Smith. 2010. Software for nonlinearly constrained optimization. *Preprint ANL/MCS-P1768-0610, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne* (2010), 183–204.

[116] Sven Leyffer and Paul Manns. 2021. Sequential Linear Integer Programming for Integer Optimal Control with Total Variation Regularization. *arXiv preprint arXiv:2106.13453* (2021).

[117] Sven Leyffer, Paul Manns, and Malte Winckler. 2021. Convergence of sum-up rounding schemes for cloaking problems governed by the Helmholtz equation. *Computational Optimization and Applications* 79, 1 (2021), 193–221.

[118] Jr-Shin Li and Navin Khaneja. 2006. Control of inhomogeneous quantum ensembles. *Phys. Rev. A* 73 (Mar 2006), 030302. Issue 3. https://doi.org/10.1103/PhysRevA.73.030302

[119] Jr-Shin Li, Justin Ruths, and Dionisis Stefanatos. 2009. A pseudospectral method for optimal control of open quantum systems. *The Journal of chemical physics* 131, 16 (2009), 164110.

[120] Kun Li and Petros Ioannou. 2004. Modeling of traffic flow of automated vehicles. *IEEE Transactions on Intelligent Transportation Systems* 5, 2 (2004), 99–113.

[121] Lubing Li, Wei Huang, Andy HF Chow, and Hong K Lo. 2021. Two-stage stochastic program for dynamic coordinated traffic control under demand uncertainty. *IEEE Transactions on Intelligent Transportation Systems* (2021).

[122] Lubing Li, Wei Huang, and Hong K Lo. 2018. Adaptive coordinated traffic control for stochastic demand. *Transportation Research Part C: Emerging Technologies* 88 (2018), 31–51.

[123] Li Li and Saif Eddin Jabari. 2019. Position weighted backpressure intersection control for urban networks. *Transportation Research Part B: Methodological* 128 (2019), 435–461.

[124] Daniel Liang, Li Li, and Stefan Leichenauer. 2020. Investigating quantum approximate optimization algorithms under bang-bang protocols. *Physical Review Research* 2, 3 (2020), 033402. https://doi.org/10.1103/physrevresearch.2.033402

[125] Xiao Liang, S Ilgin Guler, and Vikash V Gayah. 2021. Decentralized arterial traffic signal optimization with connected vehicle information. *Journal of Intelligent Transportation Systems* (2021), 1–16.

[126] Michael James Lighthill and Gerald Beresford Whitham. 1955. On kinematic waves II. A theory of traffic flow on long crowded roads. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 229, 1178 (1955), 317–345.

[127] Chungwei Lin, Yebin Wang, Grigory Kolesov, and Uroš Kalabić. 2019. Application of Pontryagin's Minimum Principle to Grover's Quantum Search Problem. *Physical Review A* 100 (2019), 022327.

[128] Chungwei Lin, Yebin Wang, Grigory Kolesov, and Uro š Kalabić. 2019. Application of Pontryagin's minimum principle to Grover's quantum search problem. *Phys. Rev. A* 100 (Aug 2019), 022327. Issue 2. https://doi.org/10.1103/PhysRevA.100.022327

[129] John DC Little and Stephen C Graves. 2008. Little's law. In *Building intuition.* Springer, 81–100.

[130] Yang Liu, Hongyi Li, Renquan Lu, Zongyu Zuo, and Xiaodi Li. 2022. An overview of finite/fixed-time control and its application in engineering systems. *IEEE/CAA Journal of Automatica Sinica* 9, 12 (2022), 2106–2120.

[131] Hong K Lo. 1999. A novel traffic signal control formulation. *Transportation Research Part A: Policy and Practice* 33, 6 (1999), 433–448.

[132] Hong K Lo, Elbert Chang, and Yiu Cho Chan. 2001. Dynamic network traffic control. *Transportation Research Part A: Policy and Practice* 35, 8 (2001), 721–744.

[133] Jeffery S Logsdon and Lorenz T Biegler. 1989. Accurate solution of differential-algebraic optimization problems. *Industrial & engineering chemistry research* 28, 11 (1989), 1628–1639.

[134] Sindri Magnússon, Pradeep Chathuranga Weeraddana, Michael G Rabbat, and Carlo Fischione. 2015. On the convergence of alternating direction lagrangian methods for nonconvex structured optimization problems. *IEEE Transactions on Control of Network Systems* 3, 3 (2015), 296–309.

[135] Paul Manns and Christian Kirches. 2020. Multidimensional sum-up rounding for elliptic control systems. *SIAM J. Numer. Anal.* 58, 6 (2020), 3427–3447. https://doi.org/10.1137/19M1260682

[136] Ivan I Maximov, Zdeněk Tošner, and Niels Chr Nielsen. 2008. Optimal control design of NMR and dynamic nuclear polarization experiments using monotonically convergent algorithms. *The Journal of Chemical Physics* 128, 18 (2008), 05B609.

[137] Glen Bigan Mbeng, Rosario Fazio, and Giuseppe Santoro. 2019. Quantum Annealing: A journey through Digitalization, Control, and hybrid Quantum Variational schemes. *arXiv preprint arXiv:1906.08948* (2019).

[138] Glen Bigan Mbeng, Rosario Fazio, and Giuseppe Santoro. 2019. Quantum Annealing: a journey through Digitalization, Control, and hybrid Quantum Variational schemes. https://doi.org/10.48550/ARXIV.1906.08948

[139] Elke-Barbara Meier and Arthur E Bryson Jr. 1990. Efficient algorithm for time-optimal control of a two-link manipulator. *Journal of Guidance, Control, and Dynamics* 13, 5 (1990), 859–866.

[140] Pitu Mirchandani and Larry Head. 2001. A real-time traffic signal control system: architecture, algorithms, and analysis. *Transportation Research Part C: Emerging Technologies* 9, 6 (2001), 415–432.

[141] Brian E. Mischuck, Seth T. Merkel, and Ivan H. Deutsch. 2012. Control of inhomogeneous atomic ensembles of hyperfine qudits. *Phys. Rev. A* 85 (Feb 2012), 022302. Issue 2. https://doi.org/10.1103/PhysRevA.85.022302

[142] Rasool Mohebifard and Ali Hajbabaie. 2019. Optimal network-level traffic signal control: A benders decomposition-based solution algorithm. *Transportation Research Part B: Methodological* 121 (2019), 252–274.

[143] Simone Montangero, Tommaso Calarco, and Rosario Fazio. 2007. Robust Optimal Quantum Gates for Josephson Charge Qubits. *Physical Review Letters* 99 (2007), 170501.

[144] Oleg V Morzhin and Alexander N Pechen. 2019. Krotov method for optimal control of closed quantum systems. *Russian Mathematical Surveys* 74, 5 (2019), 851.

[145] F. Motzoi, J. M. Gambetta, P. Rebentrost, and F. K. Wilhelm. 2009. Simple Pulses for Elimination of Leakage in Weakly Nonlinear Qubits. *Physical Review Letters* 103, 11 (2009). https://doi.org/10.1103/physrevlett.103.110501

[146] Heinz Mühlenbein, Martina Gorges-Schleuter, and Ottmar Krämer. 1988. Evolution algorithms in combinatorial optimization. *Parallel computing* 7, 1 (1988), 65–85.

[147] Matthias M Müller, Ressa S Said, Fedor Jelezko, Tommaso Calarco, and Simone Montangero. 2022. One decade of quantum optimal control in the chopped random basis. *Reports on Progress in Physics* 85, 7 (2022), 076001.

[148] Ajith Muralidharan, Ramtin Pedarsani, and Pravin Varaiya. 2015. Analysis of fixed-time control. *Transportation Research Part B: Methodological* 73 (2015), 81–90.

[149] Kai Nagel. 1996. Particle hopping models and traffic flow theory. *Physical review E* 53, 5 (1996), 4655.

[150] Michael J Neely. 2010. Stochastic network optimization with application to communication and queueing systems. *Synthesis Lectures on Communication Networks* 3, 1 (2010), 1–211.

[151] Michael A Nielsen and Isaac L Chuang. 2010. *Quantum computation and quantum information.* Cambridge university press.

[152] Murphy Yuezhen Niu, Sergio Boixo, Vadim N Smelyanskiy, and Hartmut Neven. 2019. Universal quantum control through deep reinforcement learning. *npj Quantum Information* 5, 1 (2019), 1–8.

[153] Jorge Nocedal and Stephen Wright. 2006. *Numerical Optimization.* Springer Science & Business Media.

[154] A. Omran, H. Levine, A. Keesling, G. Semeghini, T. T. Wang, S. Ebadi, H. Bernien, A. S. Zibrov, H. Pichler, S. Choi, J. Cui, M. Rossignolo, P. Rembold, S. Montangero, T. Calarco, M. Endres, M. Greiner, V. Vuletić, and M. D. Lukin. 2019. Generation and manipulation of Schrödinger cat states in Rydberg atom arrays. *Science* 365, 6453 (2019), 570–574.

[155] Philip Owrutsky and Navin Khaneja. 2012. Control of inhomogeneous ensembles on the Bloch sphere. *Phys. Rev. A* 86 (Aug 2012), 022315. Issue 2. https://doi.org/10.1103/PhysRevA.86.022315

[156] Guido Pagano, Aniruddha Bapat, Patrick Becker, Katherine S. Collins, Arinjoy De, Paul W. Hess, Harvey B. Kaplan, Antonis Kyprianidis, Wen Lin Tan, Christopher Baldwin, Lucas T. Brady, Abhinav Deshpande, Fangli Liu, Stephen Jordan, Alexey V. Gorshkov, and Christopher Monroe. 2020. Quantum Approximate Optimization of the Long-Range Ising Model with a Trapped-Ion Quantum Simulator. *PNAS* 117, 41 (2020), 25396–25401.

[157] José P. Palao and Ronnie Kosloff. 2002. Quantum Computing by an Optimal Control Algorithm for Unitary Transformations. *Physical Review Letters* 89 (2002), 188301.

[158] José P. Palao and Ronnie Kosloff. 2003. Optimal control theory for unitary transformations. *Physical Review A* 68 (2003), 062308.

[159] José P Palao, Ronnie Kosloff, and Christiane P Koch. 2008. Protecting coherence in optimal control theory: State-dependent constraint approach. *Physical Review A* 77, 6 (2008), 063412.

[160] Łukasz Pawela and Przemysław Sadowski. 2016. Various methods of optimizing control pulses for quantum systems with decoherence. *Quantum Information Processing* 15, 5 (2016), 1937–1953. https://doi.org/10.1007/s11128-016-1242-y

[161] Harold J Payne. 1979. FREFLO: A macroscopic simulation model of freeway traffic. *Transportation Research Record* 722 (1979).

[162] Anthony P. Peirce, Mohammed A. Dahleh, and Herschel Rabitz. 1988. Optimal control of quantum-mechanical systems: Existence, numerical approximation, and applications. *Phys. Rev. A* 37 (Jun 1988), 4950–4964. Issue 12. https://doi.org/10.1103/PhysRevA.37.4950

[163] Anthony P. Peirce, Mohammed A. Dahleh, and Herschel Rabitz. 1988. Optimal control of quantum-mechanical systems: Existence, numerical approximation, and applications. *Physical Review A* 37 (1988), 4950.

[164] Georg Ch Pflug. 2000. Some remarks on the value-at-risk and the conditional value-at-risk. *Probabilistic constrained optimization: Methodology and applications* (2000), 272–281.

[165] Louis A Pipes. 1953. An operational analysis of traffic dynamics. *Journal of applied physics* 24, 3 (1953), 274–281.

[166] Joseph N Prashker and Shlomo Bekhor. 2004. Route choice models used in the stochastic user equilibrium problem: a review. *Transport reviews* 24, 4 (2004), 437–463.

[167] Thomas Propson, Brian E Jackson, Jens Koch, Zachary Manchester, and David I Schuster. 2022. Robust quantum optimal control with trajectory optimization. *Physical Review Applied* 17, 1 (2022), 014036.

[168] Brent Pryor and Navin Khaneja. 2006. Fourier decompositions and pulse sequence design algorithms for nuclear magnetic resonance in inhomogeneous fields. *The Journal of chemical physics* 125, 19 (2006), 194111. https://doi.org/10.1063/1.2390715

[169] Herschel Rabitz, Regina De Vivie-Riedle, Marcus Motzkus, and Karl Kompa. 2000. Whither the future of controlling quantum phenomena? *Science* 288, 5467 (2000), 824–828.

[170] Ragheb Rahmaniani, Teodor Gabriel Crainic, Michel Gendreau, and Walter Rei. 2017. The Benders decomposition algorithm: A literature review. *European Journal of Operational Research* 259, 3 (2017), 801–817.

[171] Viswanath Ramakrishna, Murti V. Salapaka, Mohammed Dahleh, Herschel Rabitz, and Anthony Peirce. 1995. Controllability of molecular systems. *Physical Review A* 51, 2 (1995), 960.

[172] Patrick Rebentrost and Frank K Wilhelm. 2009. Optimal control of a leaking qubit. *Physical Review B* 79, 6 (2009), 060507. https://doi.org/10.1103/physrevb.79.060507

[173] Bilal Riaz, Cong Shuang, and Shahid Qamar. 2019. Optimal control methods for quantum gate preparation: a comparative study. *Quantum Information Processing* 18 (2019), 1–26.

[174] R Tyrrell Rockafellar and Stanislav Uryasev. 2002. Conditional value-at-risk for general loss distributions. *Journal of banking & finance* 26, 7 (2002), 1443–1471.

[175] R Tyrrell Rockafellar, Stanislav Uryasev, et al. 2000. Optimization of conditional value-at-risk. *Journal of risk* 2 (2000), 21–42.

[176] Paul Rodríguez. 2013. Total variation regularization algorithms for images corrupted with different noise models: A review. *Journal of Electrical and Computer Engineering* 2013 (2013).

[177] Jonathan Romero, Ryan Babbush, Jarrod R McClean, Cornelius Hempel, Peter J Love, and Alán Aspuru-Guzik. 2018. Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz. *Quantum Science and Technology* 4, 1 (2018), 014008. https://doi.org/10.1088/2058-9565/aad3e4

[178] Leonid I Rudin, Stanley Osher, and Emad Fatemi. 1992. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena* 60, 1-4 (1992), 259–268.

[179] Justin Ruths and Jr-Shin Li. 2011. A multidimensional pseudospectral method for optimal control of quantum ensembles. *The Journal of chemical physics* 134, 4 (2011), 044128. https://doi.org/10.1063/1.3541253

[180] Sebastian Sager. 2005. *Numerical Methods for Mixed-Integer Optimal Control Problems*. Der Andere Verlag Tönning.

[181] Sebastian Sager, Hans Georg Bock, and Moritz Diehl. 2012. The integer approximation error in mixed-integer optimal control. *Mathematical programming* 133, 1 (2012), 1–23. https://doi.org/10.1007/s10107-010-0405-3

[182] Sebastian Sager, Michael Jung, and Christian Kirches. 2011. Combinatorial integral approximation. *Mathematical Methods of Operations Research* 73, 3 (2011), 363–380. https://doi.org/10.1007/s00186-011-0355-4

[183] Sebastian Sager and Clemens Zeile. 2021. On mixed-integer optimal control with constrained total variation of the integer control. *Computational Optimization and Applications* 78, 2 (2021), 575–623.

[184] Nikolaos V Sahinidis. 1996. BARON: A general purpose global optimization software package. *Journal of global optimization* 8, 2 (1996), 201–205.

[185] Sergey Sarykalin, Gaia Serraino, and Stan Uryasev. 2008. Value-at-risk vs. conditional value-at-risk in risk management and optimization. In *State-of-the-art decision-making tools in the information-intensive age*. Informs, 270–294.

[186] D. Schrank, B. Eisele, and Lomax T. 2019. *2019 Urban mobility report*. Technical Report. Texas A&M Transportation Institute.

[187] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. 2014. *Lectures on Stochastic Programming: Modeling and Theory*. SIAM.

[188] Yosef Sheffi. 1985. *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*.

[189] Shenghua Shi and Herschel Rabitz. 1989. Selective excitation in harmonic molecular systems by optimally designed fields. *Chemical Physics* 139 (1989), 185–199. Issue 1.

[190] Shenghua Shi, Andrea Woody, and Herschel Rabitz. 1988. Optimal control of selective vibrational excitation in harmonic linear chain molecules. *Journal of Chemical Physics* 88 (1988), 6870.

[191] Chaitrali Shirke, Nasser Sabar, Edward Chung, and Ashish Bhaskar. 2022. Metaheuristic approach for designing robust traffic signal timings to effectively serve varying traffic demand. *Journal of Intelligent Transportation Systems* 26, 3 (2022), 343–355.

[192] Arthur G Sims and Kenneth W Dobinson. 1980. The Sydney coordinated adaptive traffic (SCAT) system philosophy and benefits. *IEEE Transactions on vehicular technology* 29, 2 (1980), 130–137.

[193] Marius Sinclair. 1986. An exact penalty function approach for nonlinear integer programming problems. *European journal of operational research* 27, 1 (1986), 50–56.

[194] VV Sivak, A Eickbusch, H Liu, B Royer, I Tsioutsios, and MH Devoret. 2022. Model-free quantum control with reinforcement learning. *Physical Review X* 12, 1 (2022), 011059.

[195] Thomas E Skinner, Timo O Reiss, Burkhard Luy, Navin Khaneja, and Steffen J Glaser. 2003. Application of optimal control theory to the design of broadband excitation pulses for high-resolution NMR. *Journal of Magnetic Resonance* 163, 1 (2003), 8–15.

[196] Shlomo E Sklarz and David J Tannor. 2002. Loading a Bose-Einstein condensate onto an optical lattice: An application of optimal control theory to the nonlinear Schrödinger equation. *Physical Review A* 66, 5 (2002), 053619.

[197] József Somlói, Vladimir A Kazakov, and David J Tannor. 1993. Controlled dissociation of I2 via optical transitions between the X and B electronic states. *Chemical physics* 172, 1 (1993), 85–98.

[198] JJWH Sørensen, MO Aranburu, Till Heinzel, and JF Sherson. 2018. Quantum optimal control in a chopped basis: Applications in control of Bose-Einstein condensates. *Physical Review A* 98, 2 (2018), 022119. https://doi.org/10.1103/PhysRevA.98.022119

[199] Lorenzo Stella, Andreas Themelis, Pantelis Sopasakis, and Panagiotis Patrinos. 2017. A simple and efficient algorithm for nonlinear model predictive control. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 1939–1944.

[200] Bartolomeo Stellato, Sina Ober-Blöbaum, and Paul J Goulart. 2017. Second-order switching time optimization for switched dynamical systems. *IEEE Trans. Automat. Control* 62, 10 (2017), 5407–5414.

[201] Aleksandar Stevanovic, Peter T Martin, and Jelka Stevanovic. 2007. VisSim-based genetic algorithm optimization of signal timings. *Transportation Research Record* 2035, 1 (2007), 59–68.

[202] Mehrdad Tajalli, Mehrzad Mehrabipour, and Ali Hajbabaie. 2020. Network-level coordinated speed optimization and traffic light control for connected and automated vehicles. *IEEE Transactions on Intelligent Transportation Systems* 22, 11 (2020), 6748–6759.

[203] Keshuang Tang, Manfred Boltze, Hideki Nakamura, and Zong Tian. 2019. *Global Practices on Road Traffic Signal Control: Fixed-time Control at Isolated Intersections*. Elsevier.

[204] David J Tannor, Vladimir Kazakov, and Vladimir Orlov. 1992. Control of photochemical branching: Novel procedures for finding optimal pulses and global upper bounds. *Time-dependent quantum molecular dynamics* (1992), 347–360.

[205] Leandros Tassiulas and Anthony Ephremides. 1990. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. In *29th IEEE Conference on Decision and Control*. IEEE, 2130–2132.

[206] Andreas Themelis, Lorenzo Stella, and Panagiotis Patrinos. 2018. Forward-backward envelope for the sum of two nonconvex functions: Further properties and nonmonotone linesearch algorithms. *SIAM Journal on Optimization* 28, 3 (2018), 2274–2303.

[207] Stelios Timotheou, Christos G Panayiotou, and Marios M Polycarpou. 2014. Distributed traffic signal control using the cell transmission model via the alternating direction method of multipliers. *IEEE Transactions on Intelligent Transportation Systems* 16, 2 (2014), 919–933.

[208] Yue Tong, Lei Zhao, Li Li, and Yi Zhang. 2015. Stochastic programming model for oversaturated intersection signal timing. *Transportation Research Part C: Emerging Technologies* 58 (2015), 474–486.

[209] Thomas Urbanik, Alison Tanaka, Bailey Lozner, Eric Lindstrom, Kevin Lee, Shaun Quayle, Scott Beaird, Shing Tsoi, Paul Ryus, Doug Gettman, et al. 2015. *Signal timing manual*. Vol. 1. Transportation Research Board Washington, DC.

[210] R. M. W. van Bijnen and T. Pohl. 2015. Quantum Magnetism and Topological Ordering via Rydberg Dressing near Förster Resonances. *Physical Review Letters* 114 (2015), 243002.

[211] Pravin Varaiya. 2013. Max pressure control of a network of signalized intersections. *Transportation Research Part C: Emerging Technologies* 36 (2013), 177–195.

[212] Davide Venturelli, Minh Do, Eleanor Rieffel, and Jeremy Frank. 2018. Compiling quantum circuits to realistic hardware architectures using temporal planners. *Quantum Science and Technology* 3, 2 (2018), 025004.

[213] Lorenzo Campos Venuti, Domenico D'Alessandro, and Daniel A. Lidar. 2021. Optimal Control for Closed and Open System Quantum Optimization. *arXiv preprint arXiv:2107.03517* (2021).

[214] Lorenzo Campos Venuti, Domenico D'Alessandro, and Daniel A. Lidar. 2021. Optimal Control for Quantum Optimization of Closed and Open Systems. *Physical Review Applied* 16, 5 (nov 2021). https://doi.org/10.1103/physrevapplied.16.054023

[215] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* 17 (2020), 261–272. https://doi.org/10.1038/s41592-019-0686-2

[216] Ryan H. Vogt and N. Anders Petersson. 2021. Binary Optimal Control Of Single-Flux-Quantum Pulse Sequences. arXiv:2106.10329 [quant-ph]

[217] Ryan H. Vogt and N. Anders Petersson. 2022. Binary Optimal Control of Single-Flux-Quantum Pulse Sequences. *SIAM Journal on Control and Optimization* 60, 6 (2022), 3217–3236. https://doi.org/10.1137/21M142808X

[218] Gottfried Vossen. 2010. Switching time optimization for bang-bang and singular controls. *Journal of optimization theory and applications* 144, 2 (2010), 409–429.

[219] Andreas Wächter and Lorenz T Biegler. 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming* 106, 1 (2006), 25–57.

[220] G. Waldherr, Y. Wang, S. Zaiser, M. Jamali, T. Schulte-Herbrüggen, H. Abe, T. Ohshima, J. Isoya, J. F. Du, P. Neumann, and J. Wrachtrup. 2014. Quantum error correction in a solid-state hybrid spin register. *Nature* 506 (2014), 204–207.

[221] Xingmin Wang, Zachary Jerome, Chenhao Zhang, Shengyin Shen, Vivek Vijaya Kumar, and Henry X Liu. 2022. Trajectory Data Processing and Mobility Performance Evaluation for Urban Traffic Networks. *Transportation Research Record* (2022), 03611981221115088.

[222] Xingmin Wang, Yafeng Yin, Yiheng Feng, and Henry X Liu. 2022. Learning the max pressure control for urban traffic networks considering the phase switching loss. *Transportation Research Part C: Emerging Technologies* 140 (2022), 103670.

[223] Fo Vo Webster. 1958. *Traffic signal settings.* Technical Report.

[224] J. Werschnik and E. K. U. Gross. 2007. Quantum optimal control theory. *Journal of Physics B* 40, 18 (2007), R175–R211.

[225] J Werschnik and E K U Gross. 2007. Quantum optimal control theory. *Journal of Physics B: Atomic, Molecular and Optical Physics* 40, 18 (sep 2007), R175. https://doi.org/10.1088/0953-4075/40/18/R01

[226] Janus H. Wesenberg. 2004. Designing robust gate implementations for quantum-information processing. *Phys. Rev. A* 69 (Apr 2004), 042323. Issue 4. https://doi.org/10.1103/PhysRevA.69.042323

[227] Hadley Wickham and Lisa Stryjewski. 2011. 40 years of boxplots. *Am. Statistician* (2011), 2011.

[228] Laurence A Wolsey. 2020. *Integer Programming.* John Wiley & Sons.

[229] Jian Wu, Dipak Ghosal, Michael Zhang, and Chen-Nee Chuah. 2017. Delay-based traffic signal control for throughput optimality and fairness at an isolated intersection. *IEEE Transactions on Vehicular Technology* 67, 2 (2017), 896–909.

[230] Ning Wu and Werner Brilon. 1999. Cellular Automata for Highway Traffic Flow Simulation. *Proceedings 14th International Symposium on Transportation and Traffic Theory (Abbreviated presentations)* (01 1999).

[231] Re-Bing Wu, Haijin Ding, Daoyi Dong, and Xiaoting Wang. 2019. Learning robust and high-precision quantum controls. *Physical Review A* 99, 4 (2019), 042327.

[232] Xiaodong Yang, Jun Li, and Xinhua Peng. 2019. An improved differential evolution algorithm for learning high-fidelity quantum controls. *Science Bulletin* 64, 19 (2019), 1402–1408.

[233] Zhi-Cheng Yang, Armin Rahmani, Alireza Shabani, Hartmut Neven, and Claudio Chamon. 2017. Optimizing Variational Quantum Algorithms Using Pontryagin's Minimum Principle. *Physical Review X* 7, 2 (may 2017). https://doi.org/10.1103/physrevx.7.021027

[234] Zhi-Cheng Yang, Armin Rahmani, Alireza Shabani, Hartmut Neven, and Claudio Chamon. 2017. Optimizing Variational Quantum Algorithms using Pontryagin's Minimum Principle. *Physical Review X* 7 (2017), 021027.

[235] Yafeng Yin. 2008. Robust optimal traffic signal timing. *Transportation Research Part B: Methodological* 42, 10 (2008), 911–924.

[236] Fengqi You and Sven Leyffer. 2011. Mixed-integer dynamic optimization for oil-spill response planning with integration of a dynamic oil weathering model. *AIChE Journal* 57, 12 (2011), 3555. https://doi.org/10.1002/aic.12536

[237] Isaak Yperman, Steven Logghe, and Ben Immers. 2005. The link transmission model: An efficient implementation of the kinematic wave theory in traffic networks. In *Proceedings of the 10th EWGT Meeting*. Poznan Poland, 122–127.

[238] X-H Yu and Wilfred W Recker. 2006. Stochastic adaptive control model for traffic signal systems. *Transportation Research Part C: Emerging Technologies* 14, 4 (2006), 263–282.

[239] Ehsan Zahedinejad, Sophie Schirmer, and Barry C Sanders. 2014. Evolutionary algorithms for hard quantum control. *Physical Review A* 90, 3 (2014), 032310. https://doi.org/10.1103/PhysRevA.90.032310

[240] Ali A Zaidi, Balázs Kulcsár, and Henk Wymeersch. 2016. Back-pressure traffic signal control with fixed and adaptive routing for urban vehicular networks. *IEEE Transactions on Intelligent Transportation Systems* 17, 8 (2016), 2134–2143.

[241] Xianyuan Zhan, Yu Zheng, Xiuwen Yi, and Satish V Ukkusuri. 2016. Citywide traffic volume estimation using trajectory data. *IEEE Transactions on Knowledge and Data Engineering* 29, 2 (2016), 272–285.

[242] Guohui Zhang and Yinhai Wang. 2010. Optimizing minimum and maximum green time settings for traffic actuated control at isolated intersections. *IEEE Transactions on Intelligent Transportation Systems* 12, 1 (2010), 164–173.

[243] Lihui Zhang, Yafeng Yin, and Yingyan Lou. 2010. Robust signal timing for arterials under day-to-day demand variations. *Transportation Research Record* 2192, 1 (2010), 156–166.

[244] Xiao-Ming Zhang, Zezhu Wei, Raza Asad, Xu-Chen Yang, and Xin Wang. 2019. When does reinforcement learning stand out in quantum control? A comparative study on state preparation. *npj Quantum Information* 5, 1 (2019), 85.

[245] Yan Zhao, Jianfeng Zheng, Wai Wong, Xingmin Wang, Yuan Meng, and Henry X Liu. 2019. Estimation of queue lengths, probe vehicle penetration rates, and traffic volumes at signalized intersections using probe vehicle trajectories. *Transportation Research Record* 2673, 11 (2019), 660–670.

[246] Yan Zhao, Jianfeng Zheng, Wai Wong, Xingmin Wang, Yuan Meng, and Henry X Liu. 2019. Various methods for queue length and traffic volume estimation using probe vehicle trajectories. *Transportation Research Part C: Emerging Technologies* 107 (2019), 70–91.

[247] Jianfeng Zheng and Henry X Liu. 2017. Estimating traffic volumes for signalized intersections using connected vehicle data. *Transportation Research Part C: Emerging Technologies* 79 (2017), 347–362.

[248] Ciyou Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. 1997. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on mathematical software (TOMS)* 23, 4 (1997), 550–560.