8-2022

Optimal scheduling of connected and autonomous vehicles at a reservation-based intersection.

Muting Ma
University of Louisville

OPTIMAL SCHEDULING OF CONNECTED AND AUTONOMOUS VEHICLES AT A RESERVATION-BASED INTERSECTION

By

Muting Ma

B.Eng., Liaoning Police College, China, 2014
M.Eng., People's Public Security University of China, China, 2017

A Dissertation
Submitted to the Faculty of the
J. B. Speed School of Engineering of the University of Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy
in Civil Engineering

Department of Civil and Environmental Engineering
University of Louisville
Louisville, Kentucky

August 2022

OPTIMAL SCHEDULING OF CONNECTED AND AUTONOMOUS VEHICLES AT A RESERVATION-BASED INTERSECTION

By

Muting Ma

B.Eng., Liaoning Police College, China, 2014

M.Eng., People's Public Security University of China, China, 2017

A Dissertation Approved on

July 25, 2022

By the following Dissertation Committee:

Dr. Zhixia Li, Dissertation Chair

Dr. Zhihui Sun

Dr. Mark French

Dr. Olfa Nasraoui

DEDICATION

To my mother, my grandmother and my family.

To my father passed away in 2020.

ACKNOWLEDGEMENTS

During the past five years pursuing my PhD degree, I was given and taken away.

I will miss these five years in the rest of my life and remember what I learned for a better future.

First, I would like to appreciate my dissertation committee members, Dr. Zhihui Sun, Dr. Mark French, and Dr. Olfa Nasraoui for their kindly support and constructive insight while serving on my dissertation committee. I sincerely admire their personality and academic achievement. They are my role models in my future career.

I especially pay my thanks to my transportation lab colleagues, including but not limited to Dr. Song Wang, Dr. Majeed Algomaiah, and Mr. Yingfan Gu, for their companion and help. I also want to thank Mr. Bernie Miles for his enthusiasm and passion. I enjoy working with these folks and learned a lot of lessons from them. There were too many people that I met, worked and laughed with. I will not list names of each one of them here but will remember every second I spent with them together.

Lastly, I want to thank my PhD advisor, Dr. Zhixia Li. He is a role model for me from both perspective of academic research and living as a person with integrity. I may not achieve what he has accomplished but will feel encouraged with his guidance at front of me. I learned a lot from him and will cherish this opportunity forever being working with him. I can still see a lot of opportunities of working with and learning from him in the future.

In the end, for my mother and my grandmother, which are only two persons left in my family, I will do my best to repay them with what they gave me, including my birth, my opportunities, and my life in this world. I would not survive in this world without them.

To my father passed away in 2020 during my third year of PhD study, I did not understand the value of what I had been given at no cost until he passed away. My father would never see my career, marriage and children, but I will let my children know how they come to this world and treasure everything in this world, which cannot be taken for granted.

If my father could see these words in heaven, please rest in peace and guide his son toward his future.

ABSTRACT

OPTIMAL SCHEDULING OF CONNECTED AND AUTONOMOUS VEHICLES AT A RESERVATION-BASED INTERSECTION

Muting Ma

July 23, 2022

Reservation-based intersection control has been evaluated with better performance over traditional signal controls in terms of intersection safety, efficiency and emission. Controlling connected and autonomous vehicles (CAVs) at a reservation-based intersection in terms of improving intersection efficiency is performed via two factors: trajectory (speed profile) and arrival time of CAVs at the intersection. In an early stage of the reservation-based intersection control, an intersection controller at the intersection may fail to find a feasible solution for both the trajectory and arrival time for a CAV at a certain planning horizon. Leveraging deeper understanding of the control problem, reservation-based intersection control methods are able to optimize both trajectory and arrival time simultaneously while overcome the infeasible condition. Furthermore, in order to achieve a real-time control at the reservation-based intersection, a scheduling problem of CAV crossing the intersection has been widely modeled to optimize the intersection efficiency. Efficient solution algorithms have been proposed to overcome the curse of dimensionality. However, a control methodology consisting of trajectory planning and arrival time scheduling that can overcomes the infeasible condition has not been explicitly explained and defined. Furthermore, an optimal control framework for a joint control of the trajectory

planning and arrival time scheduling in terms of global intersection efficiency has not been theoretically established and numerically validated; and mechanisms of how to reduce the time complexity meanwhile solve the scheduling problem to an optimal solution are not fully understood and rigorously defined.

In this dissertation, a control method that eliminates the infeasible problem at any planning horizon is first explicitly explained and defined based on a time-speed-independent trajectory planning and scheduling model. Secondly, this dissertation theoretically defines the optimal control framework via analyzing various control methods in terms of intersection capacity, throughput and delay. Furthermore, this dissertation theoretically analyzes the mechanism of the scheduling problem and designs an exact algorithm to further reduce the time complexity. Through theoretical analyses of properties of the scheduling problem, reasons that the time complexity can be reduced are fundamentally explained.

The results first validate that the defined control framework can adapt to extremely high traffic demand scenario with feasible solutions at any planning horizon for all CAVs. Under extensive sensitivity analyses, the theoretical definition on the optimal control framework is validated in terms of maximizing the intersection efficiency. Moreover, numerical examples validate that a proposed scheduling algorithm finds an optimal solution with lower computation time and time complexity.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

1.1 Background

Following autonomous intersection management (AIM) concept or reservation-based intersection was proposed by Dresner and Stone (2004), multiple control policies were further developed including first-come-first-serve (FCFS), Auction (Schepperle and Böhm, 2007, 2008), and Batch (Tachet et al., 2016) etc. to improve intersection efficiency comparing with the conventional signalized control. Most previous studies have investigated into several AIM control policies and evaluated that AIM is superior than signalized control in terms of safety, efficiency and emission (Li et al., 2013a; Lin et al., 2017; Wu et al., 2019).

Controlling connected and autonomous vehicles (CAVs) at a reservation-based intersection in terms of improving intersection efficiency is performed via two factors: trajectory (speed profile) and arrival time of CAVs at the intersection. Trajectory or acceleration of CAVs approaching to the intersection is modelled as a continuous function over time (Dresner and Stone, 2008; Lee and Park, 2012) or a piecewise function over time (Liu et al., 2022; Tajalli and Hajbabaie, 2021; Yang et al., 2021). Along the trajectory, arrival speed of CAVs at the intersection is determined as a fixed value independent of the arrival time (Li et al., 2019; Rios-Torres and Malikopoulos, 2016; Yu et al., 2019) or modelled as a variable depending on the arrival time of CAVs (Chalaki and Malikopoulos, 2021; Malikopoulos et al., 2018; Zhao et al., 2021). The intersection efficiency, including capacity, throughput and delay, is optimized via these various control methods.

Furthermore, specifically, scheduling of CAVs crossing a reservation-based intersection is of great importance for improving intersection efficiency and safety (Zhang et al., 2022). Regarding the scheduling problem, rule-based conflict solving techniques (Dresner and Stone, 2008; Tachet et al., 2016) have been replaced by optimization-based techniques. In terms of an objective, such as minimizing intersection delay, the scheduling problem is widely modeled via mixed integer linear programming (MILP) (Liu et al., 2022; Lu et al., 2022; Ma and Li, 2020, 2021). However, it takes exponential growth time complexity for state-of-art solvers, such as Cplex or Gurobi, to find an optimal solution of the MILP model (Morrison et al., 2016; Xu et al., 2021). It is a challenge to develop an efficient algorithm in order to achieve a better tradeoff between the optimality and the time complexity.

1.2 Research Gap

In an early stage for an intersection control problem for CAVs, the vehicles follow the trajectory planned by an intersection controller and arrive at the intersection with a given arrival time and speed. However, when the traffic condition varies over time, i.e., a priority passing sequence to the intersection is shifted from one earlier coming vehicle to another later coming vehicle, or the traffic demand is increasing sharply from one approach, the vehicle may not follow the planned trajectory and arrive at the intersection with an expected time and speed (Au et al., 2012; Au and Stone, 2010). In addition, when the trajectory planning only consists of a few segments, it renders trajectory solutions to arriving with an expected time and speed infeasible if the traffic condition varies sharply. Vehicles that fail to follow the planned arrival time and speed each planning horizon are re-scheduled at next horizon. In other words, vehicles that is not assigned a feasible solution by an intersection central controller is re-planned by the controller at next horizon. However, once a vehicle's

speed is compromised for trajectory solution feasibility, the optimality is eliminated for overall intersection efficiency since the arrival speed of following vehicles is also compromised. Moreover, conflicts between vehicles from conflicting approaches might happen within the intersection when vehicles cannot arrive at the intersection with the assigned arrival time and speed. This kind of problem with infeasible solutions is called fail-follow problem.

Other than the fail-follow problem, an open question still exists in the following: by using which control method can the optimal intersection efficiency be achieved. Specifically, three challenges have not been understood and solved in terms of answering this question: (1) how trajectory control of CAVs affects the intersection efficiency? (2) how to control arrival speed of CAVs at the intersection in terms of the intersection efficiency, especially under (un)balanced or high traffic demand scenarios? (3) how to model arrival time and arrival speed of CAVs in a way that the intersection efficiency can reach a maximum?

Especially for a scheduling problem of CAVs crossing the intersection, an efficient solution algorithm is critical for the central controller to implement real-time control. To date, at a general reservation-based intersection with two conflicting approaches, an approximation algorithm is able to take linear time complexity (Xu et al., 2019a) to find a near optimal solution, whereas an exact algorithm takes quadratic time complexity to find an optimal solution (Pei et al., 2019). However, the mechanisms of how to reduce the time complexity meanwhile solve the scheduling problem to an optimal solution are not fully understood yet. It is critical to theoretically explain the mechanisms in order to further reduce the time complexity at an intersection with two or more conflicting approaches.

Overall, this dissertation identifies three fundamental research gaps in the control and scheduling problem at a reservation-based intersection.

1. A control methodology consisting of trajectory planning and arrival time scheduling that can overcomes the fail-follow problem has not been explicitly explained and defined;
2. An optimal control framework for a joint control of the trajectory planning and arrival time scheduling in terms of global intersection efficiency has not been theoretically established and numerically validated; and
3. The mechanisms of how to reduce the time complexity meanwhile solve the scheduling problem to an optimal solution are not fully understood and rigorously defined.

CHAPTER 2. RESEARCH OBJECTIVES AND CONTRIBUTIONS

2.1 Objectives

In order to fill the research gaps identified at Chapter 1, this dissertation aims to solve the following objectives.

1. To overcome the fail-follow problem caused by an assigned arrival time and speed in a certain planning horizon, meanwhile explicitly explain and define a control method that eliminates the fail-follow problem at any planning horizon;
2. To propose an optimal control framework for CAVs at a reservation-based intersection in terms of maximizing the intersection efficiency; and
3. To theoretically analyze the mechanism of scheduling problem and design an exact algorithm to further reduce the time complexity.

2.2 Contributions

Specifically, the contributions of this dissertation are holistically summarized as follows. In terms of the first objective in the dissertation that will be introduced in Chapter 4, the contributions are as follows.

1. Separated the formulation and optimization between arrival time, speed and trajectory planning by optimizing the trajectory without arrival time and speed predetermined at any planning horizon;
2. Found the optimal solutions in terms of overall intersection efficiency by relaxing the constraint of speed at any time under varying traffic condition;

3. Kept trajectory solutions feasible by formulating the variation of acceleration rate and breaking a whole trajectory into an enlarged set of segments; and
4. Explicitly explained and defined a joint control method that overcomes the fail-follow problem at any planning horizon.

In terms of the second objective in the dissertation that will be introduced in Chapter 5, the contributions are as follows.

5. I first define the trajectory control problem at a reservation-based intersection in a general form. Based on the definition, the relationship between arrival speed and arrival time of CAVs under the continuous or piecewise function is analytically investigated. Based on the discovered relationship, the intersection efficiency, including capacity and delay, is approximated by a greedy algorithm with arrival speed modelling independently of arrival time. Furthermore, the intersection efficiency is quantified via queue theory with considering arrival speed as a variable and analyzed under balanced and unbalanced traffic scenarios.
6. I then theoretically define the optimal framework of controlling the trajectory, arrival speed and arrival time simultaneously in terms of maximizing the intersection efficiency. Based on the framework, an integrated control method is modelled via the MILP. In the MILP, arrival speed, as well as arrival time, is modelled as a variable in order to validate the findings of the defined framework. Moreover, both continuous and piecewise trajectory control methods are performed on the integrated MILP.
7. Lastly, I test the modelling methods under extensive numerical simulations. The theoretical definition on the optimal control framework is validated through various

traffic scenarios, including high traffic demand scenarios with 1800 veh/h/ln. The optimal solution of the intersection efficiency is found under the optimal control framework with zero optimality gap.

In terms of the third objective in the dissertation that will be introduced in Chapter 6, the contributions are as follows.

8. The problem properties of the scheduling problem are first investigated in nature from the perspective of four critical decision variables. Any scheduling problem at a general reservation-based intersection can be modeled by these four variables in essence. An optimal substructure of the scheduling problem is then identified. Based on the optimal substructure, dynamic programming (DP) or branch & bound (B&B) algorithms can be utilized to solve such a problem. Further, it is proven that a conflict-order-based property is inherited in the problem. Based on the property, integer-infeasible solutions can be eliminated during solving the problem.
9. Through relaxing the original problem to subproblems, Markov property is first identified, based on which an objective of the problem is designed to minimize the maximal arrival time of all CAVs at the intersection. While adding subproblems, two overlapping properties are theoretically analyzed. Based on the overlapping properties, one subproblem can be dominated by another such that why the time complexity of solving the problem can be reduced is fundamentally explained.
10. To investigate more possibilities of reducing the time complexity, a platoon of CAVs is first defined. Based on platooning of CAVs, it is rigorously proven that number of subproblems can be further reduced. In addition, a total number of

subproblems/nodes of the original problem is analytically derived with and without the platooning. An optimal platooning property is subsequently identified.

11. To solve such a problem, a MILP model is first formulated. A customized DP algorithm is further designed to incorporate all the identified properties. Based on the customized DP algorithm, an upper bound of the time complexity is first derived. Based on the optimal platooning property, a lower bound of the time complexity is then derived. The DP algorithm solves the problem to an optimal solution. The lower bound is reduced to linear time complexity from quadratic time complexity.
12. A control framework for the scheduling problem with stochastic arrivals of CAVs from two conflicting approaches is first developed. Through numerical examples, the proposed DP algorithm is compared with a state-of-art DP algorithm in terms of computation time and solution. The proposed DP algorithm finds an optimal solution with lower computation time. Interesting findings prove that CAVs tend to dynamically batch as a platoon in the optimal solution.

2.3 Organization of the Dissertation

The remainder of the dissertation is organized as follows. Chapter 3 summarizes the state-of-art literatures on the topic of the reservation-based intersection control. Chapter 4 explicitly explains and defines a control method that eliminates the fail-follow problem at any planning horizon. Chapter 5 proposes an optimal control framework for CAVs at a reservation-based intersection in terms of maximizing the intersection efficiency. Chapter 6 theoretically analyzes the mechanism of scheduling problem and designs an exact algorithm to further reduce the time complexity. Chapter 7 concludes the dissertation and introduces future research direction on this topic.

CHAPTER 3. LITERATURE REVIEW

3.1 Reservation-based Intersection Control

3.1.1 Reservation-based intersection formulation

By leveraging the technology of vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), infrastructure-to-vehicle (I2V) and connected and autonomous vehicle (CAV), autonomous intersection management (AIM) is substituting the role of traditional traffic signal (Wuthishuwong and Traechtler, 2013). AIM was first proposed by Dresner and Stone (2004, 2005), where an intersection is divided to a grid consisting of cells with n x n granularity and controlled by an intersection controller. The temporal and spatial resources of the intersection are therefore fully utilized by allowing conflicting vehicles arriving into the intersection simultaneously only if they are not colliding at certain cells and at periods of time they requested to occupy, whereas under traditional signal controller, vehicles cannot enter the intersection unless conflicting vehicles leave the intersection. AIM was further investigated to implement AIM communication protocols between agents of vehicles and intersections (Dresner and Stone, 2008). They also proposed ideas of allowing AIM to control human-driven vehicles and prioritizing emergency vehicles at no cost of civilian vehicles. The results indicated that their mechanism significantly outperforms traffic lights and stop signs. Similar results showing better efficiency and lower emission were also demonstrated by Fajardo et al. (2011) and Li et al. (2013a). However, according to the control policy of the AIM in earlier years, the reservation-based intersection control could overlook the requests of vehicles from minor roads if the traffic

demand from main and minor roads is unbalanced and cause the intersection to a gridlock (Au et al., 2011).

Zhu and Ukkusuri (2015) proposed a conflict point-based reservation system based on dynamic traffic assignment, in which the vehicle was also formulated as a point so that conflicts are avoided only if conflicting vehicles are not occupying at certain points at same time. Levin and Rey (2017) proposed a mixed integer linear programming to optimize the travel time based on conflict point formulation of the intersection. Furthermore, Li et al. (2019) improved the conflict point formulation by considering the shape (length and width) of the vehicle into a conflict-time list and avoiding the conflicts within the intersection by solving the conflict-time list via heuristics. Besides, conflict cell-based formulation is also considered in general to separate conflicts between through, left-turn, right-turn vehicles and pedestrians (Bichiou and Rakha, 2018; Niels et al., 2020; Yu et al., 2019; Zhang et al., 2016).

3.1.2 Reservation-based intersection control strategies

Reservation-based intersection was first proposed by Dresner and Stone (2004), dividing an intersection to a grid consisting of cells with $n \times n$ granularity and distributing temporal and spatial resources for intersection control. The system was further investigated systematically by Dresner and Stone (2008), Au et al. (2011), Au et al. (2012), Li et al. (2013a, 2013b), Li et al. (2015) and Levin and Boyles (2015). Tachet et al. (2016) proposed a dynamic batch heuristic and mainly focused on the properties of intersection capacity while utilizing the batch strategy. Other control policies, such as Auction (Schepperle and Böhm, 2007) and csPriorFCFS (Zhang et al., 2016), were also proposed based on the reservation-based control. Besides priority-based reservation-based strategies, Levin and

Rey (2017) started to introduce optimization strategies into the reservation-based intersection control in order to improve the intersection efficiency. Other optimization-based strategies were also proposed based on the reservation-based system, such as Zhu and Ukkusuri (2015), Levin et al. (2016), Lin et al. (2017), Bichiou and Rakha (2018), Wu et al. (2019) and Ma and Li (2020). Taking pedestrians with the reservation-based system into account, Niels et al. (2020) integrated pedestrian control with the reservation-based control of vehicles and tested the integrated under different scenarios. Furthermore, Chen et al. (2020) utilized max pressure control to analyze the stability of the reservation-based system when pedestrians are involved. Wu et al. (2022a) also modelled the pedestrians with CAVs and optimized arrival time of both of them simultaneously.

Building upon idea of the reservation-based system, multiple AIM protocols/policies were proposed successively to overcome the drawbacks existed in the original version of the AIM. Reservation-based system and first-come-first-serve (FCFS) policy were first proposed by Dresner and Stone (2004), where vehicles are served in a sequence ordered by their request time. Dresner and Stone (2008)further improved the policy by combining traffic light with FCFS to accommodate CAV and Human-driven vehicles (HDV) at the same time. Sharon and Stone (2017) further proposed a new protocol named hybrid autonomous intersection management (H-AIM) to improve intersection performance under mixed traffic conditions. Schepperle and Böhm (2007, 2008) proposed an auction mechanism to improve intersection efficiency meanwhile consider drivers' valuation by processing bids from each driver. Tachet et al. (2016) proposed a new intersection management policy called BATCH of reservation in AIM, where vehicles are batched and processed together from any approaching lanes if the vehicles arrive within an interval,

which is defined as the delay between earliest arrival time and actual/assigned arrival time of the first vehicle in the batch. Wu et al. (2012) presented an AIM strategy based on an ant colony system and discrete optimization algorithm to solve real-time control problems. Lin et al. (2017) proposed a vehicle-intersection coordination method that divides a network into three logical sections with optimizing vehicle trajectories in a buffer-assignment mechanism. Wu et al. (2019) proposed a decentralized coordination learning of autonomous intersection management (DCL-AIM) with benefit of real time implementation. Overall, two main problems are needed to be investigated with the AIM: the one is how and when vehicles should arrive at the intersection, and the other is how to avoid conflicts between conflicting vehicles within the intersection. Regarding these two main problems, trajectory planning methods are proposed to solve the first, and cell- or point-based reservation formulation are proposed to solve the second.

3.2 Trajectory Planning of CAVs at a Reservation-based Intersection

3.2.1 Integrated trajectory planning with arrival time

Since the reservation-based system was proposed Dresner and Stone (2004), the acceleration rate along a road to an intersection was formulated by a linear function over time. Dresner and Stone (2008) further integrated the continuous trajectory modelling with FCFS strategy to compare with signalized control method in terms of intersection efficiency. The linear trajectory modelling method was also utilized by Lee (2010), where an optimization problem rather than FCFS was formulated to minimize the overlapping trajectories of conflicting vehicles. Lee and Park (2012) fixed the acceleration rate in overall control from initial position to the end of a communication range. Lee et al. (2013) further extended their linear trajectory modeling method to multiple intersections on a

corridor and investigated the mobility, safety and energy improvements. However, Kamal et al. (2014) pointed out that the optimization problem (Lee and Park, 2012) may fail to find a feasible solution for some vehicles in an optimization cycle, thus a recovery mode was introduced for these vehicles, which eventually leads to a piecewise trajectory modeling method. Different from the recovery mode, Li et al. (2013a) introduced an advanced stop location strategy to maintain a high arrival speed at an intersection whenever reservations cannot be accepted. Overall, the linear trajectory modeling method may not find feasible or optimal solutions for CAVs in terms of intersection efficiency.

As opposed to the linear trajectory modelling, He et al. (2015) derived a multi-stage optimization model and approximated trajectory solutions over sub-stages. Wu et al. (2015) also applied the multi-stage optimization on a signalized corridor. Wan et al. (2016) analytically derived solutions for piecewise trajectory functions based on Pontryagin's minimum principle (PMP). Fayazi and Vahidi (2018) also modelled trajectories of CAVs as a piecewise function over time, where arrival time of CAVs is first optimized, then trajectory is optimized based on the assigned arrival time. Different from heuristic piecewise trajectory modelling, Malikopoulos et al. (2018) analytically derived the optimal solutions of acceleration/deceleration rate based on PMP, where arrival speed of each CAV can be different. Zhou et al. (2019) also applied the PMP to analytically derive 18 kinds of acceleration/deceleration control sequences after introducing speed bounds on CAVs. Chalaki and Malikopoulos (2021) derived the piecewise trajectory solutions based on PMP and varied arrival speed of each CAV at merging area as long as its average speed inside the area keeps as same as a constant value.

Apart from using PMP to derive trajectory solutions, Barth et al. (2011) proposed an arrival time-based trajectory modelling method based on piecewise trigonometric-linear function. Altan et al. (2017) further tested the trajectory modeling method on a signalized corridor. Yang et al. (2019) also utilized this method by considering queue along a corridor. Yang et al. (2021) and Du et al. (2021) respectively used the trajectory modeling method in a mixed traffic environment.

In addition, trajectory solutions are also found through numerical optimization based methods by discretizing simulation times. Mirheli et al. (2019) proposed a mixed-integer nonlinear programming (MINLP) method to minimize travel time and speed variation among CAVs. Niroumand et al. (2020) further solved the MINLP by receding horizon control in a mixed traffic environment. Tajalli and Hajbabaie (2021) further proposed a MINLP model with a multi-objective of maximizing travel distance and minimizing speed difference among CAVs. Yao and Li (2020) proposed a discrete-time decentralized trajectory control method at a single-lane segment for a mixed traffic environment. Mu et al. (2021) proposed an event-triggered rolling horizon based trajectory planning method to optimize the merging process in a mixed traffic environment based on MINLP. Xiong and Jiang (2021) proposed a piecewise trajectory optimization problem solved by dynamic programming in a mixed traffic environment. Zhao et al. (2021) modelled a bilevel scheduling and trajectory planning model and solved the model using a heuristic; Yang et al. (2016) also proposed a bilevel programming model and solved it using branch-and-bound algorithm. Liu et al. (2022) proposed a discrete-time trajectory MILP optimization model using each signal cycle as a prediction horizon.

Note that all above methods model the trajectory control problem in an integrated way, where the arrival speed of CAVs is dependent on their assigned arrival time at an intersection or a merging point.

3.2.2 Independent trajectory planning with arrival time

As opposed to integrated trajectory modelling with arrival time, Liu et al. (2011) designed an independent model that finds trajectory solutions with a given arrival speed based on PMP. Rios-Torres and Malikopoulos (2016) applied the PMP on a merging point where arrival time of CAVs is assigned based on First-in-first-out and arrival speed of them is assigned with their initial speed. Feng et al. (2018) also found trajectory solutions with a given arrival speed based on PMP; Yu et al. (2018) further simplified the derivation of trajectory solutions to six scenarios based on an assigned arrival time; Yu et al. (2019) then applied the model on a reservation-based intersection. Wang et al. (2020) developed a framework on a network where planned arrival time of CAVs based on a heuristic and found trajectory solutions based on PMP. Dollar and Vahidi (2021) also utilized the Hamiltonian analysis combining with receding horizon control to derive trajectory solutions over long- and short-term planning horizons.

Moreover, Zhou et al. (2017) and Ma et al. (2017) subsequently proposed and finalized a piecewise trajectory planning model to smooth a stream of CAVs based on shooting heuristic. Li et al. (2018) further simplified the trajectory model by confining each trajectory to consist of no more than five quadratic sections and improved the solutions to near-optimum. Soleimaniamiri et al. (2020) further improved an analytical algorithm by allowing different accelerations in a platoon of vehicles. Guo et al. (2019) then integrated

the shooting heuristic with dynamic programming to optimize the vehicle trajectory and arrival time simultaneously and applied the model to a mixed traffic environment.

In addition to deriving trajectory solutions based on PMP or shooting/meta heuristic, Li et al. (2014) derived a trajectory planning model that contains at most four segments for each trajectory based on a heuristic. Stebbins et al. (2017) designed a speed advisory model that aims to minimize delay and applied it to accommodate a platoon of vehicles; Zhang et al. (2020) further applied and extended the model to a network of reservation-based intersections. Xu et al. (2018a) found solutions of arrival time and trajectory of CAVs simultaneously and solved the problem using an enumeration and pseudo-spectral method. Li et al. (2019) developed a temporal and spatial trajectory planning heuristic and assigned arrival time of CAVs by two heuristics.

Furthermore, the trajectory solutions with assigned arrival time can also be found via optimization methods. Han et al. (2020) proposed a platoon control method and found trajectory solutions for the platoon by numerically solving a MINLP. Xu et al. (2020b) also designed a discrete-time exact trajectory model based on nonlinear programming and proved it as a convex optimization model. Yao et al. (2022) designed a hierarchical stage programming model that finds trajectory solutions and arrival time of CAVs simultaneously by a rolling horizon. It is remarkably noted that Ma and Li (2021) built a single-layer MILP model with both arrival time and arrival speed as variables, such that both of them can be optimized simultaneously and theoretical properties can be derived from the model; in addition, a rolling horizon was applied based on number of vehicles in each optimization cycle.

3.3 Arrival Sequence Scheduling of CAVs at a Reservation-based Intersection

3.3.1 MILP modeling of a scheduling problem

A scheduling problem of CAVs for crossing a reservation-based intersection aims to solve conflicts of crossing. Conflict solving techniques can stem from first-in-first-out strategy (Dresner and Stone, 2004, 2008; Li et al., 2015; Li et al., 2013a). However, such strategy cannot guarantee optimality of solutions of the scheduling problem in terms of overall travel time or average delay of the intersection. Zhu and Ukkusuri (2015) started to model the scheduling problem via linear programming, however, the mode is based on a lane level rather than from individual vehicle's perspective. Levin and Rey (2017) then modeled the problem via MILP and solved the MILP using a rolling horizon as well as a fixed horizon method. Li and Zhou (2017) also modeled the problem as a MILP and solved it using branch-and-bound algorithm. Fayazi and Vahidi (2018) solved such MILP model using commercial solver Cplex and extended it into a mixed traffic environment. Yu et al. (2019) also modeled the problem as a MILP and investigated its performance under various traffic demands. Xu et al. (2020a) incorporated lane changes into the MILP model and solved it in a bilevel planning method. Zhang et al. (2020) solved such MILP using solver Gurobi in a nearly real time because of simplified constraints in the model. Ge et al. (2021) also proposed a lane-based MILP and solved it by clustering vehicle subgraphs in a directed graph. Ma and Li (2021) modeled the MILP from individual vehicle's perspective and solved it using a rolling horizon method. Tajalli and Hajbabaie (2021) transformed a mixed integer nonlinear programming model to a MILP, then solved it by decomposing it into subproblems and tightened solution space by introducing a set of cliques. Yang et al. (2021) also modeled a MILP to optimize arrival times of CAVs at an intersection in a mixed traffic

environment and solved it at an approach level. Zhao et al. (2021) modeled a vehicle-based MILP in the upper level of a bilevel programming and solved it using a heuristic. Liu et al. (2022) modeled a MILP for optimizing arrival time and trajectory of CAVs simultaneously. Lu et al. (2022) modeled three MILPs to evaluate performances among different strategies under a fully CAV environment.

It is shown above that solving conflicts between CAVs has been mainly modeled via MILP and the objective is to optimize the intersection efficiency. Regarding different modeling methods, such as based on individual lane- or vehicle-level, Ma and Li (2022a) pointed out necessary vehicle-based conditions in order to reach an optimality of the intersection efficiency. Under such optimal conditions, however, it takes exponential growth time complexity for commercial solvers to solve such a MILP model. Efficient solution algorithms are followingly proposed to overcome this problem.

3.3.2 Solution algorithms

In the reservation-based intersection scheduling problem, solution algorithms include approximation algorithms and exact algorithms: the former finds a near optimum solution and the latter finds an optimal solution to the problem. Regarding the approximation algorithms, Xu et al. (2018b) and Hu et al. (2021) both utilized a Depth-first search (DFS) to solve a depth-first spanning tree problem in a directed graph and found a near optimal solution. Ding et al. (2019) proposed a group-based heuristic to balance a trade-off between optimality and computation time. Li et al. (2019) proposed a Tabu search heuristic to find near-optimal solutions of the scheduling problem. Xu et al. (2019a) also proposed a heuristic algorithm that groups vehicles if time headways between any pair of leading and following vehicles in the group are less than a specific value. Xu et al. (2019b) applied a

Monte Carlo tree search (MCTS) method with two heuristic rules to find a near optimum solution. Xu et al. (2021) further compared four heuristics in terms of computation time and analyzed time complexity for these strategies. Tang et al. (2022) also applied the MCTS to solve the scheduling problem to a near optimal solution.

Regarding the exact algorithms, Li and Wang (2006) initiated a modified brute-force tree search algorithm. Wu et al. (2009) started to apply dynamic programming (DP) to minimize the maximal arrival time of CAVs and analyzed the time complexity. Yan et al. (2011) developed a branch & bound (B&B) algorithm with heuristic lower and upper bounds to solve such a scheduling problem. Yang et al. (2016) also utilized the B&B algorithm based on first-in-first-out rule under depth-first search. Besa Vial et al. (2016) analyzed the time complexity of the scheduling problem by using dynamic programming algorithm for different types of reservation-based intersections. Pei et al. (2019) applied the DP and reduced the time complexity at a reservation-based intersection with two approaches; Pei et al. (2021) further extended the problem to an intersection with four approaches. Sun et al. (2020) utilized the DP and solved the scheduling problem in a mixed traffic environment.

In summary, the time complexity of the above algorithms is illustrated in

***Table* 1** that details the number of approaches at an intersection and upper and lower bounds of the time complexity.

Table 1 Summary of time complexity of algorithms

	Algorithm	No. of approaches	Time complexity		Optimal
			Upper bound	Lower bound	
Wu et al. (2009)	DP	4	$\mathcal{O}(\mathcal{V}^{10} log \mathcal{V})$	/	Yes
Besa Vial et al. (2016)	DP	k	$\mathcal{O}(\min(\mathcal{V}'^{2k}, \mathcal{V}'^{k} log T))$	/	Yes

Xu et al. (2018b)	DFS	4	$\mathcal{O}(\mathcal{V}^2)$	/	No
Pei et al. (2019)	DP	2	$\mathcal{O}(\mathcal{V}^2)$	/	Yes
Xu et al. (2019a)	Heuristic	2	$\mathcal{O}(\mathcal{k}! \cdot \mathcal{V})$	/	No
Pei et al. (2021)	DP	4	$\mathcal{O}(\mathcal{V}^6)$	$\mathcal{O}(\mathcal{V}^4)$	Yes
This paper	DP	2	$\mathcal{O}(\mathcal{V}^2)$	$\mathcal{O}(\mathcal{V})$	Yes

Note in an optimization: $\mathcal{V}$ denotes number of CAVs, $\mathcal{V}'$ denotes number of platoons of CAVs, k denotes number of approaches, T denotes an upper bound of an optimal solution and $\mathcal{k}$ denotes number of groups.

3.3.3 Platooning of CAVs

Platoon-based intersection control can be traced back to Jiang et al. (2006), where platoons are identified and utilized to minimize intersection delay. Wu et al. (2013) then started to mathematically analyze the platoon-based control based on Petri Nets model and identified a method to increase throughput by platooning. Chen and Kang (2015) proposed a dynamic platooning and scheduling algorithm and analyzed the time complexity. Tachet et al. (2016) also proposed a dynamic platooning method and analyzed its performance via simulations and queue theory. Lioris et al. (2017) further identified that the platoons of CAVs can double intersection capacity via queuing theory. Liu et al. (2018) modeled a framework for platoons in a mixed traffic environment considering lane changing behaviors. An et al. (2021) further proposed an analytical model to investigate the effect of platoons on capacity in a mixed traffic environment. Chen and Mårtensson (2021) proposed a MILP to simultaneously schedule platoons and maintain integrity of platoons. Kumaravel et al. (2021) proposed a scheduling method for platoons via modeling a job-shop scheduling problem with forming cliques. Timmerman and Boon (2021) also proposed a method to schedule CAVs as platoons crossing the intersection and analyzed intersection delay under the algorithm by queue theory. Zhou and Zhu (2021) analyzed the effect of a maximum

size of a platoon on capacity and stability. Wu et al. (2022b) further modeled and analyzed the effect of the platoons on the intersection capacity in a mixed traffic environment. Zhou et al. (2022) proposed a virtual platooning coordination strategy and two heuristics to schedule the platoons crossing the intersection.

It is concluded from above that platooning of CAVs is beneficial in maximizing the intersection capacity and minimizing the intersection delay. It is also promising and yet to be investigated to incorporate the platooning strategies into solution algorithms of the aforementioned MILP in order to reduce the time complexity. Especially, batching CAVs in a platoon during stochastic arrivals is significant for implementing solution algorithms in a scheduling problem.

3.4 Summary

In summary, the temporal and spatial resources of the intersection are fully utilized by allowing conflicting vehicles arriving into the intersection simultaneously only if they are not colliding at certain cells and at periods of time they requested to occupy, whereas under traditional signal controller, vehicles cannot enter the intersection unless conflicting vehicles leave the intersection. Therefore, the capacity as well as efficiency can be improved by discretizing the intersection into a set of cells temporally and spatially.

A list of average vehicle delay under different traffic condition and control strategy of typical scenarios is provided in **Table *2***, where the delay ranges from 2.6 to 164 seconds per vehicle. Note here that if optimization strategy is used, turning movements are considered, or conflict point/cell-based formulation is utilized, intersection granularity is set, and how intersection demand is varied would highly impact the average vehicle delay.

Table 2 Average vehicle delay under different conditions of typical scenarios

	Yu et al. (2019)	Li et al. (2019)	Yao et al. (2022)	Tachet et al. (2016)	Bashiri et al. (2018)
Intersection demand (veh/h)	4000	11520	4200	1800	2600
Optimization	Y	Y	Y	N	Y
Turning movements	Y	Y	Y	N	Y
Conflict measurement	Cell	Point	Cell	Cell	Cell
Granularity (cells)	48	0	1	1	1
CAV penetration rate (%)	100	100	100	100	100
Average delay (s/veh)	35	4.1	164	2.6	20

However, although the vehicle arrival time and trajectory can be jointly optimized to minimize the travel delay, the trajectory planning only can be realized when the arrival time and arrival speed are given to the CAVs. As noted, the arrival time and arrival speed would not be maintained when traffic condition varies (Au et al., 2012), thus the planned trajectory would fail to be followed and vehicles would collide with vehicles from conflicting approaches within the intersection.

Under various traffic scenarios, the trajectory, especially the arrival speed, is modelled either in an integrated or independent way with the arrival time of CAVs. Under such modelling methods, the intersection efficiency is optimized to different levels of operational efficiency. Nonetheless, the optimal control method has not been defined yet in terms of finding the optimal intersection efficiency.

In order to achieve the optimum of intersection performances, it is necessary to take every CAV into consideration in the modeling and scheduling problem at a reservation-based intersection. To solve such a scheduling problem, efficient solution algorithms are required to deal with the curse of dimensionality. In order to achieve a better tradeoff between the optimal solution and the computational demand, platooning strategies can be leveraged and are yet to be discussed.

CHAPTER 4. A TIME-INDEPENDENT TRAJECTORY OPTIMIZATION APPROACH

4.1 Reservation-based Control Concept

The reservation system was first proposed by Dresner and Stone (2004)and stemmed from an attempt that people make reservations before they arrive in a hotel. In this case, the people are the vehicles, and the hotel is an intersection manager (IM). Vehicles send a reservation request and the IM determines whether/when/how vehicles can cross the intersection without conflicts. Vehicles will go through following travelling instructions from IM if the request is approved, otherwise vehicles will receive a "counter-offer".

An intersection is divided to an n by n grid of tiles. Each tile can be occupied by one vehicle per simulation step. One of the motivations of reservation-based system is to utilize all intersection tile resources temporarily and spatially so as to improve the intersection efficiency. No double or more occupation requests on one tile by two or more vehicles simultaneously can be accepted. **Figure *1*** illustrates that a vehicle occupies certain tiles at a certain time instant under reservation-based system.

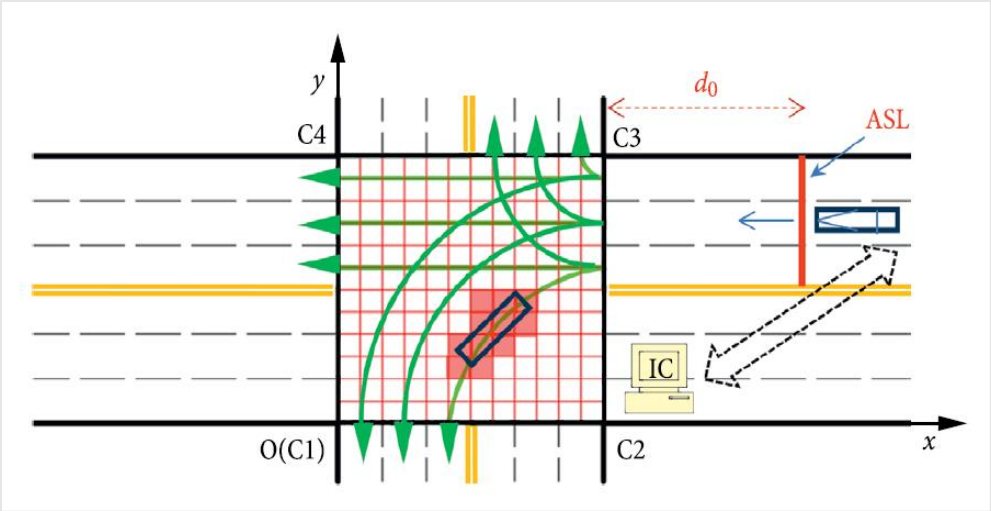


Figure 1 Intersection cells occupied by one vehicle at a time instant

4.2 Arrival Time-speed-independent Trajectory (TSIT) Planning Method

In an AIM environment, the vehicles follow the trajectory planned by an intersection controller and arrive at the intersection with a given arrival time and speed (Fayazi and Vahidi, 2018; Feng et al., 2018; Malikopoulos et al., 2018; Soleimaniamiri et al., 2020; Yao et al., 2022). However, when the traffic condition varies over time, i.e., a priority passing sequence to the intersection is shifted from one earlier coming vehicle to another later coming vehicle, or the traffic demand is increasing sharply from one approach, the vehicle may not follow the planned trajectory and arrive at the intersection with an expected time and speed (Au et al., 2012; Au and Stone, 2010). To overcome this fail-follow problem, an arrival time-speed-independent trajectory (TSIT) optimization approach is proposed to adapt to varying condition without arrival time or speed predetermined.

In the TSIT, the arrival time and speed at the intersection are not predetermined and optimized simultaneously with the trajectory along the road. Furthermore, the acceleration rate varies along the trajectory over each simulation time, 0.1 second. The formulation approach renders instant changes of arrival time and speed available if the traffic condition

is changing or the traffic demand is increasing sharply at certain periods of time, such that a vehicle will be assigned with a new arrival time and speed and be able to follow a new trajectory to the intersection. The variable arrival time and speed at the intersection are formulated in a mixed integer linear programming (MILP) approach, where they are only measured and updated by a possible maximum travel time of each vehicle in one optimization cycle, rather than fixed with predetermined numbers. Therefore, the objective of this formulation is to minimize the maximum travel time of each vehicle. Furthermore, the conflicts are avoided within the intersection by ensuring each cell within the intersection can only be occupied by one vehicle at any time instant. The formulation details are explained in the following section.

Before introducing the formulation approach, some assumptions are considered throughout this section:

1. Signal controller as well as the timing and phasing design is not considered in the reservation-based system;
2. All vehicles are CAV and fully controlled by an intersection controller, i.e., 100% CAV penetration rate;
3. No right or left turn movements are considered within the intersection; and
4. No communication latency is considered between the vehicle and the infrastructure.

4.3 TSIT Formulation

4.3.1 Notation

Decision variables and parameters applied hereafter are summarized in **Table *3***.

Table 3 Decision variables and parameters

Decision variables	
z	Maximum value of travel time of all vehicles in each simulation round, s

$x_{t,p,i}$	Binary variable, vehicle i from approach p at time t travel status within the intersection
$lx_{t,p,i}$	Horizontal coordinate of vehicle i from approach p at time t , *ft*
$ly_{t,p,i}$	Vertical coordinate of vehicle i from approach p at time t , *ft*
$v_{t,p,i}$	Speed of vehicle i from approach p at time t , *ft/s*
$a_{t,p,i}$	Acceleration rate of vehicle i from approach p at time t , *ft/s*^{2}
$itf_{t,p,i}$	Binary variable, intermediate variable to quantify cell (m,n) occupation status by front bumper of vehicle i from approach p at time t
$itr_{t,p,i}$	Binary variable, intermediate variable to quantify cell (m,n) occupation status by rear bumper of vehicle i from approach p at time t
$ito_{t,p,i}$	Binary variable, intermediate variable to quantify cell (m,n) occupation status by center of vehicle i from approach p at time t
$c_{t,p,i}^{m,n}$	Binary variable, cell (m,n) occupation status by vehicle i from approach p at time t
$c_t^{m,n}$	Binary variable, cell (m,n) occupation status at time t
$c_t^{m,n,G}$	Binary variable, global cell (m,n) occupation status at time t
Sets	
T	Maximum time range of time step t, *second*
P	Sets of approaches p
I	Sets of vehicles i
H	Sets of horizontal cells m
V	Sets of vertical cells n
Parameters	
$lx_{0,p,i}$	Initial horizontal coordinate of vehicle i from approach p at time 0, *ft*
$ly_{0,p,i}$	Initial vertical coordinate of vehicle i from approach p at time 0, *ft*
$v_{0,p,i}$	Initial speed of vehicle i from approach p at time 0, *ft/s*
$\ell_{p,i}$	Vehicle length of vehicle i from approach p *ft*
h	Length of a square intersection, 40 *ft*
d	Length of an optimization range before boundaries of an intersection, 600 *ft*
s	Length of a square cell, 10 ft for BATCH, 20 ft for ZONE

4.3.2 Objective function

The objective of this problem to minimize the maximum travel time of all vehicles entering into an optimization cycle. The travel time is measured by the binary variable $x_{t,p,i}$, which equals 1 if the vehicle is not cleared by the intersection and 0 if the vehicle leaves the intersection.

$$\min z = max \sum_{0}^{T} x_{t,p,i} \quad (1)$$

4.3.3 Traffic simulation constraints

From very beginning, all vehicles with coordinates ($lx_{t,p,i}, ly_{t,p,i}$) will be programmed to enter the simulation from boundary of optimization range as defined in Eqn. (2) and (3); their initial speed is assigned with an initial speed as defined in Eqn. (4). The acceleration or deceleration rate, as a decision variable, keeps the same through all simulation time as defined in Eqn. (5); this constraint is used to compare with the other condition in Eqn. (6), where the acceleration rate varies all the simulation time if needed, in terms of travel time minimization. The optimal solutions of the acceleration rate along with the arrival time/speed are found via the MILP, which eliminates the requirements of a predetermined arrival time/speed, and then the solutions of the trajectory can adapt to varying traffic conditions by relaxing the constraints of acceleration rate along the trajectory.

$$lx_{t,p,i} = lx_{0,p,i} \quad t = 0, \forall i \in I, \forall p \in P \quad (2)$$

$$ly_{t,p,i} = ly_{0,p,i} \quad t = 0, \forall i \in I, \forall p \in P \quad (3)$$

$$v_{t,p,i} = v_{0,p,i} \quad t = 0, \forall i \in I, \forall p \in P \quad (4)$$

$$a_{t,p,i} = a_{t-1,p,i} \quad t \geq 1, \forall i \in I, \forall p \in P \quad (5)$$

$$|a_{t,p,i} - a_{t-1,p,i}| \leq 3 \quad t \geq 1, \forall i \in I, \forall p \in P \quad (6)$$

Vehicle trajectory coordinates and speed variations are optimized per 0.1 second as defined in Eqns. (7) to (9); Eqns. (10) and (11) constrain range of acceleration rate and speed with

unit *ft/s*^{2} and *ft/s*. Therefore, the speed of each vehicle can vary within the intersection area and is not a fixed number while vehicles travel through the intersection.

$$lx_{t,p,i} = lx_{t-1,p,i} + 0.1v_{t-1,p,i} + 0.5a_{t-1,p,i} \times 0.01 \quad t \geq 1, \forall i \in I, \forall p \in (1,2) \quad (7)$$

$$ly_{t,p,i} = ly_{t-1,p,i} + 0.1v_{t-1,p,i} + 0.5a_{t-1,p,i} \times 0.01 \quad t \geq 1, \forall i \in I, \forall p \in (3,4) \quad (8)$$

$$v = v_{t-1,p,i} + 0.1a_{t-1,p,i} \quad t \geq 1, \forall i \in I, \forall p \in P \quad (9)$$

$$-15 \leq 0.1a_{t,p,i} \leq 10 \quad \forall t \in T, \forall i \in I, \forall p \in P \quad (10)$$

$$0 \leq v_{t,p,i} \leq 60 \quad \forall t \in T, \forall i \in I, \forall p \in P \quad (11)$$

Regarding the basic car following model, in this scenario, it is simplified for safety distance between front bumper of leading and following vehicles by using 18 feet, as defined in Eqns. (12) and (13).

$$lx_{t,p,i} \leq lx^L_{t,p,i} - 18 \quad \forall t \in T, \forall i \in I, \forall p \in (1,2) \quad (12)$$

$$ly_{t,p,i} \leq ly^L_{t,p,i} - 18 \quad \forall t \in T, \forall i \in I, \forall p \in (3,4) \quad (13)$$

4.3.4 Travel time measurement constraints

To measure travel time of vehicles in each simulation round before they leave the intersection, time represented by $x_{t,p,i}$ equals 1 as long as those vehicles are not leaving out of the boundary of intersection, otherwise equals 0, as defined in Eqns. (14) and (15).

$$\begin{cases} lx_{t,p,i} \geq h + d + \ell_{p,i} - Mx_{t,p,i} \quad \forall t \in T, \forall i \in I, \forall p \in (1,2) \\ lx_{t,p,i} \leq h + d + \ell_{p,i} + M(1 - x_{t,p,i}) \quad \forall t \in T, \forall i \in I, \forall p \in (1,2) \end{cases} \quad (14)$$

$$\begin{cases} ly_{t,p,i} \geq h + d + \ell_{p,i} - Mx_{t,p,i} \quad \forall t \in T, \forall i \in I, \forall p \in (3,4) \\ ly_{t,p,i} \leq h + d + \ell_{p,i} + M(1 - x_{t,p,i}) \quad \forall t \in T, \forall i \in I, \forall p \in (3,4) \end{cases} \quad (15)$$

4.3.5 Conflict avoidance constraints

In this simplified scenario, the intersection is divided into 16 or 4 cells identified by m and n, each of which is a square with length s; $itf_{t,p,i}$, $itr_{t,p,i}$ and $ito_{t,p,i}$ are formulated as intermediate variables to measure values of $c_{t,p,i}^{m,n}$; $c_{t,p,i}^{m,n}$ equals 1 as long as any part of a vehicle i from approach p is occupying cell (m,n), otherwise equals 0, as defined in Eqs. (16) and (17).

$$\begin{cases} d+sn \le lx_{t,p,i} + M(1-itf_{t,p,i}) \\ d+sn \ge lx_{t,p,i} + M1 - itf_{t,p,i} \\ lx_{t,p,i} \le d+s(n+1) + \ell_{p,i} + M(1-itr_{t,p,i}) \\ lx_{t,p,i} \ge d+s(n+1) + \ell_{p,i} - Mitr_{t,p,i} \\ d+sm \le ly_{t,p,i} + M(1-ito_{t,p,i}) \\ d+sm \ge ly_{t,p,i} - Mito_{t,p,i} \\ 0 \le itf_{t,p,i} + itr_{t,p,i} + ito_{t,p,i} - 3c_{t,p,i}^{m,n} \le 1 \\ \forall t \in T, \forall m \in H, \forall n \in V, \forall i \in I, \forall p \in (1,2) \end{cases} \tag{16}$$

$$\begin{cases} d+sm \le ly_{t,p,i} + M(1-itf_{t,p,i}) \\ d+sm \ge ly_{t,p,i} + M1 - itf_{t,p,i} \\ ly_{t,p,i} \le d+s(m+1) + \ell_{p,i} + M(1-itr_{t,p,i}) \\ ly_{t,p,i} \ge d+s(m+1) + \ell_{p,i} - Mitr_{t,p,i} \\ d+sn \le lx_{t,p,i} + M(1-ito_{t,p,i}) \\ d+sn \ge lx_{t,p,i} - Mito_{t,p,i} \\ 0 \le itf_{t,p,i} + itr_{t,p,i} + ito_{t,p,i} - 3c_{t,p,i}^{m,n} \le 1 \\ \forall t \in T, \forall m \in H, \forall n \in V, \forall i \in I, \forall p \in (3,4) \end{cases} \tag{17}$$

Each cells' occupation status $c_t^{m,n}$ at time *t* is formatted as a 0-1 matrix, which is updated by vehicles' trajectory coordinates, as defined in Eqn. (18).

$$c_t^{m,n} = \sum_{p}^{P} \sum_{i}^{I} c_{t,p,i}^{m,n} \le 1 \;\; \forall t \in T, \forall m \in H, \forall n \in V, \forall i \in I, \forall p \in P \tag{18}$$

Furthermore, $c_t^{m,n}$ will be combined into a global matrix $c_t^{m,n,G}$ and any elements in the global matrix should not exceed 1. Such that, conflict avoidance between any vehicles at any time is accomplished in Eqn. (19).

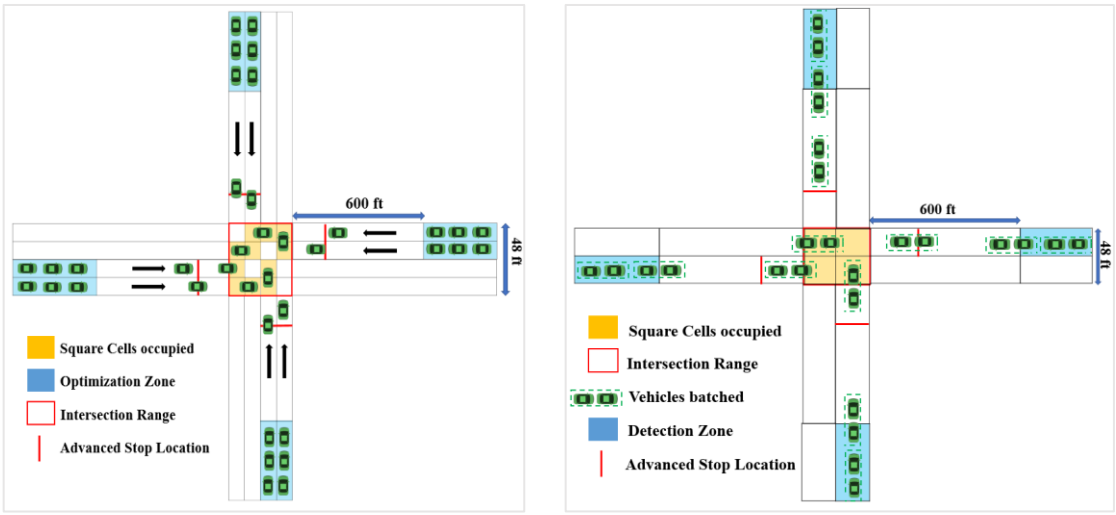
$$c_t^{m,n} + c_t^{m,n,G} \leq 1 \;\; \forall t \in T, \forall m \in H, \forall n \in V \tag{19}$$

4.4 Performance Evaluation

4.4.1 Experimental design

The intersection is divided into 4 x 4 cells or 2 x 2 cells for implementing different optimization strategy BATCH and ZONE, where the first strategy collects all the vehicles information from any lanes of any approaches once they enter into an optimization cycle, which is determined by the arrival rate, and processes them all as a batch; and the second strategy also processes all the vehicle together once they enter into an optimization cycle, but there is only one lane per approach, which is shown in Figure 2. Once vehicles are batched together, the intersection controller processes all the vehicles via a mixed integer linear programming approach to minimize the average intersection delay.

An optimization cycle detects vehicles if they enter per 0.1 second. All vehicles detected will be included in an optimization iteration at their instant time step. The two lanes-four approaches intersection design is shown in Figure 2 (a), where the vehicles are processed within a batch. The one lane-four approaches intersection is shown in Figure 2(b), where vehicles are processed together once they enter into the detection zone of the optimization range. The difference between the BATCH (Fig. 2(a)) and the ZONE (Fig. 2(b)) is mainly about the intersection design, which is to test the optimization results under different intersection granularity and different formulation approach, i.e., with or without variable acceleration rate, in terms of travel delay.



(a) (b)

Figure 2 Optimization strategies: (a) BATCH; (b) ZONE

Once any vehicles enter those zones highlighted in blue as illustrated in Figure 2, an optimization iteration will be activated for those vehicles at that simulation step. From each approach, vehicles will only reserve tiles at front of them and cells reserved are highlighted in yellow. An intersection range (a red frame) is used to determine when vehicles enter and leave the intersection so as to measure travel time. Both scenarios have 600 feet communication range at upstream of boundary of intersection and 40 feet square intersection design. Only through movements are simulated in these two scenarios.

For these two scenarios, the formulation of BATCH/ZONE with variable acceleration rate uses Eqn. (1) as objective function, Eqns. (2) through (4) and (6) through (19) as constraints; the formulation of BATCH/ZONE with constant acceleration rate uses the same objective function as defined by Eqn. (1) and applies Eqns. (2) through (5) and (7) through (19) as constraints.

4.4.2 Intersection granularity

As Fig. 2 shown above, two different intersection granularities are designed for each scenario. There are 16 cells in BATCH, 4 cells in ZONE strategy and same intersection

dimension for each. Therefore, the TSIT can be evaluated under different control strategy and different intersection granularity in terms of travel delay and computation time.

4.4.3 Simulation environment

As formulated in above section, one of key decision variables, acceleration rate, is investigated in terms of its effect on the performances of the TSIT. Parameter values of acceleration rate range and other variables are illustrated in **Table *3***.

For these two scenarios, travel demands range from 300 to 1800 veh/hr/ln; each scenario under different demands and strategies is simulated 10 minutes including 5 minutes warm-up session. Cplex was used to implement the simulation and optimization at an Intel Core i3-3220 CPU computer with 3.30 GHz processor and 8 GB RAM.

To further evaluate the performance of the TSIT, a state-of-art rule-based approach, Dynamic Batch (Tachet et al., 2016) (DB), is performed under the same traffic demand with two different maximum batch number, 8 (DB8) and 20 (DB20). Correspondingly, the headway gap of vehicles at the same approach and different approach is computed by the intersection width, vehicle length and arrival speed.

4.4.4 Solution approach

The optimization is formulated by a MILP and can be solved easily through commercial optimization package, such as Cplex, to a globally optimal solution by enumerating all feasible solutions via their solution approaches, e.g., Branch & Cut, enhanced heuristic. In terms of each cell within the intersection, every vehicle will be assigned with a binary variable, $c_{t,p,i}^{m,n}$, o or 1, so as to order the sequence of which vehicle will occupy the cell (m, n) at time t, and if a vehicle (p, i) occupies it, all other vehicles will correspond to this decision by being assigned a new $c_{t,p,i}^{m,n}$ value until all vehicles are cleared by the current

optimization iteration, and then a travel time solution $\sum_0^T x_{t,p,i}$ can be obtained; however, this value may not be optimal, therefore, until all combinations of the $c_{t,p,i}^{m,n}$ of each vehicle are searched, an optimal solution will be found via this enumeration process, during this process, in order to facilitate the computation, some branches will be cut once no any other better solutions under the trees can be found than the current incumbent. Furthermore, Cplex can utilize different search strategies among their libraries automatically over different situations to balance the optimality and computation time. The details of the solution approach used in this paper can be referred to Yang et al. (2016).

4.5 Results

4.5.1 Computation time comparison

Totally, four scenarios were compared in terms of computation time consumed to lead to optimization convergence. The computation time for each scenario is an average of all optimization iterations with 0.1 second simulation step. These four scenarios are Zone strategy with constant acceleration rate (ZONE), Zone strategy with variable acceleration rate (ZONEV), BATCH strategy with constant acceleration rate (BATCH), and BATCH strategy with variable acceleration rate (BATCHV), respectively. The comparison result is illustrated in Fig. 3.

All convergence gaps of different scenarios at first start with 100%. Before the branch-and-bound algorithm starts, there exists a presolving process to enumerate feasible solutions in order to reduce the size of the problem in the optimization so that the formulation of the problem can be tightened. The computation time starts to kick in when the presolving processes begins.

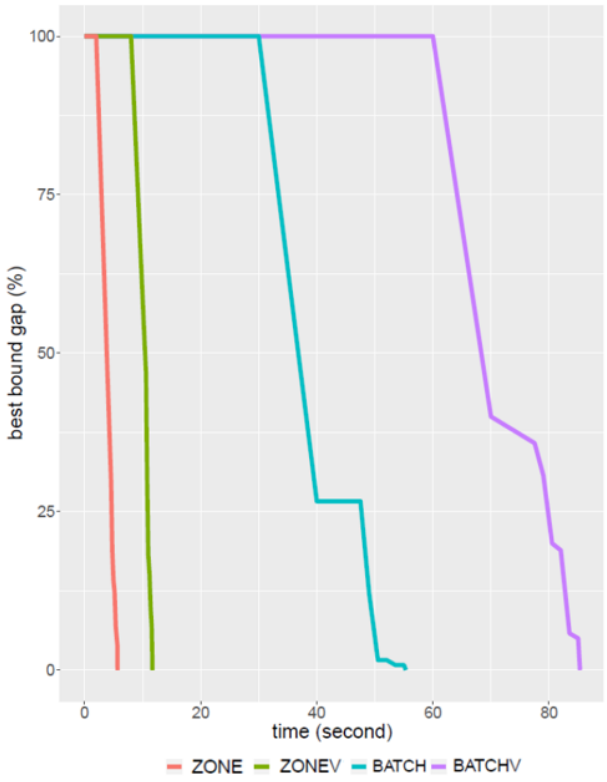


Figure 3 Convergence computation time under different scenarios

As shown in Fig. 3 above, ZONE costs least time and BATCHV costs the most. BATCHV costs more time than BATCH, either does ZONEV than ZONE. The computation time of BATCH and BATCHV is much more than that of ZONE and ZONEV.

It is because that reservation-based system with small granularity is more efficient than large granularity regarding the convergence time. It is also because that computation time increases for scenarios under variable acceleration rate as decision variable. It is noted that all best bound gaps between upper bound and lower bound converges to 0%, indicating the optimality of the solution to the TSIT problem.

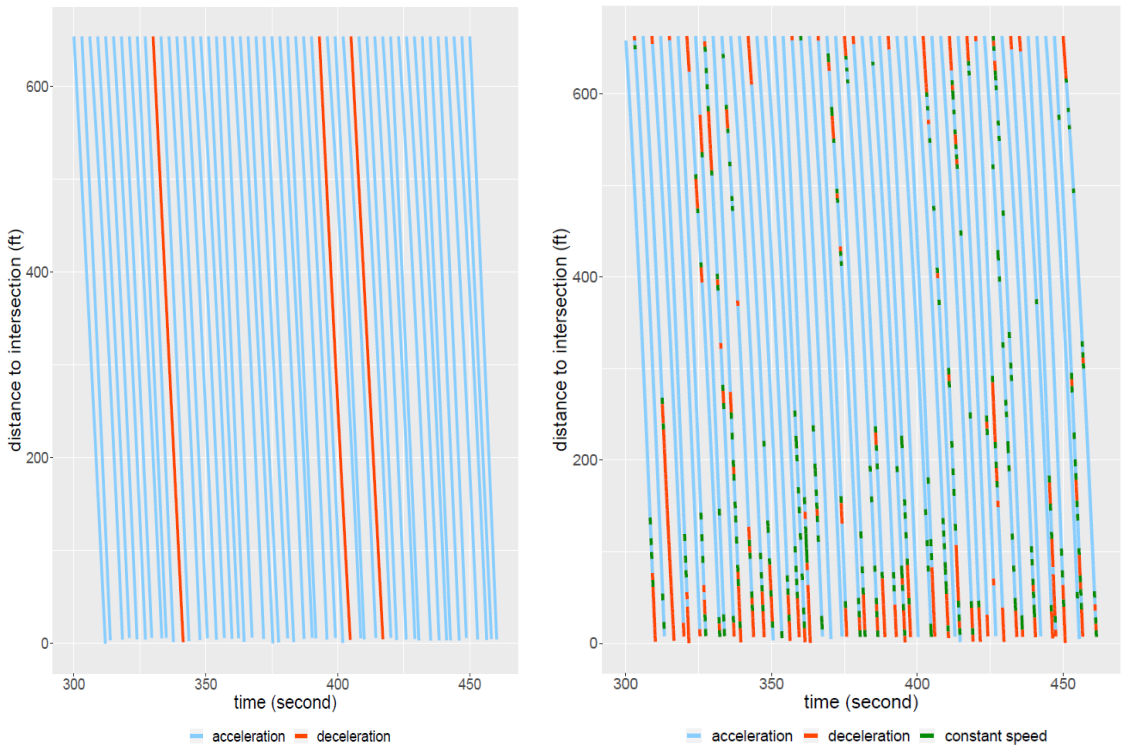
4.5.2 Trajectory analysis

In order to evaluate the simulation results of the formulation in this paper before comparing the travel efficiency of each optimization strategy, a trajectory diagram between distance from vehicle to intersection and simulation time is visualized for each scenario. Trajectory

diagrams of BATCH and BATCHV under 1200 veh/hr/ln travel demand are shown in Fig. 4.

As shown in Fig. 4, blue lines indicate accelerating, red segments/lines indicate decelerating and green segments indicate constant speed during their simulation time. Y-axis indicates the distance from vehicles' current position to downstream intersection boundary and X-axis indicates the simulation time.

As of BATCH strategy shown in Fig. 4(a), first, no conflicts are identified along the trajectory or within the intersection, because any lines would intersect with others if the vehicles represented by the lines collide with the others; second, all vehicles besides three of them were accelerating, which indicates a consistently free flow condition; third, some of vehicles had lower acceleration rates to hold constraints of safety distance and conflict avoidance, showing as bigger gaps between certain lines in the Fig.4 (a); fourth, some vehicles were travelling as a platoon while holding constraints, showing as smaller gaps between certain lines. Overall, a consistently free flow condition without any conflicts indicates the effectiveness of the formulation with BATCH strategy regarding travel efficiency.



(a) (b)

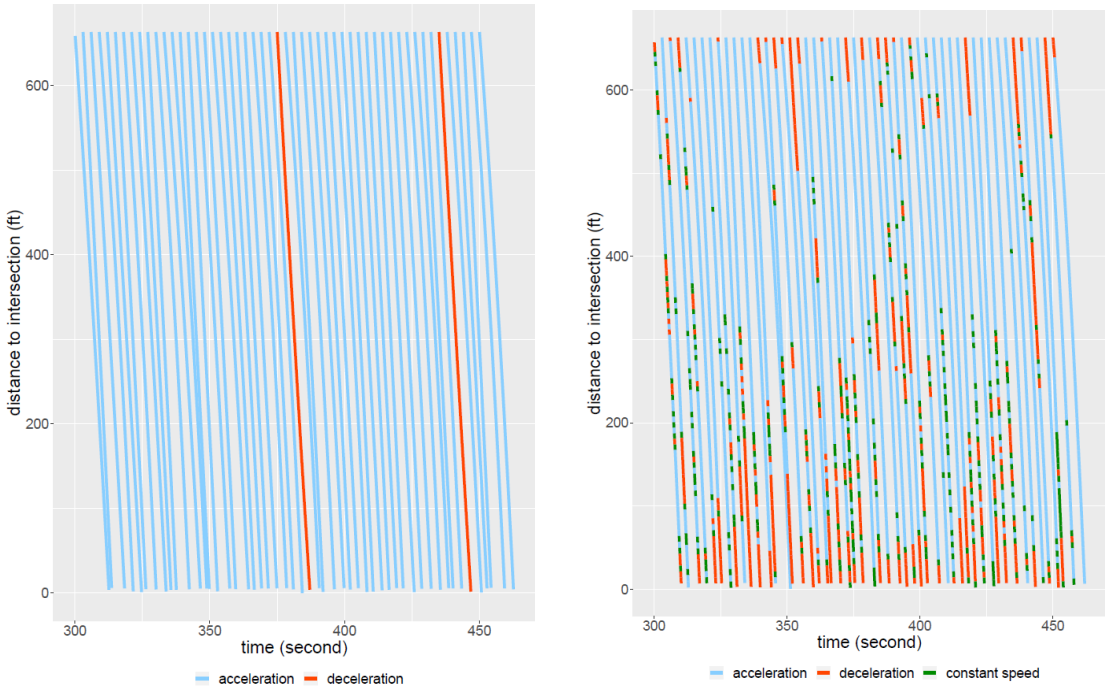
Figure 4 Trajectory diagrams of (a) BATCH and (b) BATCHV

As of BATCHV strategy shown in Fig. 4(b), first, no conflicts happened; second, vehicles were always accelerating before upstream intersection boundary while decelerating around intersection range considering reasonable vehicle dynamics; third, longer red segments might indicate a potential conflict was eliminated or a maximum speed was happened to reach; fourth, shorter green segments indicate that the room for an acceleration rate is acceptable, considering all constraints; fifth, acceleration was taken at most of simulation time, indicating an efficient and safe traffic flow. Overall, this formulation outputs a stable simulation status even under high travel demand, i.e., the TSIT can accommodate all vehicles without explicit delay in the trajectory diagram.

Trajectory diagrams of ZONE and ZONEV under 1200 veh/hr/ln travel demand are shown in Fig. 5. As ZONE strategy shown in Fig. 5(a), first, no conflicts happened during this

simulation range; second, all vehicles besides two of them were accelerating, which indicates a consistently free flow condition; third, some of vehicles had lower acceleration rates to hold constraints of safety distance and conflict avoidance, showing as bigger gaps between certain lines in the Fig.5 (a); fourth, some vehicles were travelling as a platoon while holding constraints, showing as smaller gaps between certain lines. Overall, a consistently free flow condition without any conflicts indicates the effectiveness of the formulation with ZONE strategy regarding travel efficiency.

As of ZONEV strategy shown in Fig. 5(b), first, no conflicts happened; second, vehicles were always accelerating before upstream intersection boundary while decelerating around intersection range considering reasonable vehicle dynamics; third, longer red segments might indicate a potential conflict was eliminated or a maximum speed was happened to reach; fourth, shorter green segments indicate the room for an acceleration rate is acceptable. considering all constraints; fifth, acceleration was taken at most of simulation time, indicating an efficient and safe traffic flow. Overall, this formulation outputs a stable simulation status even under high travel demand.



(a) (b)

Figure 5 Trajectory diagrams of (a) ZONE and (b) ZONEV

Furthermore, it is also indicated that a significant difference between Fig. 4(b) and Fig. 5(b) is that deceleration actions were taken more by ZONEV than by BATCHV, probably due to smaller granularity diminishing shift frequency of cell occupation status temporarily and spatially, whereas no significant variance between Figs. 4(a) and 5(a) was observed.

4.5.3 Delay analysis

The average intersection delay under different demands with 12 feet vehicle length fixed is compared between the BATCHV, ZONEV, BATCH, ZONE, DB8 and DB20. As shown in Figure 6, The BATCHV and ZONEV are insensitive to the varying traffic condition even if the traffic demand of the whole intersection reaches to 14400 veh/h, result of which, less than 2.3 s/veh, is better than all the performances listed in Table 1 in terms of the intersection delay. Meanwhile, the BATCH and ZONE perform worse than those with variable acceleration rate over each trajectory segment in terms of the intersection delay,

indicating the variable acceleration rate improve the solutions to the trajectory optimization problem under reservation-based control. The DB8 and DB20 performs better than the BATCH and ZONE under low traffic demand conditions, 300 and 600 veh/h/ln, with an average delay 3.3 s/veh. However, the performances of DB8 and DB20 deteriorate sharply from 900 veh/h/ln and reach to 140 and 160 s/veh delay respectively, indicating the rule-based control policy cannot adapt to the high traffic demand condition but perform better at low traffic demand condition. Overall, the BATCHV and ZONEV compete against others under any traffic condition, whereas the BATCH and ZONE perform worse than the formers but keep the same performances under all traffic demand condition, in addition, the DB8 and DB20 performs better under low traffic demand condition than high traffic demand condition. The BATCHV performs the best and DB8 performs the worst throughout all traffic demands in terms of average intersection delay

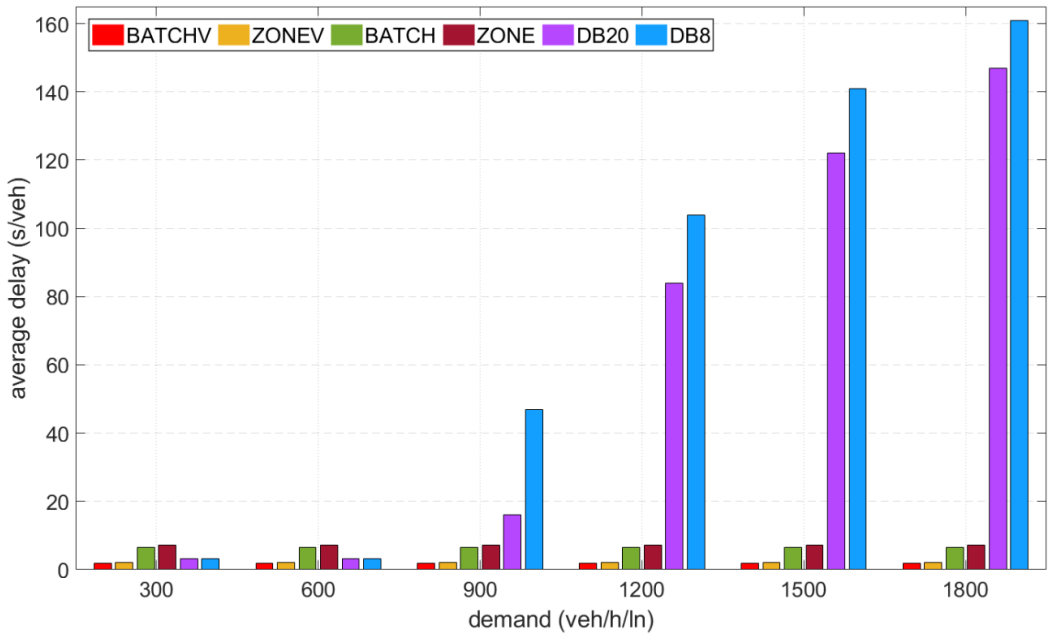


Figure 6 Average intersection delay under different traffic demand and different control strategies

4.5.4 Sensitivity analysis

After evaluating the stability and efficiency of this simulation and formulation through trajectory diagrams, average intersection delay was compared from 300 to 1800 veh/hr/ln traffic demand for BATCH, BATCHV, ZONE, ZONEV, DB8 and DB20 under different vehicle length 10 to 18 feet with an interval of 2 feet.

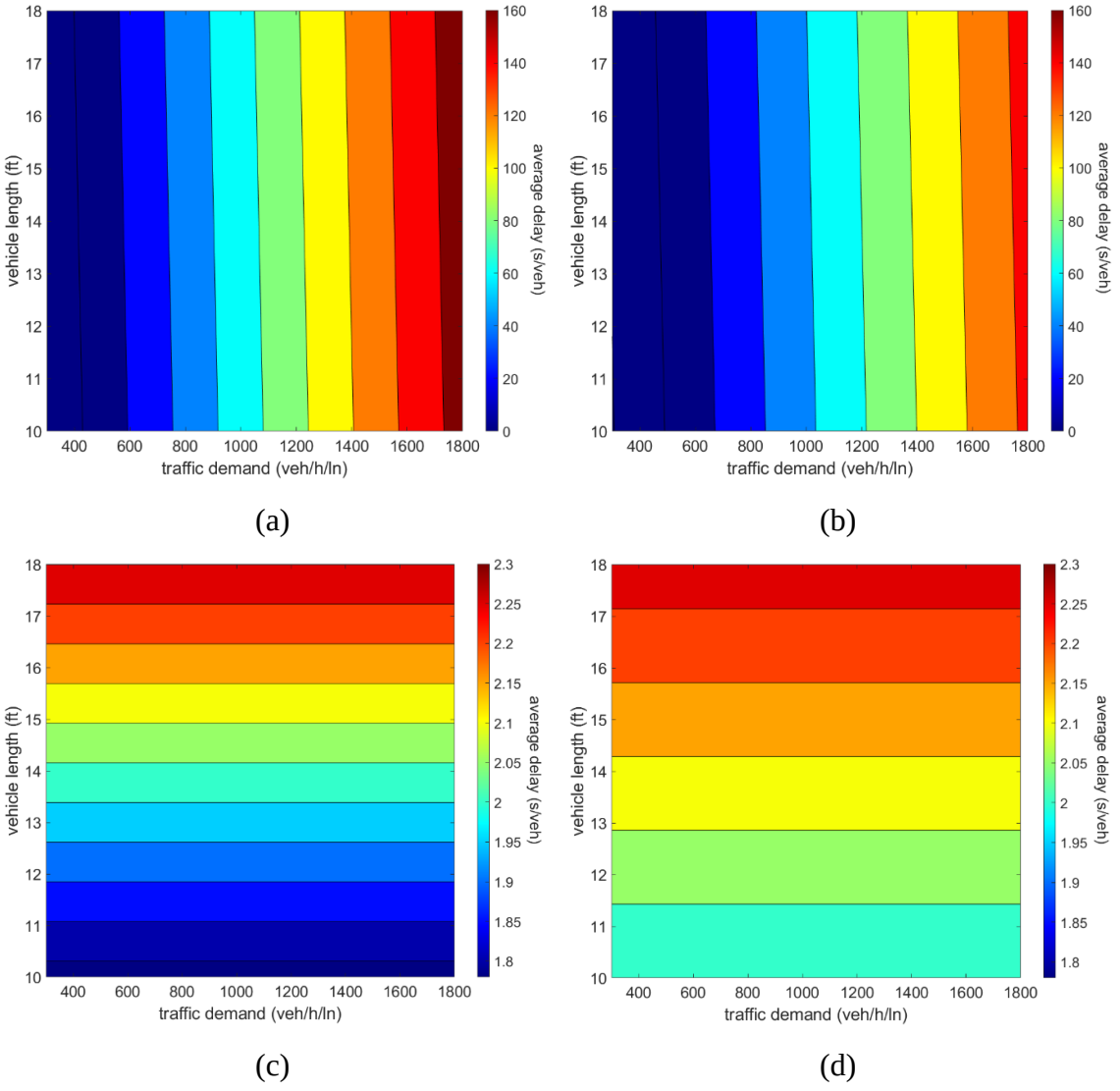
As shown in Fig. 7, the average intersection delay is annotated with different colors. First of all, the delay is worst in DB8 and the delay is best in BATCHV. In Fig.7 (a), the delay is highly related with the traffic demand under the DB8, similarly under DB 20 in Fig.7 (b). Under DB8, the delay ranges from 3.3 to 169 s/veh while the traffic demand is increasing. Under DB20, the delay is a bit better than DB8, ranging from 3.3 to 151 s/veh. Each scenario is not too much sensitive to the vehicle length. However, as mentioned in Section 4.3, the vehicle length also effects the arrival speed and headway gap, and the sensitivity between the delay and vehicle length is therefore eliminated.

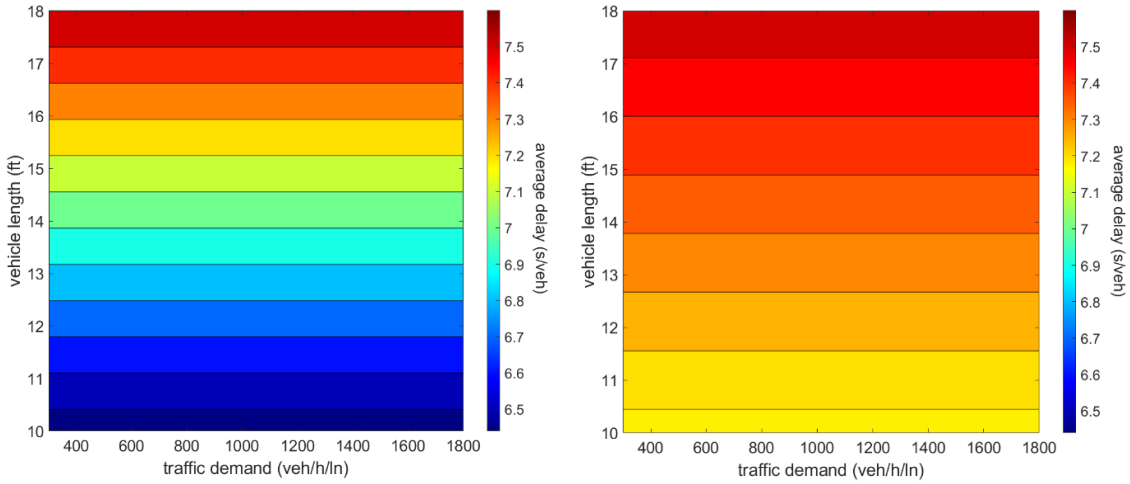
As shown in Fig.7 (c) and (d), the BATCHV competes over the ZONEV under short vehicle length scenarios. For each scenario, the relationship between the delay and traffic demand is insignificant because the capacity for the intersection under TSIT formulation is not reached, and the intersection capacity of BATCHV is larger than 14400 veh/h in total, and that of ZONEV is larger than 7200 veh/h in total. The average delay of BATCHV ranges from 1.8 to 2.3 s/veh and that of ZONEV ranges from 2 to 2.3 s/veh under different scenarios of vehicle length.

As shown in Fig.7 (e) and (f), the delay is worse than that under variable acceleration rate formulation. The delay of BATCH ranges from 6.5 to 7.5 s/veh and that of ZONE ranges from 7.2 to 7.5 s/veh under different scenarios of vehicle length. Both strategy is sensitive

to the variation of vehicle length but not to the traffic demand. However, since the delay is increasing from variable to constant acceleration rate formulation, the latter performs worse than the former even though the intersection capacity is not reached.

Overall, the average intersection delay is highly effected by some parameters, including the length of optimization range, traffic demands, intersection width, vehicle length, arrival speed and vehicle dynamics.





(e) (f)

Figure 7 Sensitivity analyses under different traffic demands and different vehicle length of (a) DB8, (b) DB20, (c) BATCHV, (d) ZONEV, (e) BATCH and (f) ZONE

CHAPTER 5. AN OPTIMAL CONTROL FRAMEWORK

5.1 Notations

Before introducing the optimal framework and modelling method, the sets, decision variables including control, state and binary variables, and parameters are defined in **Table 4**. The units of each variable and parameter are also listed.

Table 4 Notations of the optimal framework

Sets	
P	Approaches of the intersection
L	Lanes of each approach
I	Vehicles on each lane of each approach
C	Cells within the intersection
O	Number of cells ahead of vehicles
Decision variables	
$a_{t,p,l,i}$	Acceleration or deceleration rate at time t of vehicle (batch) i on lane l of approach p, m/s^2
$t_{p,l,i}^{in}$	Entry time into the communication range of vehicle (batch) i on lane l of approach p, s
$t_{p,l,i}^{out}$	Upper bound of the end time instant of the optimization cycle of vehicle (batch) i on lane l of approach p, s
$x_{t,p,l,i}$	Travel distance at time t of vehicle (batch) i on lane l of approach p, m
$v_{t,p,l,i}$	Travel speed at time t of vehicle (batch) i on lane l of approach p, km/h
$b_{t,p,l,i,c}^{1}$	Binary variable to determine if the head of the vehicle (batch) has crossed the lower bound (front boundary) of cell c at time t
$b_{t,p,l,i,c}^{2}$	Binary variable to determine if the rear of the vehicle (batch) has crossed the upper bound (rear boundary) of cell c at time t
$\varphi_{t,p,l,i,c}$	Binary variable to determine if the vehicle (batch) occupies the cell c at time t
$\varphi_{t,p,l,i,c}^{s}$	Binary variable that keeps the occupation status at cell c of vehicle (batch) i on lane l of approach p of any finished optimization cycles
$\delta_{t,p,l,i}$	Binary variable to determine if the vehicle (batch) is not cleared by the intersection at time t
Parameters	
X	Upper bound of travel distance, m
v_{max}	Upper bound of travel speed, km/h
a_{min}	Lower bound of deceleration rate (negative value), m/s^2

a_{max}	Upper bound of acceleration rate (positive value), m/s^2
n	Batch number
ℓ	Zone range, m
$intv$	Optimization interval, s
$d_{p,l,i,c}^{U,o}$	Position of the upper bound of o_{th} cell c ahead of vehicle (batch) $i$on lane l of approach p
$d_{p,l,i,c}^{L,o}$	Position of the lower bound of o_{th} cell c ahead of vehicle (batch) $i$on lane l of approach p
D	Communication range, m
g	Length of a square cell, m
m	Infinite small positive value
M	Infinite big positive value
$vehL$	Length of a vehicle, m
ϑ	Bounds of variation of acceleration or deceleration along with time, m/s^2
x_0	Initial position of vehicle (batch) i on lane l of approach p entering the communication range, m
v_0	Initial speed of vehicle (batch) i on lane l of approach p entering the communication range, km/h
bor	Distance of a vehicle's (batch's) rear to be totally cleared by the intersection, m

5.2 Problem Description

In a reservation-based intersection where all vehicles are CAVs, i.e., the penetration rate of CAVs is 100%, trajectory, arrival time and arrival speed of CAVs can be controlled by a central controller as shown in **Figure 8**.

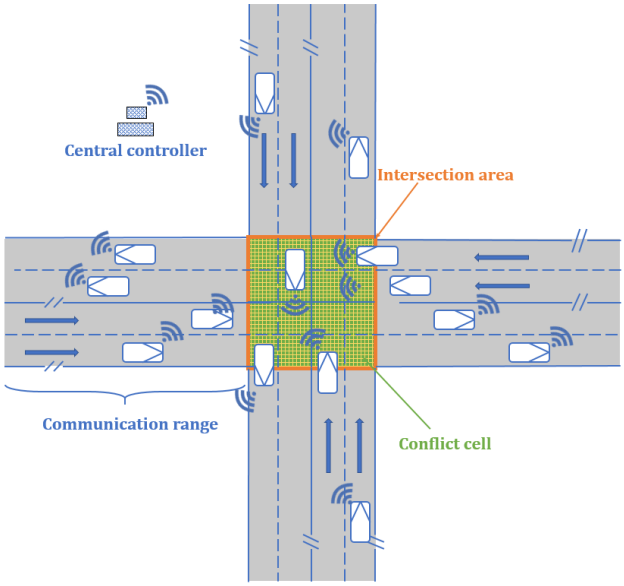
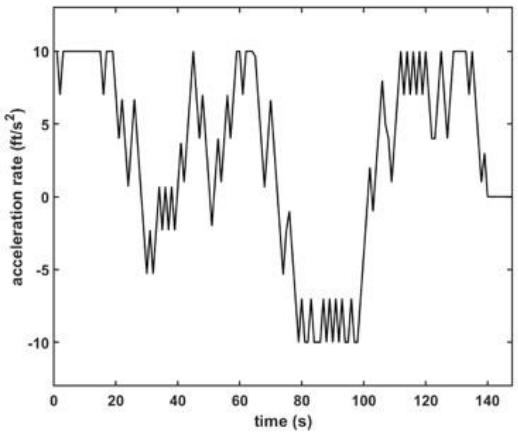
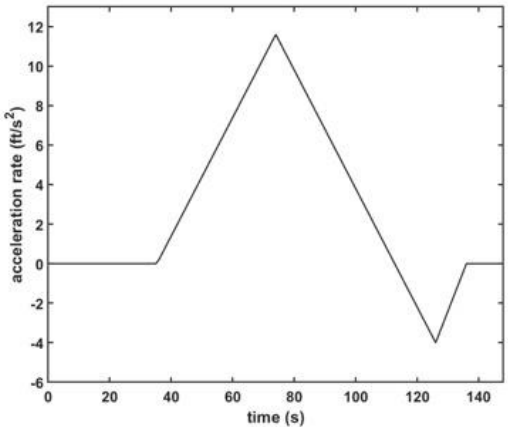


Figure 8 The reservation-based intersection control system

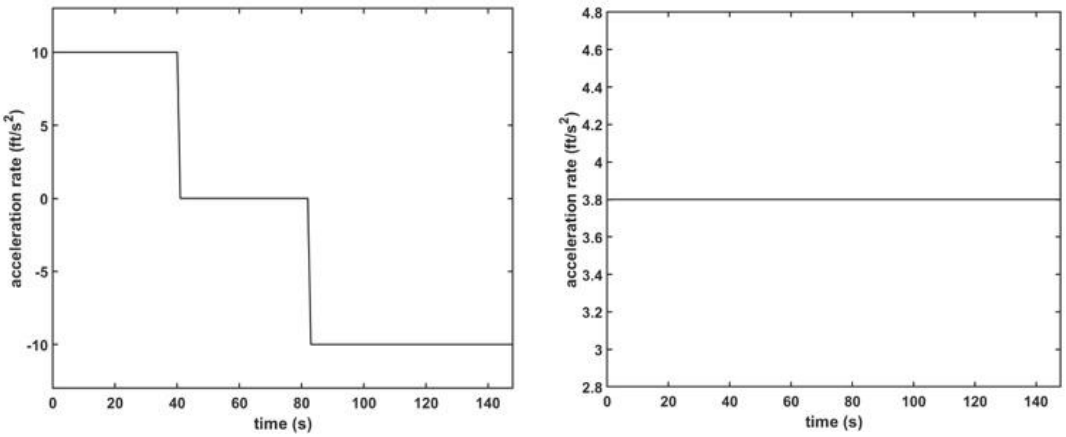
The intersection efficiency depends on the control methods of trajectory, arrival time and arrival speed of CAVs. There are mainly four categories in terms of controlling CAVs to arrive at the intersection as shown in Figure 9.



(a) (Ma and Li, 2021)



(b) (Rios-Torres and Malikopoulos, 2016)



(c) (Au and Stone, 2010) (d) (Lee et al., 2013)

Figure 9 Various control methods at a reservation-based intersection

The trajectory of CAVs can be planned based on a numerical solution of an optimization as in Figure 9 (a), or solved via a function over time based on control theory as in Figure 9 (b), or planned by an analytical solution as in Figure 9 (c) or solved via a continuous function over time as in Figure 9 (d). As opposed to Figure 9 (d), first three methods model a trajectory as a piecewise function over time. Furthermore, arrival speeds of CAVs are modelled as a variable as in Figure 9 (a) and (d) and as a constant as in Figure 9 (b) and (c). The arrival speed can be modelled dependently or independently of arrival time of each CAV. However, when delay occurs, the arrival speed may be decreased if it is modelled as a variable, such as in Figure 9 (d) or Malikopoulos et al. (2018). The decrease of the arrival speed of CAVs then leads to the decrease of the intersection efficiency, such as throughput. The relationship between the arrival time and arrival speed is not investigated yet in terms of the intersection efficiency.

Furthermore, the arrival time also depends on the planning methods of trajectory and arrival speed. If the arrival speed is increasing, the minimum of arrival time at the intersection then can be decreased (Xu et al., 2018a; Yu et al., 2019). In addition, it is

basically to optimize the passing order or arrival time for each CAV such that the intersection efficiency is optimized (Ma and Li, 2022b; Pei et al., 2019; Zhang et al., 2022). Therefore, how to model the arrival time and speed simultaneously such that the intersection efficiency reaches a maximum is also not analyzed yet.

Overall, in order to define the optimal control framework, the relationship among trajectory, arrival speed and arrival time is investigated followingly in terms of optimizing the intersection efficiency.

Before introducing the theoretical analysis and numerical modeling, some assumptions are given as follows in the context of the optimal control framework for the reservation-based intersection.

1. Vehicle-to-everything communication latency is not considered;
2. Vehicle-to-everything communication is reliable and protected from cyber-physical attacks;
3. Road surface condition is not considered;
4. It is assumed the reservation-based intersection is undersaturated during theoretical proof; and
5. Vehicle-to-pedestrian communication is not considered.

5.3 Theoretical Definition on the Optimal Control Framework

5.3.1 Analysis on the relationship among trajectory, arrival time and arrival speed

A trajectory modelling problem is generalized as follows with constraints of arrival speed and arrival time of each CAV.

Lemma 5.1 (Feasibility of a trajectory modeling problem). *Given a trajectory modeling problem* $\boldsymbol{J}(\cdot)$ *with the following constraints,* $\boldsymbol{J}(\cdot)$ *is feasible.*

$$\begin{cases} v\left(t_{p,l,i}^{in}\right) = v_0 \\ \int_{t_{p,l,i}^{in}}^{t_f} v(t)dt = D \\ v_0 + \int_{t_{p,l,i}^{in}}^{t_f} a(t)dt = v\left(t_f\right) \le v_f \\ 0 \le v(t) \le v_{max} \\ a_{min} \le a(t) \le a_{max} \end{cases} \tag{20}$$

Proof. It is feasible to find a solution of $\boldsymbol{J}(\cdot)$ in terms of the control variable $a(t)$ as long as the above constraints have been met, where $a(t)$ and $v(t)$ are acceleration rate and speed of a CAV at time instant t, and t_f and v_f are assigned arrival time and maximum arrival speed of a CAV at the intersection. ■

Proposition 5.1. Assume $\boldsymbol{J}(\cdot)$ is feasible when $a(t) = f_1(t)$ is a continuous function, $v\left(t_f\right) = v_0 + \int_{t_{p,l,i}^{in}}^{t_f} a(t)dt$.

Proof. The proposition proves itself. ■

Remark 5.1. Based on **Proposition 5.1**, since $a(t)$, $t_{p,l,i}^{in}$ and v_0 are given, $v\left(t_f\right)$ depends on t_f when $a(t) = f_1(t)$ is a continuous function.

Example 5.1. Based on **Proposition 5.1**, since $D = v_0\Delta t + \frac{a^c\Delta t^2}{2}$, where $a(t) = f_1(t) = a^c$, $t \in [t_{p,l,i}^{in}, t_f]$ and $\Delta t = t_f - t_{p,l,i}^{in}$, then $a^c = \frac{2(D - v_0\Delta t)}{\Delta t^2}$ and $v\left(t_f\right) = v_0 + \frac{2(D - v_0\Delta t)}{\Delta t^2}t_f$. It can be seen that $v\left(t_f\right)$ depends on t_f when $a(t) = f_1(t)$ is a continuous function.

Proposition 5.2. Assume $\boldsymbol{J}(\cdot)$ is feasible when $a(t) = f_2(t)$ is a piecewise function, $t_f \ge t_{f,min} = f_3(v_f)$.

Proof. When $a(t) = f_2(t)$ is a piecewise function,

$$t_{f,min} = f_3(v_f)$$

$$= \begin{cases} \dfrac{v_i - v_o}{a_{max}} - \dfrac{v_i - v_f}{a_{min}} & if\ \dfrac{{v_{max}}^2 - {v_0}^2}{2a_{max}} - \dfrac{{v_{max}}^2 - {v_f}^2}{2a_{min}} > D \\ \dfrac{v_{max} - v_o}{a_{max}} - \dfrac{v_{max} - v_f}{a_{min}} + \dfrac{D - \frac{{v_{max}}^2 - {v_0}^2}{2a_{max}} + \frac{{v_{max}}^2 - {v_f}^2}{2a_{min}}}{v_{max}} & otherwise \end{cases} \tag{21}$$

where v_i is an intermediate value that can be referred to Yu et al. (2018). ■

Remark 5.2. Based on **Proposition 5.2**, it can be seen that t_f depends on v_f, and $v(t_f)$ is independent of t_f when $a(t) = f_2(t)$ is a piecewise function. In such case, $v(t_f) = v_f$.

Example 5.2. Assume at a reservation-based intersection where $\boldsymbol{J}(\cdot)$ has the parameters as in Table 5, $t_{f,min}$ can be calculated as in **Figure 10**, where t_f' denotes the minimum achievable arrival time of **Example 1** if $\boldsymbol{J}(\cdot)$ is feasible. It can be seen from **Figure 10** that regardless of the given arrival time of a CAV, the arrival speed keeps unchanged when $a(t) = f_2(t)$ is a piecewise function. However, $v(t_f)$ decreases as t_f increases when $a(t) = f_1(t)$ is a continuous function, in addition, if $t_f < t_f'$, $\boldsymbol{J}(\cdot)$ is infeasible under **Example 5.1**.

Table 5 Initial settings at a reservation-based intersection

$\boldsymbol{D(m)}$	$\boldsymbol{v_0(km/h)}$	$\boldsymbol{t_{p,l,i}^{in}(s)}$	$\boldsymbol{a_{max}(m/s^2)}$	$\boldsymbol{v_{max}(km/h)}$	$\boldsymbol{v_f(km/h)}$
182.88	21.85	0	3.05	65.84	65.84

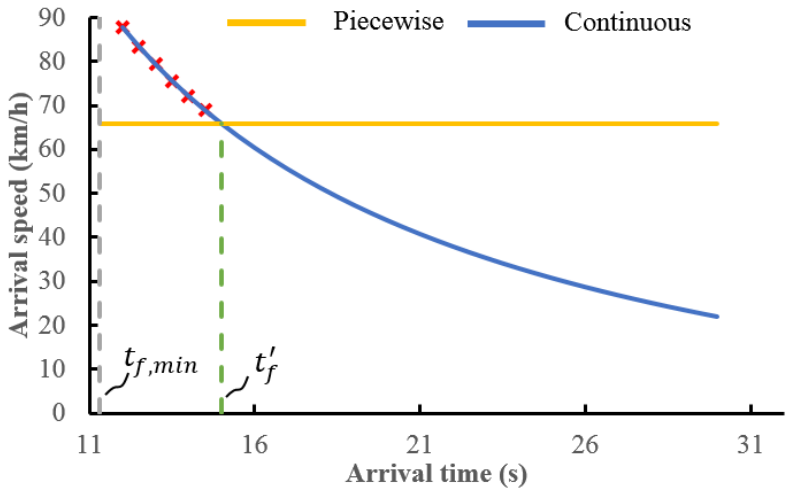


Figure 10 The relationship between arrival speed and arrival time

5.3.2 Analysis on the reservation-based intersection efficiency

As analyzed above, arrival speed can be either dependent or independent of arrival time based on continuous or piecewise trajectory modelling methods. Furthermore, in terms of the intersection efficiency, such as capacity or delay, scheduling of arrival times and controlling of arrival speeds of CAVs are essential factors. The intersection efficiency is respectively modelled by considering the scheduling and quantified based on queue theory via considering arrival speeds as follows.

Lemma 5.2. The optimal solution to the intersection efficiency can be approximated via greedy algorithm.

Proof. To achieve the optimal intersection efficiency, such as capacity or delay, it is essentially to optimize passing orders of CAVs approaching to the intersection (Ma and Li, 2022b; Zhang et al., 2022). In terms of optimizing the passing order, greedy algorithm can be used to approximate the optimal solution (Besa Vial et al., 2016). ■

Remark 5.3. First-come-first-serve (FCFS) strategy assigns passing orders of CAVs based on their arrival times at the beginning of a communication range. Under FCFS, the earlier

time a CAV arrives at, the earlier time the CAV crosses through the intersection. Therefore, FCFS is a greedy algorithm of approximating the optimal intersection efficiency.

Proposition 5.3. The average service time $E(S)$ of a CAV at a reservation-based intersection under FCFS strategy is $y_d - (y_d - y_s)\sum_{j\in J}(P_j)^2$.

Proof. The detailed proof process can be referred to Yu et al. (2018), where y_d denotes the safety crossing gap between two vehicles from conflicting approaches, y_s measures the saturation headway between two vehicles from a same approach, J denotes a set of approaches, $P_j = \frac{\lambda_j}{\sum_{j'\in J}\lambda_{j'}}$ denotes the probability of a CAV crossing the reservation-based intersection from approach $j \in J$, λ_j denotes an average arrival rate of CAVs from approach j, and $\lambda_j > 0(veh/s)$ conforms to Poisson distribution. ■

Remark 5.4. The $E(S)$ in **Proposition 5.3** is derived based on an M/G/1 queue system (Ross, 2014), where the intersection capacity and delay can be further derived based on the average service time.

The arrival speed is assumed as a constant and independent of the average service time in **Proposition 5.3**. However, y_d depends on the arrival speed of each CAV. To analyze the effect of the arrival speed of CAVs on the intersection efficiency, the arrival speed is introduced as a variable and the average service time is redefined as follows.

Corollary 5.1. The average service time (s) of a CAV is

$$E(S) = \frac{L + l + \varepsilon}{v_f} - (\frac{L + l + \varepsilon}{v_f} - y_s)\sum\nolimits_{j\in J}(P_j)^2 \tag{22}$$

when v_f is introduced as a variable under the M/G/1 queue system.

Proof. Based on **Proposition 5.3**, y_d is a variable when v_f is introduced as a variable in $\boldsymbol{J}(\cdot)$, and $y_d = \frac{L+l+\varepsilon}{v_f}$, where L denote an intersection length, l denotes an average CAV

length, and ε denotes a safety buffer length. The larger the v_f of a CAV is, the shorter the y_d is. ■

Furthermore, the capacity C of the reservation-based intersection can be followingly based on **Corollary 5.1**.

Corollary 5.2. The intersection capacity (veh/h) is

$$C = \frac{3600 v_f}{L + l + \varepsilon - (L + l + \varepsilon - y_s v_f) \sum_{j \in J} (P_j)^2} \tag{23}$$

when v_f is introduced as a variable under the M/G/1 queue system.

Proof.

$$\begin{aligned} C &= \frac{3600}{y_d - (y_d - y_s) \sum_{j \in J} (P_j)^2} \\ &= \frac{3600}{\frac{L + l + \varepsilon}{v_f} - (\frac{L + l + \varepsilon}{v_f} - y_s) \sum_{j \in J} (P_j)^2} \\ &= \frac{3600 v_f}{L + l + \varepsilon - (L + l + \varepsilon - y_s v_f) \sum_{j \in J} (P_j)^2} \end{aligned} \tag{24}$$

■

Proposition 5.4. The intersection delay D (s/veh) is

$$\frac{\lambda E[S^2]}{2(1 - \lambda E[S])} \tag{25}$$

under the M/G/1 queue system.

Proof. The average intersection delay is derived based on Pollaczek–Khinchine formula (Ross, 2014), where $\lambda = \sum_{j \in J} \lambda_j$, $E[S^2]$ denotes the mean of the square of S, and S denotes a service time of a CAV spending to cross the intersection. ■

Remark 5.5. $E[S^2]$ cannot be calculated directly if the service time of each CAV is different; however, based on **Proposition 5.2**, the arrival speed of every CAV can be given

independently of t_f, i.e., $v(t_f)$ of each CAV can be equal. Therefore, the average intersection delay can be further derived based on the arrival speed as follows.

Corollary 5.3. The intersection delay D (s/veh) is

$$D = \frac{\lambda(L+l)^2}{2(v_f^2 - \lambda v_f[L+l+\varepsilon-(L+l+\varepsilon-y_s v_f)\sum_{j\in J}(P_j)^2])} \tag{26}$$

when v_f is introduced as a variable under the M/G/1 queue system.

Proof. Since $v(t_f) = v_f$ for each CAV in $\boldsymbol{J}(\cdot)$ based on **Proposition 5.2**, $E[S^2] = S^2$. Therefore, we have

$$\begin{aligned} D &= \frac{\lambda E[S^2]}{2(1-\lambda E[S])} = \frac{\lambda S^2}{2(1-\lambda E[S])} \\ &= \frac{\lambda\left(\frac{L+l}{v_f}\right)^2}{2\left(1-\lambda\left[\frac{L+l+\varepsilon}{v_f}-(\frac{L+l+\varepsilon}{v_f}-y_s)\sum_{j\in J}(P_j)^2\right]\right)} \\ &= \frac{\lambda(L+l)^2}{2(v_f^2 - \lambda v_f[L+l+\varepsilon-(L+l+\varepsilon-y_s v_f)\sum_{j\in J}(P_j)^2])} \end{aligned} \tag{27}$$

where $S = \frac{L+l}{v_f}$. ■

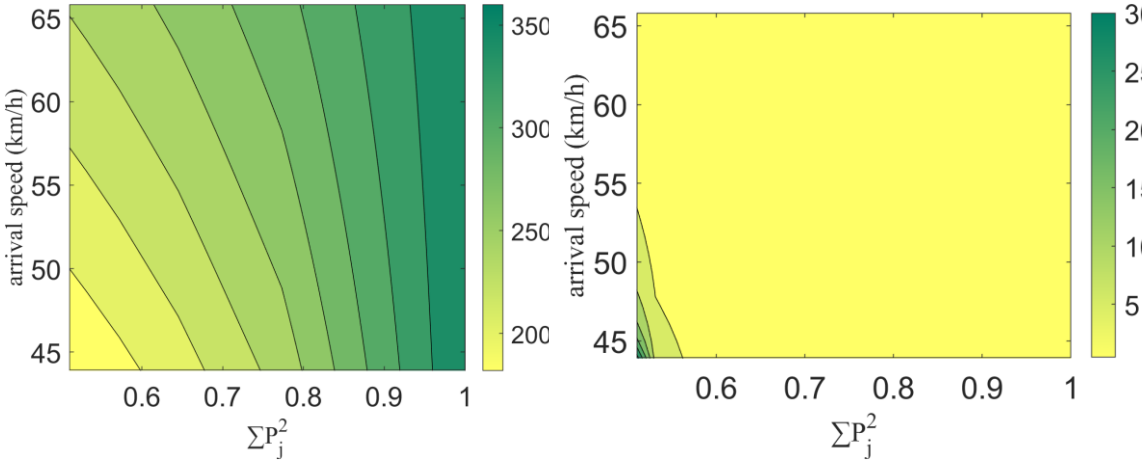
v_f is given as the maximum allowable crossing speed at the intersection for each CAV in $\boldsymbol{J}(\cdot)$. It is validated that the larger the v_f is, the more efficient the intersection is based on Little's Law (Au and Stone, 2010). However, when the arrival rates from different approaches are unbalanced, crossing gaps from major roads are created for accommodating vehicles from minor roads (Au et al., 2011; Tachet et al., 2016). Under such unbalanced traffic, whether arrival speeds of vehicles from minor roads should be larger or lower than vehicles from major roads is not investigated thoroughly in terms of optimizing the

intersection efficiency. Based on **Corollary 5.2 & 5.3**, an example is given followingly to analyze the effect of arrival speeds from different approaches on the intersection efficiency.

Example 5.3. Without loss of generosity, a reservation-based intersection with two conflicting approaches is given, where $L + l = 27.43\,m$, $\varepsilon = 9.14\,m$, and $y_s = 1\,s$, in addition, $v_f = 65.84\,\mathrm{km/h}$. The arrival rates of each approach and corresponding $\sum_{j\in J}(P_j)^2$ are given in Table 6. The intersection capacity and average delay are given in **Figure 11**. Note that under each scenario listed in Table 6. $\lambda E[S] < 1$ so that the intersection is under-saturated.

Table 6 Arrival rates of a unbalanced intersection

$\boldsymbol{\lambda_{j_1}}$ (veh/h)	1000	1000	1000	1000	1000	1000
$\boldsymbol{\lambda_{j_2}}$ (veh/h)	0	150	300	450	600	750
$\boldsymbol{\sum_{j\in J}(P_j)^2}$	1	0.77	0.64	0.57	0.53	0.51



(a) (b)

Figure 11 Capacity and delay analysis under a unbalanced intersection

Remark 5.6. It can be seen from **Figure 11** that under a unbalanced traffic scenario, e.g., $\sum_{j\in J}(P_j)^2 = 0.77$, the larger the arrival speed of both approaches is, the more the intersection capacity is, in addition, under such scenario, the average delay increases from 0.59 s/veh to 1.51 s/veh while the arrival speed decreases from 65.82 km/h to 43.88 km/h,

i.e., the intersection efficiency has a positive relationship with the arrival speed from both approaches when $\lambda E[S] < 1$. The same pattern is also found under a balanced traffic scenario when $\sum_{j \in J}(P_j)^2 = 0.51$. It can be concluded that the larger the arrival speed is for both approaches, the more efficient the intersection is under a balanced or unbalanced intersection.

5.3.3 Discussion on the optimal framework

To maximize the reservation-based intersection efficiency, it is analytically proven that the larger the arrival speed is, the more efficient the intersection is regardless of that arrival rates from conflicting approaches are balanced or unbalanced based on **Corollary 5.2 & 5.3**. To achieve a maximum allowable arrival speed, trajectory of each CAV should be controlled in a piecewise manner based on **Proposition 5.2**, where the arrival speed is independent of the assigned arrival time. Overall, a theoretical definition on the optimal control framework is given as follows.

Proposition 5.5. The optimal control framework for a reservation-based intersection in terms of intersection efficiency can be defined as:

1. $a(t) = f(t)$ is a piecewise function in $\boldsymbol{J}(\cdot)$.
2. $v(t_f) = v_f$ for each CAV in $\boldsymbol{J}(\cdot)$.
3. $v(t_f)$ is modelled independently of t_f in $\boldsymbol{J}(\cdot)$.

Proof. **Proposition 5.1 & 5.2** and **Corollary 5.1 & 5.2 & 5.3** prove this as above. ■

5.4 Numerical Modelling on the Optimal Control Framework

Based on the theoretical definition on the optimal control framework, the framework is modelled via an MILP as follows to further validate the optimality in terms of the intersection efficiency. Specifically, the optimality of the proposed framework is validated

by comparing different modelling methods of trajectory, arrival time and speed. The modelling methods consist of the continuous and piecewise trajectory control, based on which arrival speed can be either modelled dependently or independently of arrival time. Furthermore, the arrival speed is modelled as a variable rather than a constant such that whether $v(t_f)$ of each CAV should equal v_f can be tested in terms of achieving the optimal intersection efficiency under various traffic scenarios.

The numerical modelling is given as follows starting with basic environment constraints of the optimal control framework. Control variables include the acceleration/deceleration rate over time and the state variables include the travel distance and speed of a CAV over time. Besides, the travel distance is measured from the start of the communication range, denoted as 0, to the upper bound where a vehicle can travel, *X*, to the downstream of the intersection. The lower and upper bound of the control variable respectively defines the admissible deceleration rate a_{min} and acceleration rate a_{max}. Note here that $t_{p,l,i}^{in}$ and $t_{p,l,i}^{out}$ respectively indicates the entry time that a CAV enters the start of the communication range and the time instant that the vehicle leaves off of the downstream of the intersection for the vehicle i from lane l of approach p. Every CAV would have different $t_{p,l,i}^{in}$ and $t_{p,l,i}^{out}$.

$$\begin{cases} 0 \le x_{t,p,l,i} \le X \\ 0 \le v_{t,p,l,i} \le v_{max} \\ a_{min} \le a_{t,p,l,i} \le a_{max} \\ \forall t \in [t_{p,l,i}^{in}, t_{p,l,i}^{out}], p \in P, l \in L, i \in I \end{cases} \tag{28}$$

Within the numerical modelling, the simulation time is discretized over simulation time t and the total number of segments equals $(t_{p,l,i}^{out} - t_{p,l,i}^{in})/t$. The control variable, acceleration rate, is optimized over every segment such that $a(t) = f(t)$ is a piecewise function in $\boldsymbol{J}(\cdot)$. In addition, the length of each segment is varied since the state and the

control variables are varied either. Specifically, the variation between the control variable between time t and $t-1$ of each trajectory segment is constrained by a parameter ϑ in order to avoid sharp changes in the acceleration rate.

$$\begin{cases} a_{t,p,l,i} - a_{t-1,p,l,i} \leq \vartheta \\ a_{t,p,l,i} - a_{t-1,p,l,i} \geq -\vartheta \\ \forall t \in \left[t_{p,l,i}^{in} + 1, t_{p,l,i}^{out}\right] \\ p \in P, l \in L, i \in I \end{cases} \tag{29}$$

For validation purpose, $a(t) = f(t)$ as a continuous function in $\boldsymbol{J}(\cdot)$ is also numerically modelled. The continuous trajectory modelling method is formulated in the following equation by making $a_{t,p,l,i}$ equal its first acceleration rate until the CAV is cleared by the intersection. Note that $a_{t_{p,l,i}^{in},p,l,i}$ would vary among different CAVs.

$$\begin{cases} a_{t,p,l,i} = a_{t_{p,l,i}^{in},p,l,i} \\ \forall t \in \left[t_{p,l,i}^{in}, t_{p,l,i}^{out}\right], p \in P, l \in L, i \in I \end{cases} \tag{30}$$

To accommodate both continuous and piecewise trajectory modelling in the MILP, vehicle dynamic equations are linearized from $\int_{t_{p,l,i}^{in}}^{t_{p,l,i}^{out}} v(t)dt = X$ and updated over time recursively. The travel distance at time instant t is updated from the position at instant $t-1$ based on vehicle dynamic equations as follows.

$$\begin{cases} x_{t,p,l,i} = x_{t-1,p,l,i} + intv \times v_{t-1,p,l,i} + 0.5 \times intv^2 \times a_{t-1,p,l,i} \\ \forall t \in \left[t_{p,l,i}^{in} + 1, t_{p,l,i}^{out}\right], p \in P, l \in L, i \in I \end{cases} \tag{31}$$

Similarly, the speed is also linearly formulated with the acceleration rate via the recursive procedure. In addition, the planning method is recursive during $[t_{p,l,i}^{in} + 1, t_{p,l,i}^{out}]$ along each sub-segment, and the values of travel distance and speed at time $t_{p,l,i}^{in}$ are initialized with given values as follows.

$$\begin{cases} v_{t,p,l,i} = v_{t-1,p,l,i} + intv \times a_{t-1,p,l,i} \\ \forall t \in \left[t_{p,l,i}^{in} + 1, t_{p,l,i}^{out}\right], p \in P, l \in L, i \in I \end{cases} \quad (32)$$

$$x_{t,p,l,i} = x_0 \quad \forall t = t_{p,l,i}^{in}, p \in P, l \in L, i \in I \quad (33)$$

$$v_{t,p,l,i} = v_0 \quad \forall t = t_{p,l,i}^{in}, p \in P, l \in L, i \in I \quad (3)$$

As shown in **Figure 8**, only through movements are considered from each lane of each approach, therefore the conflict-free crossing and coordination are reduced from two-dimension to one-dimension formulation. It is sufficient to only consider the through movements for validating the optimal control framework proposed in this paper. As indicated in the following equations, only one-dimension position of the head of each vehicle is needed, thus, the time complexity is simplified. Specifically, the front and rear boundary position of each cell within the intersection area are pre-measured and stored without consuming online computational resources, i.e., $d_{p,l,i,c}^{U,o}$ and $d_{p,l,i,c}^{L,o}$ are defined symmetrically same for each vehicle from any lanes or approaches. The variable $c \in C$ is used to determine which specific cell within the intersection is ahead of a CAV and whether there are conflicts on certain cell c occupying by vehicles from same or conflicting approaches, then the central controller identifies if conflicts occur at certain cells based on the information p, l, i and c.

$$\begin{cases} d_{p,l,i,c}^{U,o} = D + g \times o \\ \forall p \in P, l \in L, i \in I, c \in C, o \in O \end{cases} \quad (35)$$

$$\begin{cases} d_{p,l,i,c}^{L,o} = D + g \times (o - 1) \\ \forall p \in P, l \in L, i \in I, c \in C, o \in O \end{cases} \quad (36)$$

Since the $d_{p,l,i,c}^{U,o}$ and $d_{p,l,i,c}^{L,o}$ are defined for each vehicle from any lanes or approaches, once any parts of a vehicle occupy a cell at time t, no matter completely or only partially, $b_{t,p,l,i,c}^{1}$ and $b_{t,p,l,i,c}^{2}$ both equal 1. The shape of the vehicle is measured only by $vehL$ in one-

dimension coordinate system. The following equation indicates that a vehicle could occupy one or more cells at the same time.

$$\begin{cases} x_{t,p,l,i} \geq d_{p,l,i}^{L,o} + m - M \times (1 - b_{t,p,l,i,c}^{1}) \\ x_{t,p,l,i} \leq d_{p,l,i}^{L,o} + M \times b_{t,p,l,i,c}^{1} \\ d_{p,l,i}^{U,o} \geq x_{t,p,l,i} - vehL + m - M \times (1 - b_{t,p,l,i,c}^{2}) \\ d_{p,l,i}^{U,o} \leq x_{t,p,l,i} - vehL + M \times b_{t,p,l,i,c}^{2} \\ \forall t \in \left[t_{p,l,i}^{in}, t_{p,l,i}^{out}\right], p \in P, l \in L, i \in I, c \in C, o \in O \end{cases} \tag{37}$$

If $b_{t,p,l,i,c}^{1}$ and $b_{t,p,l,i,c}^{2}$ both equal 1 at the same time t $\varphi_{t,p,l,i,c}$ equals 1; otherwise $\varphi_{t,p,l,i,c}$ equals 0. The if-else condition is linearized as follows.

$$\begin{cases} b_{t,p,l,i,c}^{1} + b_{t,p,l,i,c}^{2} - 2 \times \varphi_{t,p,l,i,c} \leq 1 \\ b_{t,p,l,i,c}^{1} + b_{t,p,l,i,c}^{2} - 2 \times \varphi_{t,p,l,i,c} \geq 0 \\ \forall t \in \left[t_{p,l,i}^{in}, t_{p,l,i}^{out}\right], p \in P, l \in L, i \in I, c \in C \end{cases} \tag{38}$$

In addition to identifying the value of $\varphi_{t,p,l,i,c}$ at time t for vehicle (p, l, i) at cell c, any pre-stored values of $\varphi_{t,p,l,i,c}$ denoted as $\varphi_{t,p,l,i,c}^{s}$ at time t for other vehicles (p', l', i') at the same cell c are used to identify if conflicts occur between vehicles in the current optimization cycle and vehicles in last optimization cycles. More specifically, $\varphi_{t,p,l,i,c}$ ensures no conflicts occur in the current optimization cycle and $\varphi_{t,p,l,i,c} + \varphi_{t,p,l,i,c}^{s}$ ensures no conflicts occur over all optimization cycles. It is noted that $t \in \left[t_{p,l,i}^{in}, t_{p,l,i}^{out}\right]$ rather than $t \in \left[t_{p^0,l^0,i^0}^{in}, t_{p^f,l^f,i^f}^{out}\right]$ simplifies the time complexity of the computation, where (p^0, l^0, i^0) denotes the first vehicle entering in all cycles and (p^f, l^f, i^f) denotes the last vehicle leaving in all cycles.

$$\begin{cases} \sum_{p \in P} \sum_{l \in L} \sum_{i \in I} (\varphi_{t,p,l,i,c} + \varphi_{t,p,l,i,c}^{s}) \leq 1 \\ \forall t \in \left[t_{p,l,i}^{in}, t_{p,l,i}^{out}\right], p \in P, l \in L, i \in I, c \in C \end{cases} \tag{39}$$

As indicated earlier, $v(t_f)$ is modelled independently of t_f in $\boldsymbol{J}(\cdot)$ and is a variable, so is the t_f. Different from the t_f, the $t_{p,l,i}^{out}$ is first estimated with a value as long as the vehicle can reach the downstream of the intersection, i.e., cross the intersection. In case the $t_{p,l,i}^{out}$ is not enough for the vehicle to cross the intersection, $t_{p,l,i}^{out}$ will be added by a buffer to re-initialize the recursive update and find the solution of the optimization problem. The buffer is introduced in **Algorithm 5.1**.

To further measure arrival and travel times of a CAV in an optimization cycle, an alternative method of measuring them is to add another decision variable $\delta_{t,p,l,i}$ that measures the current position of a vehicle. The variable is an integer variable and accumulates as long as the vehicle has not crossed the intersection.

$$\begin{cases} bor \geq x_{t,p,l,i} - vehL + m - M \times (1 - \delta_{t,p,l,i}) \\ bor \leq x_{t,p,l,i} - vehL + M \times \delta_{t,p,l,i} \\ \forall t \in \left[t_{p,l,i}^{in}, t_{p,l,i}^{out}\right], p \in P, l \in L, i \in I \end{cases} \tag{40}$$

While accumulating the integer variable $\delta_{t,p,l,i}$, the total travel time is finalized once a vehicle crosses the intersection. Note here that maybe $t_{p,l,i}^{out} - t_{p,l,i}^{in} \geq \sum_{t \in \left[t_{p,l,i}^{in}, t_{p,l,i}^{out}\right]} \delta_{t,p,l,i}$. The smaller $t_{p,l,i}^{out}$ is, the less the computational demand requires. The objective of the optimization problem is to minimize the maximum travel time of any vehicles in the current optimization cycle.

$$minmax \sum_{t \in \left[t_{p,l,i}^{in}, t_{p,l,i}^{out}\right]} \delta_{t,p,l,i} \quad \forall p \in P, l \in L, i \in I \tag{41}$$

To validate the proposed optimal control framework, both piecewise and continuous trajectory modellings are formulated separately. The problem $\boldsymbol{J}_1$ of piecewise trajectory

modelling and problem $\boldsymbol{J}_2$ of continuous trajectory modelling are respectively shown as follows.

$$\begin{aligned} &\boldsymbol{J}_1\colon\ minmax \sum_{t\in\left[t_{p,l,i}^{in},t_{p,l,i}^{out}\right]} \delta_{t,p,l,i} \\ &s.t.\colon\ (28),(29),(31)-(41) \\ &\forall p\in P, l\in L, i\in I \end{aligned} \tag{42}$$

$$\begin{aligned} &\boldsymbol{J}_2\colon\ minmax \sum_{t\in\left[t_{p,l,i}^{in},t_{p,l,i}^{out}\right]} \delta_{t,p,l,i} \\ &s.t.\colon\ (28),(30),(31)-(41) \\ &\forall p\in P, l\in L, i\in I \end{aligned} \tag{43}$$

Furthermore, the recursive trajectory optimization algorithm for numerically modelling the optimal control framework is illustrated in **Algorithm 5.1**.

Algorithm 5.1. Recursive trajectory optimization algorithm

// Set up a total simulation time T
Begin
New $\varphi_{t,p,l,i,c}^{s}$
  While $t_{p,l,i}^{in} \le T$ Do
New Model with objective J_1, J_2
New Data with $P, L, C, O, d_{p,l,i,c}^{U,o}$ and $d_{p,l,i,c}^{L,o}$
Let $\varphi_{t,p,l,i,c} = \varphi_{t,p,l,i,c}^{s}$
Generate Model
    If Model is solved, Then
Get solution $\sum_{t\in\left[t_{p,l,i}^{in},t_{p,l,i}^{out}\right]} \delta_{t,p,l,i}$
If $\sum_{t\in\left[t_{p,l,i}^{in},t_{p,l,i}^{out}\right]} \delta_{t,p,l,i} < t_{p,l,i}^{out} - t_{p,l,i}^{in}$ Then
Add $\varphi_{t,p,l,i,c}$ to $\varphi_{t,p,l,i,c}^{s}$
New stochastic $t_{p,l,i}^{in}$
Let $t_{p,l,i}^{out} = t_{p,l,i}^{in} + \sum_{t\in\left[t_{p,l,i}^{in},t_{p,l,i}^{out}\right]} \delta_{t,p,l,i} + 10$
      Else If Then
        Let $t_{p,l,i}^{out} = t_{p,l,i}^{in} + \sum_{t\in\left[t_{p,l,i}^{in},t_{p,l,i}^{out}\right]} \delta_{t,p,l,i} + 10$
        Let $t_{p,l,i}^{in} = t_{p,l,i}^{in}$
      End If
    Else If Then
      Get Model conflicts

20.	Break
21.	End If
22.	End While
23.	End

Note here that in Step 10 in **Algorithm 5.1**, when $\sum_{t\in\left[t_{p,l,i}^{in},t_{p,l,i}^{out}\right]}\delta_{t,p,l,i} < t_{p,l,i}^{out} - t_{p,l,i}^{in}$, vehicle (p,l,i) is totally cleared by the intersection boundary, and the optimal solution is $\sum_{t\in\left[t_{p,l,i}^{in},t_{p,l,i}^{out}\right]}\delta_{t,p,l,i}$ for vehicle (p,l,i); otherwise $t_{p,l,i}^{out}$ will be updated in Step 15 and the trajectory will be re-optimized in the current cycle via Step 16, where **10** in Step 13 and 15 means 1 second because 1 second is discretized to 10 simulation steps. Overall, the trajectory can be optimized recursively to get the optimal solutions with respect to any vehicles in the current optimization cycle. Therefore, the optimal control framework can be validated in terms of the intersection efficiency.

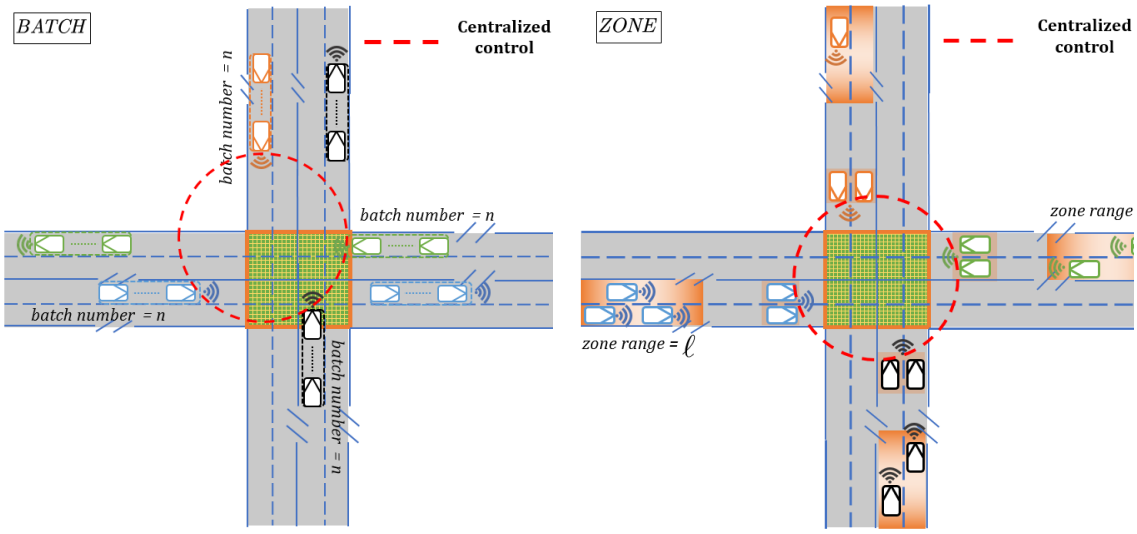
5.5 Results and Discussions

5.5.1 Numerical simulation

The simulation platform for the numerical modelling operates on a desktop computer with an Intel i3-3220 CPU with 3.3 GHz and 8 GB memory. The platform is coded in the off-the-shelf optimization package CPLEX Studio IDE 12.10.0.

Two test scenarios BATCH and ZONE are respectively created in the simulation environment. The BATCH strategy is performed for a scenario where a group of vehicles with given numbers coming from each lane of an approach is optimized as a platoon of vehicle. The ZONE strategy is otherwise performed for a scenario where a group of vehicles inside a dynamic zone that is created for each approach is optimized together as a platoon of vehicle. Note here that BATCH processes the platoon per lane, whereas ZONE processes the platoon per approach.

Moreover, the batch number n under BATCH can group six vehicles at most and two vehicles at least per lane, and the zone range ℓ under ZONE can extend from 11 to 33 meters at each approach. Length of a vehicle is 3.66 meters and safety headway distance is 1.8 meters. The two strategies are illustrated in **Figure 12**. Again, the BATCH strategy is applied per lane whereas the ZONE is applied per approach, i.e., at each optimization cycle, the batch strategy can process at most eight platoons from all approaches while the zone can process at most four groups of vehicles from all approaches.



(a) (b)

Figure 12 Various test scenarios for optimal control framework

Table 7 Values of formulation parameters

Parameter	Value	Parameter	Value
$\boldsymbol{P}$	4	D	183
$\boldsymbol{L}$	2	g	3
$\boldsymbol{C}$	16	$vehL$	5.5
$\boldsymbol{O}$	4	ϑ	0.9
$\boldsymbol{X}$	305	x_0	0
$\boldsymbol{v_{max}}$	66	v_0	22
$\boldsymbol{a_{max}}$	3	bor	200
$\boldsymbol{a_{min}}$	-3	$intv$	0.1

As indicated in **Table 7**, the values of P, L, C and O define the size of the sets and other values only define parameters themselves. In addition, in the sensitivity analysis, the batch number n ranges from 2 to 6 with an interval 1, and the zone range ℓ extends from 11 to 33 meters with an interval 5.5 meters. For comparative analyses and simplified notations in the following sections, let BATCH-**V**4 denotes piecewise trajectory modelling with batch number **4**, BATCH-**C**6 denotes continuous trajectory modelling with batch number **6**, ZONE-**V**5 denotes piecewise trajectory modelling with zone range **27** meters, and ZONE-**C**2 denotes continuous trajectory modelling with zone range **11** meters. For each scenario, it runs 10 minutes on CPLEX multiple times with random seeds. The arrival times of each vehicle in the scenarios are stochastically generated.

5.5.2 Trajectory modelling analysis

The trajectory modelling methods under various traffic scenarios are shown as follows. Note that only the trajectory under **1800** veh/h/ln is shown for a worst-case/highest-demand analysis. Therefore, the overall intersection demand is up to **14400** veh/h. The trajectories of piecewise trajectory modelling with BATCH strategy are shown in **Figure 13**, where trajectories from conflicting approaches 1 and 4 are illustrated respectively by arrows.

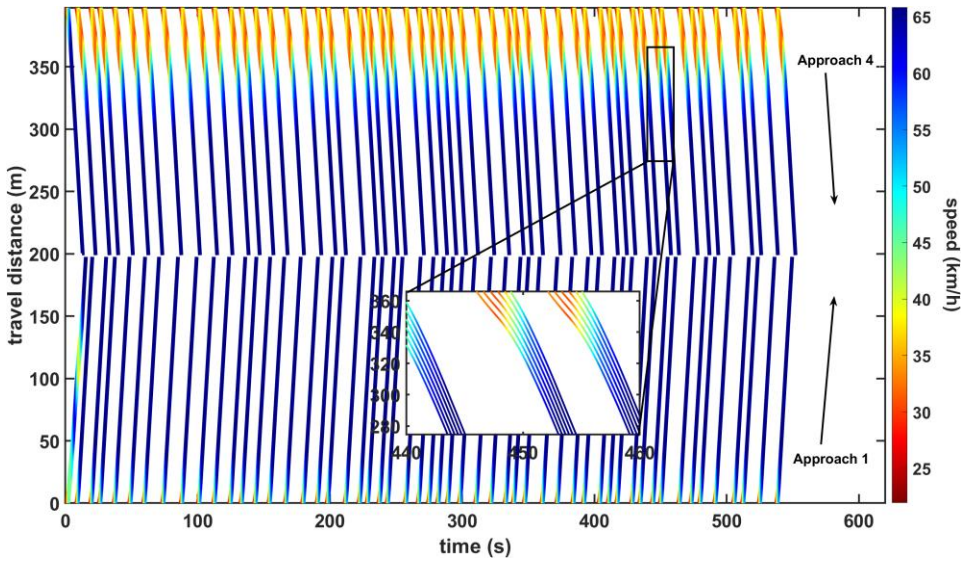


Figure 13 The trajectories of piecewise trajectory modelling with BATCH strategy

To have a closer view on the trajectories, trajectories generated between time 440 to 460 seconds and travel distance 274 to 366 meters are amplified for approach 4. From that, the trajectory becomes loose to dense, indicating an increasing trend in speed. It is also shown that all the trajectories have the same and smooth pattern without queue propagation or oscillation, indicating piecewise trajectory modelling under BATCH strategy can efficiently coordinate all the vehicles until they cross the intersection.

The trajectories of continuous trajectory modelling with BATCH strategy are shown in **Figure 14**. From **Figure 14**, no conflicts are identified either. However, some vehicles cannot reach the maximal design speed when arriving at the intersection.

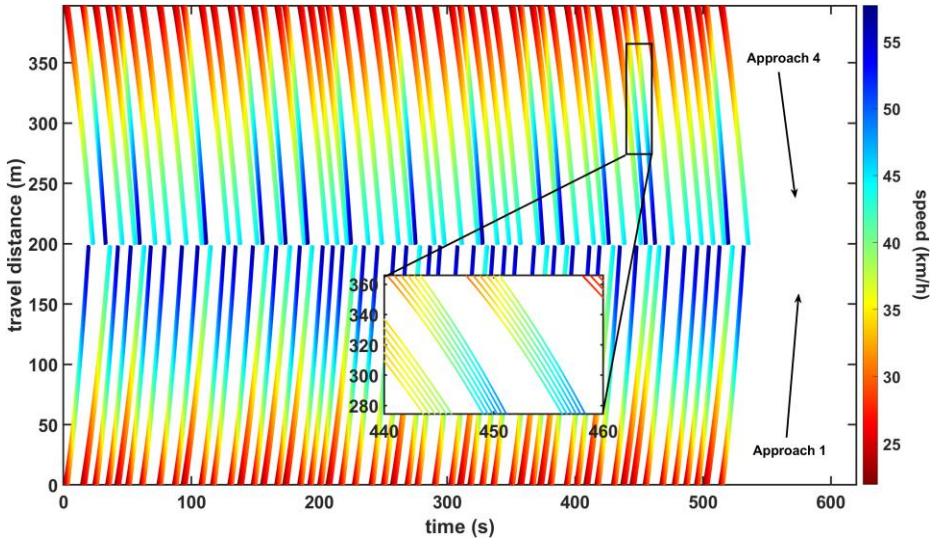


Figure 14 The trajectories of continuous trajectory modelling with BATCH strategy

The trajectories of piecewise trajectory modelling with ZONE strategy are shown in **Figure 15**. Similarly, no conflicts are identified in the ZONE strategy. But during time 440 to 460 seconds and between travel distance 274 to 366 meters, the trajectories become dense to loose, indicating a decreasing trend in speed. Similarly, as in **Figure 13**, a smooth trajectory pattern is again identified for piecewise trajectory modelling with ZONE strategy. In this case, the zone range is set to 33 meters, which can process at most six vehicles per lane per approach.

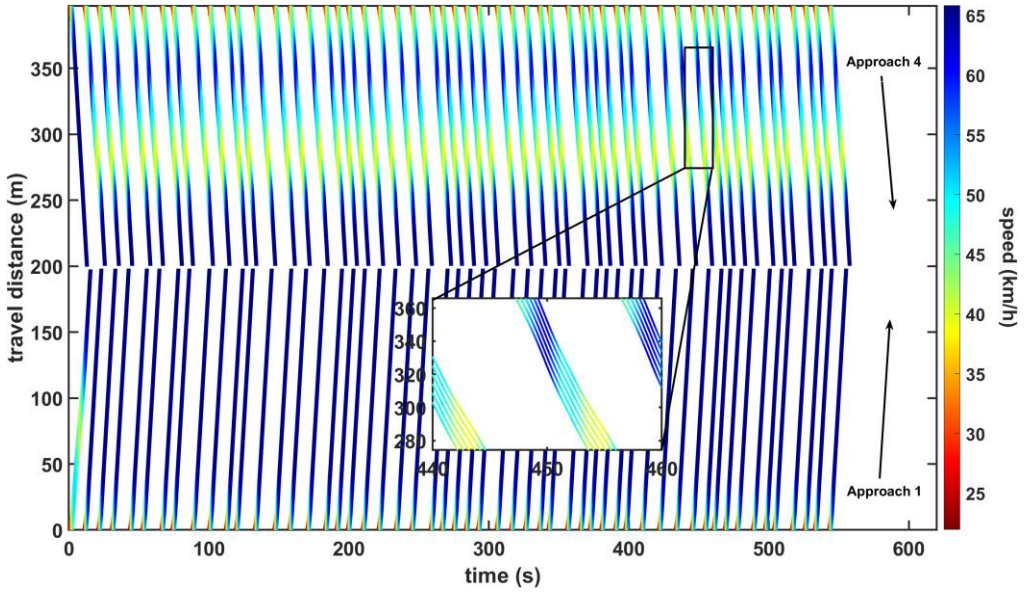


Figure 15 The trajectories of piecewise trajectory modelling with ZONE strategy

The trajectories of continuous trajectory modelling with ZONE strategy are shown in **Figure 16**. Similarly as **Figure 14**, some vehicle cannot reach the maximal design speed when arriving at the intersection.

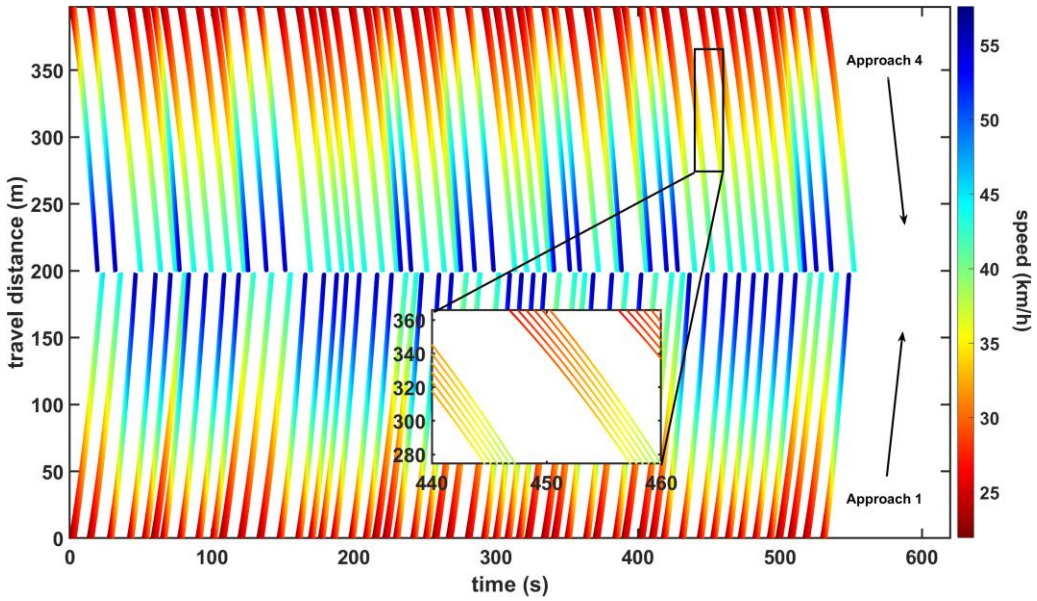
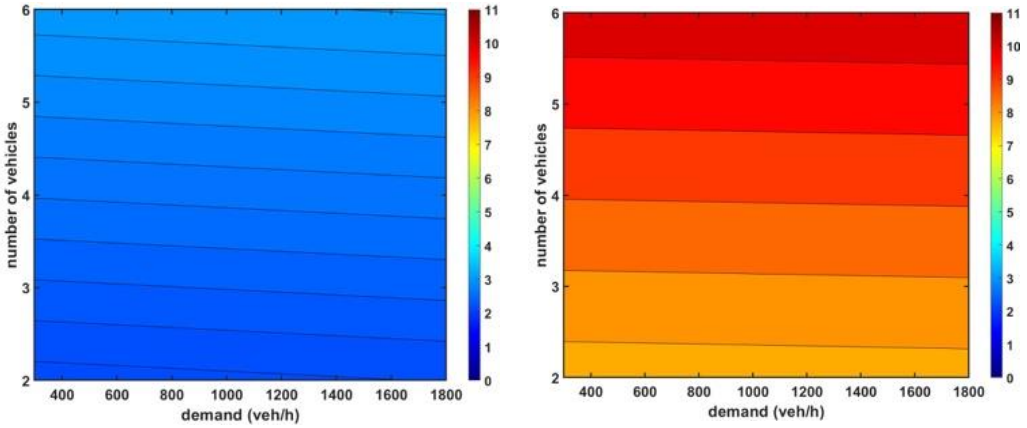


Figure 16 The trajectories of continuous trajectory modelling with ZONE strategy

Overall, under various traffic scenarios, $v(t_f) = v_f$ for each CAV when $v(t_f)$ is modelled independently of t_f in $\boldsymbol{J}(\cdot)$, i.e., $a(t) = f(t)$ is a piecewise function in $\boldsymbol{J}(\cdot)$; whereas $v(t_f) < v_f$ for each CAV when $a(t) = f(t)$ is a continuous function in $\boldsymbol{J}(\cdot)$. It is concluded from the numerical simulations that the variable $v(t_f)$ of each CAV should equal v_f so that the optimal intersection efficiency can be achieved, i.e., **Corollary 5.3** is validated.

5.5.3 Sensitivity analysis for intersection delay

The sensitivity analysis for intersection delay under different traffic demands and different batch numbers is shown in **Figure 17** for evaluating the performance of piecewise trajectory modelling over continuous modelling.

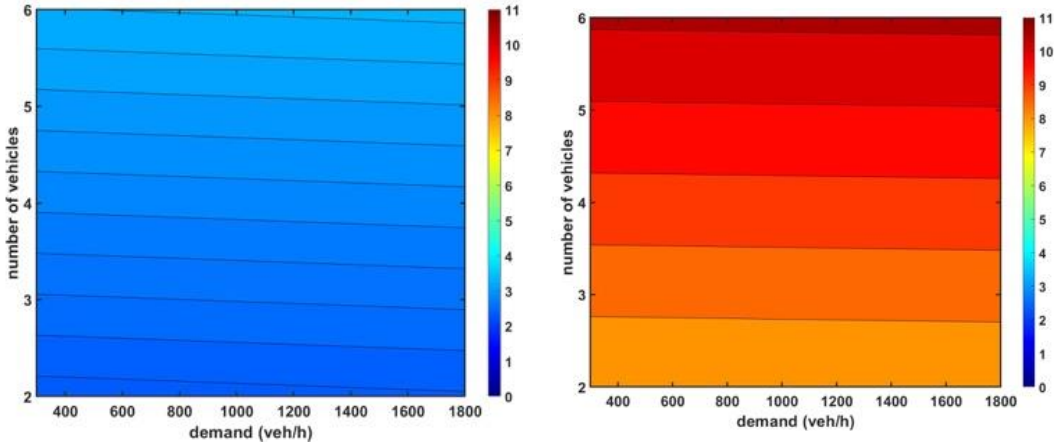


(a) (b)

Figure 17 Average intersection delay (s/veh) under BATCH strategy and different demands: (a) piecewise modelling; and (b) continuous modelling

As shown in **Figure 17**, the average intersection delay of continuous trajectory modelling is higher than that of piecewise trajectory modelling under all levels of demands and batch numbers, indicating the higher efficiency of piecewise trajectory modelling for intersection coordination based on **Proposition 5.1 & 5.2** and **Corollary 5.3**. In addition, the average intersection delay is increasing while the batch number is increasing under all levels of

demands, which is because the average service time is increasing based on **Corollary 5.1 & 5.3**. Besides, no significant variation of average intersection delay between different traffic demands is identified for both continuous and piecewise trajectory modelling, which is because the intersection demand is lower than the capacity. Overall, the average delay of piecewise trajectory modelling is substantially lower than that of continuous modelling. The sensitivity analysis for intersection delay under different traffic demands and different zone ranges is shown in **Figure 18** for evaluating the performance of piecewise trajectory modelling over continuous modelling.



(a) (b)

Figure 18 Average intersection delay (s/veh) under ZONE strategy and different demands: (a) piecewise modelling; and (b) continuous modelling

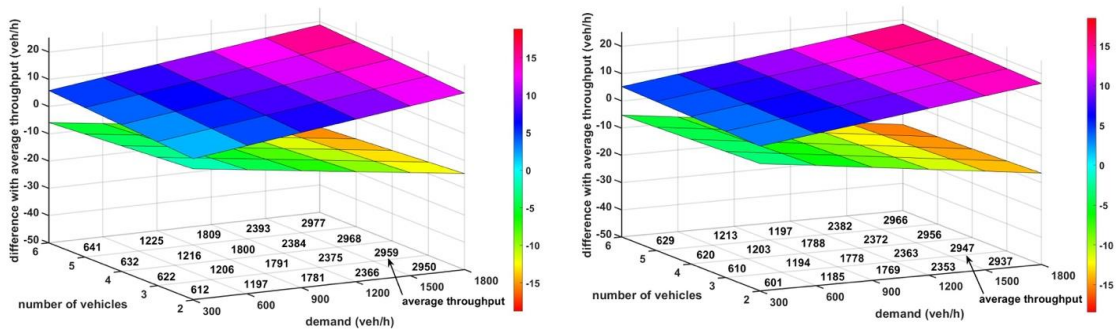
As shown in **Figure 18**, the average intersection delay of continuous modelling is higher than that of piecewise trajectory modelling under all levels of demands and batch numbers, indicating the higher efficiency of piecewise trajectory modelling for intersection coordination.

Similarly, the average intersection delay is increasing while the zone range is increasing under all levels of demands. Besides, no significant variation of average intersection delay between different traffic demands is identified for both continuous and piecewise trajectory

modelling. However, the average delay under ZONE strategy is slightly higher than that under BATCH strategy in terms of all levels of demands. This is probably because vehicles grouped by ZONE require to occupy at least two cells versus at least one cell required by vehicles grouped by BATCH at each simulation step, such that BATCH has more and better solutions based on available intersection resources, i.e., cells within the intersection. This is also reflected from **Figure 13** to **Figure 16** where BATCH has larger average travel speed than ZONE. Overall, the average delay of piecewise trajectory modelling is again substantially lower than that of the continuous modelling under the ZONE strategy.

5.5.4 Sensitivity analysis for intersection throughput

The average intersection approach throughput for each control scenario is shown in **Figure 19** for evaluating the performance of piecewise trajectory modelling over continuous modelling under all levels of demands. As shown in **Figure 19**, the average intersection approach throughput of piecewise trajectory modelling is higher than that of continuous modelling under all traffic demands and all batch numbers/zone ranges scenarios. The numbers in the bottom indicate the mean of the average intersection approach throughput of piecewise and continuous modelling. For **Figure 19** (a) or (b), the upper surface with positive numbers indicates the difference with the mean value of piecewise modelling, and the lower surface with negative numbers indicates the difference of continuous modelling.



(a) (b)

Figure 19 Difference with average approach throughput (veh/h) for piecewise (upper surface) and continuous (lower surface) modelling under: (a) BATCH strategy; and (b) ZONE strategy

As indicated earlier, as shown in **Figure 14** and **Figure 16**, the arrival speed of some vehicles under continuous modelling cannot reach the maximal design speed, and the average service time of the intersection under continuous modelling is larger than piecewise modelling based on **Corollary 5.1**. Furthermore, the capacity of the intersection under continuous modelling is lower than piecewise modelling based on **Corollary 5.2**. Therefore, the throughput under continuous modelling is lower than that under piecewise modelling across all test scenarios.

CHAPTER 6. AN OPTIMAL SCHEDULING MECHANISM

6.1 Problem Description

6.1.1 Problem definition

In a reservation-based intersection, passing orders of each CAV coming from different approaches can be scheduled by a central controller. The central controller assigns arrival time at the intersection to each CAV within a certain communication range in order to achieve a global optimal solution in terms of the intersection efficiency, e.g., overall travel time or average delay. The reservation-based intersection that is investigated in this paper is shown in **Figure 20**.

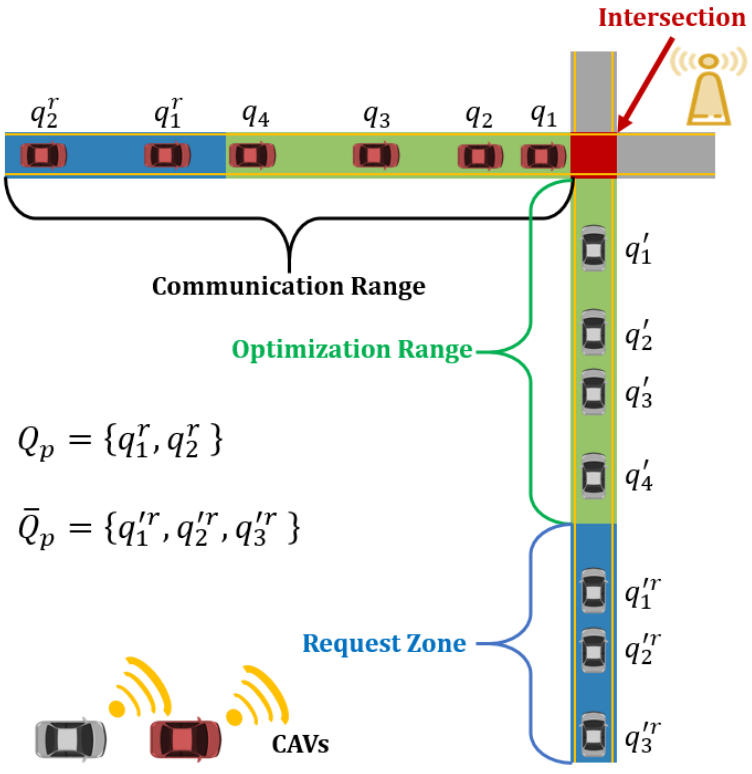


Figure 20 Scheduling of CAVs crossing a reservation-based intersection

This kind of scheduling problem is always modelled by a mixed integer linear programming (MILP) approach due to binary decisions of resolving each conflict between conflicting CAVs within the intersection area. Because of the branching strategy utilized by state-of-art solvers, such as Cplex or Gurobi, it takes exponential growth time to solve such problem with respect to the number of conflicts between conflicting CAVs at worst case.

Overall, the objective of this problem is to find the global optimal solution in terms of minimizing the maximal arrival time of all CAVs at the intersection, in the meantime, to develop a customized algorithm that can reduce the time complexity of the scheduling problem.

In the context of the scheduling problem, some assumptions are made for theoretical derivation and numerical modeling as follows.

1. A platoon of CAVs is more efficient than an orderly mix of conflicting CAVs crossing the intersection in terms of intersection throughput or delay;
2. A platoon of CAVs is consisted of at least 2 CAVs;
3. An initial speed for all CAVs from entering the communication range is same; and
4. No lane-changing and overtaking are allowed along the communication range.

6.1.2 Decision variables

Under the reservation-based control, four decision variables should be defined regarding the scheduling problem. The definitions of the decision variables are used to derive important properties.

Definition 6.1. Decision variable, **crossing time**, $\boldsymbol{\alpha}$, is used to measure a length of time that a CAV requires to cross the intersection.

The crossing time that a CAV requires to cross a cell is illustrated in **Figure 21**. Based on the **Definition 6.1**, an assumption of the cell within the intersection is given as follows.

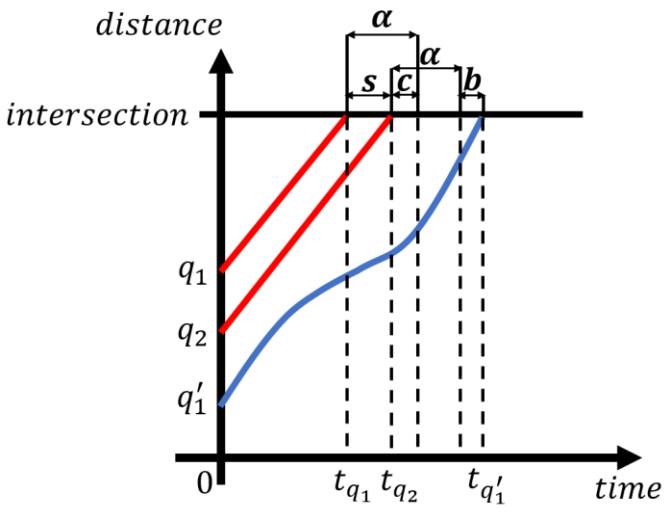


Figure 21 Illustration of decision variables in a space-time diagram

Assumption 6.1. Any one cell within the reservation-based intersection is a *square* with a fixed size.

Based on the **Assumption 6.1**, it is given that the crossing time for each CAV is same as long as an arrival speed of CAVs is same and the speed is fixed during crossing. The arrival speed will be discussed later. Furthermore, second decision variable related to the cell is defined as follows.

Definition 6.2. Decision variable, **safety buffer**, ***b***, is used to measure a minimum safety time gap for one of two conflicting CAVs arriving at the intersection after the other crossed the intersection.

The safety buffer is illustrated in **Figure 21**. Additionally, third decision variable related to the vehicle is defined as follows.

Definition 6.3. Decision variable, **saturation headway**, $\boldsymbol{s}$, is used to measure the minimum headway that can be achieved between a leading and following vehicle from a same lane.

Assumption 6.2. The saturation headway is same for all pairs of leading and following vehicles from any lane of any approach.

Definition 6.4. Decision variable, **crossing gap**, $\boldsymbol{c} \coloneqq \boldsymbol{\alpha} - \boldsymbol{s}$, is used to measure a time difference between the crossing time and the saturation headway.

Note that the crossing gap may be larger than, equal to, or less than zero. The crossing gap is illustrated in **Figure 21**.

Assumption 6.3. $\boldsymbol{\alpha} > \boldsymbol{c} > -\boldsymbol{b}$.

Remark 6.1. The **Assumption 6.3** is a *weak* assumption in that a platoon of CAVs is more efficient than an orderly mix of conflicting CAVs crossing the intersection in terms of intersection throughput or delay (Lioris et al., 2017; Tachet et al., 2016; Yu et al., 2019; Zhou and Zhu, 2021).

6.1.3 Notations

All the sets, variables and parameters that will be used to model the scheduling problem and derive the problem properties are listed below. Note here that specific functions, denoted as *symbols* in **Table 8**, are detailed as well.

Table 8 Sets, variables, parameters and symbols

Sets	
$\mathbb{A}$	All vehicles from all approaches
$\boldsymbol{L}$	Levels of the tree of a scheduling problem, $L = \{l \mid l \in \mathbb{Z}^{0+}\}$
$\boldsymbol{Z_l}$	Nodes at level l, $Z_l = \{i \mid i \in \mathbb{Z}^{+}\}$
$\mathbb{R}$	All nodes of a tree, $\mathbb{R} = \{r_{l,i} \mid i \in Z_l, l \in L\}$
$\boldsymbol{P}$	Sets of approaches, $P = \{p\}$
$\boldsymbol{Q_p}$	Sets of vehicles from approach p at a current optimization cycle, $Q_p = \{q\}$

$\bar{\boldsymbol{Q}}_{\boldsymbol{p}}$	Sets of vehicles from approach p' at a current optimization cycle, $\bar{Q}_p = \{q' \mid p \neq p'\}$
$\mathbb{Q}$	A platoon of vehicles from an approach
$\mathbb{C}$	Sets of all conflict pairs
$\mathbb{C}_{r_{l,i}}$	A first conflict pair at node $r_{l,i}$, $\mathbb{C}_{r_{l,i}} \in \mathbb{C}$
$\mathbb{O}_{r_{l,i}}$	Sets of vehicles that have been ordered at node $r_{l,i}$
Variables	
$\boldsymbol{t_q}$	Assigned arrival time at intersection of vehicle q
$\boldsymbol{t_q^{r_{l,i}}}$	Assigned arrival time at intersection of *current* vehicle q ordered at node $r_{l,i}$
$\boldsymbol{T}$	Minimum of maximal arrival time of all vehicles at a current optimization cycle
$\boldsymbol{T}_{r_{l,i}}$	Minimum of maximal arrival time of all unordered vehicles at node $r_{l,i}$
$\boldsymbol{\delta}_{r_{l,i}}$	Binary variable for determining a passing order of a conflict pair $\mathbb{C}_{r_{l,i}}$ at node $r_{l,i}$
Parameters	
$\boldsymbol{t_q^m}$	Minimum arrival time at intersection of vehicle q in terms of itself
$\boldsymbol{t^o}$	Arrival time at the intersection of a *last passing* vehicle o at last optimization cycle
$\boldsymbol{t_q^{m,c}}$	Updated minimum arrival time at intersection of vehicle q after considering t^o at a current optimization cycle
$\boldsymbol{\tau}$	A difference of assigned arrival time between a first vehicle and a last vehicle from a same approach in a *partially* ordered set of an optimal solution under a given node
Symbols	
$\prec$	Relation between two vehicles, where $q \prec \varphi$ denotes vehicle φ immediately follows vehicle q from a same approach
$\subset$	Relation between a child node and a parent node, where $r_{l+1,i} \subset r_{l,\bar{i}}$ denotes node $r_{l+1,i}$ is a child of node $r_{l,\bar{i}}$
$\perp$	Relation between a vehicle and a node, where $q \perp r_{l,i}$ denotes the vehicle q is the *current* vehicle ordered at node $r_{l,i}$
$<>$	Counting vehicles in order from a same approach, where $\langle q, \varphi \rangle$ denotes the number of vehicles from q to φ in order including φ but not q

6.2 Problem Properties

6.2.1 Optimal substructure

If the original MILP is regarded as a root node $r_{0,1}$ of a tree of nodes $\mathbb{R}$, solving the MILP is then regarded as adding child nodes $\{r_{l,i} \mid l \in L, i \in Z_l\}$ of each level l to $\mathbb{R}$, then

$$\mathbb{R} \coloneqq \left\{r_{0,1}, r_{1,1}, r_{1,2}, \dots, r_{\max(L),Z_{\max(L)}}\right\} = \{r_{l,i} | \ i \in Z_l, l \in L\} \tag{44}$$

Each child $r_{l,i}$ denotes a subproblem of the root node $r_{0,1}$ with a binary variable $\delta_{r_{l,i}}$ fixed as 0 or 1.

$$\delta_{r_{l,i}} \coloneqq \begin{cases} 1 & if\ q \perp r_{l,i} \\ 0 & if\ q' \perp r_{l,i} \end{cases} \tag{45}$$

where $r_{l,i} \subset r_{l-1,\bar{i}}$ and without loss of generality (w.l.o.g.) $\mathbb{C}_{r_{l-1,\bar{i}}} = \left\{q, q' | q \in Q_p, q' \in \bar{Q}_p\right\}$. While $r_{l,i}$ is adding to $\mathbb{R}$ from upper levels to lower levels, passing orders of a subset of all approaching vehicles are fixed subsequently at node $r_{l,i}$. When passing orders of all vehicles are fixed at nodes $\{r_{\max(L),i} | i \in Z_{\max(L)}\}$, the global optimal solution of the original MILP can be found from a subset of $\mathbb{R}$, then this kind of problem has an optimal substructure (Cormen et al., 2009). Techniques of solving such problem with the optimal substructure include Dynamic Programming (DP) and Brand & Bound/Cut (B&B/C) algorithms.

Dealing with a general scheduling problem as defined in Section 6.1.1, state-of-art solvers, such as Cplex, take exponential-growth time while the problem size, i.e., number of vehicles and binary variables, is increasing (Xu et al., 2021). The reason of that is due to a hybrid strong and pseudo-cost branching strategy for the binary variables utilized by commercial solvers (Morrison et al., 2016). Specifically, the branching strategies in general neglect vehicle orders in a single lane. However, by considering the vehicle or conflict orders of the scheduling problem, the time complexity of finding a global optimal solution can be thereby reduced, which is introduced in the following section.

6.2.2 Conflict order

Before giving the definition of the conflict order, we first denote $\overrightarrow{q(\mathbb{S})}$ as the vehicle order of vehicle q in a given set $\mathbb{S}$, and $1 \le \overrightarrow{q(\mathbb{S})} \le |\mathbb{S}|$, where $|\cdot|$ denotes the number of vehicles in a given set.

Definition 6.5. $\mathbb{C}_{r_{l,i}} \coloneqq \left\{\{q, q'\} \middle| q \in \{\mathbb{A} - \mathbb{O}_{r_{l,i}}\} \cap Q_p, q' \in \{\mathbb{A} - \mathbb{O}_{r_{l,i}}\} \cap \bar{Q}_p\right\}$, where $\overrightarrow{q(\{\mathbb{A} - \mathbb{O}_{r_{l,i}}\} \cap Q_p)} = 1$ and $\overrightarrow{q'(\{\mathbb{A} - \mathbb{O}_{r_{l,i}}\} \cap \bar{Q}_p)} = 1$.

Remark 6.2. It is defined as above that a first conflict pair of a node $r_{l,i}$ consists of the first vehicle that has not been ordered at the node from each approach in conflict. $\mathbb{C}_{r_{l,i}}$ is also determined by the conflict order of vehicles based on a first-in-first-out (FIFO) rule for each approach.

Since the first conflict pair of each node searches vehicles based on FIFO, the reason that the time complexity of the scheduling problem based on searching the conflict order can be reduced is given as follows. Before explaining the reason, we first define a situation where a solution of the scheduling problem at a node is infeasible as follows.

$$\begin{cases} \mathbb{t} \cap \mathbb{t}' \neq \emptyset \\ \forall \mathbb{t} \subseteq \left\{\left[t_q^{r_{l,i}}, t_q^{r_{l,i}} + \alpha + b\right] \middle| t_1 \le t_q^{r_{l,i}} \le t_2\right\} \\ \forall \mathbb{t}' \subseteq \left\{\left[t_{q'}^{r_{l,i'}}, t_{q'}^{r_{l,i'}} + \alpha + b\right] \middle| t_1' \le t_{q'}^{r_{l,i'}} \le t_2'\right\} \end{cases} \tag{46}$$

Where $\{r_{l,i}, r_{l,i'}\} \subset r_{l-1,\bar{i}}$, $\mathbb{C}_{r_{l-1,\bar{i}}} = \{q, q' | q \in Q_p, q' \in \bar{Q}_p\}$ and t_1, t_2, t_1' and t_2' are any four non-negative constants. Then if the solution at a node $r_{l-1,\bar{i}}$ is infeasible, $\mathbb{t} \cap \mathbb{t}' \neq \emptyset$.

Proposition 6.1. *Solutions of nodes* $\{r_{l,i} | l \in L, i \in Z_l\}$ *are all feasible based on the conflict order ruled by FIFO*.

Proof. We prove this by contradiction. For any node with a conflict pair and solution subsets of its conflicting vehicles as shown in Eq. (46), $t_2 \le +\infty$ and $t_2' \le +\infty$ because of

the FIFO rule on each approach. Then w.l.o.g. we assume $t_1' > t_q^{r_{l,i}} + \alpha + b$, where $t_1 \leq t_q^{r_{l,i}} \leq t_2$, i.e., $t_1' > t_2 + \alpha + b$, therefore $\mathbb{t} \cap \mathbb{t}' = \emptyset$ and solutions of nodes $\{r_{l,i} | l \in L, i \in Z_l\}$ are all feasible. ■

Remark 6.3. As opposed to wasting computation time in process of branching to nodes with infeasible solutions solved by state-of-art solvers, the time complexity and computation time of the scheduling problem can be reduced based on **Proposition 6.1** in which optimal solutions are found in a tightened subset of nodes that are all feasible.

6.2.3 Overlapping subproblems

In the process of branching from the root node of a tree, when a node is comparable with another node in terms of an objective function of a problem, then it is defined that the optimization problem has overlapping subproblems (Cormen et al., 2009). If any two nodes are comparable, one of them can be dominated by another, i.e., one node of them can be discarded in the process of branching, so can be child nodes of the node, thus the overall time complexity can be reduced.

An overlapping property is first introduced followingly and can be leveraged by the dynamic programming algorithm. Moreover, an extension of the overlapping property is later introduced, which further expedites the branching process and reduces the time complexity. The definition of the objective function of the scheduling problem is first introduced as follows.

Definition 6.6. $T_{r_{l,i}} \coloneqq minmax\{t_q | q \in \mathbb{A} - \mathbb{O}_{r_{l,i}} \neq \emptyset\} = \{t_q^{r_{l,i}} | \mathbb{A} - \mathbb{O}_{r_{l,i}} = \emptyset\}$.

Definition 6.7. $T \coloneqq minmax\{t_q | q \in \mathbb{A}\} = min\{T_{r_{l,i}} | i \in Z_l, l \in L\}$.

Remark 6.4. As indicated in **Definition 6.6**, $T_{r_{l,i}}$ can be classified by two scenarios: one is when $\mathbb{A} - \mathbb{O}_{r_{l,i}} \neq \emptyset$ and the other is when $\mathbb{A} - \mathbb{O}_{r_{l,i}} = \emptyset$. To find T, constraints of the scheduling model are to be satisfied.

Lemma 6.1. *Given a scheduling problem the optimal passing order of all vehicles is deterministic in terms of minimizing the maximal arrival time*.

Proof. The lemma proves itself. ■

Proposition 6.2. *Markov Property holds when searching the optimal solution from subproblems of the scheduling problem*.

Proof. A problem or a decision process has *Markov Property* if the following equation holds (Howard, 1960).

$$P(X_{n+1} = x_{n+1} | X_n = x_n, \cdots, X_0 = x_0) = P(X_{n+1} = x_{n+1} | X_n = x_n) \qquad (47)$$

Eq. (47) implies that future states do not depend on past states but only the present state. In the scheduling problem, X_n can be generally denoted as a state where $q \perp r_{l,\bar{i}}$. If $X_n = x_n$, then the vehicle q is assigned an arrival time $t_q^{r_{l,\bar{i}}}$ at the state X_n, i.e., $x_n = t_q^{r_{l,\bar{i}}}$. X_{n+1} denotes the future state based on the state X_n, and $x_{n+1} = t_\varphi^{r_{l+1,i}}$, where the vehicle φ immediately follows the vehicle q crossing the intersection and $r_{l+1,i} \subset r_{l,\bar{i}}$. W.l.o.g., we denote $x_{n+1} = f(x_n)$ in the scheduling problem, where $f(\cdot)$ denotes a state transition function, and

$$t_\varphi^{r_{l+1,i}} = f\left(t_q^{r_{l,\bar{i}}}\right) \coloneqq \max\left\{t_q^{r_{l,\bar{i}}} + \mathcal{F}(q, \varphi), t_\varphi^m\right\} \qquad (48)$$

where w.l.o.g. $t_\varphi^m = t_\varphi^{m,c}$ in this case, $t_\varphi^{m,c}$ is defined in Eq. (69) and $\mathcal{F}(\cdot)$ is denoted as follows as a state decision function.

$$\mathcal{F}(q,\varphi) \coloneqq \begin{cases} \alpha + b & if\ \varphi \in \bar{Q}_p, q \in Q_p \\ \alpha - c & if\ \{\varphi, q\} \in Q_p\ or\ \bar{Q}_p \end{cases} \tag{49}$$

It is shown as in Eq. (48) and (49) that $t_{\varphi}^{r_{l+1,i}}$ depends on the only one variable $t_q^{r_{l,\bar{i}}}$, in other words, the approach of the vehicle q. Therefore, the *Markov Property* holds until the $max\{t_q | q \in \mathbb{A}\}$ is minimized. ■

Note that in Eq. (48) $t_{\varphi}^{r_{l+1,i}} \geq f\left(t_q^{r_{l,\bar{i}}}\right)$ is valid as long as $t_{\varphi}^{r_{l+1,i}} \geq t_{\varphi}^{m}$. The variant of Eq. (48) is used and only used in some of the following proof processes. Before introducing the overlapping properties, a prerequisite for the properties is given as follows, where $r_{l,i}$ and $r_{l,i'}$ represent different nodes with different ID i and i' at a same level l.

Corollary 6.1. *If* $\mathbb{O}_{r_{l,i}} = \mathbb{O}_{r_{l,i'}}$ *and* $t_q^{r_{l,i}} = t_{\breve{q}}^{r_{l,i'}}$, *where* $q = \breve{q}$ *and* $i \neq i'$, *then* $T_{r_{l,i}} = T_{r_{l,i'}}$.

Proof. Since $\mathbb{O}_{r_{l,i}} = \mathbb{O}_{r_{l,i'}}$, $\mathbb{A} - \mathbb{O}_{r_{l,i}} = \mathbb{A} - \mathbb{O}_{r_{l,i'}}$. In addition, since $q = \breve{q}$, $\{q, \mathbb{A} - \mathbb{O}_{r_{l,i}}\} = \left\{\breve{q}, \mathbb{A} - \mathbb{O}_{r_{l,i'}}\right\}$, where $q \perp r_{l,i}$ and $\breve{q} \perp r_{l,i'}$. If $\mathbb{A} - \mathbb{O}_{r_{l,i}} = \mathbb{A} - \mathbb{O}_{r_{l,i'}} = \emptyset$, then $t_q^{r_{l,i}} = T_{r_{l,i}} = t_{\breve{q}}^{r_{l,i'}} = T_{r_{l,i'}}$; otherwise, based on **Lemma 6.1**, $\{q, \mathbb{A} - \mathbb{O}_{r_{l,i}}\}$ and $\left\{\breve{q}, \mathbb{A} - \mathbb{O}_{r_{l,i'}}\right\}$ are a same scheduling problem and have the same optimal passing order $\{q, q_n | 1 \leq n \leq N\}$ in the problem, where $N = \left|\mathbb{A} - \mathbb{O}_{r_{l,i}}\right| \geq 1$ and $\{q, q_n | 1 \leq n \leq N\} = \left\{q, \mathbb{A} - \mathbb{O}_{r_{l,i}}\right\} = \left\{\breve{q}, \mathbb{A} - \mathbb{O}_{r_{l,i'}}\right\}$. Therefore, it can be concluded that $T_{r_{l,i}} = T_{r_{l,i'}}$ based on **Proposition 6.2** because

$$\begin{aligned} T_{r_{l,i}} = T_{r_{l,i'}} &= minmax\{q_n | 1 \leq n \leq N\} = minmax\left\{t_{q_1}, t_{q_2}, \dots, t_{q_N}\right\} \\ &= t_{q_N} \end{aligned} \tag{50}$$

where

$$t_{q_n} = \begin{cases} \max\left\{t_{q_{n-1}} + \mathcal{F}(q_{n-1}, q_n), t_{q_n}^m\right\} & if\ n \geq 2 \\ \max\left\{\ t_q^{r_{l,i}} + \mathcal{F}(q, q_n), t_{q_n}^m\right\} & if\ n = 1 \end{cases} \tag{51}$$

■

Remark 6.5. **Corollary 6.1** implies that the precondition where two nodes with different ID at a same level are overlapping or comparable. *Markov Property* in **Proposition 6.2** also indicates that if and only if $t_q^{r_{l,\bar{i}}}$ is minimized, $t_\varphi^{r_{l+1,i}}$ is minimized, where $r_{l+1,i} \subset r_{l,\bar{i}}$; in addition, $T = min\left\{t_q^{r_{\max(L),i}} \middle| q \perp r_{\max(L),i}, i \in Z_{\max(L)}\right\}$.

Therefore, we can have the overlapping/dominance property proven as follows.

Theorem 6.1. *If and only if* $\mathbb{O}_{r_{l,i}} = \mathbb{O}_{r_{l,i'}}$ *and* $t_q^{r_{l,i}} \geq t_{\breve{q}}^{r_{l,i'}}$, *where* $q = \breve{q}$ *and* $i \neq i'$, *node* $r_{l,i}$ *can be dominated by node* $r_{l,i'}$.

Proof. Since $t_q^{r_{l,i}} \geq t_{\breve{q}}^{r_{l,i'}}$, we have $T_{r_{l,i}} \geq T_{r_{l,i'}}$ based on **Corollary 6.1** because

$$\begin{cases} T_{r_{l,i}} = minmax\{q_n | 1 \leq n \leq N\} = minmax\left\{t_{q_1}, t_{q_2}, \dots, t_{q_N}\right\} = t_{q_N} \\ T_{r_{l,i'}} = minmax\{q_n | 1 \leq n \leq N\} = minmax\left\{t'_{q_1}, t'_{q_2}, \dots, t'_{q_N}\right\} = t'_{q_N} \end{cases} \tag{52}$$

where if $n \geq 2$

$$\begin{aligned} t_{q_n} = \max\left\{t_{q_{n-1}} + \mathcal{F}(q_{n-1}, q_n), t_{q_n}^m\right\} &\geq t'_{q_n} \\ = \max\left\{t_{q_{n-1}} + \mathcal{F}(q_{n-1}, q_n), t_{q_n}^m\right\} & \end{aligned} \tag{53}$$

and if $n = 1$

$$t_{q_n} = \max\left\{t_q^{r_{l,i}} + \mathcal{F}(q, q_n), t_{q_n}^m\right\} \geq t'_{q_n} = \max\left\{t_{\breve{q}}^{r_{l,i'}} + \mathcal{F}(\breve{q}, q_n), t_{q_n}^m\right\} \tag{54}$$

where $\{q, q_n | 1 \leq n \leq N\} = \left\{q, \mathbb{A} - \mathbb{O}_{r_{l,i}}\right\} = \left\{\breve{q}, \mathbb{A} - \mathbb{O}_{r_{l,i'}}\right\}$ and $N = \left|\mathbb{A} - \mathbb{O}_{r_{l,i}}\right| \geq 1$. In addition, if $\mathbb{A} - \mathbb{O}_{r_{l,i}} = \mathbb{A} - \mathbb{O}_{r_{l,i'}} = \emptyset$, then $t_q^{r_{l,i}} = T_{r_{l,i}} \geq T_{r_{l,i'}} = t_{\breve{q}}^{r_{l,i'}}$. Therefore, it is

concluded that node $r_{l,i}$ can be dominated/discarded by node $r_{l,i'}$ based on **Proposition 6.2** and **Definition 6.7** because $t_q^{r_{l,i}} \geq t_{\breve{q}}^{r_{l,i'}}$ and $T_{r_{l,i}} \geq T_{r_{l,i'}}$. ■

Based on **Theorem 6.1**, an extension of the dominance property can be derived to further expedite the branching process. Before giving the property, a fundamental rule for comparing any two nodes is given as follows.

Lemma 6.2. *Given any two nodes $r_{l,i}$ and $r_{l,i'}$, where $i \neq i'$, if any one solution under node $r_{l,i'}$ is less than or equal to $T_{r_{l,i}}$, node $r_{l,i}$ can be dominated by node $r_{l,i'}$.*

Proof. It is easy to prove that if we have one solution under node $r_{l,i'}$ denoted as $\Upsilon_{r_{l,i'}}$, then $T_{r_{l,i'}} \leq \Upsilon_{r_{l,i'}} \leq T_{r_{l,i}}$. Based on **Definition 6.7**, node $r_{l,i}$ can be dominated/discarded by node $r_{l,i'}$. ■

Theorem 6.2. *If and only if $\mathbb{O}_{r_{l,i}} \backslash q = \mathbb{O}_{r_{l,i'}} \backslash q'$ and $t_q^{r_{l,i}} - t_{q'}^{r_{l,i'}} \geq b + c$, where $q \in Q_p$, $q' \in \bar{Q}_p$ and $i \neq i'$, node $r_{l,i}$ can be dominated by node $r_{l,i'}$.*

Proof. It can be determined that the optimal passing order under node $r_{l,i}$ has a form as

$$\left\{ \overbrace{\underbrace{q, \dots, q_n}_{|q_\mathfrak{q}| \geq 1}, q'}^{\mathbb{q}}, \overbrace{\varphi, \dots}^{\mathbb{A}-\mathbb{q}-\mathbb{O}_{r_{l,i}} \backslash q} \right\} \tag{55}$$

where $\mathbb{A} - \mathbb{q} - \mathbb{O}_{r_{l,i}} \backslash q \neq \emptyset$, $q_\mathfrak{q} = \{q, \dots, q_n\} \subseteq Q_p$ and $|q_\mathfrak{q}| \geq 1$, i.e., when $|q_\mathfrak{q}| = n = 1$, $q = q_n$. Since $\mathbb{O}_{r_{l,i}} \backslash q = \mathbb{O}_{r_{l,i'}} \backslash q'$, Eq. (55) is valid because q' has not been ordered under $r_{l,i}$ at a same level l with node $r_{l,i'}$. We further denote $\tau = t_{q_n} - t_q^{r_{l,i}}$ and $\tau \geq 0$ based on Eq. (55). Then it can be given that a passing order under node $r_{l,i'}$ has a form as

$$\left\{\overbrace{q', \underbrace{q, \dots, q_n}_{|q|\geq 1}}^{\mathbb{q}'}, \overbrace{\breve{\varphi}, \dots}^{\mathbb{A}-\mathbb{q}'-\mathbb{O}_{r_{l,i'}}\backslash q'}\right\} \tag{56}$$

where $\mathbb{A}-\mathbb{q}'-\mathbb{O}_{r_{l,i'}}\backslash q' \neq \emptyset$. We let $\breve{\varphi}=\varphi$ and it is valid because $\mathbb{A}-\mathbb{q}-\mathbb{O}_{r_{l,i}}\backslash q = \mathbb{A}-\mathbb{q}'-\mathbb{O}_{r_{l,i'}}\backslash q'$, however, only the approach of φ is unknown. We further denote $\tau' = t'_{q_n} - t_q^{r_{l+1,j}}$ in Eq. (56), where $r_{l+1,j} \subset r_{l,i'}$; we let $\tau'=\tau$ and it is valid because we can have (in some cases $\tau' < \tau$ but it does not impact the proof and conclusion)

$$\begin{cases} t'_{q_n} = t_{q_n} & if\ t_q^m \geq t_{q'}^{r_{l,i'}} + \alpha + b \\ t'_{q_n} > t_{q_n} & if\ t_q^m < t_{q'}^{r_{l,i'}} + \alpha + b \end{cases} \quad \forall q_n \in q \tag{57}$$

i.e.

$$\tau = t_{q_n} - t_q^{r_{l,i}} = t'_{q_n} - t_q^{r_{l+1,j}} = t_{q_n} + \varepsilon - (t_q^{r_{l,i}} + \varepsilon) \quad \forall q_n \in q \tag{58}$$

where $\varepsilon = t_q^{r_{l+1,j}} - t_q^{r_{l,i}} \geq 0$ and in this case we let $t_{q_n}^m = t_{q_n}^{m,c}$, $\forall q_n \in q$. Furthermore, since $\mathbb{q} \cup \mathbb{O}_{r_{l,i}}\backslash q = \mathbb{q}' \cup \mathbb{O}_{r_{l,i'}}\backslash q' \neq \mathbb{A}$, if we know $t_\varphi^{r_{l+|q|+1,g}} \geq t_{\breve{\varphi}}^{r_{l+|q|+1,g'}}$, where $r_{l+|q|+1,g} \subset r_{l,i}$, $r_{l+|q|+1,g'} \subset r_{l,i'}$, $\forall g \in Z_{l+|q|+1}$, $\forall g' \in Z_{l+|q|+1}$ and $g \neq g'$, we can have node $r_{l,i}$ dominated by node $r_{l,i'}$ based on **Theorem 6.1** and **Lemma 6.2**. To validate if $t_\varphi^{r_{l+|q|+1,g}} \geq t_{\breve{\varphi}}^{r_{l+|q|+1,g'}}$, if $\{\varphi, q\} \in Q_p$ we have as follows given the situations in Eq. (57)

$$\begin{cases} t_\varphi^{r_{l+|q|+1,g}} = t_{\breve{\varphi}}^{r_{l+|q|+1,g'}} & if\ t_\varphi^m \geq t_q^{r_{l,i}} + \tau + 2\alpha + 2b \\ t_\varphi^{r_{l+|q|+1,g}} > t_{\breve{\varphi}}^{r_{l+|q|+1,g'}} & if\ t_\varphi^m < t_q^{r_{l,i}} + \tau + 2\alpha + 2b \end{cases} \tag{59}$$

where

$$t_\varphi^{r_{l+|q|+1,g}} = max\left\{t_q^{r_{l,i}} + \tau + 2\alpha + 2b, t_\varphi^m\right\} > t_{q'}^{r_{l,i'}} + \alpha + b + \tau + s \tag{60}$$

and

$$t_{\breve{\varphi}}^{r_{l+|q|+1,g'}} = max\left\{t_{q'}^{r_{l,i'}} + \alpha + b + \tau + s, t_{\varphi}^{m}\right\} \tag{61}$$

where we let $t_{\varphi}^{m} = t_{\varphi}^{m,c}$ in this case and followings. Note that Eq. (60) is valid because $c > -b$ and $t_{q}^{r_{l,i}} - t_{q'}^{r_{l,i'}} \geq b + c$. If $\{\varphi, q'\} \in \bar{Q}_p$ we have

$$\begin{cases} t_{\varphi}^{r_{l+|q|+1,g}} = t_{\breve{\varphi}}^{r_{l+|q|+1,g'}} & if\ t_{\varphi}^{m} \geq t_{q}^{r_{l,i}} + \tau + 2\alpha + b - c \\ t_{\varphi}^{r_{l+|q|+1,g}} \geq t_{\breve{\varphi}}^{r_{l+|q|+1,g'}} & if\ t_{\varphi}^{m} < t_{q}^{r_{l,i}} + \tau + 2\alpha + b - c \end{cases} \tag{60}$$

where

$$t_{\varphi}^{r_{l+|q|+1,g}} = max\left\{t_{q}^{r_{l,i}} + \tau + 2\alpha + b - c, t_{\varphi}^{m}\right\} \geq t_{q'}^{r_{l,i'}} + 2\alpha + 2b + \tau \tag{63}$$

and

$$t_{\breve{\varphi}}^{r_{l+|q|+1,g'}} = max\left\{t_{q'}^{r_{l,i'}} + 2\alpha + 2b + \tau, t_{\varphi}^{m}\right\} \tag{64}$$

Note that Eq. (63) is valid because $t_{q}^{r_{l,i}} - t_{q'}^{r_{l,i'}} \geq b + c$. Therefore, we have $t_{\varphi}^{r_{l+|q|+1,g}} \geq t_{\breve{\varphi}}^{r_{l+|q|+1,g'}}$ no matter which approach vehicle φ belongs to, then node $r_{l,i}$ can be dominated by node $r_{l,i'}$.

In case the optimal passing order under node $r_{l,i}$ has a form $\mathbb{q} \cup \mathbb{O}_{r_{l,i}} \backslash q = \mathbb{A}$ and $|q| \geq 1$, then $\mathbb{q} \cup \mathbb{O}_{r_{l,i}} \backslash q = \mathbb{q}' \cup \mathbb{O}_{r_{l,i'}} \backslash q' = \mathbb{A}$. We have $T_{r_{l,i}} = T_{r_{\max(L),u}} = t_{q'}^{r_{\max(L),u}}$ based on **Definition 6.6**, where $r_{\max(L),u} \subset r_{l,i}$ and $t_{q'}^{r_{\max(L),u}} = t_{q}^{r_{l,i}} + \tau + \alpha + b$. As shown in Eq. (56) we can have (only) one passing order under node $r_{l,i'}$ and let $\tau' = \tau$, then $T_{r_{l,i'}} = T_{r_{\max(L),u'}} \leq t_{q_n}^{r_{\max(L),u'}}$, where $r_{\max(L),u'} \subset r_{l,i'}$, $q = q_n$ when $|q| = 1$, and $t_{q_n}^{r_{\max(L),u'}} = max\left\{t_{q'}^{r_{l,i'}} + \alpha + b, t_{q}^{m}\right\} + \tau$. Therefore, we have $T_{r_{l,i}} > T_{r_{l,i'}}$ because

$$\begin{cases} t_q^m + \tau + \alpha + b > t_q^m + \tau & if\ t_q^m \geq t_{q'}^{r_{l,i'}} + \alpha + b \\ t_q^{r_{l,i}} + \tau + \alpha + b > t_{q'}^{r_{l,i'}} + \tau + \alpha + b & if\ t_q^m < t_{q'}^{r_{l,i'}} + \alpha + b \end{cases} \quad (65)$$

such that node $r_{l,i}$ can be dominated by node $r_{l,i'}$ based on **Definition 6.7** and **Lemma 6.2**. Note that $T_{r_{l,i'}} < max\left\{t_{q'}^{r_{l,i'}} + \alpha + b, t_q^m\right\} + \tau$ when $\tau' = t'_{q_n} - t_q^{r_{l+1,j}} < \tau$ in the optimal solution of node $r_{l,i'}$ and when $t_q^m < t_{q'}^{r_{l,i'}} + \alpha + b$ and $|q| > 1$.

The proof is completed. ■

Remark 6.6. W.l.o.g., any two nodes are comparable as long as **Theorem 6.1**or **Theorem 6.2** is met, then one of them can be dominated.

6.2.4 Optimal platooning

In this section, a property that defines when a platoon of vehicles can be dynamically batched in the optimal solution of the scheduling problem is first derived based on **Theorem 6.2**. Based on the property, an optimal platooning property is further derived in order to reduce the time complexity of the scheduling problem. Moreover, regarding measuring the time complexity of the proposed algorithm, the total number of nodes in $\mathbb{R}$ is analytically given with and without considering the optimal platooning property.

Before introducing the properties, a platoon of vehicles in the scheduling problem is defined as follows. If vehicles $\{q, \dots, q_n\} = \mathbb{Q}$ are a platoon, then

$$t_{q_n}^{m,c} - t_q^{r_{l,i}} \leq \langle q, q_n \rangle (\alpha - c) \quad (66)$$

where $\{q, \dots, q_n\} \subseteq Q_p$, $l \leq \max(L) - 1$, $i \in Z_l$, $\langle q, q_n \rangle \geq 1$ and $t_{q_n}^{m,c}$ and $t_q^{r_{l,i}}$ are respectively defined in Eq. (69) and (48). Followingly, we give how vehicles can be dynamically batched in the optimal solution under a node $r_{l,i}$.

Proposition 6.3. *If* $\{q, \dots, q_n\} = \mathbb{Q}$, $\{q, \dots, q_n\}$ *must be batched as a platoon in* $T_{r_{l,i}}$, *where* $q \perp r_{l,i}$.

Proof. If $\mathbb{C}_{r_{l-1,\bar{i}}} = \{q, q'\}$, then $\mathbb{C}_{r_{l,i}} = \{q', q_1\}$ based on **Definition 6.5**, where $q \prec q_1$, $q_1 \in \mathbb{Q}$, $r_{l,i} \subset r_{l-1,\bar{i}}$ and $l - 1 \geq 0$. We have $t_{q'}^{r_{l+1,j}} = max\left\{t_q^{r_{l,i}} + \alpha + b, t_{q'}^{m,c}\right\}$ and $t_{q_1}^{r_{l+1,j'}} = t_q^{r_{l,i}} + \alpha - c \geq t_{q_1}^{m,c}$, where $r_{l+1,j} \subset r_{l,i}$, $r_{l+1,j'} \subset r_{l,i}$ and $j \neq j'$. Because $t_{q'}^{r_{l+1,j}} \geq t_{q_1}^{r_{l+1,j'}} + b + c$, node $r_{l+1,j}$ can be dominated by node $r_{l+1,j'}$ based on **Theorem 6.2**. For conflict pairs

$$\left\{\mathbb{C}_{r_{l',i'}}\right\} = \left\{\{q', q_n\} \middle| \forall q_n \in \mathbb{Q} \backslash q, l' = \overrightarrow{q_n(Q_p)} - 1 + \overrightarrow{q'(\bar{Q}_p)} - 1, i' \in Z_{l'}\right\} \quad (67)$$

Theorem 6.2 always holds true such that there is only one route from level $l = \overrightarrow{q(Q_p)} + \overrightarrow{q'(\bar{Q}_p)} - 1$ to level $l' = \overrightarrow{q(Q_p)} + \langle q, q_n \rangle + \overrightarrow{q'(\bar{Q}_p)} - 1$ under node $r_{l,i}$. In addition to Eq. (66), for node $\left\{r_{l,i} \middle| q \perp r_{l,i}, \mathbb{C}_{r_{l,i}} = \emptyset, l = \overrightarrow{q(Q_p)} + \left|\bar{Q}_p\right|, i \in Z_l\right\}$, $\{q, \dots, q_n\} = \mathbb{Q}$ is still batched as a platoon under node $r_{l,i}$ by definition Eq. (66). Therefore, $\{q, \dots, q_n\}$ is batched as a platoon in $T_{r_{l,i}}$. ■

Remark 6.7. **Proposition 6.3** indicates a way to *dynamically* batch a platoon during searching the optimal solution of the scheduling problem because $t_q^{r_{l,i}}$ is a *variable* in Eq. (66). However, the $T_{r_{l,i}}$, where the $\mathbb{Q}$ is batched, is not strictly equal to T, i.e., the $\mathbb{Q}$ may be not batched as a platoon in T.

We then followingly give a rigorous proof when a $\mathbb{Q}$ must be a platoon in the optimal solution T. A new definition of the $\mathbb{Q}$, denoted as $\mathbb{Q}'$, is first given as follows

$$t_{q_n}^{m,c} - t_q^{m,c} = \langle q, q_n \rangle (\alpha - c) \quad (68)$$

where $\{q, \dots, q_n\} = \mathbb{Q}' \subseteq Q_p$ and $\langle q, q_n \rangle \geq 1$. Further, $t_{q_n}^{m,c}$ is defined as follows, where $\forall q_n \in \mathbb{A}$.

$$t_{q_n}^{m,c} = \max \begin{cases} \left(t_{q_n}^{m}, t^o + \alpha + b + \left(\overrightarrow{q_n(Q_p)} - 1\right)(\alpha - c)\right) & if\ o \in \bar{Q}_p, q_n \in Q_p \\ \left(t_{q_n}^{m}, t^o + \overrightarrow{q_n(Q_p)}(\alpha - c)\right) & if\ \{o, q_n\} \in Q_p \end{cases} \quad (69)$$

Note that $t_{q_n}^{m,c} \geq t_{q_n}^{m}$ because of t^o.

Proposition 6.4. *If* $\{q, \dots, q_n\} = \mathbb{Q}'$, $\{q, \dots, q_n\}$ *must be batched as a platoon in* T.

Proof. Since $t_q^{m,c} \leq t_q^{r_{l,i}}$ and $t_{q_n}^{m,c} - t_q^{r_{l,i}} \leq t_{q_n}^{m,c} - t_q^{m,c} = \langle q, q_n \rangle(\alpha - c) = t_{q_n}^{r_{l+\langle q,q_n \rangle,j}} - t_q^{r_{l,i}}$, where $1 \leq l \leq \max(L) - 1$, $i \in Z_l$ and $\langle q, q_n \rangle \geq 1$, $\{q, \dots, q_n\} = \mathbb{Q}'$ is batched as a platoon in $\left\{T_{r_{l,i}} \middle| 1 \leq l \leq \max(L) - 1, i \in Z_l\right\}$ based on **Proposition 6.3** and **Theorem 6.2**. Since $T = min\left\{T_{r_{l,i}} \middle| i \in Z_l, l \in L\right\}$, $\{q, \dots, q_n\}$ is batched as a platoon in T; whereas vehicles are not ordered at level $l = 0$ and are all ordered at level $l = \max(L)$, besides,

$$\begin{aligned} T = T_{r_{0,1}} &= min\left\{t_{\varphi}^{r_{\max(L),i}} \middle| \varphi \perp r_{\max(L),i}, i \in Z_{\max(L)}\right\} \\ &= min\left\{T_{r_{l,i}} \middle| 1 \leq l \leq \max(L) - 1, i \in Z_l\right\} \end{aligned} \quad (70)$$

Therefore, the proof is complete. ■

Based on **Proposition 6.4**, the optimal platooning property can be extended to more complex scenarios, where one or more $\mathbb{Q}'$ occur at both Q_p and $\bar{Q}_p$, such that the time complexity can be further reduced. Before measuring the time complexity, the total number of nodes in $\mathbb{R}$ is first given without considering **Proposition 6.4**.

Theorem 6.3. *Based on* ***Theorem 6.1***,

$$|\mathbb{R}| = 2\left|Q_p\right|\left|\bar{Q}_p\right| + \left|Q_p\right| + \left|\bar{Q}_p\right| + 1 \tag{71}$$

Proof. W.l.o.g., if $\left|Q_p\right| > \left|\bar{Q}_p\right|$, then

$$|Z_l| = \begin{cases} 1 & if\ l = 0 \\ \sum_{l=1}^{|\bar{Q}_p|}[2(l-1)+2] & if\ 1 \le l \le \left|\bar{Q}_p\right| \\ \sum_{l=|\bar{Q}_p|+1}^{|Q_p|}(2\left|\bar{Q}_p\right|+1) & if\ \left|\bar{Q}_p\right|+1 \le l \le \left|Q_p\right| \\ \sum_{l=|Q_p|+1}^{|\bar{Q}_p|+|Q_p|} 2(\left|\bar{Q}_p\right| + \left|Q_p\right| - l + 1) & if\ \left|Q_p\right|+1 \le l \le \left|\bar{Q}_p\right| + \left|Q_p\right| \end{cases} \tag{72}$$

The detailed derivation of Eq. (72) can be referred to Pei et al. (2019). Therefore, $|\mathbb{R}| = 2\left|Q_p\right|\left|\bar{Q}_p\right| + \left|Q_p\right| + \left|\bar{Q}_p\right| + 1$ by summing $|Z_l|$ at each level $l \in L$. Furthermore, Eq. (71) still holds true when $\left|Q_p\right| = \left|\bar{Q}_p\right|$ because number of terms $\left|Q_p\right| + 1 - \left|\bar{Q}_p\right| - 1 = 0$ for the third row in Eq. (72). ■

From **Theorem 6.3**, we can further obtain the number of nodes in $\mathbb{R}$ with considering the optimal platooning property.

Theorem 6.4. *Based on* ***Proposition 6.4*** *and* ***Theorem 6.1***,

$$|\mathbb{R}| = 2\left|Q_p\right|\left|\bar{Q}_p\right| + \left|Q_p\right| + \left|\bar{Q}_p\right| + 1 - \sum_{n\in\mathbb{N},\mathbb{Q}'_n\subseteq Q_p}(|\mathbb{Q}'_n|-1)\left|\bar{Q}_p\right| - \sum_{m\in\mathbb{M},\mathbb{Q}'_m\subseteq \bar{Q}_p}(|\mathbb{Q}'_m|-1)\left|Q_p\right| \tag{73}$$

where $\mathbb{Q}'_n$ *denotes* n*-th platoon of platoons* $\mathbb{N}$ *from* Q_p *and* $\mathbb{Q}'_m$ *denotes* m*-th platoon of platoons* $\mathbb{M}$ *from* $\bar{Q}_p$.

Proof. Based on **Proposition 6.3**, conflict pairs for $\mathbb{Q}'_n$ on which **Proposition 6.4** can be applied are

$$\begin{aligned}&\{\mathbb{C}_{r_{l,i}}\}_n\\&=\left\{\{q',q_n\}\middle|\forall q'\in\bar{Q}_p,\forall q_n\in\mathbb{Q}'_n\backslash q,l=\overrightarrow{q_n(Q_p)}-1+\overrightarrow{q'(\bar{Q}_p)}-1,i\in Z_l\right\}\end{aligned}\tag{74}$$

where $\mathbb{Q}'_n=\{q,\dots,q_n\}$ is defined in Eq. (68). As shown in Eq. (74), $\left|\{\mathbb{C}_{r_{l,i}}\}_n\right|=(|\mathbb{Q}'_n|-1)\left|\bar{Q}_p\right|$, i.e., $\left|\{\mathbb{C}_{r_{l,i}}\}_n\right|$ nodes for $\mathbb{Q}'_n$ can be dominated/discarded based on **Theorem 6.2**. Eq. (74) can also be applied on other platoons of platoons $\mathbb{N}$ from Q_p, therefore, we have

$$\left|\{\mathbb{C}_{r_{l,i}}\}\right|=\sum_{n\in\mathbb{N}}\left|\{\mathbb{C}_{r_{l,i}}\}_n\right|=\sum_{n\in\mathbb{N},\mathbb{Q}'_n\subseteq Q_p}(|\mathbb{Q}'_n|-1)\left|\bar{Q}_p\right|\tag{75}$$

Note that $\mathbb{Q}'_\eta\cup\mathbb{Q}'_{\eta'}\neq\mathbb{Q}'$, where $\forall\{\eta,\eta'\}\in\mathbb{N}$ or $\mathbb{M}$ and $\eta\neq\eta'$. Similarly, we can obtain that $\sum_{m\in\mathbb{M}}\left|\{\mathbb{C}_{r_{l,i}}\}_m\right|=\sum_{m\in\mathbb{M},\mathbb{Q}'_m\subseteq\bar{Q}_p}(|\mathbb{Q}'_m|-1)\left|Q_p\right|$ for $\mathbb{Q}'_m\subseteq\bar{Q}_p$ and $m\in\mathbb{M}$. Since **Proposition 6.4** is applied upon **Theorem 6.1**, $\sum_{n\in\mathbb{N},\mathbb{Q}'_n\subseteq Q_p}(|\mathbb{Q}'_n|-1)\left|\bar{Q}_p\right|+\sum_{m\in\mathbb{M},\mathbb{Q}'_m\subseteq\bar{Q}_p}(|\mathbb{Q}'_m|-1)\left|Q_p\right|$ nodes can be further reduced from **Theorem 6.3**. ■

Remark 6.8. Since **Theorem 6.2** can be applied on any conflict pairs within $\mathbb{R}$, $|\mathbb{R}|$ can be further reduced from Eq. (73) if **Theorem 6.2** can be applied on conflict pairs other than those conflict pairs on which **Proposition 6.4** can be applied. The availability of **Theorem 6.2** depends on real (simulation) data.

It can be easily seen that Eq. (73) is a monotonically decreasing function with respect to the number of vehicles in platoons since $\left|Q_p\right|$ and $\left|\bar{Q}_p\right|$ are constants at each optimization

cycle. Therefore, we can obtain the lower bound of the number of nodes in $\mathbb{R}$ based on **Theorem 6.4**.

Corollary 6.2. *The lower bound of* $|\mathbb{R}|$, $|\mathbb{R}|_{LB}$, *is*

$$|\mathbb{R}|_{LB} = 2\left|Q_p\right| + 2\left|\bar{Q}_p\right| + 1 \tag{76}$$

Proof. Since Eq. (73) is a monotonically decreasing function, besides, $\max_{n\in\mathbb{N},\mathbb{Q}'_n\subseteq Q_p} \sum|\mathbb{Q}'_n| = \left|Q_p\right|$ and $\max_{m\in\mathbb{M},\mathbb{Q}'_m\subseteq \bar{Q}_p} \sum|\mathbb{Q}'_m| = \left|\bar{Q}_p\right|$, we then obtain the $|\mathbb{R}|_{LB}$ as follows

$$\begin{aligned} |\mathbb{R}|_{LB} = 2\left|Q_p\right|\left|\bar{Q}_p\right| + \left|Q_p\right| + \left|\bar{Q}_p\right| + 1 - \left(\left|Q_p\right| - 1\right)\left|\bar{Q}_p\right| \\ - \left(\left|\bar{Q}_p\right| - 1\right)\left|Q_p\right| = 2\left|Q_p\right| + 2\left|\bar{Q}_p\right| + 1 \end{aligned} \tag{77}$$

■

Remark 6.9. **Corollary 6.2** indicates a lower bound of $|\mathbb{R}|$ when only **Proposition 6.4** is applied; as illustrated in **Remark 6.8**, there is only one conflict pair on which **Theorem 6.2** does not apply. However, if in the conflict pair, $\mathbb{C}_{r_{0,1}} = \left\{q, q' \middle| \overrightarrow{q(Q_p)} = 1, \overrightarrow{q'(\bar{Q}_p)} = 1\right\}$, w.l.o.g., $t_q^{r_{1,i}} - t_{q'}^{r_{1,i'}} \geq b + c$, then the lower bound of $|\mathbb{R}|$ can be further reduced as $|\mathbb{R}|_{LB} = \left|Q_p\right| + \left|\bar{Q}_p\right| + 1$ because any child nodes under node $r_{1,i}$ are discarded based on **Theorem 6.2**, where $i \neq i'$.

6.2.5 Summary

Based on the above subsections that define the properties of the scheduling problem, the upper bound of $|\mathbb{R}|$ is strictly proven and given in Eq. (71), which is significantly reduced than nodes that commercial solvers find using a general branch-and-bound algorithm (Xu et al., 2021). Furthermore, the lower bound of $|\mathbb{R}|$ is rigorously given in Eq. (76) based on **Theorem 6.2** and **Proposition 6.4**. Since $|\mathbb{R}|$ directly determines the time complexity and computation time of the scheduling problem (Pei et al., 2019), the time complexity analysis

is given in the next section along with the algorithm proposed in this paper. It is also remarkably noted that the time complexity of an algorithm is often used to measure a worst case scenario, however, the computation time can be greatly decreased if the platooning strategy is leveraged given that platooning is a promising strategy and quite observed at an intersection (Lioris et al., 2017).

6.3 Algorithm Design and Analysis

In this section, a MILP model is first proposed to formulate the scheduling problem proposed in the above section. The MILP conforms to the identified problem properties and can be solved via the proposed algorithm. Further, the proposed algorithm that utilizes the problem properties is given in detail. Finally, analyses of time complexity as well as computation time of the algorithm are specified.

6.3.1 Model formulation

A general reservation-based intersection where the scheduling problem occurs is shown in ***Figure 20***. As indicated earlier, the objective of the scheduling problem $\boldsymbol{\mathcal{P}}$ is to minimize the maximum arrival time of all CAVs at the intersection, which is illustrated in Eq. (78). The details of the MILP model are explained as follows.

$$\boldsymbol{\mathcal{P}} = \min_{\forall q \in \mathbb{A}} \max t_q \tag{78}$$

$$t_q \geq t_q^{m,c} \geq t_q^m \quad \forall q \in \mathbb{A} \tag{79}$$

$$t_q \geq t_{\check{q}} + s \quad \forall q \in \mathbb{A}, \check{q} \prec q \tag{80}$$

$$s \geq s_t + \frac{s_d}{v_q} \quad \forall q \in \mathbb{A} \tag{81}$$

$$v_q \equiv v_{max} \quad \forall q \in \mathbb{A} \tag{82}$$

$$t_q + \mathrm{M}(1 - \delta_{q,q'}) \geq t_{q'} + \alpha + b \quad \forall \left\{ \{q, q'\} \middle| q \in Q_p, q' \in \bar{Q}_p \right\} \in \mathbb{C} \tag{83}$$

$$t_{q'} + \mathrm{M}\delta_{q,q'} \geq t_q + \alpha + b \quad \forall \left\{\{q, q'\} \middle| q \in Q_p, q' \in \bar{Q}_p\right\} \in \mathbb{C} \tag{84}$$

As shown in Eq. (79), the assigned arrival time t_q of a CAV is larger than or equal to $t_q^{m,c}$, which is given in Eq. (69). Further, $t_q^{m,c}$ may be more than or equal to t_q^m given t^o at last optimization cycle, where t_q^m is subject to the communication range and speed constraints. The details of the derivation of t_q^m can be referred to Yu et al. (2018) and Ma and Li (2022a). Further, the difference of arrival time of any pair of a leading and a following vehicle from a same approach should be bounded by the saturation headway, s, as shown in Eq. (80). The saturation headway, s, is also bounded by a safety time headway (Newell, 2002), which is shown in Eq. (81), where it is a sum of a static time headway, s_t, and a dynamic time headway, $\frac{s_d}{v_q}$. s_t and s_d are two given constants. Note that in order to achieve a higher intersection efficiency (Ma and Li, 2022a), the arrival speed of CAVs at the intersection $v_q \equiv v_{max}$, where v_{max} is the maximum allowable speed. As shown in Eq. (83) and (84), the difference of arrival times between any conflict pair is bounded by $\alpha + b$, i.e., if vehicle q enters the intersection before q', $\delta_{q,q'} = 0$, otherwise $\delta_{q,q'} = 1$, where $\delta_{q,q'}$ is a binary variable and M is a positive and sufficiently large number. Note that $|\mathbb{C}| = |Q_p||\bar{Q}_p|$, i.e., there is at most $|Q_p||\bar{Q}_p|$ binary variables for the scheduling problem, and the number of the binary variables determines the time complexity, as does $|\mathbb{R}|$.

6.3.2 Dynamic programming algorithm

The proposed MILP model $\boldsymbol{\mathcal{P}}$ can be solved to global optimal solutions via dynamic programming (Chen et al., 2019; Pei et al., 2019), branch & bound (Chen and Li, 2021; Li and Zhou, 2017) or branch & cut (Cordeau, 2006). Dynamic programming algorithm that leverages the identified problem properties in addition to **Theorem 6.1** and **Theorem 6.2**

is utilized in this paper. Specifically, compared to B&B/C algorithms, the lower bound derivation during branching process of branch & bound algorithm is replaced by a state transition function based on **Proposition 6.2**; further, the upper bound derivation by the branch & bound algorithm is converged at level max (L) of $\mathbb{R}$ based on **Definition 6.6**. The details of the dynamic programming that incorporates the properties of conflict order, overlapping subproblems and optimal platooning are expanded as follows.

Before introducing the customized DP, all sets and variables that are utilized by the DP are defined as follows.

$$\mathbb{X}_{Q_p}^0 = \left\{ x_q^0 \middle| q \in Q_p, q \prec \varphi, 1 \le \overrightarrow{q(Q_p)} \le |Q_p| \right\} \tag{85}$$

$$\mathbb{T}_{Q_p}^m = \left\{ t_q^m \middle| q \in Q_p, q \prec \varphi, 1 \le \overrightarrow{q(Q_p)} \le |Q_p| \right\} \tag{86}$$

$$\mathbb{D}_{Q_p} = \left\{ d_q \middle| 1 \le d_q \le |Q_p|, q \in Q_p, q \prec \varphi, d_\varphi = d_q + 1 \right\} \tag{87}$$

$$\mathbb{X}_{\bar{Q}_p}^0 = \left\{ x_{q'}^0 \middle| q' \in \bar{Q}_p, q' \prec \varphi', 1 \le \overrightarrow{q'(\bar{Q}_p)} \le |\bar{Q}_p| \right\} \tag{88}$$

$$\mathbb{T}_{\bar{Q}_p}^m = \left\{ t_{q'}^m \middle| q' \in \bar{Q}_p, q' \prec \varphi', 1 \le \overrightarrow{q'(\bar{Q}_p)} \le |\bar{Q}_p| \right\} \tag{89}$$

$$\mathbb{D}_{\bar{Q}_p} = \left\{ d_{q'} \middle| |Q_p| + 1 \le d_{q'} \le |Q_p| + |\bar{Q}_p|, q' \in \bar{Q}_p, q' \prec \varphi', d_{\varphi'} = d_{q'} + 1 \right\} \tag{90}$$

$$\overline{\mathbb{R}} = \left\{ \mathbb{r}_\xi := \left\{ \overrightarrow{q(Q_p)}_\xi, \overrightarrow{q'(\bar{Q}_p)}_\xi, \delta_\xi \right\} \middle| 1 \le \xi \le |\overline{\mathbb{R}}|, \delta_\xi = 1 \; if \; q \perp r_{l,\bar{\imath}}, \delta_\xi = 0 \; if \; q' \perp r_{l,\bar{\imath}} \right\} \tag{91}$$

$$\underline{\mathbb{R}} = \left\{ \mathbb{r}_{\xi'} := \left\{ \overrightarrow{\varphi(Q_p)}_{\xi'}, \overrightarrow{\varphi'(\bar{Q}_p)}_{\xi'}, \delta_{\xi'} \right\} \middle| 1 \le \xi' \le |\underline{\mathbb{R}}|, \delta_{\xi'} = 1 \; if \; \varphi \perp r_{l+1,i}, \delta_{\xi'} = 0 \; if \; \varphi' \perp r_{l+1,i}, r_{l+1,i} \subset r_{l,\bar{\imath}} \right\} \tag{92}$$

$$\mathbb{D} = \left\{ \mathbb{d}_{\xi'} \subseteq \mathbb{D}_{\bar{Q}_p} \cup \mathbb{D}_{Q_p} \middle| 1 \le \xi' \le |\mathbb{D}| = |\underline{\mathbb{R}}| \right\} \tag{93}$$

$$\mathbb{D}' = \left\{ \mathbb{d}'_{\xi} \subseteq \mathbb{D}_{\bar{Q}_p} \cup \mathbb{D}_{Q_p} \middle| 1 \le \xi \le |\mathbb{D}'| = |\overline{\mathbb{R}}| \right\} \tag{94}$$

$$\widetilde{\mathbb{R}}^{q} = \widetilde{\mathbb{R}}^{q'} = \mathbb{D}^{q} = \mathbb{D}^{q'} = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix}_{(|Q_p|+1)(|\bar{Q}_p|+1)} \tag{95}$$

$$\mathbb{T}^{q} = \mathbb{T}^{q'} = \begin{bmatrix} -999 & 999 & \cdots & 999 \\ 999 & 999 & \cdots & 999 \\ \vdots & \vdots & \ddots & \vdots \\ 999 & 999 & \cdots & 999 \end{bmatrix}_{(|Q_p|+1)(|\bar{Q}_p|+1)} \tag{96}$$

Eq. (85)-(87) defines initial entry location at the communication range, minimum arrival time at the intersection and unique IDs for $q \in Q_p$; whereas Eq. (88)-(90) is defined correspondingly for $q' \in \bar{Q}_p$. Eq. (91) denotes parent nodes at level $l \in L$ and an indicator of which approach a CAV is currently coming/ordered at each parent node; whereas Eq. (92) denotes child nodes of the parent nodes correspondingly. Eq. (95) and (94) respectively denotes a partially/currently ordered set of CAVs of parent nodes and child nodes. In Eq. (95) and (96), respectively for $q \perp r_{l,i}$ or $q' \perp r_{l,i}$, where $q \in Q_p$ and $q' \in \bar{Q}_p$, $\widetilde{\mathbb{R}}^{q}$ and $\widetilde{\mathbb{R}}^{q'}$ denotes an indicator to utilize **Theorem 6.1**, $\mathbb{D}^{q}$ and $\mathbb{D}^{q'}$ denotes a sequence of each partially ordered set in child node lists and $\mathbb{T}^{q}$ and $\mathbb{T}^{q'}$ denotes $t_q^{r_{l,i}}$ of each child node $r_{l,i}$, $\forall i \in Z_l$, $\forall l \in L$, $\forall q \in \mathbb{A}$. Note that in Eq. (95) and (96) the number of elements is $(|Q_p| + 1)(|\bar{Q}_p| + 1)$ in total, where $\widetilde{\mathbb{R}}_{x,y}^{q} \in \widetilde{\mathbb{R}}^{q}$, $\widetilde{\mathbb{R}}_{x,y}^{q'} \in \widetilde{\mathbb{R}}^{q'}$, $\mathbb{D}_{x,y}^{q} \in \mathbb{D}^{q}$, $\mathbb{D}_{x,y}^{q'} \in \mathbb{D}^{q'}$, $\mathbb{T}_{x,y}^{q} \in \mathbb{T}^{q}$ and $\mathbb{T}_{x,y}^{q'} \in \mathbb{T}^{q'}$, and numbering of all elements starts from zero, i.e., $\min(x) = \min(y) = 0$, where $(0,0)$ denotes the root node $r_{0,1}$. In addition, numbering of row denotes $\overrightarrow{q(Q_p)}$ and column denotes $\overrightarrow{q'(\bar{Q}_p)}$, i.e., $\max(x) = |Q_p|$ and $\max(y) = |\bar{Q}_p|$. Followingly, the DP is designed to solve the model $\boldsymbol{\mathcal{P}}$ in **Algorithm 6.1**, **Algorithm 6.2** and **Algorithm 6.3**.

Before measuring the time complexity of the DP algorithm, there are *three* main procedures to find the T and the optimal passing order, which are *get* $t_q^{r_{l,i}}$ for each child node, *add* a child node to $\underline{\mathbb{R}}$ and *add* a partially/currently ordered set to $\mathbb{D}$; i.e., for each parent node in $\overline{\mathbb{R}}$, it costs *at worst three* unit/constant computation times to find one child node in the DP algorithm. Note here that a breadth-first search (BFS) method is utilized in the DP.

Algorithm 6.1. **The customized dynamic programming algorithm**

//**Initiate**
$\mathbb{D} = \{\emptyset\}$; $\mathbb{D}' = \{\emptyset\}$; $\overline{\mathbb{R}} = \{\mathbb{r}_1 = \{0,0,\emptyset\}\}$; $\underline{\mathbb{R}} = \{\emptyset\}$
$t_q^{m,c} \leftarrow t_q^m$ given t^o, $\forall q \in \mathbb{A}$ based on Eq. (69)
$t_q^m \leftarrow t_q^{m,c}$, $\forall q \in \mathbb{A}$
//**Start branching**
while $|\mathbb{d}_1'| < |Q_p| + |\bar{Q}_p|$
  for $\mathbb{r}_\xi \in \overline{\mathbb{R}}$
    if $\overrightarrow{q(Q_p)_\xi} = |Q_p|$
      get $t_{\varphi'}^{r_{l+1,i}}$ given $t_{\varphi'}^m$, $\mathbb{T}^q_{\overrightarrow{q(Q_p)_\xi},\overrightarrow{q'(\bar{Q}_p)_\xi}}$, $\mathbb{T}^{q'}_{\overrightarrow{q(Q_p)_\xi},\overrightarrow{q'(\bar{Q}_p)_\xi}}$ and δ_ξ based on Eq. (48)
      add a child node to $\underline{\mathbb{R}}$ and **add** a partially passing order to $\mathbb{D}$ based on **Algorithm 6.2**
    end
    if $\overrightarrow{q'(\bar{Q}_p)_\xi} = |\bar{Q}_p|$
      get $t_{\varphi}^{r_{l+1,i}}$ given t_{φ}^m, $\mathbb{T}^q_{\overrightarrow{q(Q_p)_\xi},\overrightarrow{q'(\bar{Q}_p)_\xi}}$, $\mathbb{T}^{q'}_{\overrightarrow{q(Q_p)_\xi},\overrightarrow{q'(\bar{Q}_p)_\xi}}$ and δ_ξ based on Eq. (48)
      add a child node to $\underline{\mathbb{R}}$ and **add** a partially passing order to $\mathbb{D}$ based on **Algorithm 6.3**
    end
    if $\overrightarrow{q(Q_p)_\xi} < |Q_p|$ & $\overrightarrow{q'(\bar{Q}_p)_\xi} < |\bar{Q}_p|$
      get $t_{\varphi'}^{r_{l+1,i'}}$ given $t_{\varphi'}^m$, $\mathbb{T}^q_{\overrightarrow{q(Q_p)_\xi},\overrightarrow{q'(\bar{Q}_p)_\xi}}$, $\mathbb{T}^{q'}_{\overrightarrow{q(Q_p)_\xi},\overrightarrow{q'(\bar{Q}_p)_\xi}}$ and δ_ξ based on Eq. (48)
      get $t_{\varphi}^{r_{l+1,i}}$ given t_{φ}^m, $\mathbb{T}^q_{\overrightarrow{q(Q_p)_\xi},\overrightarrow{q'(\bar{Q}_p)_\xi}}$, $\mathbb{T}^{q'}_{\overrightarrow{q(Q_p)_\xi},\overrightarrow{q'(\bar{Q}_p)_\xi}}$ and δ_ξ based on Eq. (48)
      if $t_{\varphi}^{r_{l+1,i}} - t_{\varphi'}^{r_{l+1,i'}} \geq \boldsymbol{b} + \boldsymbol{c}$
        add a child node to $\underline{\mathbb{R}}$ and **add** a partially passing order to $\mathbb{D}$ based on **Algorithm 6.2**
      else if $t_{\varphi}^{r_{l+1,i}} - t_{\varphi'}^{r_{l+1,i'}} \leq -\boldsymbol{b} - \boldsymbol{c}$
        add a child node to $\underline{\mathbb{R}}$ and **add** a partially passing order to $\mathbb{D}$ based on **Algorithm 6.3**
      else
        add a child node to $\underline{\mathbb{R}}$ and **add** a partially passing order to $\mathbb{D}$ based on **Algorithm 6.2**
        add a child node to $\underline{\mathbb{R}}$ and **add** a partially passing order to $\mathbb{D}$ based on **Algorithm 6.3**
      end
    end
  end
  $\overline{\mathbb{R}} \leftarrow \underline{\mathbb{R}}$
  $\mathbb{D}' \leftarrow \mathbb{D}$
  $\mathbb{D} = \{\emptyset\}$
  $\underline{\mathbb{R}} = \{\emptyset\}$
end
//Output
$\mathbb{T}^q$, $\mathbb{T}^{q'}$ and $\mathbb{D}'$

Algorithm 6.2. Add a child node to $\underline{\mathbb{R}}$ and add a partially passing order to $\mathbb{D}$ if $\boldsymbol{\varphi' \perp r_{l+1,i}}$

1 //**Add a child node to $\underline{\mathbb{R}}$ and add a partially passing order to $\mathbb{D}$ if $\boldsymbol{\varphi' \perp r_{l+1,i}}$**
2 **if** $t_{\varphi'}^{r_{l+1,i}} < \mathbb{T}^{q'}_{\overrightarrow{q(Q_p)_\xi},\overrightarrow{\varphi'(\bar{Q}_p)_\xi}}$
3 $\quad \mathbb{T}^{q'}_{\overrightarrow{q(Q_p)_\xi},\overrightarrow{\varphi'(\bar{Q}_p)_\xi}} \leftarrow t_{\varphi'}^{r_{l+1,i}}$
4 $\quad$ **if** $\widetilde{\mathbb{R}}^{q'}_{\overrightarrow{q(Q_p)_\xi},\overrightarrow{\varphi'(\bar{Q}_p)_\xi}} = 0$
5 $\quad\quad \underline{\mathbb{R}} \leftarrow \left\{\overrightarrow{q(Q_p)_\xi}, \overrightarrow{\varphi'(\bar{Q}_p)_\xi}, 0\right\}$
6 $\quad\quad \mathbb{D} \leftarrow \left\{\mathbb{d}'_\xi, d_{\varphi'}\right\}$
7 $\quad\quad \mathbb{D}^{q'}_{\overrightarrow{q(Q_p)_\xi},\overrightarrow{\varphi'(\bar{Q}_p)_\xi}} \leftarrow |\mathbb{D}|$
8 $\quad\quad \widetilde{\mathbb{R}}^{q'}_{\overrightarrow{q(Q_p)_\xi},\overrightarrow{\varphi'(\bar{Q}_p)_\xi}} \leftarrow 1$
9 $\quad$ **else**
10 $\quad\quad \mathbb{d}_{\mathbb{D}^{q'}_{\overrightarrow{q(Q_p)_\xi},\overrightarrow{\varphi'(\bar{Q}_p)_\xi}}} \leftarrow \left\{\mathbb{d}'_\xi, d_{\varphi'}\right\}$
11 $\quad$ **end**
12 **end**

Algorithm 6.3. Add a child node to $\underline{\mathbb{R}}$ and add a partially passing order to $\mathbb{D}$ if $\boldsymbol{\varphi \perp r_{l+1,i}}$

1 **//Add a child node to $\underline{\mathbb{R}}$ and add a partially passing order to $\mathbb{D}$ if $\boldsymbol{\varphi \perp r_{l+1,i}}$**
2 **if** $t_{\varphi}^{r_{l+1,i}} < \mathbb{T}^{q}_{\overrightarrow{\varphi(Q_p)_\xi},\overrightarrow{q'(\bar{Q}_p)_\xi}}$
3 $\quad \mathbb{T}^{q}_{\overrightarrow{\varphi(Q_p)_\xi},\overrightarrow{q'(\bar{Q}_p)_\xi}} \leftarrow t_{\varphi}^{r_{l+1,i}}$
4 $\quad$ **if** $\widetilde{\mathbb{R}}^{q}_{\overrightarrow{\varphi(Q_p)_\xi},\overrightarrow{q'(\bar{Q}_p)_\xi}} = 0$
5 $\quad\quad \underline{\mathbb{R}} \leftarrow \left\{\overrightarrow{\varphi(Q_p)_\xi}, \overrightarrow{q'(\bar{Q}_p)_\xi}, 1\right\}$
6 $\quad\quad \mathbb{D} \leftarrow \left\{\mathbb{d}'_\xi, d_{\varphi}\right\}$
7 $\quad\quad \mathbb{D}^{q}_{\overrightarrow{\varphi(Q_p)_\xi},\overrightarrow{q'(\bar{Q}_p)_\xi}} \leftarrow |\mathbb{D}|$
8 $\quad\quad \widetilde{\mathbb{R}}^{q}_{\overrightarrow{\varphi(Q_p)_\xi},\overrightarrow{q'(\bar{Q}_p)_\xi}} \leftarrow 1$
9 $\quad$ **else**
10 $\quad\quad \mathbb{d}_{\mathbb{D}^{q}_{\overrightarrow{\varphi(Q_p)_\xi},\overrightarrow{q'(\bar{Q}_p)_\xi}}} \leftarrow \left\{\mathbb{d}'_\xi, d_{\varphi}\right\}$
11 $\quad$ **end**
12 **end**

6.3.3 Time complexity analysis

To measure the time complexity of the DP algorithm, specifically, the state transition function is realized based on **Proposition 6.2** as shown in **Algorithm 6.1** lines (9,13,17,18),

an overlapping subproblem property is realized based on **Theorem 6.1** as shown in line (2) of **Algorithm 6.2** and **Algorithm 6.3**, and last, the other overlapping subproblem property is realized based on **Theorem 6.2** as shown in lines (19,21) in **Algorithm 6.1**. Recall that in **Theorem 6.3**, there are two cases in the branching process where a parent node can only find one child node, as shown in lines (8,12) in **Algorithm 6.1** and can find two child nodes, as shown in line (16) in **Algorithm 6.1**. As indicated earlier, for each branching it costs at worst three unit computation times. Therefore, how many times of branching occur in the branching process is first given as follows.

Theorem 6.5. *Based on **Theorem 6.3**,*

$$\left|\overleftrightarrow{\mathbb{R}}\right| = 4\left|Q_p\right|\left|\bar{Q}_p\right| \tag{97}$$

where $\left|\overleftrightarrow{\mathbb{R}}\right|$ *denotes the total times of branching in the branching process*.

Proof. W.l.o.g., if $\left|Q_p\right| > \left|\bar{Q}_p\right|$, then

$$\left|\overleftrightarrow{\mathbb{R}}_l\right| =$$

$$\begin{cases} 2 & if\ l = 0 \\ \sum_{l=1}^{\left|\bar{Q}_p\right|}[4(l-1)+4] & if\ 1 \le l < \left|\bar{Q}_p\right| \\ 4\left|\bar{Q}_p\right| - 1 & if\ l = \left|\bar{Q}_p\right| \\ \sum_{l=\left|\bar{Q}_p\right|+1}^{\left|Q_p\right|-1} 4\left|\bar{Q}_p\right| & if\ \left|\bar{Q}_p\right| + 1 \le l < \left|Q_p\right| \\ 4\left|\bar{Q}_p\right| - 1 & if\ l = \left|Q_p\right| \\ \sum_{l=\left|Q_p\right|+1}^{\left|\bar{Q}_p\right|+\left|Q_p\right|-1} 4\left(\left|\bar{Q}_p\right| + \left|Q_p\right| - l\right) & if\ \left|Q_p\right| + 1 \le l < \left|Q_p\right| + \left|\bar{Q}_p\right| \\ 0 & if\ l = \left|Q_p\right| + \left|\bar{Q}_p\right| \end{cases} \tag{98}$$

The detailed derivation of Eq. (98) can be referred to Pei et al. (2019). Therefore, $\left|\overleftrightarrow{\mathbb{R}}\right| = 4\left|Q_p\right|\left|\bar{Q}_p\right|$ by summing $\left|\overleftrightarrow{\mathbb{R}}_l\right|$ at each level $l \in L$. Furthermore, Eq. (97) still holds true when $\left|Q_p\right| = \left|\bar{Q}_p\right|$ because number of terms $\left|Q_p\right| - 1 + 1 - \left|\bar{Q}_p\right| - 1 = -1$ for the fourth row in Eq. (98) and the sum of times of branching from third row to fifth row is $4\left|\bar{Q}_p\right| - 2$.■

Given **Theorem 6.5**, we then can obtain the time complexity of the DP algorithm at worst case scenario as follows.

Corollary 6.3. *The time complexity of the DP algorithm is* $\mathcal{O}(\left|Q_p\right|^2)$.

Proof. W.l.o.g., we let $\left|Q_p\right| = \left|\bar{Q}_p\right|$, then the total computation times at worst case scenario is $3\left|\overrightarrow{\mathbb{R}}\right| = 12\left|Q_p\right|^2$ and the time complexity is $\mathcal{O}(\left|Q_p\right|^2)$. ■

Leveraging the optimal platooning property based on **Proposition 6.4** and **Theorem 6.4**, the number of (parent) nodes at each level $l \in L$ can be greatly reduced, in other words, the total times of branching can be greatly reduced, so can the time complexity and computation time. Before analyzing the time complexity and computation time based on the optimal platooning property, we first derive the reduced total times of branching $\left|\overrightarrow{\mathbb{R}}\right|'$ in the branching process.

Theorem 6.6. *Based on* ***Proposition 6.4*** *and* ***Theorem 6.4****,*

$$\begin{aligned}\left|\overrightarrow{\mathbb{R}}\right|' = 4\left|Q_p\right|\left|\bar{Q}_p\right| &- \sum_{n\in\mathbb{N},\mathbb{Q}'_n\subseteq Q_p}\left[2(|\mathbb{Q}'_n|-1)\left(\left|\bar{Q}_p\right|-1\right)+|\mathbb{Q}'_n|-1\right] \\ &- \sum_{m\in\mathbb{M},\mathbb{Q}'_m\subseteq \bar{Q}_p}\left[2(|\mathbb{Q}'_m|-1)\left(\left|Q_p\right|-1\right)+|\mathbb{Q}'_m|-1\right]\end{aligned} \tag{99}$$

Proof. Recall that in Eq. (74) $\forall q' \in \bar{Q}_p$ should be further specified in terms of $\left|\overrightarrow{\mathbb{R}}\right|'$, i.e.,

$$\begin{aligned}&\left\{\mathbb{C}_{r_{l,i}}\right\}_n \\ &= \left\{\{q', q_n\}\cup\{\varphi', q_n\}\middle|\forall q' \in \bar{Q}_p\backslash\varphi', \overrightarrow{\varphi'(\bar{Q}_p)} = \left|\bar{Q}_p\right|, \forall q_n \in \mathbb{Q}'_n\backslash q\right\}\end{aligned} \tag{100}$$

where for the conflict pairs

$$\left\{\mathbb{C}_{r_{l,i}}\right\}_n = \left\{\{\varphi', q_n\}\middle|\forall q_n \in \mathbb{Q}'_n\backslash q, l = \overrightarrow{q_n(Q_p)} - 1 + \left|\bar{Q}_p\right| - 1, i \in Z_l\right\} \tag{101}$$

nodes $\{r_{l,i}\}_n = \left\{r_{l,i} \middle| \varphi' \perp r_{l,i}, l = \overrightarrow{q_n(Q_p)} - 1 + |\bar{Q}_p|, i \in Z_l\right\}$ are dominated/discarded based on **Theorem 6.2**. As indicated in line (12) in **Algorithm 6.1**, only one child node was supposed to be found, however, for nodes $\{r_{l,i}\}_n = \left\{r_{l,i} \middle| \forall q' \in \bar{Q}_p \backslash \varphi', q' \perp r_{l,i}, l = \overrightarrow{q_n(Q_p)} - 1 + \overrightarrow{q'(\bar{Q}_p)}, i \in Z_l\right\}$, there were two child nodes supposed to be found as indicated line (16) in **Algorithm 6.1**. Therefore, for the conflict pairs in Eq. (100) there are $2(|\mathbb{Q}'_n| - 1)\left(|\bar{Q}_p| - 1\right) + |\mathbb{Q}'_n| - 1$ times of branching in total reduced for the platoon $\mathbb{Q}'_n$; then for all the platoons $\mathbb{N}$ from Q_p, $\sum_{n\in\mathbb{N},\mathbb{Q}'_n\subseteq Q_p}\left[2(|\mathbb{Q}'_n| - 1)\left(|\bar{Q}_p| - 1\right) + |\mathbb{Q}'_n| - 1\right]$ times of branching can be reduced. Similarly, $\sum_{m\in\mathbb{M},\mathbb{Q}'_m\subseteq \bar{Q}_p}\left[2(|\mathbb{Q}'_m| - 1)\left(|Q_p| - 1\right) + |\mathbb{Q}'_m| - 1\right]$ times of branching can be reduced for all the platoons $\mathbb{M}$ from $\bar{Q}_p$. Based on **Theorem 6.5**, $|\overleftrightarrow{\mathbb{R}}|'$ is derived in Eq. (99).

The proof is complete. ■

Remark 6.10. Based on **Theorem 6.6**, $3|\overleftrightarrow{\mathbb{R}}| > 3|\overleftrightarrow{\mathbb{R}}|'$, which indicates that the computation time of the DP algorithm based on the optimal platooning is reduced. However, since the platoons $\mathbb{N}$ and $\mathbb{M}$ are dynamically batched at each optimization cycle, the upper bound of the time complexity at worst case scenario, if no platoons are batched, is still $\mathcal{O}(|Q_p|^2)$. Recall **Corollary 6.2** that the lower bound of the time complexity can be achieved when $\sum_{n\in\mathbb{N},\mathbb{Q}'_n\subseteq Q_p}|\mathbb{Q}'_n| = |Q_p|$ and $\sum_{m\in\mathbb{M},\mathbb{Q}'_m\subseteq \bar{Q}_p}|\mathbb{Q}'_m| = |\bar{Q}_p|$.

Corollary 6.4. *The lower bound of the time complexity of the DP algorithm is* $\mathcal{O}(|Q_p|)$.

Proof. Based on **Corollary 6.2**,

$$\begin{aligned}\left|\overleftrightarrow{\mathbb{R}}\right|'_{LB} &\leq 4|Q_p||\bar{Q}_p| - \left[2\left(|Q_p|-1\right)\left(|\bar{Q}_p|-1\right)+|Q_p|-1\right] \\ &\quad - \left[2\left(|\bar{Q}_p|-1\right)\left(|Q_p|-1\right)+|\bar{Q}_p|-1\right] \\ &= 3|Q_p| + 3|\bar{Q}_p| - 2\end{aligned} \tag{102}$$

W.l.o.g., we let $|Q_p| = |\bar{Q}_p|$, then the total computation times $3\left|\overleftrightarrow{\mathbb{R}}\right|'_{LB} \leq 18|Q_p| - 6$ and the lower bound of the time complexity is $\mathcal{O}(|Q_p|)$. ■

6.3.4 Summary

Based on the rigorous derivations at above sections, the time complexity and computation time of the DP algorithm are compared for scenarios with and without the optimal platooning property. It is found that the lower bound of the time complexity is achieved in **Corollary 6.4**, which is reduced to *linear time complexity* from *quadratic time complexity* of **Corollary 6.3**. This significant reduction only needs two additional lines of codes as shown in lines (19,21) in **Algorithm 6.1** as compared to the state-of-art DP algorithm (Pei et al., 2019) without them. It is remarkably noted that this reduction holds true because of **Theorem 6.2** and **Proposition 6.4** and can be achieved when platoons are batched along the communication range at each optimization cycle. Also note that even though the lower bound of the time complexity cannot be achieved, the computation time can be reduced based on **Theorem 6.6** as long as at least one platoon of vehicles are batched, where the number of vehicles in the platoon is at least 2.

6.4 Numerical Evaluation

6.4.1 Control framework

A reservation-based intersection control framework that schedules arrival times of CAVs at the intersection consists of two procedures. As shown in **Figure 20**, the communication range consists of a request zone and an optimization range. The purpose of the request zone

is to reserve all requests from CAVs whose arrival conforms to a (Poisson) distribution at each optimization cycle. By collecting the arrival information of CAVs, Eq. (85) and (88) are used to formulate their initial arrival position at the request zone for each approach. After a given optimization interval or once any CAV arrives a downstream boundary of the request zone, whichever is earlier, an optimization cycle is initiated for all the CAVs at the request zone. This optimization strategy is performed as a *rolling horizon strategy*. Once an optimization cycle is initiated and finished, all the CAVs move into the optimization range as scheduled by a central controller. The t^o is collected from a last CAV that arrives the intersection. Then the t^o is used for next optimization cycle as indicated in Eq. (69).

Note that once CAVs are collected for an optimization cycle, their minimal and maximal arrival time at the intersection can be obtained, depending on their initial speed and position. For simplicity and focusing on theoretical analyses of the scheduling problem and algorithm, it is assumed that the maximal arrival time for all CAVs is $+\infty$. Furthermore, we assume the initial speed for all CAVs is same; this is a relatively *strong* assumption, however, we leave the relaxation of the assumption to our future research. Therefore, the minimal arrival time only depends on the initial position of CAVs. Since the arrival speed of CAVs at the intersection is given as the maximal allowable speed as indicated in Eq. (82), the arrival time of CAVs is subject to their minimal arrival time and their leading vehicle as indicated in Eq. (79) and (80). Overall, detailed derivations of the minimal and maximal arrival time of CAVs can be referred to (Feng et al., 2018; Ma and Li, 2022a; Yu et al., 2018).

6.4.2 Numerical examples

To validate the time complexity and computation time conclusions of **Theorem 6.6** and **Corollary 6.4**, some numerical examples are formulated as follows. In the meantime, the state-of-art DP algorithm based on **Theorem 6.5** and **Corollary 6.3** is compared with the expedited DP algorithm proposed in this paper based on **Theorem 6.2** and **Proposition 6.4** in terms of the time complexity and computation time. Specifically, note again that the comparison only depends on the lines (19,21) in **Algorithm 6.1**.

Before giving the numerical comparisons, CAVs are initiated based on Eq. (86) and (89). As indicated in Section 6.4.1, Eq. (86) and (89) can be numerically derived with constraints of Eq. (85) and (88). For simplicity, we directly initialize Eq. (86) and (89) as follows.

$$\mathbb{T}_{Q_p}^m = \{10,10.5,14\} \tag{103}$$

$$\mathbb{T}_{\bar{Q}_p}^m = \{11,13.5,14\} \tag{104}$$

Remark 6.11. Note that the decision variables $\boldsymbol{\alpha}$, $\boldsymbol{b}$ and $\boldsymbol{c}$ are not subject to any other constraints than the arrival speed v_q, $\forall q \in \mathbb{A}$ and the intersection length as mentioned by **Assumption 6.1**. Especially for $\boldsymbol{b}$ and $\boldsymbol{c}$, any values hold true as long as the weak **Assumption 6.3** is met.

In the examples, we define that $\alpha = 2$, $b = 1$ and $c = 1.5$. Simply, the unit of the decision variables is second (s). W.l.o.g., we let $t^o = -\infty$ such that $t_q^{m,c} = t_q^m$, $\forall q \in \mathbb{A}$. Therefore, it can be seen that $\mathbb{D}_{Q_p} = \left\{\{1,2,3\}\middle|\{q_1, q_2\} = \mathbb{Q}'_{Q_p}\right\}$ based on Eq. (103) and $\mathbb{D}_{\bar{Q}_p} = \left\{\{4,5,6\}\middle|\{q'_2, q'_3\} = \mathbb{Q}'_{\bar{Q}_p}\right\}$ based on Eq. (104). Then we first calculate the computation time and the optimal solution based on the state-of-art DP algorithm.

As shown in **Figure 22**, the optimal passing order is found via those *green* nodes based on the state-of-art DP algorithm, where $\mathbb{d}_2' = \{1,2,4,5,6,3\}$ and $T = 17.5$ based on **Algorithm 6.1** without executing lines (19,21). As indicated in Eq. (103) and (104), $\{q_1, q_2\} = \mathbb{Q}'_{Q_p}$ and $\{q_2', q_3'\} = \mathbb{Q}'_{\bar{Q}_p}$, however, interestingly, $\{q_1', q_2', q_3'\}$ dynamically batches as a platoon in the optimal solution, which objectively verifies **Proposition 6.3**. Furthermore, $\mathbb{Q}'_{Q_p}$ and $\mathbb{Q}'_{\bar{Q}_p}$ are also in the optimal solution based on **Proposition 6.4**. Overall, the computation time is *108* unit times at worst based on **Theorem 6.5**.

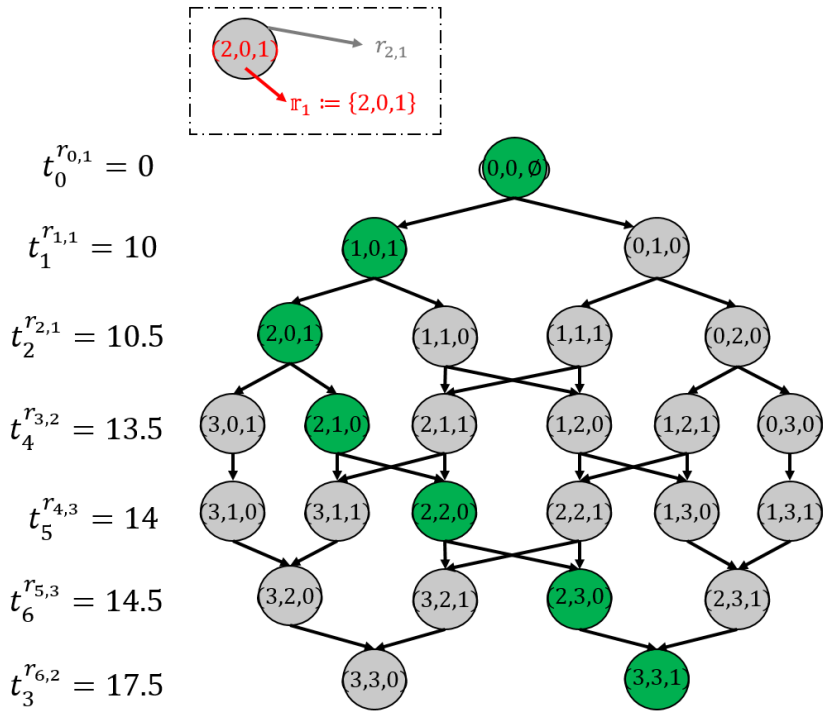
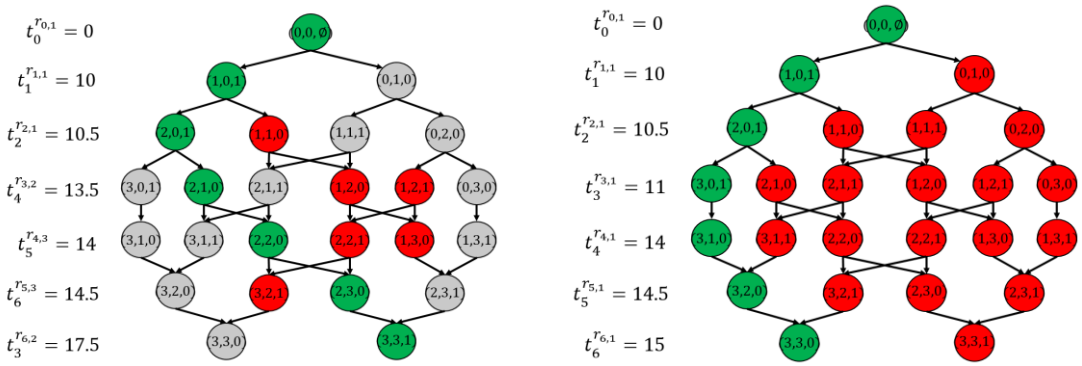


Figure 22 An optimal solution of state-of-art DP algorithm

We then calculate the computation time and the optimal passing order based on the expedited DP algorithm proposed in this paper. In this case, lines (19,21) in **Algorithm 6.1** are executed. It can be seen in **Figure 23** (a) that the $\mathbb{d}_2'$ and T are same as state-of-art DP algorithm, however, the computation time at worst is reduced to *78* unit times based on **Theorem 6.6**, where those *red* nodes are discarded while branching.



(a) (b)

Figure 23 Optimal solutions of expedited DP algorithm based on optimal platooning

Moreover, a special data initialized at a certain optimization cycle as mentioned in **Remark 6.9** is given as follows.

$$\mathbb{T}_{Q_p}^m = \{10,10.5,11\} \tag{105}$$

$$\mathbb{T}_{\bar{Q}_p}^m = \{12.5,13.5,14\} \tag{106}$$

W.l.o.g., again, we let $t_q^{m,c} = t_q^m$, $\forall q \in \mathbb{A}$. In this case, $\mathbb{d}_1' = \{1,2,3,4,5,6\}$ and $T = 15$ based on **Algorithm 6.1** with lines (19,21) executed. As shown in **Figure 23** (b), the computation time at worst is *27* unit times based on **Theorem 6.2**, whereas the computation time at worst is still *108* unit times based on **Theorem 6.1**, where lines (19,21) in **Algorithm 6.1** are not executed. In this case, the computation time can be reduced by 75%.

6.4.3 Summary

From the above numerical calculations, the computation time of algorithms can be directly obtained whether platoons are identified at initializations of each optimization cycle. As mentioned in **Remark 6.8**, the computation time partially depends on the data initialization; as shown in Eq. (105) and (106), the computation time can be greatly reduced. Moreover, the computation time directly depends on the algorithm being executed. Overall, one upper

bound of the time complexity at worst case can be obtained when the **Proposition 6.4** is not considered, whereas one lower bound can be obtained when the **Proposition 6.4** is utilized. Furthermore, the **Proposition 6.4** can be achieved only when the **Theorem 6.2** is implemented in the algorithm. Lastly, it can be concluded that platoons contribute to less computation time in the expedited DP algorithm.

Finally, note that the decision variables α, b and c only impact on the T rather than the $\mathbb{D}'$. The time complexity and computation time of the expedited DP algorithm are not affected by the values of the decision variables.

CHAPTER 7. CONCLUSIONS

7.1 Trajectory Planning at a Reservation-based Intersection

Under the TSIT, the performances of it with/without variable acceleration rate are evaluated in comparison with the Dynamic Batch in terms of average intersection delay. In addition, the performances are also evaluated in terms of computation time between BATCHV, ZONEV, BATCH and ZONE. The trajectory is also analyzed by the pattern of the variation of the acceleration rate among different vehicles under different control strategies. The results are found and summarized as follows:

1. The ZONE performs the best and the BATCHV performs the worst in terms of the computation time, and all the proposed methods under TSIT are not compared with the Dynamic Batch in terms of the computation time since the latter is a rule-based control strategy, which is fast in nature;
2. The TSIT with variable acceleration rate (BATCHV and ZONEV) is better than that (BATCH and ZONE) without it in terms of the average intersection delay; and
3. The sensitivity analyses show that the TSIT with/without variable acceleration rate is insensitive to the varying traffic condition in terms of the average intersection delay but sensitive to the vehicle length, whereas the Dynamic Batch is sensitive to the varying traffic condition, and the Dynamic Batch performs better under low traffic demand condition and with large threshold of batched vehicles.

Correspondingly, the findings of the results are reflected as follows:

1. The TSIT with variable acceleration rate results in better average intersection delay but increases the computational complexity such that the computation time is increased, since that requires more time to search more feasible solutions and find the optimal solution;
2. The TSIT can adapt to varying traffic condition even with extremely high traffic demand because the arrival time and speed, as well as the planned trajectory, can vary over the traffic condition, whereas the Dynamic Batch cannot because the arrival time and speed are predetermined ahead of entering the intersection; and
3. The capacity of the intersection with the assumptions and the parameter settings in this paper is over 14400 veh/h under the TSIT control, such that the TSIT is insensitive to the traffic demand in terms of the average intersection delay.

7.2 Optimal Control Framework for a Reservation-based Intersection

Based on the numerical simulations, it is proved that piecewise trajectory modelling performs better than continuous modelling in terms of improving intersection efficiency. The evaluations are performed through the trajectory, intersection delay and intersection throughput comparison between piecewise and continuous trajectory modelling, and the results are found as follows:

1. The traffic demand is lower than the intersection capacity under both trajectory modeling methods even when the traffic demand is as high as 1,800 veh/h/ln;
2. The average intersection delay of piecewise modelling is significantly lower than that of continuous trajectory modelling under all test scenarios;
3. The average intersection throughput of piecewise modelling is higher than that of the continuous trajectory modelling under all test scenarios;

4. The overall performances of all evaluations under BATCH strategy are slightly better than those under ZONE strategy; and
5. The average intersection delay is increasing, and the average intersection throughput is increasing while the batch number or zone range is increasing.

The findings from the above results are reflected and discussed as follows:

1. The piecewise trajectory modelling finds better solutions than continuous trajectory modelling regarding intersection delay and throughput, because the average service time under piecewise trajectory modelling is less than that under continuous trajectory modelling based on **Corollary 5.1**, such that the capacity/throughput/delay of the intersection under piecewise modelling is better than that under continuous trajectory modelling based on **Corollary 5.2 & 5.3**;
2. **Proposition 5.5** is proved through trajectory, delay and throughput analyses and that $v(t_f) = v_f$ for each CAV in $\boldsymbol{J}(\cdot)$ is also proved via modelling the $v(t_f)$ as a variable in the numerical modelling and simulations; and
3. While the batch number or zone range is increasing under piecewise or continuous trajectory modelling, the average service time is increasing based on **Corollary 5.1** such that the average delay is also increasing based on **Corollary 5.3**, and the average throughput is increasing because the demand is always lower than the capacity.

7.3 Optimal Scheduling of CAVs at a Reservation-based Intersection

An expedited DP algorithm is proposed to reduce time complexity and computation time of a scheduling problem of CAVs at a reservation-based intersection. To achieve the less time complexity and computation time, a deeper property of identifying overlapping

subproblems in the branching process of the algorithm is rigorously analyzed. Based on the deeper property of the scheduling problem, a property of batching CAVs in an optimal solution of the problem is further identified.

Numerically, it is found that platoons of CAVs can significantly reduce the time complexity and computation time, in addition, the platoons are batched in the optimal solution. Interestingly, CAVs that are not identified in a platoon at an initialization of an optimization cycle can be batched with other platoons in the optimal solution. This finding reveals that the proposed algorithm can dynamically batches CAVs in the optimal solution of the problem, however, the dynamical batch depends on data initialization of each optimization cycle.

Theoretically, based on identified properties of the scheduling problem, a lower bound of the proposed algorithm is achieved. The lower bound reduces the time complexity from quadratic to linear time growth of the problem size. In addition, the computation time of the algorithm can be directly determined based on a specific detailing of procedures of the branching process of the algorithm. Each unit computation time depends on processing speed of a computer.

7.4 Discussion

Throughout this dissertation, some research areas, within or out of the dissertation scope, should be further discussed in order to clarify the dissertation research goals and proposed further research directions.

First, safety concerns or conflicts between vehicles that are from same approach or different approach are inherently integrated in the reservation-based control models. Such conflict constraints as Eqns. (16), (17), (37), (80), (83) and (84) ensure safely crossing and

following from the perspective of modeling. Because of these integer constraints, the at-worst upper bound of the time complexity of the reservation-based control and scheduling problem is an exponential function of the problem size. In other words, the aim of reducing time complexity arises from avoiding conflicts between vehicles. Overall, this dissertation is evaluated from multiple criteria, including ensuring safety, optimizing intersection efficiency, and reducing time complexity.

Second, during simulation and test of the modeling framework in this dissertation, some issues arise in terms of computation time and memory management, especially for Section 5.5.1. Due to coding platform in Section 5.5.1, Cplex, the memory cannot be maintained or released in Cplex's interface such that memory leaks during the simulation. This issue also prevents a real-time evaluation and implementation of the modeling method. After identifying this problem, the coding platform is transferred to C++ in order to efficiently manage the memory. Furthermore, in Section 6.4, the real-time implementation is improved by updating the solution algorithm and testing on a prototype intersection. Section 6.4 proves and validates the efficacy of the modeling method and associated solution algorithm at a prototype intersection. Therefore, this contribution paves the way for the real-time control at a realistic intersection on the field.

Furthermore, vehicle-to-everything communication protocols, latency, and reliability are out of scope of this dissertation. Such research areas are emerging and required to be investigated such that the reservation-based control can be implemented in the real world (Feng et al., 2022). Especially, vehicle-to-pedestrian communication and appearance of pedestrians or bikes in the vicinity of the reservation-based intersection are out of research scope of this dissertation. However, pedestrians or bikes can be easily integrated with

existing modeling framework of the reservation-based intersection (Wu et al., 2022a). In addition, road surface condition during raining or other adverse weather conditions is not within the dissertation research area, but the controllable acceleration rate under such conditions can be compromised from the perspective of perception-reaction time of human drivers (El-Shawarby et al., 2013).

7.5 Future Research Direction

In the optimal control framework for the reservation-based intersection, the result partially shows that the BATCH strategy has better performances than the ZONE strategy, which leads to an open question of investigating the relationship between the intersection capacity and the configuration of cells within the intersection. In light of this result, the capacity is believed to increase as the number of cells within the intersection increases. Furthermore, the demand is still lower than the intersection capacity even when the demand is up to 14,400 veh/h at the reservation-based intersection. In future research, it is interesting to investigate theoretically and numerically the effect of the number/configuration of cells on the capacity of the reservation-based intersection.

Furthermore, there are still limitations in designing the optimal scheduling algorithms for the reservation-based intersection. During each optimization cycle, solutions of CAVs at last cycle are fixed in terms of next cycle. This control strategy may compromise the optimal solution in terms of all CAVs overall in a time interval. However, the computation time would increase if CAVs in last cycle are included in next cycle when those CAVs are not cleared by the intersection yet. Further, the proposed algorithm only applies to a special reservation-based intersection, where only one cell is configured. In order to achieve a wider application on a general reservation-based intersection, which allows all types of

movements in different lanes from an approach and is configured by more cells, the algorithm needs to accommodate accordingly. Since this type of algorithm that can accommodate such scenarios has been tested (Pei et al., 2021), the future research could incorporate problem properties identified into their algorithm and update the algorithm accordingly for more complex scenarios. In addition, once the strong assumption, same initial speeds for all CAVs, is lifted, more realistic scenarios can be further investigated to test the performance of the scheduling algorithm in the control environment.

Moreover, although the reservation-based intersection control has been tested and validated in a real intersection (Fayazi et al., 2019; Quinlan et al., 2010), it is still implemented through a vehicle-in-the-loop (VIL) method, which requires few CAV in a testbed. More complicated scenarios with more CAVs at a realistic intersection should be further investigated in terms of testing the reservation-based intersection control when a market penetration rate of CAVs rapidly increases. Such test and evaluation require lower communication latency and cyber-physical security among other real world variables in order to achieve real-time implementation of the reservation-based control and associated control algorithms proposed in this dissertation. In the meantime, pedestrians and other road users are critical factors that need to be included in the modeling and testing process. Safety concerns involving pedestrians in the interaction with CAVs are significant and require specific attentions from researchers and transportation agencies.

REFERENCES

Altan, O.D., Wu, G., Barth, M.J., Boriboonsomsin, K., Stark, J.A., 2017. Glidepath: Eco-friendly automated approach and departure at signalized intersections. IEEE Transactions on Intelligent Vehicles 2(4), 266-277.

An, L., Lai, J., Yang, I.X., Hu, J., 2021. Modeling Maximum Throughput of Freeway Merging Area with Partially Connected Automated Traffic, 2021 IEEE International Intelligent Transportation Systems Conference (ITSC). IEEE, pp. 3553-3557.

Au, T.-C., Quinlan, M., Stone, P., 2012. Setpoint scheduling for autonomous vehicle controllers, 2012 IEEE International Conference on Robotics and Automation. IEEE, pp. 2055-2060.

Au, T.-C., Shahidi, N., Stone, P., 2011. Enforcing liveness in autonomous traffic management, Twenty-Fifth AAAI Conference on Artificial Intelligence.

Au, T.-C., Stone, P., 2010. Motion planning algorithms for autonomous intersection management, AAAI 2010 Workshop on Bridging The Gap Between Task And Motion Planning (BTAMP).

Barth, M., Mandava, S., Boriboonsomsin, K., Xia, H., 2011. Dynamic ECO-driving for arterial corridors, 2011 IEEE Forum on Integrated and Sustainable Transportation Systems. IEEE, pp. 182-188.

Bashiri, M., Jafarzadeh, H., Fleming, C.H., 2018. Paim: Platoon-based autonomous intersection management, 2018 21st International Conference on Intelligent Transportation Systems (ITSC). IEEE, pp. 374-380.

Besa Vial, J.J., Devanny, W.E., Eppstein, D., Goodrich, M.T., 2016. Scheduling Autonomous Vehicle Platoons Through an Unregulated Intersection, 16th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2016). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Bichiou, Y., Rakha, H.A., 2018. Developing an Optimal Intersection Control System for Automated Connected Vehicles. IEEE Transactions on Intelligent Transportation Systems 20(5), 1908-1916.

Chalaki, B., Malikopoulos, A.A., 2021. Optimal control of connected and automated vehicles at multiple adjacent intersections. IEEE Transactions on Control Systems Technology.

Chen, G., Kang, K.-D., 2015. Win-fit: Efficient intersection management via dynamic vehicle batching and scheduling, 2015 International Conference on Connected Vehicles and Expo (ICCVE). IEEE, pp. 263-270.

Chen, R., Hu, J., Levin, M.W., Rey, D., 2020. Stability-based analysis of autonomous intersection management with pedestrians. Transportation Research Part C: Emerging Technologies 114, 463-483.

Chen, X., Mårtensson, J., 2021. Optimization Based Merging Coordination of Connected and Automated Vehicles and Platoons, 2021 IEEE International Intelligent Transportation Systems Conference (ITSC). IEEE, pp. 2547-2553.

Chen, Z., Li, X., 2021. Designing corridor systems with modular autonomous vehicles enabling station-wise docking: Discrete modeling method. Transportation Research Part E: Logistics and Transportation Review 152, 102388.

Chen, Z., Li, X., Zhou, X., 2019. Operational design for shuttle systems with modular vehicles under oversaturated traffic: Discrete modeling method. Transportation Research Part B: Methodological 122, 1-19.

Cordeau, J.-F., 2006. A branch-and-cut algorithm for the dial-a-ride problem. Operations Research 54(3), 573-586.

Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C., 2009. Introduction to algorithms (3rd ed.). MIT press.

Ding, J., Li, L., Peng, H., Zhang, Y., 2019. A rule-based cooperative merging strategy for connected and automated vehicles. IEEE Transactions on Intelligent Transportation Systems 21(8), 3436-3446.

Dollar, R.A., Vahidi, A., 2021. Multilane Automated Driving With Optimal Control and Mixed-Integer Programming. IEEE Transactions on Control Systems Technology 29(6), 2561-2574.

Dresner, K., Stone, P., 2004. Multiagent traffic management: A reservation-based intersection control mechanism, Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2, pp. 530-537.

Dresner, K., Stone, P., 2005. Multiagent traffic management: An improved intersection control mechanism, Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems, pp. 471-477.

Dresner, K., Stone, P., 2008. A multiagent approach to autonomous intersection management. Journal of artificial intelligence research 31, 591-656.

Du, Y., Shangguan, W., Chai, L., 2021. A Coupled Vehicle-Signal Control Method at Signalized Intersections in Mixed Traffic Environment. IEEE Transactions on Vehicular Technology 70(3), 2089-2100.

El-Shawarby, I., Abdel-Salam, A.-S.G., Rakha, H., 2013. Evaluation of driver perception–reaction time under rainy or wet roadway conditions at onset of yellow indication. Transportation research record 2384(1), 18-24.

Fajardo, D., Au, T.-C., Waller, S.T., Stone, P., Yang, D., 2011. Automated intersection control: Performance of future innovation versus current traffic signal control. Transportation Research Record 2259(1), 223-232.

Fayazi, S.A., Vahidi, A., 2018. Mixed-integer linear programming for optimal scheduling of autonomous vehicle intersection crossing. IEEE Transactions on Intelligent Vehicles 3(3), 287-299.

Fayazi, S.A., Vahidi, A., Luckow, A., 2019. A Vehicle-in-the-Loop (VIL) verification of an all-autonomous intersection control scheme. Transportation Research Part C: Emerging Technologies 107, 193-210.

Feng, Y., Huang, S.E., Wong, W., Chen, Q.A., Mao, Z.M., Liu, H.X., 2022. On the Cybersecurity of Traffic Signal Control System With Connected Vehicles. IEEE Transactions on Intelligent Transportation Systems.

Feng, Y., Yu, C., Liu, H.X., 2018. Spatiotemporal intersection control in a connected and automated vehicle environment. Transportation Research Part C: Emerging Technologies 89, 364-383.

Ge, Q., Sun, Q., Wang, Z., Li, S.E., Gu, Z., Zheng, S., Liao, L., 2021. Real‐time coordination of connected vehicles at intersections using graphical mixed integer optimization. IET Intelligent Transport Systems 15(6), 795-807.

Guo, Y., Ma, J., Xiong, C., Li, X., Zhou, F., Hao, W., 2019. Joint optimization of vehicle trajectories and intersection controllers with connected automated vehicles: Combined dynamic programming and shooting heuristic approach. Transportation research part C: emerging technologies 98, 54-72.

Han, X., Ma, R., Zhang, H.M., 2020. Energy-aware trajectory optimization of CAV platoons through a signalized intersection. Transportation Research Part C: Emerging Technologies 118, 102652.

He, X., Liu, H.X., Liu, X., 2015. Optimal vehicle speed trajectory on a signalized arterial with consideration of queue. Transportation Research Part C: Emerging Technologies 61, 106-120.

Howard, R.A., 1960. Dynamic programming and markov processes. The Technology Press of The Massachusetts Institute of Technology.

Hu, Z., Huang, J., Yang, D., Zhong, Z., 2021. Constraint-tree-driven modeling and distributed robust control for multi-vehicle cooperation at unsignalized intersections. Transportation Research Part C: Emerging Technologies 131, 103353.

Jiang, Y., Li, S., Shamo, D.E., 2006. A platoon-based traffic signal timing algorithm for major–minor intersection types. Transportation Research Part B: Methodological 40(7), 543-562.

Kamal, M.A.S., Imura, J.-i., Hayakawa, T., Ohata, A., Aihara, K., 2014. A vehicle-intersection coordination scheme for smooth flows of traffic without using traffic lights. IEEE Transactions on Intelligent Transportation Systems 16(3), 1136-1147.

Kumaravel, S.D., Malikopoulos, A., Ayyagari, R., 2021. Optimal coordination of platoons of connected and automated vehicles at signal-free intersections. IEEE Transactions on Intelligent Vehicles.

Lee, J., 2010. Assessing the potential benefits of intellidrive-based intersection control algorithms. Ph.D. Dissertation. University of Virginia, Charlottesville, VA.

Lee, J., Park, B., 2012. Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment. IEEE Transactions on Intelligent Transportation Systems 13(1), 81-90.

Lee, J., Park, B.B., Malakorn, K., So, J.J., 2013. Sustainability assessments of cooperative vehicle intersection control at an urban corridor. Transportation Research Part C: Emerging Technologies 32, 193-206.

Levin, M.W., Boyles, S.D., 2015. Intersection auctions and reservation-based control in dynamic traffic assignment. Transportation Research Record 2497(1), 35-44.

Levin, M.W., Fritz, H., Boyles, S.D., 2016. On optimizing reservation-based intersection controls. IEEE Transactions on Intelligent Transportation Systems 18(3), 505-515.

Levin, M.W., Rey, D., 2017. Conflict-point formulation of intersection control for autonomous vehicles. Transportation Research Part C: Emerging Technologies 85, 528-547.

Li, L., Wang, F., 2006. Cooperative driving at blind crossings using intervehicle communication. IEEE Transactions on Vehicular technology 55(6), 1712-1724.

Li, P.T., Zhou, X., 2017. Recasting and optimizing intersection automation as a connected-and-automated-vehicle (CAV) scheduling problem: A sequential branch-and-bound search approach in phase-time-traffic hypernetwork. Transportation Research Part B: Methodological 105, 479-506.

Li, X., Ghiasi, A., Xu, Z., Qu, X., 2018. A piecewise trajectory optimization model for connected automated vehicles: Exact optimization algorithm and queue propagation analysis. Transportation Research Part B: Methodological 118, 429-456.

Li, Z., Chitturi, M.V., Yu, L., Bill, A.R., Noyce, D.A., 2015. Sustainability effects of next-generation intersection control for autonomous vehicles. Transport 30(3), 342-352.

Li, Z., Chitturi, M.V., Zheng, D., Bill, A.R., Noyce, D.A., 2013a. Modeling reservation-based autonomous intersection control in VISSIM. Transportation Research Record 2381(1), 81-90.

Li, Z., Chitturi, M.V., Zheng, D., Bill, A.R., Noyce, D.A., 2013b. Next-generation intersection control algorithm for autonomous vehicles, 92nd Transportation Research Board Annual Meeting, No. 13-2185.

Li, Z., Elefteriadou, L., Ranka, S., 2014. Signal control optimization for automated vehicles at isolated signalized intersections. Transportation Research Part C: Emerging Technologies 49, 1-18.

Li, Z., Wu, Q., Yu, H., Chen, C., Zhang, G., Tian, Z.Z., Prevedouros, P.D., 2019. Temporal-spatial dimension extension-based intersection control formulation for connected and autonomous vehicle systems. Transportation Research Part C: Emerging Technologies 104, 234-248.

Lin, P., Liu, J., Jin, P.J., Ran, B., 2017. Autonomous vehicle-intersection coordination method in a connected vehicle environment. IEEE Intelligent Transportation Systems Magazine 9(4), 37-47.

Lioris, J., Pedarsani, R., Tascikaraoglu, F.Y., Varaiya, P., 2017. Platoons of connected vehicles can double throughput in urban roads. Transportation Research Part C: Emerging Technologies 77, 292-305.

Liu, B., Ghosal, D., Chuah, C.-N., Zhang, H.M., 2011. Reducing greenhouse effects via fuel consumption-aware variable speed limit (FC-VSL). IEEE Transactions on Vehicular Technology 61(1), 111-122.

Liu, H., Kan, X.D., Shladover, S.E., Lu, X.-Y., Ferlis, R.E., 2018. Modeling impacts of cooperative adaptive cruise control on mixed traffic flow in multi-lane freeway facilities. Transportation Research Part C: Emerging Technologies 95, 261-279.

Liu, M., Zhao, J., Hoogendoorn, S., Wang, M., 2022. A single-layer approach for joint optimization of traffic signals and cooperative vehicle trajectories at isolated intersections. Transportation Research Part C: Emerging Technologies 134, 103459.

Lu, G., Shen, Z., Liu, X., Nie, Y.M., Xiong, Z., 2022. Are autonomous vehicles better off without signals at intersections? A comparative computational study. Transportation research part B: methodological 155, 26-46.

Ma, J., Li, X., Zhou, F., Hu, J., Park, B.B., 2017. Parsimonious shooting heuristic for trajectory design of connected automated traffic part II: computational issues and optimization. Transportation Research Part B: Methodological 95, 421-441.

Ma, M., Li, Z., 2020. Optimizing Reservation-Based Intersection Control of Autonomous Vehicles: A Mixed Integer Linear Programming Approach, 99th Transportation Research Board Annual Meeting, No. 20-03140.

Ma, M., Li, Z., 2021. A time-independent trajectory optimization approach for connected and autonomous vehicles under reservation-based intersection control. Transportation Research Interdisciplinary Perspectives 9, 100312.

Ma, M., Li, Z., 2022a. An Optimal Control Framework for Connected and Autonomous Vehicles at a Reservation-Based Intersection. Available at SSRN 4095468.

Ma, M., Li, Z., 2022b. Optimal scheduling of connected and autonomous vehicles at a reservation-based intersection: optimal platooning and time complexity analysis. Available at SSRN.

Malikopoulos, A.A., Cassandras, C.G., Zhang, Y.J., 2018. A decentralized energy-optimal control framework for connected automated vehicles at signal-free intersections. Automatica 93, 244-256.

Mirheli, A., Tajalli, M., Hajibabai, L., Hajbabaie, A., 2019. A consensus-based distributed trajectory control in a signal-free intersection. Transportation research part C: emerging technologies 100, 161-176.

Morrison, D.R., Jacobson, S.H., Sauppe, J.J., Sewell, E.C., 2016. Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. Discrete Optimization 19, 79-102.

Mu, C., Du, L., Zhao, X., 2021. Event triggered rolling horizon based systematical trajectory planning for merging platoons at mainline-ramp intersection. Transportation Research Part C: Emerging Technologies 125, 103006.

Newell, G.F., 2002. A simplified car-following theory: a lower order model. Transportation Research Part B: Methodological 36(3), 195-205.

Niels, T., Mitrovic, N., Dobrota, N., Bogenberger, K., Stevanovic, A., Bertini, R., 2020. Simulation-Based Evaluation of a New Integrated Intersection Control Scheme for Connected Automated Vehicles and Pedestrians. Transportation Research Record, 0361198120949531.

Niroumand, R., Tajalli, M., Hajibabai, L., Hajbabaie, A., 2020. Joint optimization of vehicle-group trajectory and signal timing: Introducing the white phase for mixed-autonomy traffic stream. Transportation Research Part C: Emerging Technologies 116, 102659.

Pei, H., Feng, S., Zhang, Y., Yao, D., 2019. A Cooperative Driving Strategy for Merging at On-Ramps Based on Dynamic Programming. IEEE Transactions on Vehicular Technology 68(12), 11646-11656.

Pei, H., Zhang, Y., Zhang, Y., Feng, S., 2021. Optimal Cooperative Driving at Signal-Free Intersections with Polynomial-Time Complexity. IEEE Transactions on Intelligent Transportation Systems.

Quinlan, M., Au, T.-C., Zhu, J., Stiurca, N., Stone, P., 2010. Bringing simulation to life: A mixed reality autonomous intersection, 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, pp. 6083-6088.

Rios-Torres, J., Malikopoulos, A.A., 2016. Automated and cooperative vehicle merging at highway on-ramps. IEEE Transactions on Intelligent Transportation Systems 18(4), 780-789.

Ross, S.M., 2014. Introduction to probability models. Academic press, San Diego, United States.

Schepperle, H., Böhm, K., 2007. Agent-based traffic control using auctions, International Workshop on Cooperative Information Agents. Springer, pp. 119-133.

Schepperle, H., Böhm, K., 2008. Auction-based traffic management: Towards effective concurrent utilization of road intersections, 2008 10th IEEE Conference on E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services. IEEE, pp. 105-112.

Sharon, G., Stone, P., 2017. A protocol for mixed autonomous and human-operated vehicles at intersections, International Conference on Autonomous Agents and Multiagent Systems. Springer, pp. 151-167.

Soleimaniamiri, S., Ghiasi, A., Li, X., Hunag, Z., 2020. An analytical optimization approach to the joint trajectory and signal optimization problem for connected automated vehicles. Transportation Research Part C: Emerging Technologies 120, 102759.

Stebbins, S., Hickman, M., Kim, J., Vu, H.L., 2017. Characterising green light optimal speed advisory trajectories for platoon-based optimisation. Transportation Research Part C: Emerging Technologies 82, 43-62.

Sun, Z., Huang, T., Zhang, P., 2020. Cooperative decision-making for mixed traffic: A ramp merging example. Transportation Research Part C: Emerging Technologies 120, 102764.

Tachet, R., Santi, P., Sobolevsky, S., Reyes-Castro, L.I., Frazzoli, E., Helbing, D., Ratti, C., 2016. Revisiting street intersections using slot-based systems. PloS one 11(3), e0149607.

Tajalli, M., Hajbabaie, A., 2021. Traffic Signal Timing and Trajectory Optimization in a Mixed Autonomy Traffic Stream. IEEE Transactions on Intelligent Transportation Systems.

Tang, Z., Zhu, H., Zhang, X., Iryo-Asano, M., Nakamura, H., 2022. A novel hierarchical cooperative merging control model of connected and automated vehicles featuring flexible merging positions in system optimization. Transportation research part C: emerging technologies 138, 103650.

Timmerman, R., Boon, M.A., 2021. Platoon forming algorithms for intelligent street intersections. Transportmetrica A: transport science 17(3), 278-307.

Wan, N., Vahidi, A., Luckow, A., 2016. Optimal speed advisory for connected vehicles in arterial roads and the impact on mixed traffic. Transportation Research Part C: Emerging Technologies 69, 548-563.

Wang, Y., Cai, P., Lu, G., 2020. Cooperative autonomous traffic organization method for connected automated vehicles in multi-intersection road networks. Transportation Research Part C: Emerging Technologies 111, 458-476.

Wu, J., Abbas-Turki, A., El Moudni, A., 2012. Cooperative driving: an ant colony system for autonomous intersection management. Applied Intelligence 37(2), 207-222.

Wu, J., Abbas-Turki, A., Moudni, A.E., 2009. Intersection traffic control by a novel scheduling model, 2009 IEEE/INFORMS International Conference on Service Operations, Logistics and Informatics. IEEE, pp. 329-334.

Wu, J., Yan, F., Abbas-Turki, A., 2013. Mathematical proof of effectiveness of platoon-based traffic control at intersections, 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013). IEEE, pp. 720-725.

Wu, W., Liu, Y., Hao, W., Giannopoulos, G.A., Byon, Y.-J., 2022a. Autonomous intersection management with pedestrians crossing. Transportation Research Part C: Emerging Technologies 135, 103521.

Wu, X., He, X., Yu, G., Harmandayan, A., Wang, Y., 2015. Energy-optimal speed control for electric vehicles on signalized arterials. IEEE Transactions on Intelligent Transportation Systems 16(5), 2786-2796.

Wu, Y., Chen, H., Zhu, F., 2019. DCL-AIM: Decentralized coordination learning of autonomous intersection management for connected and automated vehicles. Transportation Research Part C: Emerging Technologies 103, 246-260.

Wu, Y., Wang, D.Z., Zhu, F., 2022b. Influence of CAVs platooning on intersection capacity under mixed traffic. Physica A: Statistical Mechanics and its Applications 593, 126989.

Wuthishuwong, C., Traechtler, A., 2013. Vehicle to infrastructure based safe trajectory planning for Autonomous Intersection Management, 2013 13th international conference on ITS telecommunications (ITST). IEEE, pp. 175-180.

Xiong, B.-K., Jiang, R., 2021. Speed Advice for Connected Vehicles at an Isolated Signalized Intersection in a Mixed Traffic Flow Considering Stochasticity of Human Driven Vehicles. IEEE Transactions on Intelligent Transportation Systems.

Xu, B., Ban, X.J., Bian, Y., Li, W., Wang, J., Li, S.E., Li, K., 2018a. Cooperative method of traffic signal optimization and speed control of connected vehicles at isolated intersections. IEEE Transactions on Intelligent Transportation Systems 20(4), 1390-1403.

Xu, B., Li, S.E., Bian, Y., Li, S., Ban, X.J., Wang, J., Li, K., 2018b. Distributed conflict-free cooperation for multiple connected vehicles at unsignalized intersections. Transportation Research Part C: Emerging Technologies 93, 322-334.

Xu, H., Cassandras, C.G., Li, L., Zhang, Y., 2021. Comparison of cooperative driving strategies for cavs at signal-free intersections. IEEE Transactions on Intelligent Transportation Systems.

Xu, H., Feng, S., Zhang, Y., Li, L., 2019a. A grouping-based cooperative driving strategy for CAVs merging problems. IEEE Transactions on Vehicular Technology 68(6), 6125-6136.

Xu, H., Zhang, Y., Cassandras, C.G., Li, L., Feng, S., 2020a. A bi-level cooperative driving strategy allowing lane changes. Transportation research part C: emerging technologies 120, 102773.

Xu, H., Zhang, Y., Li, L., Li, W., 2019b. Cooperative driving at unsignalized intersections using tree search. IEEE Transactions on Intelligent Transportation Systems 21(11), 4563-4571.

Xu, Z., Wang, Y., Wang, G., Li, X., Bertini, R.L., Qu, X., Zhao, X., 2020b. Trajectory Optimization for a Connected Automated Traffic Stream: Comparison Between an Exact Model and Fast Heuristics. IEEE Transactions on Intelligent Transportation Systems 22(5), 2969-2978.

Yan, F., Dridi, M., Moudni, A., 2011. A scheduling approach for autonomous vehicle sequencing problem at multi-intersections. International Journal of Operations Research 9(1).

Yang, K., Guler, S.I., Menendez, M., 2016. Isolated intersection control for various levels of vehicle technology: Conventional, connected, and automated vehicles. Transportation Research Part C: Emerging Technologies 72, 109-129.

Yang, Z., Feng, Y., Gong, X., Zhao, D., Sun, J., 2019. Eco-trajectory planning with consideration of queue along congested corridor for hybrid electric vehicles. Transportation Research Record 2673(9), 277-286.

Yang, Z., Feng, Y., Liu, H.X., 2021. A cooperative driving framework for urban arterials in mixed traffic conditions. Transportation Research Part C: Emerging Technologies 124, 102918.

Yao, H., Li, X., 2020. Decentralized control of connected automated vehicle trajectories in mixed traffic at an isolated signalized intersection. Transportation Research Part C: Emerging Technologies 121, 102846.

Yao, Z., Jiang, H., Cheng, Y., Jiang, Y., Ran, B., 2022. Integrated Schedule and Trajectory Optimization for Connected Automated Vehicles in a Conflict Zone. IEEE Transactions on Intelligent Transportation Systems 23(3), 1841 - 1851.

Yu, C., Feng, Y., Liu, H.X., Ma, W., Yang, X., 2018. Integrated optimization of traffic signals and vehicle trajectories at isolated urban intersections. Transportation Research Part B: Methodological 112, 89-112.

Yu, C., Sun, W., Liu, H.X., Yang, X., 2019. Managing connected and automated vehicles at isolated intersections: From reservation-to optimization-based methods. Transportation research part B: methodological 122, 416-435.

Zhang, J., Pei, H., Ban, X.J., Li, L., 2022. Analysis of cooperative driving strategies at road network level with macroscopic fundamental diagram. Transportation Research Part C: Emerging Technologies 135, 103503.

Zhang, K., Yang, A., Su, H., de La Fortelle, A., Miao, K., Yao, Y., 2016. Service-oriented cooperation models and mechanisms for heterogeneous driverless vehicles at continuous static critical sections. IEEE Transactions on Intelligent Transportation Systems 18(7), 1867-1881.

Zhang, Z., Liu, F., Wolshon, B., Sheng, Y., 2020. Virtual Traffic Signals: Safe, Rapid, Efficient and Autonomous Driving Without Traffic Control. IEEE Transactions on Intelligent Transportation Systems 22(11), 6954-6966.

Zhao, W., Liu, R., Ngoduy, D., 2021. A bilevel programming model for autonomous intersection control and trajectory planning. Transportmetrica A: Transport Science 17(1), 34-58.

Zhou, A., Peeta, S., Yang, M., Wang, J., 2022. Cooperative signal-free intersection control using virtual platooning and traffic flow regulation. Transportation Research Part C: Emerging Technologies 138, 103610.

Zhou, F., Li, X., Ma, J., 2017. Parsimonious shooting heuristic for trajectory design of connected automated traffic part I: Theoretical analysis with generalized time geography. Transportation Research Part B: Methodological 95, 394-420.

Zhou, J., Zhu, F., 2021. Analytical analysis of the effect of maximum platoon size of connected and automated vehicles. Transportation Research Part C: Emerging Technologies 122, 102882.

Zhou, Y., Chung, E., Bhaskar, A., Cholette, M.E., 2019. A state-constrained optimal control based trajectory planning strategy for cooperative freeway mainline facilitating and on-ramp merging maneuvers under congested traffic. Transportation Research Part C: Emerging Technologies 109, 321-342.

Zhu, F., Ukkusuri, S.V., 2015. A linear programming formulation for autonomous intersection control within a dynamic traffic assignment and connected vehicle environment. Transportation Research Part C: Emerging Technologies 55, 363-378.

CURRICULUM VITA

NAME: Muting Ma

ADDRESS: Civil and Environmental Engineering Department

218 Eastern Pkwy

Louisville, KY 40208

DOB: Shenyang, China – April 14, 1992

EDUCATION: B.Eng., Transportation Engineering

Liaoning Police College, 2010-14

M.Eng., Transportation Engineering

People's Public Security University of China, 2014-17

Ph.D., Civil Engineering

University of Louisville, 2017-2022

AWARDS: Doctoral Dissertation Completion Award, Graduate School, University of Louisville, 2022

Grosscurth Fellowship, J.B. Speed School of Engineering, University of Louisville, 2017

China National Scholarship, People's Public Security University of China, 2017

PUBLICATIONS:

Ma, M. and Li, Z., 2022. An Optimal Scheduling Mechanism of Connected and Autonomous Vehicles at a Reservation-Based Intersection: Optimal Platooning and Time Complexity Analysis. *Available at SSRN*. http://dx.doi.org/10.2139/ssrn.4097173

Ma, M. and Li, Z., 2022. An Optimal Control Framework for Connected and Autonomous Vehicles at a Reservation-Based Intersection. *Available at SSRN*. http://dx.doi.org/10.2139/ssrn.4095468

Ma, M. and Li, Z., 2021. A time-independent trajectory optimization approach for connected and autonomous vehicles under reservation-based intersection control. *Transportation Research Interdisciplinary Perspectives*, *9*, p.100312. https://doi.org/10.1016/j.trip.2021.100312

INVITED PRESENTATIONS:

Ma, M. and Li, Z., 2022. Modeling Variable Second-order Kinematics for Trajectory Optimization of Connected and Autonomous Vehicles Based on Mixed Integer Linear Programming. *The Transportation Research Board (TRB) 101st Annual Meeting*. (No. 22-04707).

Ma, M. and Li, Z., 2020. Optimizing Reservation-Based Intersection Control of Autonomous Vehicles: A Mixed Integer Linear Programming Approach. *The Transportation Research Board (TRB) 99th Annual Meeting*. (No. 20-03140).

Ma, M. and Li, Z., 2019. A Proactive Approach for Intersection Safety Visualization based on Real-Time Radar Sensor Data. *The Transportation Research Board (TRB) 98th Annual Meeting*. (No. 19-01628).