國立台灣大學電機資訊學院資訊工程學研究所

碩士論文

Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

聯網自駕車線道合併與擴增之通過順序決策

Lane-Merging and Lane-Expanding Passing-Order Decision for Connected and Autonomous Vehicles

黃紹輔

Shao-Fu Huang

指導教授：林忠緯 博士

Advisor: Chung-Wei Lin, Ph.D.

中華民國 112年8月

August 2023

doi:10.6342/NTU202303752

國立臺灣大學碩士學位論文
口試委員會審定書
MASTER'S THESIS ACCEPTANCE CERTIFICATE
NATIONAL TAIWAN UNIVERSITY

聯網自駕車線道合併與擴增之通過順序決策

Lane-Merging and Lane-Expanding Passing-Order Decision for Connected and Autonomous Vehicles

本論文係黃紹輔君（學號 R10922063）在國立臺灣大學資訊工程學系完成之碩士學位論文，於民國 112 年 7 月 24 日承下列考試委員審查通過及口試及格，特此證明。

The undersigned, appointed by the Department of Computer Science and Information Engineering on 24 July 2023 have examined a Master's thesis entitled above presented by HUANG, SHAO-FU (student ID: R10922063) candidate and hereby certify that it is worthy of acceptance.

口試委員 Oral examination committee:

林忠緯　　周詩梵　　江蕙如
（指導教授 Advisor）

黎士瑋

系主任/所長 Director: 洪士灝

doi:10.6342/NTU202303752

Acknowledgements

I would like to express my deepest appreciation to my advisor, Professor Chung-Wei Lin. Thank you for your guidance, expertise, and continuous support throughout the entire thesis process. Your insightful feedback and constructive criticism have been instrumental in shaping the direction and quality of this research. Professor Lin arranges a meeting with me every week even though it is time consuming. In the weekly meetings, I received useful advice when I either encountered problems or went out of thoughts during the research. Professor Lin answered my questions patiently to make sure I knew exactly what he said. Furthermore, Professor Lin also to care of my feelings to make sure I wasn't over-stressed. Last but not least, many solutions to my work were inspired by his advice. I would like to again extend my deepest gratitude.

Moreover, I would like to express my thanks to my partner, Hsiang-Yu Chang. The discussions with him gave me some inspiration. He also helped me verify the correctness of my programs and formulated the problems with me.

Finally, I would like to express gratitude to my family. Thanks to their support and understanding, I can fully concentrate on my research work without other pressure.

Shao-Fu Huang

National Taiwan University
August 2023

doi:10.6342/NTU202303752

聯網自駕車線道合併與擴增之通過順序決策

研究生：黃紹輔　　指導教授：林忠緯 博士

國立臺灣大學資訊工程學研究所

摘要

車道合併是導致交通擁堵的主要原因之一，因為車道數目減少，而其他車輛的行為通常是不可預測的。通過利用車輛對車輛和車輛對基礎設施的通信，以及自動駕駛車輛的特點，車輛可以以較低的時間成本達成共識，從而緩解交通擁堵。本文旨在對在兩對一車道合併情景下車輛的通過順序進行調度，其中每對車輛之間的等待時間不同。首先，我們對問題進行了形式化，提出了一種無車道變換的基於動態規劃的算法，用全局視角對所有車輛進行調度。此外，我們引入了一種考慮車道變換的基於動態規劃的算法，進一步縮短了最後一輛車的預定進入時間。受到車道合併問題的啟發，我們提出了在M-N車道擴展情景下的負載平衡調度問題。負載平衡的重要性在於，如果車道上的負載不平衡，維護不僅會變得昂貴，而且會耗費時間。因此，我們對問題進行了形式化，並首先提出了一種處理每輛車輛出車道決策的混合整數線性規劃（MILP）方法。然後，我們提出了一種啟發式方

doi:10.6342/NTU202303752

法，提高了所有車輛的通過效率。車道合併的實驗結果表明，相比於無車道變換的先來先服務（FCFS）、有車道變換的FCFS以及無車道變換的基於動態規劃的算法，考慮車道變換的基於動態規劃的算法找到了更優的解決方案。負載平衡問題的實驗結果表明，雖然我們的啟發式方法可能無法找到最優解，但它仍然比FCFS方法提供了更優的解決方案。

關鍵詞：聯網自駕車、線道合併、線道擴增、排程、混合整數線性規劃、動態規劃、負載平衡

doi:10.6342/NTU202303752

LANE-MERGING AND LANE-EXPANDING PASSING-ORDER DECISION FOR CONNECTED AND AUTONOMOUS VEHICLES

Student: Shao-Fu Huang Advisor: Dr. Chung-Wei Lin

Department of Computer Science and Information Engineering
National Taiwan University

Abstract

Lane merging is a major reason causing traffic congestion because the number of lanes decreases and the behavior of other vehicles is usually unpredictable. By taking advantage of vehicle-to-vehicle and vehicle-to-infrastructure communication, as well as the characteristics of autonomous vehicles, vehicles can reach a consensus with low time cost, and traffic congestion can be alleviated. In this thesis, we aim to schedule the passing order of vehicles in a two-to-one lane-merging scenario where the waiting times between each pair of vehicles are different. We first formulate the problem and come up with a dynamic programming (DP)-based algorithm that schedules all vehicles with a global perspective. Moreover, we introduce a DP-based algorithm that takes lane changing into consideration, further reducing the scheduled entering time of the last vehicle. Inspired by the lane-merging problem, we come up with a load-balancing scheduling problem under M-N lane-expanding scenarios. The significance of load balancing is that if the load on lanes is unbalanced, maintenance will not only be expensive but also time-consuming. Therefore, we formulate the problem and first propose a Mixed-Integer Linear Programming (MILP) approach that

doi:10.6342/NTU202303752

handles the decision of outgoing lanes for each vehicle. Then, we present a heuristic approach that reduces the scheduled entering time of the last vehicle. Experimental results for lane merging show that the *DP-Based Algorithm with Lane Changing* finds a better solution compared to *First Come First Serve (FCFS) without Lane Changing*, *FCFS with Lane Changing*, and the *DP-Based Algorithm without Lane Changing*. Experimental results for the load-balancing problem demonstrate that our heuristic approach, although it may not find the optimal solution, still provides a better solution than FCFS.

Keywords: Connected and Autonomous Vehicle, Lane Merging, Lane Changing, Lane Expanding, Load Balancing, Scheduling, Mixed-Integer Linear Programming, Dynamic Programming

doi:10.6342/NTU202303752

Table of Contents

doi:10.6342/NTU202303752

doi:10.6342/NTU202303752

List of Tables

doi:10.6342/NTU202303752

List of Figures

doi:10.6342/NTU202303752

Chapter 1

Introduction

Lane merging is a primary reason for traffic congestion. It occurs at highway ramps or when the number of lanes decreases due to road construction, terrain, and other factors. The increase in road capacity and the unpredictability of other vehicles' behavior results in congestion during lane merging. Some vehicles may accelerate or decelerate during a lane-merging scenario, but it is difficult for other vehicles to anticipate their behavior. Therefore, by utilizing vehicle-to-vehicle and vehicle-to-infrastructure communication, lane-merging problems can be addressed by exchanging vehicle information to reach a consensus.

In our work, we aim to enhance the solutions to lane-merging problems by incorporating lane-changing strategies. In a lane-merging problem, vehicles from different incoming lanes approach the merging intersection and wait to be scheduled to enter the outgoing lane. On the other hand, in lane-changing scenarios, vehicles requesting to do lane changing must move to their target lanes before reaching the end of the road segment. Recent research has focused on optimizing the scheduling problems in these scenarios with the assistance of vehicle-to-vehicle or vehicle-to-infrastructure communication.

Some previous works have contributed to solving lane-merging problems to enhance traffic management. Uno *et al.* proposed a merging control method utilizing

doi:10.6342/NTU202303752

vehicle-to-vehicle communication. The method generates a virtual vehicle and maps the virtual vehicle on the other lane for longitudinal control in a scenario where only one vehicle is trying to merge and the passing order is pre-defined [1]. Ravari *et al.* calculated the cost of the vehicle at the front of each lane and picked the one with the lowest cost to enter the outgoing lane in a two-lane scenario [2]. Awal *et al.* targeted to find out the optimal passing order by grouping vehicles, calculating the cost of passing order for each group, and pruning infeasible orders that do not have to be checked necessarily [3]. Pei *et al.* provided a dynamic programming (DP) method that finds the optimal passing order with given traffic flows in a two-lane-merging scenario [4]. Furthermore, Lin *et al.* provided an improved version to solve consecutive lane-merging scenarios [5].

Previous works have also aimed at solving scheduling problems for other different tasks. Lin *et al.* proposed a Mixed-Integer Linear Programming (MILP) algorithm to find the optimal solution for scheduling mandatory lane changing that usually happens near highway ramps [6]. Chen *et al.* used a graph method to derive the optimal passing order for vehicles to pass an intersection [7]. Xu *et al.* tried to find the optimal passing order for intersections with Monte Carlo tree search [8]. Li *et al.* proposed a tree search method with pruning strategies to decide the passing order among lane closures [9]. Atagoziyev *et al.* proposed an algorithm that considers the relative positions between the ego vehicle and its four neighbor vehicles for scheduling lane changing before a critical position in a two-lane scenario [10]. Awal *et al.* proposed a cooperative lane-changing algorithm that aims to find the suitable lane-changing slot according to the position and velocity information received from other vehicles to minimize lane-changing time [11].

Control methods have been developed focusing on lane merging with the help of vehicle-to-vehicle or vehicle-to-infrastructure communication. Milanes *et al.* cal-

doi:10.6342/NTU202303752

culated the time to merge based on the referenced distance and handled incomplete or imprecise data with fuzzy logic [12]. Ioannis *et al.* minimized the acceleration and jerk while merging into the target lane by solving Hamiltonian functions [13]. Kumar *et al.* calculated the risk of collision for each lane and picked the lane with the least risk to merge into in a multi-lane scenario [14]. Wiesner *et al.* trained a model with particle swarm optimization to control vehicle behavior based on the number of mergings and collisions in each iteration [15]. Domingues *et al.* demonstrated the advantages of platooning in a lane-merging scenario and introduced a negotiation method for deciding the merging position [16]. Li *et al.* planned merging paths by solving formulated constraints with quadratic programming [17].

Other approaches also focused on developing control strategies for lane-changing scenarios. Wang *et al.* proposed a lane changing control method that seeks the optimal angular velocity while changing lanes [18]. Yang *et al.* aimed at improving lane-changing safety by adjusting the speed of vehicles involved in a lane-changing procedure [19]. Choi *et al.* did path planning with the Dubins path algorithm utilizing the Geographic Information System [20]. Qiao and Wu planned the lane-changing trajectory using Quadratic Programming [21].

Most of the previous works of two-to-one lane-merging scheduling assume that the waiting times between all vehicles are identical and do not take lane changing into consideration. However, in reality, the performance of vehicles differs from one another, resulting in different waiting times between each pair of vehicles. When waiting times between vehicles are not identical, taking lane changing into consideration can reduce the total time needed for all vehicles to pass the merging intersection because each vehicle can choose the vehicle with a shorter waiting time to follow.

Inspired by the lane-merging problems, we also tackle the issue of load balancing in lane-expanding scenarios. While lane merging deals with a decrease in the

doi:10.6342/NTU202303752

number of lanes, lane expanding deals with an increase in the number of lanes. In real-world multi-lane scenarios, load balancing is crucial for the maintenance and sustainability of infrastructures. If there is a high difference in lane usage, lanes with higher usage deplete faster than others, leading to inefficient road usage. Although load balancing is important, ensuring efficient vehicle passage is also vital. Thus, we aim to find a solution that considers both load balancing and vehicle passing efficiency.

In this thesis, we try to bring lane-merging problems closer to reality by not assuming all vehicles are identical which was not considered in previous works. Furthermore, we take lane changing into consideration which improves the passing efficiency. On the other hand, we deal with load balancing problems on lane-expanding scenarios which was hardly conducted in previous works. We not only work on balancing the loads but also try to improve the passing efficiency.

The main contributions of this thesis are as follows:

- We use the DP-based approach in [5] to solve our lane-merging problem with different waiting times.
- We propose a DP-based algorithm that takes lane changing into consideration to further improve passing efficiency.
- We propose an MILP approach for load-balancing problems under lane-expanding scenarios.
- We propose a heuristic approach to improve passing efficiency.
- We carry out experiments that show that for two-to-one lane-merging problems and load-balancing problems, the approaches we propose outperform first come first serve (FCFS) approaches.

doi:10.6342/NTU202303752

The rest of this thesis is organized as follows: Chapter 2 introduces the terminology and formulates the lane-merging problem. Chapter 3 presents our proposed approaches for the lane-merging problem. Chapter 4 introduces the terminology and formulates the load-balancing problem. Chapter 5 presents our proposed approach for the load-balancing problem. Chapter 6 provides experimental results. Chapter 7 concludes this thesis.

doi:10.6342/NTU202303752

Chapter 2

Problem Formulation for Lane Merging

In this chapter, we first introduce the terms used in this thesis and state the assumptions we make. We use the terms mainly following the terminology in [5] and add some additional terms. After defining the terms and providing the assumptions, we describe the scenario of the problem and formulate the problem in a mathematical form.

2.1 Definitions and Assumptions

Lane Merging. We assume that all vehicles are moving in the same direction. Lane merging is the process where vehicles from different incoming lanes merge into one outgoing lane. For example, in a two-to-one lane-merging problem, we have two incoming lanes merging into one outgoing lane (as shown in Figure (2.1)).

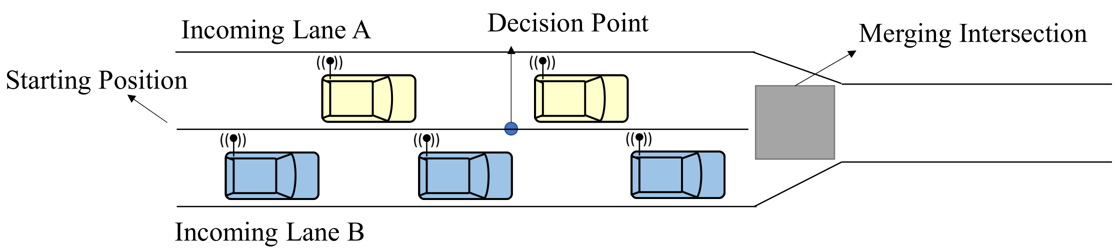


Figure 2.1: Two-to-one lane-merging scenario.

Lane Changing. Lane changing is the process that vehicles wait for a gap

doi:10.6342/NTU202303752

large enough for them to switch from one lane to another without causing collisions. For example, in a two-lane situation, vehicles on lane B changes to lane A when there is a gap large enough between two consecutive vehicles on lane A.

Merging Intersection. The merging intersection is the junction between incoming lanes and outgoing lanes.

Starting Position. The starting position is the position where vehicles enter the incoming lanes.

Decision Point. The decision point is where vehicles that decide to change their lane stop and wait for a gap large enough for them to do lane changing. Vehicles are only allowed to do lane changing at the decision point.

Decision Departure Time. The decision departure time is the time that each vehicle passes the decision point.

Waiting Time. The waiting time is the minimum time gap between the time two consecutive vehicles pass the intersection. $W^=$ is the waiting time for two consecutive vehicles coming from the same lane while W^+ is the waiting time for two consecutive vehicles coming from different lanes entering the same lane. Each pair of vehicles has its own pair of waiting times.

Scheduled Entering Time. The scheduled entering time is the time that each vehicle enters the merging intersection.

Lane-Changing Safety Gap. The lane-changing safety gap is the minimum time gap needed between the vehicle changing lane and the vehicles in the target lane.

The assumptions are as follows:

- All vehicles move in the same direction.

doi:10.6342/NTU202303752

- Vehicle can only do lane changing at the decision point and can change at most once.
- The time needed for each vehicle to pass the merging intersection is identical.
- All vehicles are connected and autonomous.

2.2 Problem Formulation

The notation used in the problem is summarized in Table 2.1. In the two-to-one lane-merging scenario, two incoming lanes, *Lane A* and *Lane B*, merge into one outgoing lane. At the decision point, somewhere before the merging intersection, vehicles decide whether or not to do lane changing to minimize the scheduled entering time for the last vehicle entering the merging intersection. To guarantee safety, vehicles that decide to do lane changing have to wait for a gap large enough on the target lane, and vehicles on the target lane will stop before the decision point to ensure that the merging vehicle can complete lane changing. Vehicles are only allowed to do lane changing at the decision point and each vehicle does lane changing at most once. In the process, the passing order manager solves the passing order, computes the scheduled entering time, and decides whether or not to do lane changing for each vehicle in the detection range.

Suppose there are N vehicles coming from *Lane A* and *Lane B*. The earliest arrival time of vehicle i is A_i, and the two-to-one lane-merging problem is to schedule the passing order of the vehicles on both lanes to minimize the scheduled entering time e_i of the last vehicle with given waiting times ($W^{+}_{i,j}$,$W^{=}_{i,j}$) for each pair of vehicles. The formulation is as follows:

doi:10.6342/NTU202303752

Index	i/j	the index of vehicles on lane A / B
Element	δ_i	the i-th vehicle
Given Constant	A_i	the earliest arrival time for δ_i at the intersection
	D_i	the earliest arrival time for δ_i at the decision point
	N	the number of vehicles
	$W^{=}_{i,j}$	the waiting time if δ_i and δ_j are from the same incoming lane
	$W^{+}_{i,j}$	the waiting time if δ_i and δ_j are from different incoming lanes
	$T_{i,j}$	the lane-changing safety gap
	S_i	the lane of δ_i from the starting position to the decision point
Decision Variable	l_i	the lane of δ_i from the decision point to the merging point
	d'_i	the departure time δ_i from the decision point
	e_i	the scheduled entering time of δ_i

Table 2.1: The notation in two-to-one lane-merging problem.

$$\text{minimize} \quad \max_i \; e_i, \tag{2.1}$$

$$\text{subject to} \quad A_i < e_i, \tag{2.2}$$

$$d'_i + B < e_i, \tag{2.3}$$

$$e_i - e_j \geq W^{=}_{i,j}, \quad \text{if } l_i = l_j \text{ and } j = \operatorname*{argmax}_k e_k \; \forall e_k < e_i, \tag{2.4}$$

$$e_i - e_j \geq W^{+}_{i,j}, \quad \text{if } l_i \neq l_j \text{ and } j = \operatorname*{argmax}_k e_k \; \forall e_k < e_i, \tag{2.5}$$

$$d'_i - d'_j \geq T_{i,j}, \; \forall j : d'_i > d'_j \text{ and } l_i = l_j, \quad \text{if } l_i \neq S_i, \tag{2.6}$$

$$d'_j - d'_i \geq T_{i,j}, \; \forall j : d'_i < d'_j \text{ and } l_i = S_j, \quad \text{if } l_i \neq S_i, \tag{2.7}$$

$$i = 1, 2, \ldots, N, \quad j = 1, 2, \ldots, N.$$

The objective function (2.1) is to minimize the scheduled entering time of the last vehicle. Constraint (2.2) makes sure that the scheduled entering time does not exceed the earliest arrival time for each vehicle. Constraint (2.3) confines the scheduled entering time of each vehicle with their decision departure time. Constraints (2.4) and (2.5) guarantee that two vehicles entering the intersection consecutively fulfill the waiting time requirements. Constraints (2.6) and (2.7) guarantee that when vehicles decide to do lane changing, there is enough space to ensure safety.

doi:10.6342/NTU202303752

Chapter 3

Proposed Approaches for Lane Merging

In this chapter, we introduce our main approaches for the problem. The main approaches include the *DP-Based Algorithm without Lane Changing*, and the *DP-Based Algorithm with Lane Changing*.

3.1 DP-Based Algorithm without Lane Changing

3.1.1 Motivation

Inspired by [5], we attempt to apply their DP approach to our problem. In [5], the waiting times between vehicles to pass the merging intersection are identical so we change part of the proposed algorithm to fulfill our situation where the waiting times for each pair of vehicles are non-identical. Considering the last vehicle going to the outgoing lane, our problem can be decomposed into the following situations:

1. The last vehicle entering the outgoing lane is from *Lane A*.

2. The last vehicle entering the outgoing lane is from *Lane B*.

3.1.2 Derivation

Based on the two situations in Table (3.1), we decompose this problem into a series of subproblems, and define $L_A(i,j)$ and $L_B(i,j)$ to represent the solutions

doi:10.6342/NTU202303752

	Last vehicle entering the outgoing lane	Solution
Situation 1	From *Lane A*	$L_A(i,j)$
Situation 2	From *Lane B*	$L_B(i,j)$

Table 3.1: Two situations of subproblems.

to the subproblems (as shown in Table (3.1)). For each solution, the subscript letter indicates which lane the last vehicle going to the outgoing lane comes from. In the parentheses are the numbers of vehicles that have passed the merging intersection on the two incoming lanes. Take $L_A(i,j)$ as an example, it means that the first i vehicles on *Lane A* and the first j vehicles on *Lane B* have passed the intersection whereas the last vehicle to pass the merging intersection comes from *Lane A*. The solution to each subproblem is a number representing the scheduled entering time of the last vehicle entering the outgoing lane. Take $L_A(1,1)$ for example, if $L_A(1,1) = 7$, the time for one vehicle on *Lane A* and one vehicle from *Lane B* to pass the merging intersection with the last passing vehicle coming from *Lane A* is 7.

To solve $L_A(i,j)$, we show that it can be derived from $L_A(i-1,j)$, $L_B(i-1,j)$. The last vehicle to pass the intersection must be the i-th vehicle from *Lane A*. Thus we consider the following situations:

(a) The last vehicle passing the merging point comes from *Lane A* and the previous vehicle passing the merging intersection is from *Lane A*.

(b) The last vehicle passing the merging point comes from *Lane A* and the previous vehicle passing the merging intersection is from *Lane B*.

When we derive from (a), we compute $L_A(i,j)$ from $L_A(i-1,j)$. We compare the earliest arrival time of the i-th vehicle from *Lane A* with $L_A(i-1,j) + W^{=}_{i,i-1}$ where $W^{=}_{i,i-1}$ is the waiting time between the i-th vehicle and the $i-1$-th from *Lane*

doi:10.6342/NTU202303752

A to pass the merging intersection. When we derive from (b), we compute $L_A(i,j)$ from $L_B(i-1,j)$. We compare the earliest arrival time of the i-th vehicle from *Lane A* with $L_B(i-1,j) + W^{+}_{i,j}$ where $W^{+}_{i,j}$ is the waiting time between the i-th vehicle on *Lane A* and the j-th vehicle on *Lane B* to pass the merging intersection. Then we get

$$\begin{aligned} & L_A(i,j) \\ &= (\max\{A_i, L_A(i-1,j) + W^{=}_{i,i-1}\}) \\ &= (\max\{A_i, L_B(i-1,j) + W^{+}_{i,j}\}). \end{aligned} \tag{3.1}$$

Similarly, we can derive $L_B(i,j)$ in the same way and get

$$\begin{aligned} & L_B(i,j) \\ &= (\max\{A_j, L_A(i,j-1) + W^{+}_{i,j}\}) \\ &= (\max\{A_j, L_B(i,j-1) + W^{=}_{j-1,j}\}). \end{aligned} \tag{3.2}$$

3.1.3 Algorithm

By combining Equations (3.1)–(3.2), we can solve the two-to-one lane-merging problem by a DP-based algorithm as listed in Algorithm 1. Since each subproblem is solved by deriving from the smaller subproblems, we need to solve some base cases first, so that we have some initial solutions to derive from. A base case is a trivial subproblem, which occurs when an incoming lane is empty, such as $L_A(i,0)$. However, some base cases are impossible cases, such as $L_B(i,0)$. $L_B(i,0)$ means the last vehicle entering the merging intersection is from *Lane B* while the number of vehicles coming from *Lane B* that has passed the merging intersection is 0, which is impossible. For these impossible base cases, we set their solutions to ∞. After computing every cell of the two tables, we can get an optimal scheduled entering

doi:10.6342/NTU202303752

Algorithm 1 DP-Based Algorithm without Lane Changing

Input: $\{a_i\}, \{b_j\}, \{A_i\}, \{B_j\}, W^{=}, W^{+}$
Output: $\min\{L(\alpha, \beta, A), L(\alpha, \beta, B)\}$
Initialization $L(0,0,A) = L(0,0,B) = 0 \quad L(1,0,A) = a_1 \quad L(0,1,B) = b_1$ **for** $i \leftarrow 2$ **to** α **do**
    $L(i,0,A) = \max\{a_i, L(i-1,0,A) + W^{=}_{A_i,A_{i-1}}\}$
end
for $j \leftarrow 2$ **to** β **do**
    $L(0,j,B) = \max\{b_j, L(0,j-1,B) + W^{=}_{B_j,B_{j-1}}\}$
end
for $i \leftarrow 1$ **to** α **do**
    for $j \leftarrow 1$ **to** β **do**
        $L(i,j,A) =$
        $\min\{\max\{a_i, L(i-1,j,A) + W^{=}_{A_i,A_{i-1}}\},$
        $\max\{a_i, L(i-1,j,B) + W^{+}_{A_i,B_j}\}\}$
        $L(i,j,B) =$
        $\min\{\max\{b_j, L(i,j-1,A) + W^{+}_{A_i,B_j}\},$
        $\max\{b_j, L(i,j-1,B) + W^{=}_{B_j,B_{j-1}}\}$
        Record the passing order for both cases
    end
end
Output the results

time of the last vehicle. Since each cell stores the last step that leads to the minimum cost of the current step, we can do backtracking to derive the optimal passing order.

3.2 DP-Based Algorithm with Lane Changing

3.2.1 Motivation

Inspired by Section 3.1 we come up with a modified version of the algorithm that takes lane changing into consideration. Vehicles can do lane changing at the decision point and a safety gap is needed for the vehicle to complete lane changing. To make it easy to understand, we abbreviate the vehicle at the decision point as LV, the most recent vehicle to pass the decision point as PV, and the vehicle that may

doi:10.6342/NTU202303752

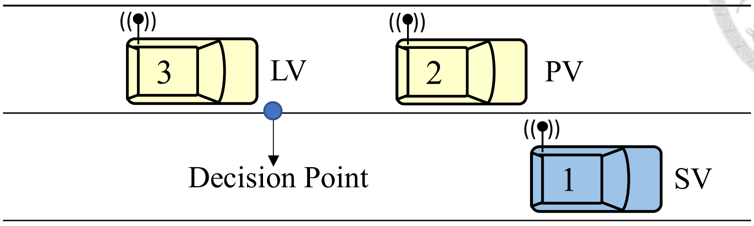


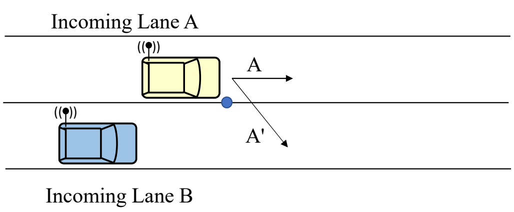
Figure 3.1: The relation between LV, PV and SV.

	Last vehicle entering the outgoing lane	Changed lane	Solution
Situation 1	From *Lane A*	No	$L_A(i,j)$
Situation 2	From *Lane A*	Yes	$L_{A'}(i,j)$
Situation 3	From *Lane B*	No	$L_B(i,j)$
Situation 4	From *Lane B*	Yes	$L_{B'}(i,j)$

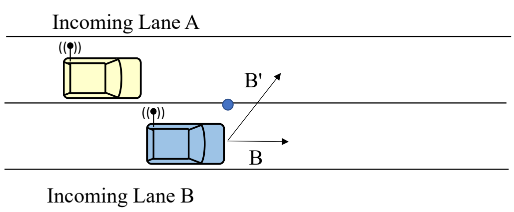
Table 3.2: Four situations of subproblems.

cause negligence as SV (as shown in Figure (3.1)). Considering LV, our problem can be decomposed as follows (as shown in Figure (3.2)):

1. LV is from *Lane A* and it did not do lane changing at the decision point.
2. LV is from *Lane A* and it did lane changing at the decision point.
3. LV is from *Lane B* and it did not do lane changing at the decision point.
4. LV is from *Lane B* and it did lane changing at the decision point.



(a)



(b)

Figure 3.2: The four cases for the next vehicle passing the merging intersection.

doi:10.6342/NTU202303752

3.2.2 Derivation

Based on the four situations, we decompose the problem and define $L_A(i,j)$, $L_{A'}(i,j)$, $L_B(i,j)$, $L_{B'}(i,j)$ to represent the subproblems (as shown in Table (3.2)). For each solution, the subscript letter indicates which lane LV is on while entering the starting position, and those with the prime sign indicate that LV did lane changing at the decision point. Take $L'_A(i,j)$ for example, it means that the first i vehicles on *Lane A* and the first j vehicles on *Lane B* have passed the intersection whereas LV comes from *Lane A* and did lane changing at the decision point. The solution to each subproblem is a number indicating the scheduled entering time of LV. Take $L_{A'}(1,1)$ for example, if $L_{A'}(1,1) = 7$, it means that one vehicle with $S_i = A$ and one with $S_i = B$ has entered the outgoing lane while the last vehicle to pass the intersection has $S_i = A$. Since the subscript letter has a prime sign, the last vehicle to pass the intersection changed its lane at the decision point, so it entered the merging intersection from *Lane B* at time 7.

To solve $L_{A'}(i,j)$, we show that it can be derived from $L_A(i-1,j)$, $L_{A'}(i-1,j)$,$L_B(i-1,j)$, $L_{B'}(i-1,j)$. LV must be on *Lane A* at the starting point and it must do lane changing at the decision point. Here we state the vehicle that passed the decision point right before PV. Thus we consider the following situations:

- The PV's $S_i = A$ and it did not do lane changing at the decision point.
- The PV's $S_i = A$ and it did lane changing at the decision point.
- The PV's $S_i = B$ and it did not do lane changing at the decision point.
- The PV's $S_i = B$ and it did lane changing at the decision point.

When we derive from (a), we compare the earliest arrival time of LV with the scheduled entering time of PV plus $W^+(k,l)$ where $W^+(k,l)$ is the waiting time needed

doi:10.6342/NTU202303752

for PV and LV to pass the merging intersection consecutively entering from different lanes where k and l are the ids for LV and PV. However, there are some situations where the lane-changing safety gap may be neglected. Figure (3.3) shows some examples that the lane-changing safety gap may be neglected. Take Figure (3.3a) for example, when we derive the solution for vehicle 3, we only consider the properties between vehicle 3 and 2. However, if the following distances of vehicles are too close, when vehicle 3 decides to do lane changing, it did not take the lane-changing safety gap of vehicle 3 and 1 into consideration, which will result in the lane-changing safety gap being neglected. Thus for each solution, we keep track of the most recent vehicles that may cause negligence to vehicles behind with decisions (a)–(d) and store their values in $p(P_1, i, j, P_2)$ where P_1 is the solution for PV and P_2 is the solution for SV. For the example in Figure (3.3a), the first vehicle's solution is *A'* and the second vehicle's solution is A, when the third vehicle is deriving its solution, its $p(A, i-1, j, A')$ is the scheduled entering time of the first vehicle and $p(A, i-1, j, A)$ is the scheduled entering time of the second vehicle. Then we use it to make sure that the previously mentioned safety concerns are not violated by comparing $max(p(A, i-1, j, A'), p(A, i-1, j, B)) + T_{k,m}$ with the two values mentioned at the beginning of the paragraph where k and m are the ids for the current vehicle i the decision-making process and the vehicle whose safety gap may be neglected. When we derive from (b), we compare the earliest arrival time of LV with the scheduled entering time of PV plus $T_{k,l}$ or $W^{=}(k, l)$ where $W^{=}(k, l)$ is the waiting time needed for PV and LV to pass the merging intersection consecutively entering from the same lane. When we derive from (c), we compare the earliest arrival time of LV with the scheduled entering time of PV plus $T_{k,l}$ or $W^{=}(k, l)$. When we derive from (d), we compare the earliest arrival time of LV with the scheduled entering time of

doi:10.6342/NTU202303752

PV plus $T_{k,l}$ or $W^+(k,l)$. Thus we get:

$$
\begin{aligned}
&L_{A'}(i,j)\\
&= (\max\{A_k, L_A(i-1,j) + W^+_{k,l}, \max(p(A,i-1,j,A'), p(A,i-1,j,B)) + T_{k,m}\})\\
&= (\max\{A_k, L_{A'}(i-1,j) + \max(W^=_{k,l}, T_{k,l})\})\\
&= (\max\{A_k, L_B(i-1,j) + \max(W^=_{k,l}, T_{k,l})\})\\
&= (\max\{A_k, L_{B'}(i-1,j) + \max(W^+_{k,l}, T_{k,l})\}).
\end{aligned}
\tag{3.3}
$$

Similarly, we can derive $L_{B'}(i,j)$ in the same way and get:

$$
\begin{aligned}
&L_{B'}(i,j)\\
&= (\max\{A_k, L_A(i,j-1) + \max(W^=_{k,l}, T_{k,l})\})\\
&= (\max\{A_k, L_{A'}(i,j-1) + \max(W^+_{k,l}, T_{k,l})\})\\
&= (\max\{A_k, L_B(i,j-1) + W^+_{k,l}, \max(p(B,i,j-1,B'), p(B,i,j-1,A)) + T_{k,m}\})\\
&= (\max\{A_k, L_{B'}(i,j-1) + \max(W^=_{k,l}, T_{k,l})\}).
\end{aligned}
\tag{3.4}
$$

For $L_A(i,j)$, when we derive from (a), we compare the earliest arrival time of LV with the scheduled entering time of PV plus $W^=(k,l)$. When we derive from (b), we compare the earliest arrival time of LV with the scheduled entering time of PV plus $W^+(k,l)$. When we derive from (c) the safety gap concerns may occur if SV has solution *B'* and PV has solution B, then the safety gap between SV and LV may be neglected (as shown in Figure 3.3c). So we compare $p(B,i-1,j,B') + T$ with the earliest arrival time of LV and the scheduled entering time of PV plus $W^+(i,j)$. When we derive from (d), we compare the earliest arrival time of LV with

doi:10.6342/NTU202303752

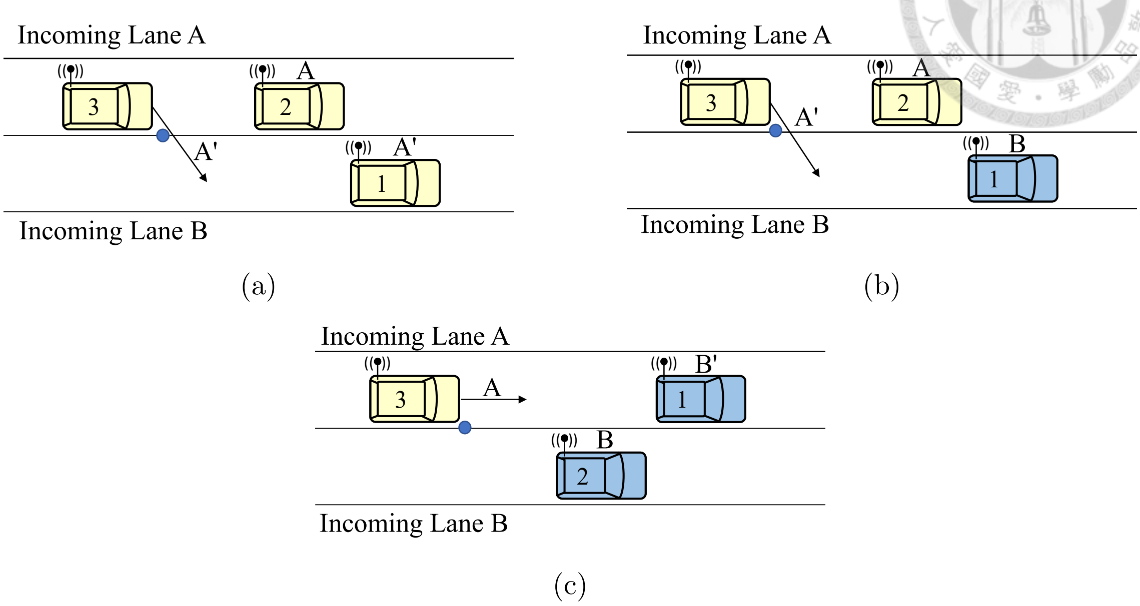


Figure 3.3: The cases the lane-changing safety gap may be neglected.

the scheduled entering time of PV plus T or $W^+(k,l)$.

$$\begin{aligned}
&L_A(i,j)\\
&= (\max\{A_k, L_A(i-1,j) + W^{=}_{k,l}\})\\
&= (\max\{A_k, L_{A'}(i-1,j) + W^{+}_{k,l})\}) \qquad (3.5)\\
&= (\max\{A_k, L_B(i-1,j) + W^{+}_{k,l}, p(B,i-1,j,B') + T_{k,m}\})\\
&= (\max\{A_k, L_{B'}(i-1,j) + \max(W^{=}_{k,l}, T_{k,l})\}).
\end{aligned}$$

Similarly, we can derive $L_B(i,j)$ in the same way and get:

$$\begin{aligned}
&L_B(i,j)\\
&= (\max\{A_k, L_A(i,j-1) + W^{+}_{k,l}, p(A,i,j-1,A')\})\\
&= (\max\{A_k, L_{A'}(i,j-1) + \max(W^{=}_{k,l}, T)\}) \qquad (3.6)\\
&= (\max\{A_k, L_B(i,j-1) + W^{=}_{k,l}\})\\
&= (\max\{A_k, L_{B'}(i,j-1) + W^{+}_{k,l}\}).
\end{aligned}$$

doi:10.6342/NTU202303752

Algorithm 2 DP-Based Algorithm with Lane Changing

Input: $\{a_i\}, \{b_j\}, \{A_i\}, \{B_j\}, \{W^{=}_{i,j}\}, \{W^{+}_{i,j}\}$,

Output: $\min\{L(\alpha, \beta, A), L(\alpha, \beta, A'), L(\alpha, \beta, B), L(\alpha, \beta, B')\}$

Initialization $L(0,0,A) = L(0,0,B) = L(0,0,A') = L(0,0,B') = 0$

$L(1,0,A) = L(1,0,A') = a_1$

$L(1,0,B) = L(1,0,B') = b_1$

for $i \leftarrow 2$ **to** α **do**

$L(i,0,A) =$
$\min\{\max\{a_{A_i}, L(i-1,0,A) + W^{=}_{A_{i-1},A_i}\},$
$\max\{a_{A_i}, L(i-1,0,A') + W^{+}_{A_{i-1},A_i}\}\}$

$L(i,0,A') =$
$\min\{\max\{a_{A_i}, L(i-1,0,A) + W^{+}_{A_{i-1},A_i}\},$
$\max\{a_{A_i}, L(i-1,0,A') + \max(T_{A_{i-1},A_i}, W^{=}_{A_{i-1},A_i})\}$

for $j \leftarrow 2$ **to** β **do**

$L(i,0,B) =$
$\min\{\max\{b_{B_i}, L(0,j-1,B) + W^{=}_{B_{j-1},B_j}\},$
$\max\{b_{B_j}, L(0,j-1,B') + W^{+}_{B_{j-1},B_j}\}\}$

$L(i,0,B') =$
$\min\{\max\{b_{B_j}, L(0,j-1,B) + W^{+}_{B_{j-1},B_j}\},$
$\max\{b_{B_j}, L(0,j-1,B') + \max(T_{B_{j-1},B_j}, W^{=}_{B_{j-1},B_j})\}\}$

for $i \leftarrow 1$ **to** α **do**

for $j \leftarrow 1$ **to** β **do**

$L(i,j,A) =$
$\min\{\max\{a_{A_i}, L(i-1,j,A) + W^{=}_{A_{i-1},A_i}\},$
$\max\{a_{A_i}, L(i-1,j,A') + W^{+}_{A_{i-1},A_i}\},$
$\max\{a_{A_i}, L(i-1,j,B) + W^{+}_{B_j,A_i}, p(B,i-1,j,B') + T_{k_{B,i-1,j,B'},A_i}\},$
$\max\{a_{A_i}, L(i-1,j,B') + \max(T_{B_j,A_i}, W^{=}_{B_j,A_i})\}$

$L(i,j,A') =$
$\min\{\max\{a_{A_i}, L(i-1,j,A) + W^{+}_{A_{i-1},A_i}, \max(p(A,i-1,j,A') + T_{k_{A,i-1,j,A'},A_i},$
$p(A,i-1,j,B) + T_{k_{A,i-1,j,B},A_i}))\},$
$\max\{a_{A_i}, L(i-1,j,A') + \max(T_{A_{i-1},A_i}, W^{=}_{A_{i-1},A_i})\},$
$\max\{a_{A_i}, L(i-1,j,B) + \max(T_{B_j,A_i}, W^{=}_{B_j,A_i})\},$
$\max\{a_{A_i}, L(i-1,j,B') + \max(T_{B_j,A_i}, W^{+}_{B_j,A_i})\}\}$

$L(i,j,B) =$
$\min\{\max\{b_{B_j}, L(i,j-1,B) + W^{=}_{B_{j-1},B_j}\},$
$\max\{b_{B_j}, L(i,j-1,B') + W^{+}_{B_{j-1},B_j}\},$
$\max\{b_{B_j}, L(i,j-1,A) + W^{+}_{B_j,A_i}, p(A,i,j-1,A') + T_{k_{A,i,j-1,A'},B_j}\},$
$\max\{b_{B_j}, L(i,j-1,A') + \max(T_{B_j,A_i}, W^{=}_{B_j,A_i})\}\}$

$L(i,j,B') =$
$\min\{\max\{b_{B_j}, L(i,j-1,B) + W^{+}_{B_{j-1},B_j}, \max(p(B,i,j-1,A) + T_{k_{B,i,j-1,A},B_j}$
$, p(B,i,j-1,B') + T_{k_{B,i,j-1,B'},B_j})\},$
$\max\{b_{B_j}, L(i,j-1,B') + \max(T_{B_{j-1},B_j}, W^{=}_{B_{j-1},B_j})\},$
$\max\{b_{B_j}, L(i,j-1,A) + \max(T_{B_j,A_i}, W^{=}_{B_j,A_i})\},$
$\max\{b_{B_j}, L(i,j-1,A') + \max(T_{B_j,A_i}, W^{+}_{B_j,A_i})\}\}$

Record the passing order for all cases

doi:10.6342/NTU202303752

Output the results

3.2.3 Algorithm

By combining Equations (3.3)–(3.6), we can solve the two-to-one lane-merging problem by a DP-based algorithm as listed in Algorithm 2. Since each subproblem is solved by deriving from the smaller subproblems, we need to solve some base cases first, so that we have some initial solutions to derive from. A base case is a trivial subproblem, which occurs when an incoming lane is empty, such as $L_A(i, 0)$, $L_{A'}(i, 0)$, $L_B(0, j)$, $L_{B'}(0, j)$. However, some base cases are impossible cases, such as $L_B(i, 0)$, $L_{B'}(i, 0)$, $L_A(0, j)$, $L_{A'}(0, j)$. $L_B(i, 0)$ means the last vehicle entering the merging intersection is from *Lane B* while the number of vehicles coming from *Lane B* that has passed the merging intersection is 0, which is impossible. For these impossible base cases, we set their solutions to ∞. We also add the ID for each vehicle to the input, for example, B_j is the ID for the j-th vehicle on *Lane B*. The reason to add the ID as input is that we have to get the waiting times of each pair of vehicles by their ID. For cases that may potentially lead to negligence, we store the solution of SV in $p(P_1, i, j, P_2)$ and the ID of the SV in $k(P_1, i, j, P_2)$ to help derive solutions for each vehicle. After computing every cell of the two tables, we can get the scheduled entering time of the last vehicle. Since each cell stores the last step that leads to the minimum cost of the current step, we can do backtracking to derive the passing order with the least scheduled entering time of the last vehicle.

doi:10.6342/NTU202303752

Chapter 4

Problem Formulation for Load Balancing

Inspired by lane-merging scenarios, lane-expanding scheduling problems occur to us. We then come up with a load-balancing problem for lane-expanding scenarios. In this problem, we wish to accomplish load balancing on the outgoing lanes. Then we wish to minimize the average delay of all vehicles to pass the expanding intersection.

4.1 Definitions and Assumptions

First we provide the additional definitions for the extended problem.

Lane Expanding. We assume that all vehicles are moving in the same direction. Lane expansion is the process where the number of lanes increases and vehicles on each incoming lane decide which outgoing lane to go to (as seen in Figure (4.1)).

Expanding Intersection. The merging intersection is the junction between incoming lanes and outgoing lanes in a lane-expanding scenario.

Decision Number. The decision number is a number that represents which decision point will each vehicle first encounter based on its initial position.

Decision Bound. The decision bound is the bound that decides how many times a vehicle can change its lane. For example, if the decision bound is 15, vehicles

doi:10.6342/NTU202303752

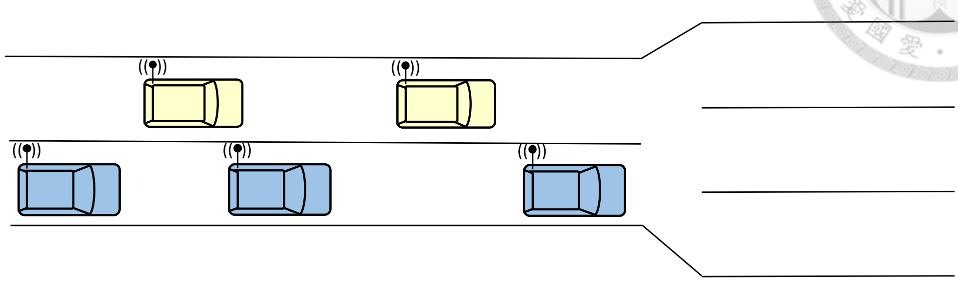


Figure 4.1: Two-to-three lane-expanding scenario.

with initial position under 15 cannot do lane changing whereas vehicles with initial position more than $15 \times k$ can do lane changing at most k times where k is the index of the decision points.

4.2 Problem Formulation

In a lane-expanding scenario, the vehicle can choose which outgoing lane to go to depending on its initial position, its initial lane, and its decision number. Vehicles can only approach an outgoing lane from certain incoming lanes. Tables (4.2) and (4.3) show some examples of the relationships between incoming lanes and outgoing lanes. If the value of $c_{l,j,0}$ is 1, it means that outgoing lane j accepts vehicles that exit from incoming lane l. Conversely, if the value is 0, then vehicles that come from incoming lane l cannot go to outgoing lane j.

Suppose there are M incoming lanes and N outgoing lanes and V vehicles. The load-balancing problem is formulated as follows.

doi:10.6342/NTU202303752

Index	i	the index of vehicles
	j	the index of outgoing lanes
	k	the index of decision points
	l	the index of incoming lanes
Element	δ_i	the i-th vehicle
Given Constant	M	the number of incoming lanes
	N	the number of outgoing lanes
	V	the number of vehicles
	B	the bound of a changing interval
	P_i	the position of δ_i
	S_i	the lane of δ_i at the starting position
Decision Variable	o_i	the outgoing lane of δ_i
	k_i	the decision bound number of δ_i
	l_i	the lane of δ_i at the intersection
	$c_{i,j,k}$	the outgoing lane decision indicator
	$a_{i,j}$	the indicator for $o_i = j$
	u_j	the total number of vehicles with outgoing lane j
Objective	R	the variance of u_0 to u_{N-1}

Table 4.1: The notation in load-balancing problem.

	N=2		N=3		
j	l=0	1	0	1	2
0	1	0	1	0	0
1	1	0	1	1	0
2	0	1	0	1	1
3	0	1	0	0	1

Table 4.2: $c_{l,j,0}$ for 4 outgoing lane scenarios.

	N=2		N=3			N=4			
j	l=0	1	0	1	2	0	1	2	3
0	1	0	1	0	0	1	0	0	0
1	1	0	1	1	0	1	1	0	0
2	0	1	0	1	0	0	1	1	0
3	0	1	0	0	1	0	0	1	1
4	0	1	0	0	1	0	0	0	1

Table 4.3: $c_{l,j,0}$ for 5 outgoing lane scenarios.

doi:10.6342/NTU202303752

$$
\begin{aligned}
\text{minimize} \quad & R, && (4.1)\\
\text{subject to} \quad & 0 \leq o_i < N, && (4.2)\\
& k_i = \left\lceil \frac{P_i}{B} \right\rceil, && (4.3)\\
& c_{l,j,k} = 1, \quad \text{if } \exists\ l' : |l - l'| <= k \text{ and } c_{l',j,0} = 1, && (4.4)\\
& o_i \neq j, \quad \text{if } c_{S_i,j,k_i} = 0, && (4.5)\\
& a_{i,j} = 1, \quad \text{if } o_i = j, && (4.6)\\
& a_{i,j} = 0, \quad \text{if } o_i \neq j, && (4.7)\\
& u_j = \Sigma_i a_{i,j}, && (4.8)\\
& R = \frac{(\Sigma_{j=0}^{N-1} u_j^2)}{N} + \left(\frac{(\Sigma_{j=0}^{N-1} u_j)}{N} \right)^2, && (4.9)\\
& i = 1, 2, \ldots, V, \quad j = 0, 1, \ldots, N-1, &&\\
& l = 1, 2, \ldots, M-1 \quad k = 0, 1, \ldots, M-1. &&
\end{aligned}
$$

The objective function (4.1) is to minimize the variance of the numbers of vehicles on each outgoing lane to achieve load balancing. Constraint (4.2) states that the outgoing lane for each vehicle cannot the total number of outgoing lanes. Constraint (4.3) shows how the decision number of each vehicle is calculated. Constraints (4.4) and (4.5) state that whether a vehicle can select an outgoing lane based on its initial lane and decision number. Constraints (4.6) – (4.8) state how the number of vehicles on each outgoing lane is calculated. Constraint (4.9) states the calculation of the variance of the numbers of vehicles on each outgoing lane.

After all vehicles have chosen their outgoing lane to achieve load balance, we then cope with the second half of the problem which is to minimize the average delay of all vehicles. Vehicles can only change lanes at decision points and the number

doi:10.6342/NTU202303752

Index	i/j	the index of vehicles
	n	the index of decision points
Element	δ_i	the i-th vehicle
Given Constant	M	the number of incoming lanes
	N	the number of outgoing lanes
	V	the number of vehicles
	B	the bound of a changing interval
	S_i	the lane of δ_i at the starting position
Decision Variable	o_i	the outgoing lane of δ_i
	c_i	the change lane indicator of δ_i
	l_i	the lane of δ_i at the intersection
	d'_i	the departure time δ_i from the decision point
	e_i	the scheduled entering time of δ_i

Table 4.4: The notation in load-balancing problem.

of decision points equals the number of incoming lanes − 1. The lane-changing constraints are similar to the lane-merging problem. To minimize the average delay of vehicles and ensure that all vehicles have changed to their target lanes so they can enter their selected outgoing lane, the second half of the problem is formulated as follows.

doi:10.6342/NTU202303752

$$\text{minimize} \quad \max_i \; e_i, \tag{4.10}$$

$$\text{subject to} \quad d'_{i,0} + B < e_i, \tag{4.11}$$

$$c_{i,n} = 1, \quad \text{if } l_{i,n+1} \neq l_{i,n}, \tag{4.12}$$

$$c_{i,n} = 0, \quad \text{if } l_{i,n+1} = l_{i,n}, \tag{4.13}$$

$$|l_{i,n-1} - l_{i,n}| \leq 1, \tag{4.14}$$

$$d'_{i,n} - d'_{j,n} \geq T_{i,j} \; \forall j : d'_{i,n} > d'_{j,n} \text{ and } l_{i,n} = l_{j,n}, \quad \text{if } c_{i,n} = 1, \tag{4.15}$$

$$d'_{j,n} - d'_{i,n} \geq T_{i,j} \; \forall j : d'_{i,n} < d'_{j,n} \text{ and } l_{i,n} = l_{j,n+1}, \quad \text{if } c_{i,n} = 1, \tag{4.16}$$

$$d'_{i,n} - d'_{j,n} \geq 1, \quad \text{if } l_{i,n+1} = l_{j,n+1} \text{ and } d'_{i,n} > d'_{j,n}, \tag{4.17}$$

$$d'_{i,n} - d'_{i,n+1} \geq B, \quad \text{if } d'_{i,n+1} > 0, \tag{4.18}$$

$$e_i - e_j \geq W^{=}_{i,j}, \quad \text{if } l_{i,0} = l_{j,0} \text{ and } j = \operatorname*{argmax}_k e_k \; \forall e_k < e_i, \tag{4.19}$$

$$e_i - e_j \geq W^{+}_{i,j}, \quad \text{if } l_{i,0} \neq l_{j,0} \text{ and } o_i = o_j \text{ and } j = \operatorname*{argmax}_k e_k \; \forall e_k < e_i, \tag{4.20}$$

$$i = 1, 2, \ldots, V, \quad j = 1, 2, \ldots, V,$$

$$n = 1, 2, \ldots, M - 1.$$

The objective function (4.10) is to minimize the scheduled entering time of the last vehicle. Constraint (4.11) states that the scheduled entering time cannot be greater than the departure time at the decision point closest to the intersection plus the decision bound. Constraints (4.12) and (4.13) derive the change lane indicator. Constraint (4.14) states that each vehicle can only stay on its previous lane or change to its neighbor lanes at a decision point. Constraints (4.15) – (4.17) restrict the behavior of all vehicles at each decision point. Constraint (4.18) confines the speed of each vehicle between decision points. Constraints (4.19) and (4.20) guarantee that two vehicles entering the intersection consecutively fulfill the waiting time requirements.

doi:10.6342/NTU202303752

Chapter 5

Proposed Approaches for Load Balancing

5.1 Our Heuristic Approach

5.1.1 Phase 1

As mentioned in the previous chapter, this problem consists of two phases. For the first phase, since the constraints aren't too complicated, we decide to use MILP to decide which vehicles exit from an incoming lane and how many of them enter an outgoing lane. To meet the constraints, each vehicle can only select certain outgoing lanes. For example, in a four-to-six lane-expanding scenario, vehicles exiting from *Lane A* can only go to *Lane E* and *Lane F*, vehicles exiting from *Lane B* can only go to *Lane F* and *Lane G*, and so on (as shown in Figure (5.1)).

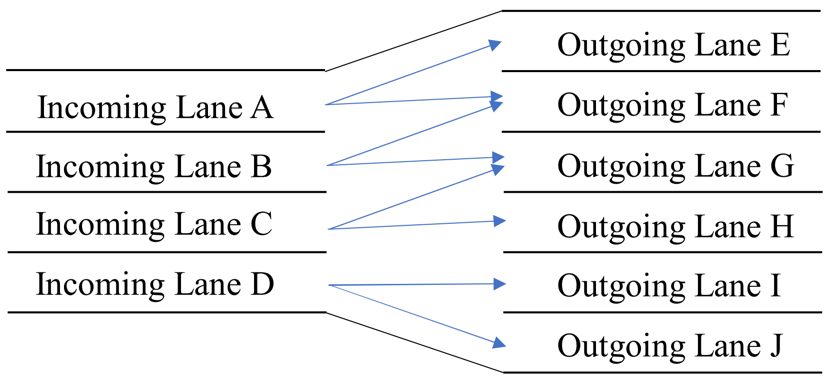


Figure 5.1: The outgoing lane constraints for incoming lanes.

doi:10.6342/NTU202303752

5.1.2 Phase 2

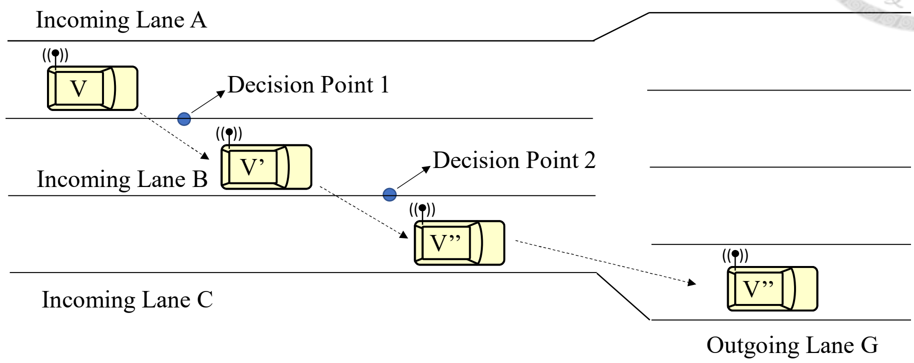
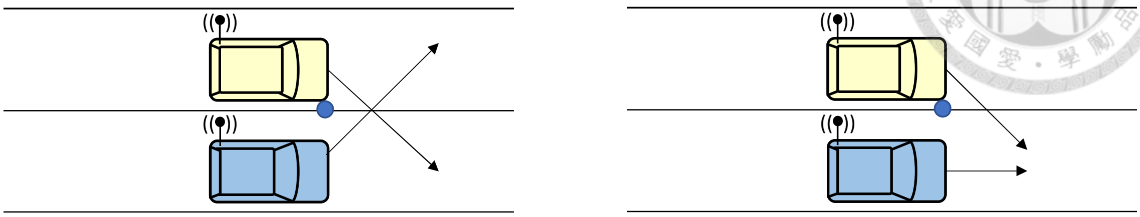


Figure 5.2: In this example, the source lane of V is *Lane A*, and *Lane G* is assigned as its outgoing lane, so it has to change to *Lane C* to enter *Lane G*.

After assigning the incoming lane from which each vehicle enters the intersection, each vehicle has to change to it is assigned incoming lane (as shown in Figure (5.2)). Since lane changing can only be done at decision points, we first assign priorities to each vehicle based on its initial position, and whenever there is a conflict, the vehicle with higher priority gets to go first. A conflict at the decision point happens when vehicles arriving at the decision point may cause collisions if they take action simultaneously. For example, if the vehicle in the bottom lane wants to either remain in the same lane or change its lane to the top lane, and the vehicle in the top lane wishes to change to the bottom lane, they will collide if they taking actions simultaneously (as shown in Figure (5.3)).

After each vehicle changes to their assigned incoming lane and arrives at the intersection. We assign them their outgoing lanes. In our approach, if there are v vehicles exiting from *Lane A*, we assign the first i vehicles to enter *Lane E* and the other $v - i$ vehicles to enter *Lane F*. For example, if five vehicles are exiting from *Lane A*, three of them are going to *Lane E*, and two of them are going to *Lane F*, then *Lane E* is assigned to the first three vehicles entering the intersection as

doi:10.6342/NTU202303752



(a) (b)

Figure 5.3: Example of conflicts in a lane-changing scenario.

their outgoing lane and *Lane F* is assigned to the other two vehicles (as shown in Figure (5.4)).

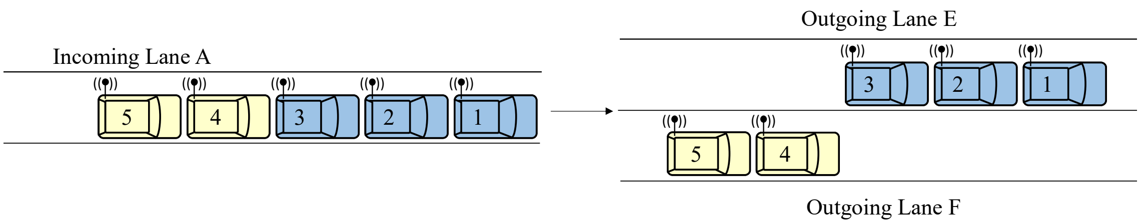


Figure 5.4: The example of the decision of a single lane.

doi:10.6342/NTU202303752

Chapter 6

Experimental Results

In this chapter, we demonstrate the experimental results of our proposed approaches. In Section (6.1), we introduce the experimental setup. In Section (6.2), we show the experimental results of the three main approaches, including the *FCFS with Lane Changing* approach, the *DP-Based Algorithm without Lane Changing*, and the *DP-Based Algorithm with Lane Changing*. In Section (6.3) we compare our heuristic approach with FCFS.

6.1 Experimental Setup

We implement the proposed algorithms with Python. All the experiments are run on a macOS Catalina notebook with dual Core 2.1-GHz Intel i7-CPU and 16 GB memory.

6.2 Experiments for Lane Merging

In the experiments for lane merging, we analyze the results of our two main approaches: (1) *DP-Based Algorithm without Lane Changing* and (2) *DP-Based Algorithm with Lane Changing* and compare them with comparative approaches mentioned in Section (6.2.1). The experiments are conducted under the following scenarios: (1) different densities (λ), (2) different numbers of vehicles (N). The

doi:10.6342/NTU202303752

results are shown in Tables (6.1) and (6.2), respectively. Each value in the tables is the average value of 10 test cases.

For the inputs of the algorithms, we randomly generate the earliest arrival time of each vehicle following a Poisson distribution, and the waiting times for each pair of vehicles are randomly assigned. We here set the change lane safety distance for each pair of vehicles to 4. Since our algorithm focuses on finding an optimal passing order of vehicles, we assume the earliest arrival time of each vehicle is given and fixed in our experiments. However, in practice, the earliest arrival time of each vehicle will be periodically computed according to its position, kinematics, and dynamics. To compare the performance of different algorithms, we report three metrics in the results: (1) T_{last}, (2) T_{delay}, and (3) T_{exec}. All the metrics are in seconds, and their meanings are as follows:

- T_{last}: The scheduled entering time of the last vehicle. It implies the total time needed for all vehicles to pass the merging point.
- T_{delay}: The average difference between each vehicle's scheduled entering time and its earliest arrival time. It reflects the delay time of each vehicle, not only the last one.
- T_{exec}: The execution time of the program, representing the computational efficiency of the algorithms.

6.2.1 Comparative Approaches

The approaches we use to compare with our proposed approaches are (1) *FCFS without Lane Changing* and (2) *FCFS with Lane Changing*. For *FCFS without Lane Changing*, the passing order of each vehicle is assigned based on its initial position, and no vehicles are not allowed to do lane changing. For *FCFS with Lane*

doi:10.6342/NTU202303752

	FCFS without Lane Changing			*FCFS with Lane Changing*			*DP without Lane Changing*			*DP with Lane Changing*		
λ	T_{last}	T_{delay}	T_{exec}	T_{last}	T_{delay}	T_{exec}	T_{last}	T_{delay}	T_{exec}	T_{last}	T_{delay}	T_{exec}
0.4	703.50	311.46	2.61E-04	504.00	218.76	2.61E-04	477.00	201.46	2.71E-02	424.00	169.86	1.22E-01
0.6	702.00	344.00	2.73E-04	489.50	239.72	7.71E-04	456.00	218.08	2.37E-02	423.00	194.55	1.25E-01
0.8	727.00	327.80	2.85E-04	522.50	255.88	8.43E-04	469.50	224.80	2.61E-02	414.00	196.51	1.30E-01

Table 6.1: Results (in second) for different traffic densities.

Changing, we follow a greedy rule and also assig the passing order of each vehicle based on their arrival time at the starting position. The difference is when a vehicle arrives at the decision point, it can decide whether or not to do lane changing by comparing the waiting times between the ego vehicle and the vehicle that last passed the decision point and the time needed for lane changing. For example, if the vehicle with passing order i is at the decision point, it compares the waiting time and the lane-changing safety gap between itself and the vehicle with passing order $i-1$. Then it decides whether to do lane changing or not.

6.2.2 Experimental Results with Different Traffic Densities

In this scenario, we set $N = 60$. Table 6.1 shows that T_{last} and T_{delay} decrease from *FCFS without Lane Changing* to our *FCFS with Lane Changing* approach because each ego vehicle has the opportunity to do lane changing if lane changing reduces the passing time for the ego vehicle at the merging intersection. T_{last} and T_{delay} decrease more from our *FCFS with Lane Changing* approach to our *DP-Based Algorithm without Lane Changing* because the passing order of the *FCFS with Lane Changing* approach is assigned only considering each vehicle's earliest arrival time and thus the *DP-Based Algorithm without Lane Changing* has better performance because it takes all possible passing orders into consideration. Then between the *DP-Based Algorithm without Lane Changing* and the *DP-Based Algorithm with Lane Changing*, T_{last} and T_{delay} decrease even more because the *DP-Based Algorithm with Lane Changing* approach was based on the *DP-Based Algorithm without Lane*

doi:10.6342/NTU202303752

	FCFS without Lane Changing			*FCFS with Lane Changing*			*DP without Lane Changing*			*DP with Lane Changing*		
N	T_{last}	T_{delay}	T_{exec}	T_{last}	T_{delay}	T_{exec}	T_{last}	T_{delay}	T_{exec}	T_{last}	T_{delay}	T_{exec}
20	289.00	167.94	5.94E-05	230.00	135.79	5.65E-04	205.50	124.98	3.92E-03	195.00	119.23	2.24E-02
40	512.00	250.03	1.50E-04	369.50	185.47	5.13E-04	345.00	172.95	1.21E-02	308.50	154.31	6.86E-02
60	702.00	344.00	2.73E-04	489.50	239.72	7.71E-04	456.00	218.08	2.37E-02	423.00	194.55	1.25E-01

Table 6.2: Results (in second) for different numbers of vehicles.

Changing and it also allowed vehicles to do lane changing if the time cost is reduced. T_{exec} increases from *FCFS without Lane Changing* to our *FCFS with Lane Changing* approach because we needed time to compute whether each vehicle will do lane changing. For *DP-Based Algorithm without Lane Changing*, T_{exec} increases even more because a DP table had to be filled to consider all possible passing orders. The *DP-Based Algorithm with Lane Changing* increases even more in T_{exec} because it takes lane changing into consideration and thus a more complicated DP table has to be filled. When λ increases, our approaches still have overwhelming advantages over T_{last} and T_{delay}.

6.2.3 Experimental Results with Different Numbers of Vehicles

In this scenario, we set $\lambda = 0.6$. Compared with *FCFS without Lane Changing*, our approaches have a much greater advantage over T_{last} and T_{delay} when the number of vehicles increases, while the execution time of each approach increases in different degrees as shown in Table (6.2). The reasons are similar to the results of different traffic densities.

6.3 Experiments for Load Balancing

In the experiments for lane changing, we analyze the results of the our heuristic approach and compare it with a FCFS approach. For the number of incoming lanes and outgoing lanes, we consider the following cases: (1) 3 to 4, (2) 3 to 5, (3) 4 to 5, and (4) 4 to 6 where the first number is the number of incoming lanes and the

doi:10.6342/NTU202303752

second number is the number of outgoing lanes. The experiments are conducted under the following scenarios: (1) different densities (λ), (2) different detection Ranges (R). The results are shown in Tables (6.3) – (6.8), respectively. Each value in the tables is the average value calculated from the results of 10 test cases. For the inputs of the algorithms, we randomly generate the initial positions of vehicles following a Poisson distribution, and the waiting times for each pair of vehicles are randomly assigned. The lane-changing safety distance for each pair of vehicles are also set to 4. In the experiments, we compare our heuristic approach with FCFS. To compare the performance between the two methods, we report three metrics in the result: (1) D_{max}, (2) D_{avg}, (3) T_{last}. All the metrics are in seconds, and their meanings are as follows:

- D_{max}: The maximum delay of all vehicles. It implies whether or not a vehicle will have the potential of suffering starvation.
- D_{avg}: The average delay of all vehicles. It considers the delay time of each vehicle, not only the one with the maximum delay.
- T_{last}: The scheduled entering time of the last vehicle. It implies the total time needed for all vehicles to pass the merging point.

6.3.1 Comparative Approaches

For the load balancing problem, we compare our heuristic approach with FCFS. In the FCFS approach, each vehicle will be assigned the incoming lane it will enter the intersection from and the outgoing lane it will go to. After being assigned, they have to change to their assigned incoming lane and enter their assigned outgoing lane. When conflicts occur at the decision point or the intersection, the decision

doi:10.6342/NTU202303752

of which vehicle goes first is based on their passing orders. The passing orders of vehicles will be assigned based on their initial position.

6.3.2 Experimental Results with Different Traffic Densities

In this scenario, we set $R = 60$. Tables (6.3) and (6.4) show that D_{max}, D_{avg} and T_{last} decrease from FCFS to our heuristic approach because our heuristic approach reduces the chance for all vehicles of choose W^+ over $W^=$. The decreasing amplitude increases when the density grows because more vehicles will appear in our detection range and thus more vehicles that had to choose W^+ in FCFS can then choose $W^=$ in our heuristic approach. Furthermore, we observe that the decreasing amplitudes for N to N+1 cases are greater than N to N+2 cases. It is because when more outgoing lanes exist, the number of lane changes has to increase to fulfill load balancing. This also results in the performance of N to N+1 cases being better than N to N+2 cases in our heuristic approach. Table (6.5) shows the runtime of each approach. For Phase 1, the MILP approach takes more time when the traffic density increases since the number of vehicles increases, and more time is needed to find the optimal solution. For Phase 2, we can see that for all cases, the runtime for FCFS and our heuristic approach is almost the same and our heuristic approach has a better performance. Thus we believe our heuristic approach performs better than FCFS.

	FCFS						Heuristic					
	3 to 4			3 to 5			3 to 4			3 to 5		
λ	D_{max}	D_{avg}	T_{last}	D_{max}	D_{avg}	T_{last}	D_{max}	D_{avg}	T_{last}	D_{max}	D_{avg}	T_{last}
0.4	36.18	12.50	94.54	26.54	10.37	84.77	20.50	7.91	78.24	24.40	8.80	82.35
0.6	71.90	28.92	131.10	52.06	21.53	110.98	45.50	18.94	104.37	49.95	19.29	108.62
0.8	98.29	40.09	157.53	77.55	32.55	136.30	60.46	27.42	118.93	71.06	28.77	129.94

Table 6.3: Results (in second) for different traffic densities.

doi:10.6342/NTU202303752

	FCFS						Heuristic					
	4 to 5			4 to 6			4 to 5			4 to 6		
λ	D_{max}	D_{avg}	T_{last}	D_{max}	D_{avg}	T_{last}	D_{max}	D_{avg}	T_{last}	D_{max}	D_{avg}	T_{last}
0.4	45.14	15.81	102.88	31.49	12.63	89.47	23.95	9.01	80.11	28.85	10.61	86.84
0.6	85.54	32.06	144.39	68.65	27.33	126.97	45.30	19.23	102.47	62.17	21.54	120.60
0.8	119.19	46.48	178.49	92.78	37.56	151.84	63.91	28.12	122.08	84.74	30.50	143.94

Table 6.4: Results (in second) for different traffic densities.

	3 to 4			3 to 5			4 to 5			4 to 6		
λ	MILP	FCFS	Heuristic	MILP	FCFS	Heuristic	MILP	FCFS	Heuristic	MILP	FCFS	Heuristic
0.4	1.65	1.19e-2	1.21e-2	4.21	1.11e-2	1.16e-2	4.15	2.66e-2	2.65e-2	8.80	2.65e-2	2.66e-2
0.6	2.09	2.26e-2	2.34e-2	4.75	2.23e-2	2.19e-2	3.86	3.46e-2	3.36e-2	7.68	3.42e-2	3.39e-2
0.8	4.14	5.93e-2	5.94e-2	8.34	5.51e-2	5.86e-2	6.97	6.06e-2	6.17e-2	16.88	7.04e-2	6.46e-2

Table 6.5: Runtime (in second) for different traffic densities.

6.3.3 Experimental Results with Different Detection Range

In this scenario, we set $\lambda = 0.8$. Tables (6.6) and (6.7) show that similar to different traffic densities, the advantages of our approach grow when the detection range increases. Due to the increase in the number of vehicles considered, more vehicles have reduced their waiting time from W^+ to $W^=$ which reduces the values of the three metrics even more. Table (6.8) shows the runtime for different detection ranges. What the results imply is similar to different traffic densities too.

	FCFS						Heuristic					
	3 to 4			3 to 5			3 to 4			3 to 5		
R	D_{max}	D_{avg}	T_{last}	D_{max}	D_{avg}	T_{last}	D_{max}	D_{avg}	T_{last}	D_{max}	D_{avg}	T_{last}
20	30.81	12.19	49.79	26.43	10.52	45.00	21.65	9.16	39.80	24.76	9.54	43.67
40	59.71	23.57	99.33	49.28	20.30	87.80	38.72	17.68	77.40	46.59	18.28	85.82
60	98.29	40.59	157.53	77.55	32.55	136.30	60.46	27.42	118.93	71.06	28.77	129.94

Table 6.6: Results (in second) for different detection range.

doi:10.6342/NTU202303752

	FCFS						Heuristic					
	4 to 5			4 to 6			4 to 5			4 to 6		
R	D_{max}	D_{avg}	T_{last}	D_{max}	D_{avg}	T_{last}	D_{max}	D_{avg}	T_{last}	D_{max}	D_{avg}	T_{last}
20	31.85	11.27	50.19	26.25	9.83	44.60	19.65	8.22	37.19	21.47	7.80	40.26
40	68.62	24.37	107.66	55.18	21.06	94.45	38.06	16.30	75.90	49.83	17.01	89.04
60	119.19	46.48	178.49	92.78	37.56	151.84	63.91	28.12	122.08	84.74	30.50	143.94

Table 6.7: Results (in second) for different detection range.

	3 to 4			3 to 5			4 to 5			4 to 6		
R	MILP	FCFS	Heuristic	MILP	FCFS	Heuristic	MILP	FCFS	Heuristic	MILP	FCFS	Heuristic
20	2.66e-1	2.72e-3	2.81e-3	0.54	2.84e-3	3.40e-3	4.47e-1	5.99e-3	6.00e-3	1.22	6.72e-3	6.55e-3
40	9.74e-1	1.11e-2	1.12e-2	2.25	9.92e-3	1.03e-2	1.85	2.40e-2	2.40e-2	4.34	2.54e-2	2.49e-2
60	2.09	2.26e-2	2.34e-2	4.75	2.23e-2	2.19e-2	3.86	3.46e-2	3.36e-2	7.68	3.42e-2	3.39e-2

Table 6.8: Runtime (in second) for different detection range.

6.3.4 Summary

With the result of experiments, we can see that for lane-merging scenarios, *DP-Based Algorithm with Lane Changing* is the best approach because it takes passing orders of all possible cases into consideration. Though the *DP-Based Algorithm without Lane Changing* performs well, it still in some cases can't avoid long waiting times. The *FCFS with Lane Changing* approach spends less time than previous approaches but has a worse performance. *FCFS without Lane Changing* takes too long for all vehicles to pass since no additional computing is done.

For the load balancing problem, results show that our heuristic approach reduces the time consumption compared with FCFS. Our approach performs better on N to N+1 than N to N+2 scenarios due to additional lane changing needed to fulfill load balancing.

doi:10.6342/NTU202303752

Chapter 7

Conclusions

As most existing works about lane merging assume that all vehicles are identical, we aim to solve lane-merging problems by considering different waiting times for each pair of vehicles. What's more, we take lane changing into consideration since a long waiting time may be avoided if the vehicle decides to change lane.

Inspired by lane-merging problems, we also come up with a load-balancing scheduling problem for lane-expanding scenarios that not only seeks for load balancing on the outgoing lanes but also takes the passing time of all vehicles into consideration. For the purpose of taking advantage of the global view and cooperative driving, we focus on optimizing lane-merging problems with different waiting times and the load-balancing problem for lane-expanding scenarios.

The experimental results for lane merging show that the DP-Based Algorithm without Lane Changing reduces the time cost by scheduling all vehicles in a global view. Furthermore, the DP-Based Algorithm with Lane Changing improves the passing efficiency of vehicles by taking lane changing into consideration. On the other hand, the experimental results for the load-balancing problem show that our heuristic approach is more efficient in scheduling vehicles to pass the expanding intersection compared to FCFS.

In this thesis, we mainly formulate and solve a two-to-one lane-merging prob-

doi:10.6342/NTU202303752

lem with different waiting times. We propose a DP-based algorithm considering lane changing to solve the problem and try to optimize the passing times of all vehicles. We also formulate and solve the M-N load-balancing problem. We propose an MILP approach to assign outgoing lanes for all vehicles and come up with a heuristic approach to improve the passing time.

For future work, we consider the following directions:

- *Simulation for approaches.* So far, we have only done experiments with code by considering every assumption we make will work. However, there may be unconsidered situations that may lead to bad results. Therefore, simulating our approaches on simulators such as SUMO would bring our approaches closer to reality.

- *Extend the approaches for lane-merging problems.* In this thesis, we only consider two-to-one lane-merging scenarios, but in reality, there may also be three-to-two or four-to-three lane-merging scenarios, and thus scheduling approaches for those scenarios are also worthy of research.

- *Seek a better approach for load-balancing problems.* We only come up with a heuristic approach that reduces the time cost compared to FCFS, so we wish to find a better approach that may reduce more time cost than our heuristic approach.

- *Extend the scenarios to traffic in the real world.* In our approaches, we assume that all vehicles are connected and autonomous. However, there is still a long way to go to reach the time when all vehicles are autonomous vehicles, and thus our approaches may also need adjustment to fit in the real world.

doi:10.6342/NTU202303752

Bibliography

[1] A. Uno, T. Sakaguchi, and S. Tsugawa. A merging control algorithm based on inter-vehicle communication. In *Proceedings 199 IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems (Cat. No.99TH8383)*, pages 783–787, 1999.

[2] Gurulingesh Raravi, Vipul Shingde, Krithi Ramamritham, and Jatin Bharadia. Merge algorithms for intelligent vehicles. In S. Ramesh and Prahladavaradan Sampath, editors, *Next Generation Design and Verification Methodologies for Distributed Embedded Control Systems*, pages 51–65, Dordrecht, 2007. Springer Netherlands.

[3] Tanveer Awal, Lars Kulik, and Kotagiri Ramamohanrao. Optimal traffic merging strategy for communication- and sensor-enabled vehicles. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 1468–1474, 2013.

[4] Huaxin Pei, Shuo Feng, Yi Zhang, and Danya Yao. A cooperative driving strategy for merging at on-ramps based on dynamic programming. *IEEE Transactions on Vehicular Technology*, 68(12):11646–11656, 2019.

[5] Shang-Chien Lin, Hsiang Hsu, Yi-Ting Lin, Chung-Wei Lin, Iris Hui-Ru Jiang, and Changliu Liu. A dynamic programming approach to optimal lane merging of connected and autonomous vehicles. In *2020 IEEE Intelligent Vehicles*

doi:10.6342/NTU202303752

Symposium (IV), pages 349–356, 2020.

[6] Shang-Chien Lin, Chia-Chu Kung, Lee Lin, Chung-Wei Lin, and Iris Hui-Ru Jiang. Efficient mandatory lane changing of connected and autonomous vehicles. In *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*, pages 1–7, 2021.

[7] Chaoyi Chen, Qing Xu, Mengchi Cai, Jiawei Wang, Biao Xu, Xiangbin Wu, Jianqiang Wang, Keqiang Li, and Chunyu Qi. A graph-based conflict-free cooperation method for intelligent electric vehicles at unsignalized intersections. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 52–57, 2021.

[8] Huile Xu, Yi Zhang, Li Li, and Weixia Li. Cooperative driving at unsignalized intersections using tree search. *IEEE Transactions on Intelligent Transportation Systems*, 21(11):4563–4571, 2020.

[9] Li Li, Fei-Yue Wang, and Yi Zhang. Cooperative driving at lane closures. In *2007 IEEE Intelligent Vehicles Symposium*, pages 1156–1161, 2007.

[10] Maksat Atagoziyev, Klaus W. Schmidt, and Ece G. Schmidt. Lane change scheduling for autonomous vehicles**this work was supported by the scientific and technological research council of turkey (tubitak) [award 115e372]. *IFAC-PapersOnLine*, 49(3):61–66, 2016.

[11] Tanveer Awal, Manzur Murshed, and Mortuza Ali. An efficient cooperative lane-changing algorithm for sensor- and communication-enabled automated vehicles. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 1328–1333, 2015.

doi:10.6342/NTU202303752

[12] Vicente Milanes, Jorge Godoy, Jorge Villagra, and Joshué Perez. Automated on-ramp merging system for congested traffic situations. *IEEE Transactions on Intelligent Transportation Systems*, 12(2):500–508, 2011.

[13] Ioannis A. Ntousakis, Ioannis K. Nikolos, and Markos Papageorgiou. Optimal vehicle trajectory planning in the context of cooperative merging on highways. *Transportation Research Part C: Emerging Technologies*, 71:464–488, 2016.

[14] A. V. S. Sai Bhargav Kumar, Adarsh Modh, Mithun Babu, Bharath Gopalakrishnan, and K. Madhava Krishna. A novel lane merging framework with probabilistic risk based lane selection using time scaled collision cone. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1406–1411, 2018.

[15] Na'Shea Wiesner, John Sheppard, and Brian Haberman. Using particle swarm optimization to learn a lane change model for autonomous vehicle merging. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8, 2021.

[16] Gil Domingues, João Cabral, João Mota, Pedro Pontes, Zafeiris Kokkinogenis, and Rosaldo J. F. Rossetti. Traffic simulation of lane-merging of autonomous vehicles in the context of platooning. In *2018 IEEE International Smart Cities Conference (ISC2)*, pages 1–6, 2018.

[17] Shurong Li, Chong Wei, and Ying Wang. A ramp merging strategy for automated vehicles considering vehicle longitudinal and latitudinal dynamics. In *2020 IEEE 5th International Conference on Intelligent Transportation Engineering (ICITE)*, pages 441–445, 2020.

[18] Zhaohui Wang, Shengmin Cui, and Tianyi Yu. Automatic lane change control for intelligent connected vehicles. In *2019 4th International Conference on*

doi:10.6342/NTU202303752

Electromechanical Control Technology and Transportation (ICECTT), pages 286–289, 2019.

[19] Mei Yang, Chunxiao Li, Wen Wu, and Chunyan Qi. A cooperative lane change strategy for improving road safety through V2V communications. In *2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1415–1418, 2022.

[20] Yong-Geon Choi, Kyung-Il Lim, and Jung-Ha Kim. Lane change and path planning of autonomous vehicles using GIS. In *2015 12th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 163–166, 2015.

[21] Bangjun Qiao and Xiaodong Wu. Lane change control of autonomous vehicle with real-time rerouting function. In *2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 1317–1322, 2019.

doi:10.6342/NTU202303752