**Report No. UT-21.04**

# METHODOLOGY FOR EVALUATING INTERSECTION SAFETY & OPERATIONAL PERFORMANCE WITH LEFT-TURN PHASING

**Prepared For:**

Utah Department of Transportation
Research & Innovation Division

**Final Report**
**March 2021**

RESEARCH

UDOT Keeping Utah Moving
Utah Department of Transportation - Research Division
4501 South 2700 West - P.O. Box 148410 - SLC,UT 84114-8410

## DISCLAIMER

The authors alone are responsible for the preparation and accuracy of the information, data, analysis, discussions, recommendations, and conclusions presented herein. The contents do not necessarily reflect the views, opinions, endorsements, or policies of the Utah Department of Transportation or the U.S. Department of Transportation. The Utah Department of Transportation makes no representation or warranty of any kind, and assumes no liability therefore.

## ACKNOWLEDGMENTS

The authors acknowledge the Utah Department of Transportation (UDOT) for funding this research, and the following individuals from UDOT on the Technical Advisory Committee for helping to guide the research:

- Glenn Blackwelder – Safety Operations Engineer
- Lisa Zundel – Traffic Management Division Director
- Travis Jensen – Research Consultant Project Manager
- Jeff Lewis – Safety Programs Engineer
- Ivana Vladisavljevic – Data & Safety Engineer
- Travis Evans – Active Transportation Safety Program Manager
- Matt Luker – Statewide Signal Engineer
- Brian Phillips – UDOT Region 3 Traffic Operations Engineer

# TECHNICAL REPORT ABSTRACT

| 1. Report No. | 2. Government Accession No. | 3. Recipient's Catalog No. |
|---|---|---|
| UT-21.04 | N/A | N/A |
| 4. Title and Subtitle<br>Methodology for Evaluating Intersection Safety & Operational Performance with Left-Turn Phasing | | 5. Report Date<br>March 2021<br>6. Performing Organization Code<br>N/A |
| 7. Author(s)<br>Zhao Zhang, Xianfeng Terry Yang | | 8. Performing Organization Report No.<br>N/A |
| 9. Performing Organization Name and Address<br>University of Utah<br>Department of Civil & Environmental Engineering<br>110 Central Campus Drive, Suite 2000<br>Salt Lake City, UT 84112 | | 10. Work Unit No.<br>5H08441H<br>11. Contract or Grant No.<br>20-8312 |
| 12. Sponsoring Agency Name and Address<br>Utah Department of Transportation<br>4501 South 2700 West<br>P.O. Box 148410<br>Salt Lake City, UT 84114-8410 | | 13. Type of Report & Period Covered<br>Final<br>Sept 2019 to March 2021<br>14. Sponsoring Agency Code<br>UT19.322 |

15. Supplementary Notes

Prepared in cooperation with the Utah Department of Transportation and the U.S. Department of Transportation, Federal Highway Administration

16. Abstract

According to the National Highway Traffic Safety Administration (NHTSA), many fatal and severe injury crashes involve left-turn traffic. Based on the left-turn crash record from the Utah Department of Transportation (UDOT) during 2010-2017, 24,860 crashes involved left-turning vehicles, including 714 suspected serious injuries and 82 fatalities. Moreover, about 88% of them are angle incidents that sometimes cause severe crashes. Therefore, considering permitted left-turn phasing can lead to high crash risks, it is critical to investigate the need of adopting protected left-turn phasing at some intersections. This research project proposes a method to evaluate intersection safety and operational performance with different left-turn phasing treatments. For comparison purposes, the method also quantifies both crash and operational performance by US dollar equivalents. The overall analysis framework includes three core steps when considering a phase change from permitted to protected: 1) Estimate the safety benefit (i.e., reduced left-turn crash cost), 2) Calculate the operational cost (i.e., increased delay cost), and 3) Compute the benefit/cost (B/C) ratio and make the phasing design recommendation. For testing, the proposed method is implemented on four signalized intersections in Utah. The research outcomes could help UDOT make better decisions when facing the tradeoff between operational efficiency and intersection safety in left-turn phasing designs.

| 17. Key Words | | 18. Distribution Statement | | 23. Registrant's Seal |
|---|---|---|---|---|
| Left-turn phasing, traffic signal, permitted-only, protected-only, crash cost, delay cost. | | Not restricted. Available through:<br>UDOT Research Division<br>4501 South 2700 West<br>P.O. Box 148410<br>Salt Lake City, UT 84114-8410<br>www.udot.utah.gov/go/research | | N/A |
| 19. Security Classification (of this report) | 20. Security Classification (of this page) | 21. No. of Pages | 22. Price | |
| Unclassified | Unclassified | 95 | N/A | |

# TABLE OF CONTENTS

## LIST OF TABLES

# LIST OF FIGURES

## LIST OF ACRONYMS

| | |
|---|---|
| ATSPM | Automated Traffic Signal Performance Measures |
| B/C | Benefit/Cost Ratio |
| CMF | Crash Modification Factor |
| CMFC | Crash Modification Factors Clearinghouse |
| CRF | Crash Reduction Factor |
| DOT | Department of Transportation |
| FHWA | Federal Highway Administration |
| FYA | Flashing Yellow Arrow |
| HCM | Highway Capacity Manual |
| HSM | Highway Safety Manual |
| MI | Minor Injury |
| MUTCD | Manual on Uniform Traffic Control Devices |
| NHTSA | National Highway Transportation Safety Administration |
| NI | No Injury |
| NLT | No Left Turn |
| NR | No Road |
| PRM | Permitted Left Turn |
| SI | Severe Injury |
| TAC | Technical Advisory Committee |
| UDOT | Utah Department of Transportation |

# EXECUTIVE SUMMARY

In the United States, about 7% of fatal and 12% of injury crashes involve left-turning vehicles. Based on the left-turn crash data from the Utah Department of Transportation (UDOT) during 2010-2017, 24,860 crashes involved left-turn traffic, including 714 suspected serious injuries and 82 fatalities. Moreover, 88% of left-turn crashes are angle incidents that lead to a disproportionately large number of severe crashes when compared to other crash types. Because permitted left-turn phasing can lead to increased crash risks, it is critical to investigate the need for adopting protected left-turn phasing at some intersections. This research proposes a comprehensive method to evaluate intersection safety and operational performance with different left-turn phasing treatments.

Compared with permitted left-turn phasing, protected left-turn phasing can help reduce the crash rate, but may increase intersection delay. There is a tradeoff between intersection operational efficiency and safety in left-turn phasing designs. Therefore, there is a need to develop a method to assist in the decision-making process. Based on the historical crash data and the delay model in the *Highway Capacity Manual* (HCM), this research introduces an approach to quantify the operation and safety costs by US dollar equivalents. The first step is to estimate the safety benefit (i.e., reduced crash cost) when permitted phasing is replaced by protected phasing at the studied intersection. The second step is to compute the corresponding operational cost (i.e., increased delay cost). Then, the last step is to calculate the benefit/cost (B/C) ratio. If the ratio is larger than one, the developed method would recommend the replacement of permitted left-turn phasing.

For testing the developed method, this research conducted case studies at four intersections in Utah. The results reveal that the delay and safety cost can be quantified and the obtained B/C ratio can help the decision-making process in left-turn phasing designs. Moreover, for intersections with high crash rates, the recommendation is likely to be changing the permitted left-turn signal to protected only. For intersections with low crash rates and high traffic demand, it may be appropriate to operate permitted left-turn phasing during peak hours and protected phasing during off-peak hours.

# 1.0 INTRODUCTION

## 1.1 Problem Statement

How to effectively design traffic signal control systems to improve the safety and mobility of urban streets has long been recognized as a vital issue by the traffic community. As shown in Figure 1.1, the two most common types of crashes experienced at signalized intersections are side-angle crashes and rear-end crashes. Rear-end crashes may be caused by driver distraction or a sudden drop in vehicle speeds. Compared with rear-end crashes, side-angle crashes are more likely to pose higher fatality risks and can cause more severe traffic crashes. One primary cause of side-angle crashes is that left-turning vehicles during permitted phasing time fail to yield to the opposing through traffic. This research project focuses on the discussion of left-turn phasing designs.

**Side-Angle Crash**

**Rear-End Crash**

**Figure 1.1 Two primary types of crashes at intersections**

According to the Federal Highway Administration (FHWA), more than 20% of traffic fatalities in the United States occur at intersections. Although traffic signals can help eliminate most traffic conflicts, left-turn crash risk remains due to the implementation of permitted left-turn phasing. Hence, replacing permitted left-turn phasing with protected phasing can greatly improve intersection safety performance. However, such action would increase the intersection delay in the meantime. To deal with the tradeoff between safety and operational efficiency, there is a need to develop a comprehensive method that can compare intersection safety and operational performance before and after the implementation of protected left-turn phasing.

In practice, left-turn movements are recognized as a type of high-risk movement at signalized intersections, which may result in a disproportionately large number of traffic injuries and fatalities when compared to other movement types. The injury severity and fatality probability in left-turn crashes are relatively high because most of them are side-angle crashes (Wang and Abdel-Aty, 2008). According to the National Highway Traffic Safety Administration (NHTSA), 2,577 (6.8% of total) fatal crashes and 29,500 (12.1% of total) injury crashes involved left-turn movements in 2014 (NHTSA 2015), 2,831 (6.9% of total) fatal crashes and 312,000 (12.3% of total) injury crashes occurred due to left turns in 2015 (NHTSA 2016), and 3,151 (7.2% of total) fatal crashes were made by left-turn movements in 2016 (NHTSA 2018). Based on the left-turn crash data from the Utah Department of Transportation (UDOT), 24,860 crashes involved left-turning traffic during 2010-2017 in Utah, which includes 714 suspected serious injuries and 82 fatalities. Notably, about 88% of left-turn crashes are of the side-angle variety. Therefore, there are some ongoing discussions in UDOT's Traffic and Safety and Traffic Management Divisions regarding the implementation of more protected left-turn phasing at intersections, and thus an evaluation tool that can help decision-making is needed.

## 1.2 Objectives

This research aims to develop a method to make more informed decisions about left-turn phasing designs and to balance operations and safety at signalized intersections in Utah. Strategically, it may help resolve potential conflicts between UDOT's Zero Fatalities and Optimize Mobility goals. Furthermore, this research aims to develop a decision support tool grounded on intersection safety and operational analysis to help UDOT address safety and operational issues. The outcome of this research will help UDOT assess the safety and operational performances of signalized intersections when left-turn phasing is changed from permitted to protected. It will also assist in the decision-making process in terms of when (time of day, day of week), how (with specific traffic demand and speed level), and where (the intersection location) the protected phasing may be adopted.

## 1.3 Scope

The scope of this research is divided into several phases, including preliminary investigation, literature review, safety cost estimation, intersection delay calculation, experimental study and results analysis, and conclusions. Each of these is described in the following subsections.

#### 1.3.1 Preliminary Investigation

In the early stage of the project, the research team and the Technical Advisory Committee (TAC) members discussed the data availability for the project, reviewed the scope and schedule, conducted preliminary investigations on the expected project outcomes, and determined the potential risks associated with the project. This meeting included members of the University of Utah and engineers from UDOT.

#### 1.3.2 Literature Review

The second phase of this project is to conduct a comprehensive literature review of existing studies related to effective left-turn signal treatments and crash modification factors (CMF) for left-turn countermeasures. Results of this phase are presented in Chapter 2.

#### 1.3.3 Safety Cost Estimation

The third phase of this project involves the estimation of the safety benefit of adopting protected left-turn phasing. The tasks include:

- Summarize left-turn phasing types that are widely used in Utah.
- Collect left-turn crash data and determine the costs for each type of crash.
- Compute the safety benefit (i.e., reduced crash cost) when left-turn phasing is changed from permitted to protected.

#### 1.3.4 Intersection Delay Calculation

The fourth phase of this project develops a method to compute the increased intersection delay when replacing permitted phasing with protected phasing. Delay equations from the Highway Capacity Manual (HCM) are implemented to estimate intersection average delay with different signal phasing plans. Results of this phase are presented in chapter 3.

#### 1.3.5 Experimental Study and Results Analysis

The fifth phase of this project is to conduct case studies and perform results analysis. The research team implemented the proposed method to study four signalized intersections in Utah and compared the reduced crash costs and increased delay costs after changing from permitted to protected left-turn phasing. Results of this phase are presented in chapter 4.

1.3.6 Recommendations and Conclusions

In this phase, the research team summarizes the key research findings and makes recommendations for left-turn phasing design, based on results of the above phases. Conclusions and recommendations are provided in Chapter 5.

## 1.4 Outline of Report

This project report is organized with the following chapters:

- Introduction
- Literature Review
- Methodology
- Experimental Analysis
- Conclusions

# 2.0 LITERATURE REVIEW

## 2.1 Overview

The research team conducted a comprehensive literature review related to left-turn signal phasing designs. This chapter summarizes the findings of existing studies in three areas: (1) left-turning traffic signal control strategies for reducing intersection crash risks; (2) determination of CMFs for left-turn crash analysis; and (3) the frequency and severity of crashes that involve left-turn traffic in Utah.

## 2.2 Studies on Left-Turn Traffic Signal Changing Strategies

Left-turn signal control phases are commonly implemented at signalized intersections with high left-turning traffic volumes (Agent and Deen, 1979; Stamatiadis et al., 1997; Al-Kaisy and Stewart, 2001). Common left-turn phasing designs include permitted-only, protected-only, and protected/permitted. According to the *Manual on Uniform Traffic Control Devices* (MUTCD), left-turning vehicles always need to yield to opposing traffic when the permitted-only phase is implemented. The protected-only phase provides an exclusive duration for left-turn traffic, which allows vehicles to make left turns only when a green signal indication (e.g., green left-turn arrow) is displayed. The protected/permitted phase combines the permitted-only and protected-only phases in the same signal cycle. Vehicles are allowed to make left-turns either on a green arrow indication (protected-only phase) or on the circular green (permitted-only phase), during which they must yield to opposing traffic.

The protected-only phase can effectively avoid conflicts between left-turn vehicles and opposing through traffic and can eliminate vehicle conflicts with pedestrians. Hence, to reduce the potential risks of causing side-angle crashes at intersections, three possible signal-changing strategies can be implemented: (1) change permitted-only to protected-only, (2) change permitted-only to protected/permitted, and (3) change protected/permitted to protected-only.

Several existing studies demonstrate that protected-only and protected/permitted phasing designs at signalized intersections are effective for reducing the left-turn crash rate. Researchers at the FHWA (FHWA 2010) conducted some “before-and-after” case studies by installing protected/permitted left-turn signals at three intersections. The study results showed that protected/permitted left-turn phasing cumulatively decreased left-turn head-on crashes by 84%, injury crashes by 58.9%, and total crashes by 32% over the one-year study period. It demonstrated that the effects of implementing protected/permitted signal phasing are consistent with the projection of the *Highway Safety Manual* (HSM), in terms of crash rate reduction.

Chen et al. (2015) evaluated the safety impacts of changing permitted left-turn phasing to protected/permitted or protected-only at 68 intersections in New York City. Using a rigorous quasi-experimental design accompanied with regression modeling, their study results showed that protected left-turn phasing is valid for reducing left-turn crashes and pedestrian-involved crashes. Srinivasan et al. (2008) conducted a "before-and-after" evaluation using the empirical Bayes methodology at signalized intersections, based on data collected from Winston-Salem, North Carolina. The results demonstrated that changing from permitted or protected/permitted to protected-only can significantly reduce left-turn crashes.

One UDOT project completed by Medina et al. (2018) showed that changing from protected-only to a flashing yellow arrow would increase crashes. Other studies (Al-Kaisy and Stewart, 2001; Golias and Porikou, 2004; Ozmen et al., 2014; Stamatiadis et al., 2015; Stamatiadis et al., 2016; Yang et al., 2018) focused on the study of benefits of protected left-turn phasing from different perspectives. All studies are very helpful in assessing the effectiveness of protected left-turn phasing and determining how, where, and when to implement it.

Moreover, Hedges (2014) conducted a review of 40 state signal design policies, which indicated that 24 Departments of Transportation (DOTs) have such a policy in place and 17 DOTs use crash records and intersection volumes for left-turn phasing selection. Left-turn volume, opposing through volume, cross product of left-turn and opposing through volume, delay, crash history, number of left-turn lanes, and number of opposing through lanes are important factors for selecting appropriate left-turn signal phasing (Agent, 1979, 1985; Agent et al., 1995; Asante et al., 1993; Cottrell, 1986; ITE Florida Section, 1982; Lalani et al., 1986; McShane and Roess, 1990; Stamatiadis et al., 1997; Upchurch, 1986). The summary of widely-used criteria related to selecting left-turn phasing is provided as follows:

1. Traffic volume: Agent (1985) suggested that protected-only phasing should be used if the volume cross product is greater than 50,000 vph for one opposing through lane and greater than 100,000 vph for two opposing through lanes. However, Upchurch (1986) used a different critical value of 144,000 for two opposing through lanes and 100,000 for three opposing through lanes.

2. Crash history: The number of historical crashes at signalized intersections is a fundamental factor in determining left-turn phasing. Efficiency and safety must be weighed against one another for choosing appropriate left-turn phasing. The number of historical crashes directly or indirectly caused by left-turn movements is a basic criterion in most guidelines (Agent, 1979, 1985; Cottrell, 1986; Stamatiadis et al., 1997). However, there is no uniform threshold of historical crashes for determining protected-only left-turn phasing.

3. Number of lanes: Protected-only phasing should be installed when the number of exclusive left-turn lanes is greater than one (Agent, 1985; ITE Florida Section, 1982). Additionally, protected-only phasing is also recommended when there are three or more opposing through lanes (Agent, 1985).

4. Delay: Left-turn delay is identified as a critical factor for selecting left-turn phasing. For example, protected-only phasing is required when there are greater than two vehicle-hours of left-turn delay or if the average delay is greater than 35 s/veh for left-turn vehicles (Agent, 1979).

### 2.3 CMF for Left-Turn Crash Analysis

Countermeasures for reducing left-turn crashes need to be assessed after their implementation. In practice, traffic safety divisions/offices in state DOTs often use CMFs or Crash Reduction Factors (CRFs) for such needs. CMFs are implemented to compute the expected number of crashes after implementing a countermeasure on a road or an intersection. CRF measures associated crash numbers to be reduced. The Crash Modification Factors Clearinghouse (CMFC) provides guidance to researchers on best practices for developing high quality CMFs. In the literature, there is a set of existing studies that focuses on the investigation of different countermeasures for reducing left-turn crashes and some corresponding CMFs and CRFs are generated for left-turn crash study. Tables 2.1-2.6 summarize existing countermeasure studies, where most show a CMF value of 0.01 for left-turn crashes and a CRF value of 99%. In Utah, a 90% CRF is commonly used and it is used in this project.

**Table 2.1 Change from permitted-only or protected/permitted to protected-only**

| CMF | CRF (%) | Quality | Crash type | Crash severity | Area type | Reference | Comments |
|---|---|---|---|---|---|---|---|
| 0.01 | 99 | **** | Angle | All | Urban | (Harkey et al., 2008) | Left-turn crashes significantly decreased. |

**Table 2.2 Change from permitted-only to protected-only on minor approaches**

| CMF | CRF (%) | Quality | Crash type | Crash severity | Area type | Reference | Comments |
|---|---|---|---|---|---|---|---|
| 0.01 | 99 | *** | Angle | All | Urban | (Davis and Aul, 2007) | Left-turn crashes significantly decreased. |

**Table 2.3 Change from protected/permitted to protected on major approaches**

| CMF | CRF (%) | Quality | Crash type | Crash severity | Area type | Reference | Comments |
|---|---|---|---|---|---|---|---|
| 0.01 | 99 | **** | Angle | All | Urban | (Davis and Aul, 2007) | Left-turn crashes significantly decreased. |

**Table 2.4 Change from protected/permitted to protected-only on minor approaches**

| CMF | CRF (%) | Quality | Crash type | Crash severity | Area type | Reference | Comments |
|---|---|---|---|---|---|---|---|
| 0.03 | 97 | *** | Angle | All | Urban | (Davis and Aul, 2007) | Left-turn crashes significantly decreased. |

**Table 2.5 Change from permitted-only to protected-only on one or more approaches**

| CMF | CRF (%) | Quality | Crash type | Crash severity | Area type | Reference | Comment |
|---|---|---|---|---|---|---|---|
| 0.01 | 99 | *** | Left turn | All | Urban | (AASHTO, 2010) | Left-turn crashes significantly decreased. |

**Table 2.6 Change from permitted-only to protected-only on all approaches**

| CMF | CRF (%) | Quality | Crash type | Crash severity | Area type | Reference | Comment |
|---|---|---|---|---|---|---|---|
| 0.021 | 97.9 | **** | Angle | All | Urban | (Srinivasan et al., 2008) | Left-turn crashes significantly decreased. |

## 2.4 Left-Turn Crashes in Utah

Zhang et al. (2020) conducted research to examine the injury severity in left-turn crashes. That study studied left-turn crashes from 2010 to 2017 in Utah. In this study, possible and suspected minor injury are combined as minor injury level, and suspected severe injuries and fatal are combined as severe injury level for yielding a statistically meaningful sample size. Therefore, the driver injury severity is recategorized into three levels including NI (no injury), MI (Minor injury), and SI (severe injury). The descriptive statistical summary of left-turn crash data is presented in Table 2.7.

## Table 2.7 Statistical description of crash injury severity

| Variable | Driver Injury Severity | | | | | | Total |
|---|---|---|---|---|---|---|---|
| | NI | | MI | | SI | | |
| Severity | 10875 | 53.22% | 8844 | 43.28% | 714 | 3.49% | 20433 |
| **Driver characteristics** | | | | | | | |
| Age | | | | | | | |
| Young (< 25 years) | 4195 | 38.57% | 3359 | 37.98% | 226 | 31.65% | 7780 |
| Middle (25 - 55 years) | 4590 | 42.21% | 3805 | 43.02% | 329 | 46.08% | 8724 |
| Old (> 55 years) | 2090 | 19.22% | 1680 | 19.00% | 159 | 22.27% | 3929 |
| Gender | | | | | | | |
| Male | 5661 | 52.06% | 4333 | 48.99% | 345 | 48.32% | 10339 |
| Female | 5214 | 47.94% | 4511 | 51.01% | 369 | 51.68% | 10094 |
| Alcohol | 130 | 1.20% | 167 | 1.89% | 29 | 4.06% | 326 |
| Drug | 39 | 0.36% | 57 | 0.64% | 25 | 3.50% | 121 |
| Disregard traffic control device | 2492 | 22.91% | 2256 | 25.51% | 181 | 25.35% | 4929 |
| **Crash Characteristics** | | | | | | | |
| Day | | | | | | | |
| Sunday | 663 | 6.10% | 590 | 6.67% | 54 | 7.56% | 1307 |
| Monday | 1708 | 15.71% | 1356 | 15.33% | 104 | 14.57% | 3168 |
| Tuesday | 1716 | 15.78% | 1470 | 16.62% | 108 | 15.13% | 3294 |
| Wednesday | 1767 | 16.25% | 1404 | 15.88% | 128 | 17.93% | 3299 |
| Thursday | 1744 | 16.04% | 1440 | 16.28% | 103 | 14.43% | 3287 |
| Friday | 1928 | 17.73% | 1486 | 16.80% | 132 | 18.49% | 3546 |
| Saturday | 1349 | 12.40% | 1098 | 12.42% | 85 | 11.90% | 2532 |
| Time Period | | | | | | | |
| Peak hour (7:00am – 10:00am & 4:00pm – 7:00pm) | 4233 | 38.92% | 3344 | 37.81% | 254 | 35.57% | 7831 |
| Daytime (10:00am – 4:00pm) | 3811 | 35.04% | 3118 | 35.26% | 278 | 38.94% | 7207 |
| Night (7:00pm – 7:00am) | 2823 | 25.96% | 2382 | 26.93% | 182 | 25.49% | 5387 |
| Collision type | | | | | | | |
| Angle | 9886 | 90.91% | 7607 | 86.01% | 578 | 80.95% | 18071 |
| Head-on | 989 | 9.09% | 1237 | 13.99% | 136 | 19.05% | 2362 |
| Number of vehicles involved | | | | | | | |
| Two vehicles | 10284 | 94.57% | 7771 | 87.87% | 582 | 81.51% | 18637 |
| Multiple vehicles (> 2 vehicles) | 591 | 5.43% | 1073 | 12.13% | 132 | 18.49% | 1796 |
| Speed limit | | | | | | | |
| Low (< 35mph) | 999 | 9.19% | 660 | 7.46% | 39 | 5.46% | 1698 |
| Middle (35 – 45mph) | 8622 | 79.28% | 6964 | 78.74% | 529 | 74.09% | 16115 |
| High (> 45mph) | 1254 | 11.53% | 1220 | 13.79% | 146 | 20.45% | 2620 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Traffic control device | | | | | | | |
| Flashing Traffic Control Signal | 7 | 0.06% | 2 | 0.02% | 0 | 0.00% | 9 |
| Traffic Control Signal | 10868 | 99.91% | 8838 | 99.93% | 714 | 10.00% | 20417 |
| Log AADT | | | | | | | |
| Less than 3.5 | 246 | 2.26% | 223 | 2.52% | 21 | 2.94% | 490 |
| Larger than 3.5 | 10629 | 97.74% | 8621 | 97.48% | 693 | 97.06% | 19943 |
| Protected left-turn phasing | 7394 | 67.99% | 6194 | 70.04% | 508 | 71.15% | 14096 |
| Right-turn overlap | 493 | 4.53% | 443 | 5.01% | 26 | 3.64% | 962 |
| **Environmental factors** | | | | | | | |
| Weather | | | | | | | |
| Clear | 8020 | 73.75% | 6576 | 74.36% | 550 | 77.03% | 15146 |
| Rain | 594 | 5.46% | 481 | 5.44% | 32 | 4.48% | 1107 |
| Snow | 445 | 4.09% | 217 | 2.45% | 10 | 1.40% | 672 |
| Cloudy/windy | 1783 | 16.40% | 1543 | 17.45% | 121 | 16.95% | 3447 |
| Fog | 33 | 0.30% | 27 | 0.31% | 1 | 0.14% | 61 |
| Light condition | | | | | | | |
| light | 9833 | 90.42% | 8028 | 90.77% | 638 | 89.36% | 18499 |
| Dark | 1042 | 9.58% | 816 | 9.23% | 76 | 10.64% | 1934 |
| **Roadway attributes** | | | | | | | |
| Road surface Condition | | | | | | | |
| Dry | 9264 | 85.19% | 7736 | 87.47% | 643 | 90.06% | 17643 |
| Wet | 1611 | 14.81% | 1108 | 12.53% | 71 | 9.94% | 2790 |
| Roadway junction feature | | | | | | | |
| 4-leg intersection | 9230 | 84.84% | 7580 | 85.71% | 622 | 87.11% | 17428 |
| 5-leg or more intersection | 68 | 0.63% | 55 | 0.62% | 5 | 0.70% | 128 |
| Ramp intersection with crossroad | 559 | 5.14% | 376 | 4.25% | 22 | 3.08% | 957 |
| T intersection | 1010 | 9.29% | 826 | 9.34% | 65 | 9.10% | 1901 |
| Y intersection | 8 | 0.07% | 6 | 0.07% | 0 | 0.00% | 14 |
| Road type | | | | | | | |
| State | 7743 | 71.20% | 6514 | 73.65% | 539 | 75.49% | 14796 |
| Federal Aid | 3132 | 28.80% | 2330 | 26.35% | 175 | 24.51% | 5637 |
| Area | | | | | | | |
| Urban | 10851 | 99.78% | 8832 | 99.86% | 713 | 99.86% | 20396 |
| Rural | 24 | 0.22% | 12 | 0.14% | 1 | 0.14% | 37 |
| Number of left-turn lanes | | | | | | | |
| 1 – 3 lanes | 10133 | 93.18% | 8338 | 94.28% | 684 | 95.80% | 19155 |
| More than 3 lanes | 742 | 6.82% | 506 | 5.72% | 30 | 4.20% | 1278 |

## 2.5 Summary

This chapter conducted a comprehensive literature review on the studies that analyzed left-turn crashes and evaluated crash reduction countermeasures. The reviewed research is categorized into three areas. The first area covers existing studies relevant to left-turn traffic signal control strategies. The second area involves the CMF and CRF calculations for left-turn crash reduction countermeasures. The third area includes a comprehensive study of the historical left-turn crashes in Utah.

In summary, despite many related studies that have been conducted in the literature, how to develop an effective tool to assist the left-turn phasing selection remains unsolved in Utah. To fill this research gap, this research project focuses on developing a methodology to help evaluate intersection safety and operational performances with different left-turn phasing designs. In the remainder of this report, Chapter 3 introduces the method for computing benefits and costs of replacing permitted phasing with protected phasing, and the corresponding B/C ratio, Chapter 4 analyzes the case studies results. Chapter 5 summarizes all the findings of this research.

# 3.0 METHODOLOGY

## 3.1 Overview

As shown in the literature review, safety performance at intersections is greatly impacted by the signal phasing design. Particularly, the selection of left-turn phasing plays a key role in affecting left-turn crash frequency and severity. Hence, for intersections with a large number of historical left-turn crashes, it is often suggested to change the left-turn phasing from permitted to protected during high-risk periods. To quantify the benefit and cost of making such a change, both crash data and traffic signal settings in Utah are collected to estimate the reduced crash cost and the increased operational cost. The reduced crash cost can be directly measured with the number of left-turn crashes and the average cost per crash. The operational cost is calculated by accounting for both signal replacement cost and increased delay cost, where the delay cost is often hard to estimate.

To tackle this difficulty, this research first proposes a method to calculate the intersection total delay under different signal timing plans. Then, the increased delay after the replacement of left-turn phasing is obtained and the corresponding delay cost is calculated by using an average monetary value of time. When both safety benefit (reduced crash cost) and operational costs are available, the left-turn phasing decision can be informed by examining the B/C ratio.

## 3.2 Safety Benefit (Reduced Crash Cost) Calculation

### 3.2.1 Left-Turn Crash Data

To support the reduced crash cost calculation, this research obtained eight-year (2010-2017) left-turn crash data from the UDOT Traffic and Safety Division. In the safety database, crash injury severity is categorized by five groups: “Fatal”, “Suspected Serious Injury”, “Suspected Minor Injury”, “Possible Injury”, and “No Injury”. All crash injury severities are mapped to specific intersections by signal ID. Some left-turn data samples are shown in Table 3.1.

**Table 3.1 Left-Turn crash data sample**

| Signal ID | Fatal | Suspected Serious Injury | Suspected Minor Injury | Possible Injury | No Injury |
|---|---|---|---|---|---|
| 1001 | 0 | 1 | 3 | 3 | 4 |
| 1022 | 0 | 0 | 0 | 1 | 1 |
| 1031 | 0 | 0 | 0 | 0 | 0 |
| 1125 | 0 | 0 | 1 | 0 | 2 |
| 1129 | 0 | 0 | 1 | 1 | 2 |
| 1215 | 0 | 0 | 1 | 1 | 4 |
| 5000 | 0 | 1 | 1 | 3 | 6 |
| 5001 | 0 | 2 | 16 | 14 | 36 |
| 5002 | 0 | 2 | 7 | 8 | 21 |
| 5003 | 0 | 0 | 1 | 0 | 5 |
| 5004 | 0 | 1 | 13 | 17 | 21 |
| 5005 | 0 | 0 | 3 | 3 | 7 |
| 5006 | 1 | 1 | 9 | 4 | 14 |
| 5007 | 0 | 0 | 2 | 2 | 3 |
| 5008 | 0 | 1 | 1 | 1 | 4 |
| 5009 | 0 | 0 | 3 | 6 | 13 |
| 5010 | 0 | 0 | 1 | 1 | 2 |
| 5011 | 0 | 1 | 1 | 0 | 3 |
| 5012 | 0 | 0 | 7 | 4 | 16 |
| 5013 | 0 | 0 | 0 | 1 | 1 |
| 5014 | 0 | 2 | 6 | 5 | 7 |
| 5015 | 0 | 1 | 1 | 0 | 2 |
| 5016 | 0 | 1 | 3 | 0 | 9 |

### 3.2.2 Crash Cost Standard

As the crash data are collected over eight years, the left-turn crash cost calculation is based on the crash cost values. Table 3.2 shows the historical crash cost values implemented by UDOT since 2010. In this study, the most recent values (2018) are used.

**Table 3.2 Utah historical crash cost by year**

| Evaluation Year | Fatal | Suspected Serious Injury | Suspected Minor Injury | Possible Injury | No Injury |
|---|---|---|---|---|---|
| 2010 | $785,000 | $785,000 | $80,000 | $42,000 | $4,400 |
| 2011 | $1,879,400 | $1,879,400 | $117,300 | $59,900 | $3,000 |
| 2012 | $1,899,400 | $1,899,400 | $118,500 | $60,500 | $3,100 |
| 2013 | $1,915,700 | $1,915,700 | $119,600 | $61,000 | $3,100 |
| 2014 | $1,940,300 | $1,940,300 | $121,100 | $61,800 | $3,100 |
| 2015 | $1,961,100 | $1,961,100 | $122,400 | $62,500 | $3,200 |
| 2016 | $1,962,100 | $1,962,100 | $122,400 | $62,500 | $3,200 |
| 2017 | $2,064,000 | $2,064,000 | $128,900 | $65,800 | $3,300 |
| 2018 | $2,133,100 | $2,133,100 | $133,100 | $68,000 | $3,400 |

### 3.2.3 Reduced Crash Cost Calculation

Based on the left-turn crash data and 2018 crash cost values, the reduced left-turn crash cost can be calculated by multiplying the number of each severity of left-turn crashes by the associated crash cost value and CRF as shown by the following equation:

$$\text{Reduced crash cost} = \sum_i^5 I_i * r_i * CRF \quad (3.1)$$

where $I_i$ denotes the number of each severity of crashes and $r_i$ denotes the associated crash cost value. The CRF is 0.9, which assumes that 90% of left-turn crashes can be prevented by protected left-turn phasing. Table 3.3 shows the calculated results based on the crash data sample shown in Table 3.1 and crash cost values shown in Table 3.2.

**Table 3.3 Sample of reduced crash cost calculation**

| Signal ID | Fatal | Suspected Serious Injury | Suspected Minor Injury | Possible Injury | No Injury | Reduced Crash Cost |
|---|---|---|---|---|---|---|
| 1001 | 0 | 1 | 3 | 3 | 4 | $2,475,000 |
| 1022 | 0 | 0 | 0 | 1 | 1 | $64,260 |
| 1031 | 0 | 0 | 0 | 0 | 0 | $0 |
| 1125 | 0 | 0 | 1 | 0 | 2 | $125,910 |
| 1129 | 0 | 0 | 1 | 1 | 2 | $187,110 |
| 1215 | 0 | 0 | 1 | 1 | 4 | $193,230 |
| 5000 | 0 | 1 | 1 | 3 | 6 | $2,241,540 |
| 5001 | 0 | 2 | 16 | 14 | 36 | $6,723,180 |
| 5002 | 0 | 2 | 7 | 8 | 21 | $5,231,970 |
| 5003 | 0 | 0 | 1 | 0 | 5 | $135,090 |
| 5004 | 0 | 1 | 13 | 17 | 21 | $4,581,720 |
| 5005 | 0 | 0 | 3 | 3 | 7 | $564,390 |
| 5006 | 1 | 1 | 9 | 4 | 14 | $5,205,330 |
| 5007 | 0 | 0 | 2 | 2 | 3 | $371,160 |
| 5008 | 0 | 1 | 1 | 1 | 4 | $2,113,020 |
| 5009 | 0 | 0 | 3 | 6 | 13 | $766,350 |
| 5010 | 0 | 0 | 1 | 1 | 2 | $187,110 |
| 5011 | 0 | 1 | 1 | 0 | 3 | $2,048,760 |
| 5012 | 0 | 0 | 7 | 4 | 16 | $1,132,290 |
| 5013 | 0 | 0 | 0 | 1 | 1 | $64,260 |
| 5014 | 0 | 2 | 6 | 5 | 7 | $4,885,740 |
| 5015 | 0 | 1 | 1 | 0 | 2 | $2,045,700 |
| 5016 | 0 | 1 | 3 | 0 | 9 | $2,306,700 |

## 3.3 Replacement Cost Calculation

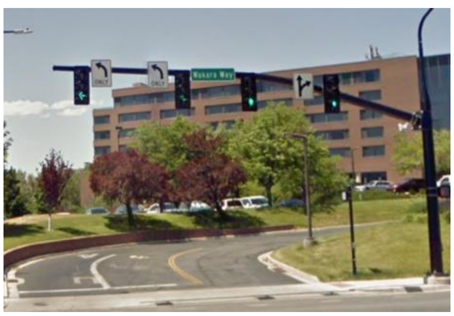
### 3.3.1 Summary of Left-Turn Signal Types in Utah

Commonly used left-turn signals in Utah include “doghouse” style protected/permitted signal, protected-only signals, flashing yellow arrow (FYA) signals, permitted-only signals, and no left turn. Table 3.4 lists the reference code and definition of each left-turn signal type. Figure 3.1 shows left-turn signal designs.

**Table 3.4 All types of left-turn signal in Utah**

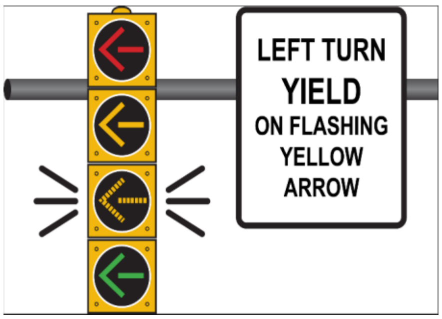
| Signal type | Definition |
|---|---|
| PRM | Permitted 3-section head |
| P1 | Protected 3-section head, 1 lane |
| P2 | Protected 3-section head, 2 lanes |
| T5 | Doghouse protected/permitted |
| FYA | Flashing yellow arrow |
| TPROT | Tee approach (no opposing through, green arrow) |
| TPRM | Tee approach (no opposing through, ped phase w/LT) |
| NLT | No left turn |
| NR | No road (includes opposing direction on one-way) |

(a) Permitted only



(b) Protected only

(c) Doghouse protected/permitted




(d) FYA

(e) No left turn

**Figure 3.1 The most common types of left-turn signals in Utah**

### 3.3.2 Left-Turn Signal Information and Signal ID

This research extracts all required left-turn signal information, along with signal ID, from the UDOT Signal Timing Routine Maintenance database. Table 3.5 shows some collected data samples.

**Table 3.5 Left-turn signal type by signal ID in Utah**

| Signal ID | NB | SB | EB | WB |
|---|---|---|---|---|
| 1001 | PRM | PRM | PRM | PRM |
| 1022 | FYA | FYA | FYA | FYA |
| 1031 | NLT | PRM | TEE | NR |
| 1125 | PRM | PRM | T5 | T5 |
| 1129 | PRM | PRM | T5 | T5 |
| 1215 | PRM | FYA | P1 | P1 |
| 5000 | P1 | P1 | FYA | P2 |
| 5001 | T5 | T5 | T5 | T5 |
| 5002 | P2 | P1 | FYA | P2 |
| 5003 | P2 | P2 | P2 | P2 |
| 5004 | FYA | FYA | FYA | FYA |
| 5005 | T5 | NLT | NR | TPROT |
| 5006 | T5 | T5 | T5 | T5 |
| 5007 | P2 | P1 | PRM | NR |
| 5008 | T5 | T5 | T5 | T5 |
| 5009 | P2 | P2 | P2 | P1 |
| 5010 | PRM | PRM | PRM | PRM |
| 5011 | NLT | NLT | TPROT | NR |
| 5012 | PRM | PRM | PRM | PRM |
| 5013 | PRM | PRM | PRM | PRM |
| 5014 | T5 | T5 | NR | PRM |
| 5015 | NLT | PRM | PRM | PRM |
| 5016 | PRM | PRM | PRM | PRM |

### 3.3.3 Replacement Cost Calculation

When changing the left-turn phasing from one type to another, some signal replacement costs may be involved. Table 3.6 shows the cost rate of each phasing replacement strategy that has been adopted by UDOT.

**Table 3.6 Replacement cost rate**

| Replacement Strategy | Cost Rate |
|---|---|
| PRM to Prot | $24,000 |
| T5 to Prot | $9,500 |
| FYA to Prot | $0 |

Based on the listed cost rate, the left-turn phasing replacement cost can be calculated by the following equation:

$$\text{Replacement cost} = \sum_i^4 s_i * c_i \tag{3.2}$$

where $s_i$ denotes the replacement strategy for intersection approach *i* and $c_i$ denotes the corresponding replacement cost rate. Given the equation, Table 3.7 presents the replacement costs of installing protected left-turn signals at the sample intersections.

**Table 3.7 Replacement cost by signal ID in Utah**

| Signal ID | NB | SB | EB | WB | Replacement cost |
|---|---|---|---|---|---|
| 1001 | PRM | PRM | PRM | PRM | $96,000 |
| 1003 | T5 | T5 | T5 | T5 | $38,000 |
| 1004 | PRM | PRM | PRM | PRM | $96,000 |
| 1005 | PRM | PRM | PRM | PRM | $96,000 |
| 1006 | PRM | PRM | PRM | PRM | $96,000 |
| 1010 | PRM | PRM | PRM | PRM | $96,000 |
| 1022 | FYA | FYA | FYA | FYA | $0 |
| 1125 | PRM | PRM | T5 | T5 | $67,000 |
| 1129 | PRM | PRM | T5 | T5 | $67,000 |
| 1152 | P2 | FYA | FYA | FYA | $0 |
| 1213 | PRM | PRM | P1 | P1 | $48,000 |
| 1215 | PRM | FYA | P1 | P1 | $24,000 |
| 5000 | P1 | P1 | FYA | P2 | $0 |
| 5001 | T5 | T5 | T5 | T5 | $38,000 |
| 5002 | P2 | P1 | FYA | P2 | $0 |
| 5003 | P2 | P2 | P2 | P2 | $0 |
| 5004 | FYA | FYA | FYA | FYA | $0 |
| 5006 | T5 | T5 | T5 | T5 | $38,000 |
| 5008 | T5 | T5 | T5 | T5 | $38,000 |
| 5009 | P2 | P2 | P2 | P1 | $0 |
| 5010 | PRM | PRM | PRM | PRM | $96,000 |

| | | | | | |
|---|---|---|---|---|---|
| 5012 | PRM | PRM | PRM | PRM | $96,000 |
| 5013 | PRM | PRM | PRM | PRM | $96,000 |
| 5015 | NLT | PRM | PRM | PRM | $72,000 |
| 5016 | PRM | PRM | PRM | PRM | $96,000 |

## 3.4 Intersection Delay Calculation

When changing the left-turn phasing from one type to another, the intersection delay will be impacted. Also, it is expected that the intersection delay will be increased when a permitted phase is updated to a protected phase.

### 3.4.1 The Framework of Intersection Delay Calculation

To estimate the increased intersection delay when changing the left-turn phasing from permitted to protected, the proposed method leverages delay equations introduced in the HCM (HCM, 2016). Figure 3.2 shows the flow chart of the intersection delay calculation method. This method includes traffic volumes, phase plans, lane utilization, and left-turn treatment alternatives as the inputs.

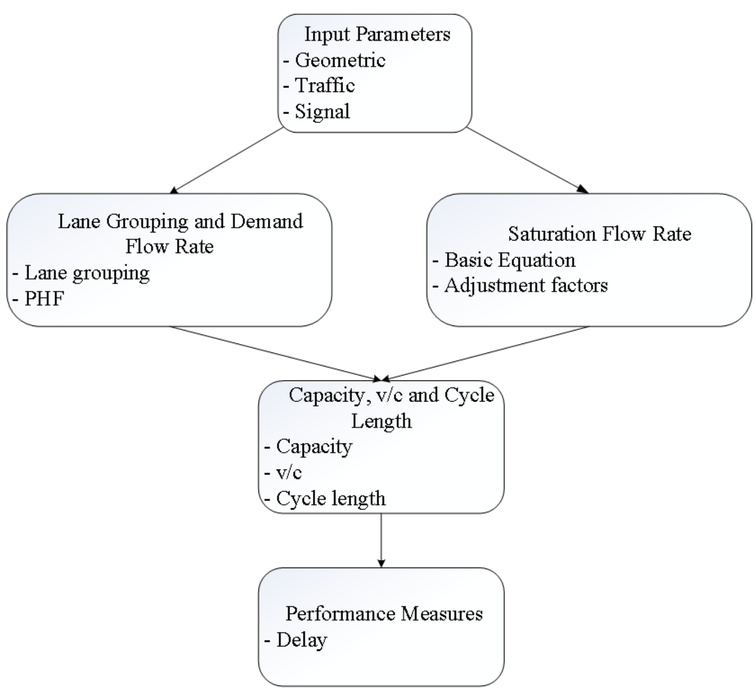



**Figure 3.2 Signalized intersection delay calculation (HCM, 2016)**

Table 3.8 provides a summary of the required input information for calculating signalized intersection delays. The data can be generally classified into three main categories: geometric conditions, traffic conditions, and signalization conditions.

**Table 3.8 Input data needs for each analysis lane group (HCM, 2016)**

| Type of Condition | Parameter |
| --- | --- |
| Geometric conditions | Area type<br>Number of lanes, N<br>Average lane width, W (ft)<br>Grade, G (%)<br>Existence of exclusive LT or RT lanes<br>Length of storage bay, LT or RT lane, $L_s$ (ft)<br>Parking |
| Traffic conditions | Demand volume by movement, V (veh/h)<br>Base saturation flow rate, $s_o$ (pc/h/ln)<br>Peak-hour factor, PHF<br>Percent heavy vehicles, HV (%)<br>Approach pedestrian flow rate, $v_{ped}$ (p/h)<br>Local buses stopping at intersection, $N_B$ (buses/h)<br>Parking activity, $N_m$ (maneuvers/h)<br>Arrival type, AT<br>Proportion of vehicles arriving on green, P<br>Approach speed, $S_A$ (mi/h) |
| Signalization conditions | Cycle length, C (s)<br>Green time, G (s)<br>Yellow-plus-all-red change-and-clearance interval (intergreen), Y (s)<br>Actuated or pretimed operation<br>Pedestrian push-button<br>Minimum pedestrian green, $G_p$ (s)<br>Phase plan<br>Analysis period, T (h) |

3.4.2 Signal Settings

The delay calculation also requires the signal timing plan, which includes a phase diagram illustrating the phase plan, cycle length, green times, and change-and-clearance intervals. The signal timing plan can be collected from UDOT for pretimed signalized intersections. For actuated signalized intersections, the minimum cycle length is computed by Equation (3.3):

$$C_{\min} = \frac{L * X_c}{X_c - \sum_{i=1}^{n} \left(\frac{v}{s}\right)_{ci}} \tag{3.3}$$

where $C_{\min}$ is the minimum necessary cycle length in seconds (typically rounded up to the nearest 5-second increment); $L$ is the total lost time for the cycle in seconds; $X_{\mathrm{c}}$ is the critical $v/c$ ratio for the intersection; $(\frac{v}{s})_{ci}$ is the flow ratio for the critical lane group $i$; and n is the number of critical lane groups. The optimum cycle length is calculated by Equation (3.4):

$$C_{\mathrm{opt}} = \frac{1.5*L+5}{1.0-\sum_{i=1}^{n}(\frac{v}{s})_{ci}} \tag{3.4}$$

3.4.3 Lane Grouping

Segmenting the intersection into lane groups considers both the geometric features of the intersection and the channelization/lane group design. The applied lane grouping guideline based on HCM (2016) is as follows and Table 3.9 shows some common lane groups used for delay analysis:

- An exclusive left-turn lane or lanes should normally be designated as a separate lane group unless there is also a shared left-through lane present, in which case the proper lane grouping will depend on the distribution of traffic volume between the movements. The same is true of an exclusive right-turn lane.
- On approaches with exclusive left-turn or right-turn lanes, or both, all other lanes on the approach would generally be included in a single lane group.
- When an approach with more than one lane includes a lane that may be used by both left-turning vehicles and through vehicles, it is necessary to determine whether equilibrium conditions exist or whether there are so many left turns that the lane essentially acts as an exclusive left-turn lane, which is referred to as a de facto left-turn lane.

**Table 3.9 Typical lane groups for analysis (HCM, 2016)**

| Number of Lanes | Movements by Lanes | Number of Possible Lane Groups |
|---|---|---|
| 1 | LT + TH + RT | ① (Single-lane approach) |
| 2 | EXC LT<br>TH + RT | ② |
| 2 | LT + TH<br>TH + RT | ①<br>OR<br>② |
| 3 | EXC LT<br>TH<br>TH + RT | ②<br>OR<br>③ |

3.4.4 Determining Flow Rate

A peak 15-min flow rate is derived from an hourly volume by dividing the movement volumes by an appropriate PHF, using Equation (3.5).

$$v_p = \frac{V}{PHF} \qquad (3.5)$$

where $v_p$ denotes the flow rate during the peak 15-min period (veh/h); $V$ denotes the hourly volume (veh/h); and $PHF$ denotes the peak-hour factor.

3.4.5 Determining Saturation Flow Rate

A saturation flow rate for each lane group is generated by Equation (3.6). The saturation flow rate is the flow in vehicles per hour that can be accommodated by the lane group assuming that the green phase is displayed 100 percent of the time (i.e., g/C = 1.0).

$$s = s_o N f_w f_{HV} f_g f_p f_{bb} f_a f_{LU} f_{LT} f_{RT} f_{Lpb} f_{wRpb} \tag{3.6}$$

where $s$ denotes the saturation flow rate for the subject lane group, expressed as a total for all lanes in the lane group (veh/h); $s_o$ denotes the base saturation flow rate per lane (pc/h/ln); $N$ is the number of lanes in the lane group; $f_w$ denotes the adjustment factor for lane width; $f_{HV}$ denotes the adjustment factor for heavy vehicles in the traffic stream; $f_g$ denotes the adjustment factor for approach grade; $f_p$ denotes the adjustment factor for existence of a parking lane bus and parking activity adjacent area; $f_{bb}$ denotes the adjustment factor for the blocking effect of local buses that stop within the intersection area; $f_a$ denotes the adjustment factor for area type; $f_{LU}$ denotes the adjustment factor for lane utilization; $f_{LT}$ denotes the adjustment factor for left turns in the lane group; $f_{RT}$ denotes the adjustment factor for right turns in the lane group; $f_{Lpb}$ denotes the pedestrian adjustment factor for left-turn movements; and $f_{wRpb}$ denotes the pedestrian-bicycle adjustment factor for right-turn movements. Computations begin with the selection of a base saturation flow rate, usually 1,900 passenger cars per hour per lane (pc/h/ln). This value can be adjusted for a variety of conditions. The adjustment factors are shown in Table 3.10.

3.4.6 Determining Capacity and V/C Ratio

***Capacity***

Capacity at signalized intersections can be calculated based on flow ratios and the saturation flow rate. The flow ratio for a given lane group is defined as the ratio of the actual or projected demand flow rate for the lane group $v_i$ and the saturation flow rate $s_i$. The flow ratio is denoted as $(\frac{v}{s})_i$ for lane group $i$. The capacity of a given lane group can be calculated by Equation (3.7).

$$c_i = s_i \frac{g_i}{C} \tag{3.7}$$

where $c_i$ denotes the capacity of lane group $i$ (veh/h); $s_i$ denotes the saturation flow rate for lane group $i$ (veh/h); and $\frac{g_i}{C}$ denotes the effective green ratio for lane group $i$.

**Table 3.10 Adjustment factors for saturation flow rate (HCM, 2016)**

| Factor | Formula | Definition of Variables | Notes |
|---|---|---|---|
| Lane width | $f_w = 1 + \frac{(W-12)}{30}$ | W = lane width (ft) | $W \geq 8.0$<br>If W > 16, a two-lane analysis may be considered |
| Heavy vehicles | $f_{HV} = \frac{100}{100 + \%HV(E_T - 1)}$ | % HV = % heavy vehicles for lane group volume | $E_T$ = 2.0 pc/HV |
| Grade | $f_g = 1 - \frac{\%G}{200}$ | % G = % grade on a lane group approach | $-6 \leq \% G \leq +10$<br>Negative is downhill |
| Parking | $f_p = \frac{N - 0.1 - \frac{18N_m}{3600}}{N}$ | N = number of lanes in lane group<br>$N_m$ = number of parking maneuvers/h | $0 \leq N_m \leq 180$<br>$f_p \geq 0.050$<br>$f_p$ = 1.000 for no parking |
| Bus blockage | $f_{bb} = \frac{N - \frac{14.4N_B}{3600}}{N}$ | N = number of lanes in lane group<br>$N_B$ = number of buses stopping/h | $0 \leq N_B \leq 250$<br>$f_{bb} \geq 0.050$ |
| Type of area | $f_a$ = 0.900 in CBD<br>$f_a$ = 1.000 in all other areas | | |
| Lane utilization | $f_{LU} = v_g/(v_{g1}N)$ | $v_g$ = unadjusted demand flow rate for the lane group, veh/h<br>$v_{g1}$ = unadjusted demand flow rate on the single lane in the lane group with the highest volume<br>N = number of lanes in the lane group | |
| Left turns | Protected phasing:<br>Exclusive lane:<br>$f_{LT} = 0.95$<br>Shared lane:<br>$f_{LT} = \frac{1}{1.0 + 0.05P_{LT}}$ | $P_{LT}$ = proportion of LTs in lane group | See Exhibit C16-1, Appendix C, for nonprotected phasing alternatives |
| Right turns | Exclusive lane:<br>$f_{RT} = 0.85$<br>Shared lane:<br>$f_{RT} = 1.0 - (0.15)P_{RT}$<br>Single lane:<br>$f_{RT} = 1.0 - (0.135)P_{RT}$ | $P_{RT}$ = proportion of RTs in lane group | $f_{RT} \geq 0.050$ |

***v/c ratio***

For a given lane group $i$, v/c ratio is defined as the ratio of flow rate to capacity (v/c), which is computed using Equation (3.8).

$$X_i = \left(\frac{v}{c}\right)_i = \frac{v_i}{s_i\left(\frac{v}{c}\right)} = \frac{v_i C}{s_i g_i} \tag{3.8}$$

where $X_i$ denotes the ratio for lane group $i$; $v_i$ denotes the actual or projected demand flow rate for lane group $i$ (veh/h); $s_i$ denotes the saturation flow rate for lane group $i$ (veh/h); $g_i$ denotes the effective green time for lane group $i$ (s); and $C$ denotes the cycle length (s).

***Critical Lane Group***

Another concept used for analyzing signalized intersections is the critical v/c ratio, $X_c$. This is the v/c ratio for the intersection as a whole, only considering the lane groups that have the highest flow ratio (v/s) for a given signal phase. Each signal phase has a critical lane group that determines the green-time requirements for the phase. The critical v/c ratio for the intersection is determined by Equation (3.9):

$$X_c = \sum \left(\frac{v}{c}\right)_{ci} \frac{C}{C-L} \tag{3.9}$$

where $X_c$ denotes the critical v/c ratio for intersection; $\sum \left(\frac{v}{c}\right)_{ci}$ denotes the summation of flow ratios for all critical lane groups $i$; $C$ denotes the cycle length (s); and $L$ denotes the total lost time per cycle.

3.4.7 Determining Delay

The values derived from the delay calculations represent the average control delay experienced by all vehicles that arrive within the analysis period, including delays incurred beyond the analysis period when the lane group is oversaturated. Control delay includes movements at slower speeds and stop on intersection approaches, when vehicles move up in queue position or slow down at the upstream of an intersection.

The average control delay per vehicle for a given lane group is given by Equation (3.10).

$$d = d_1(PF) + d_2 + d_3 \tag{3.10}$$

where $d$ denotes the control delay per vehicle (s/veh); $d_1$ denotes the uniform control delay (s/veh); PF denotes the uniform delay progression adjustment factor; $d_2$ denotes the incremental delay (s/veh); and $d_3$ denotes the initial queue delay (s/veh). More detailed descriptions can be found in HCM (2016).

***Progression Adjustment Factor***

The progression adjustment factor, *PF*, applies to all coordinated lane groups, including both pretimed control and non-actuated lane groups in semi-actuated control systems. Progression primarily affects uniform delay, and for this reason, the adjustment is applied only to $d_1$. The value of *PF* may be determined using Equation (3.11).

$$PF = \frac{(1-P)f_{PA}}{1-\left(\frac{g}{C}\right)} \tag{3.11}$$

where *PF* denotes the progression adjustment factor; *P* denotes the proportion of vehicles arriving on green; g/C denotes the proportion of green time available; and $f_{PA}$ denotes the supplemental adjustment factor for platoon arriving during green.

***Uniform Delay***

Assuming uniform arrivals, stable flow, and no initial queue, Equation (3.12) can be applied to estimate the uniform control delay.

$$d_1 = \frac{0.5C\left(1-\frac{g}{C}\right)^2}{1-\left[\min(1,X)\frac{g}{C}\right]} \tag{3.12}$$

where $d_1$ denotes the uniform control delay assuming uniform arrivals (s/veh); $C$ denotes the cycle length (s); $g$ denotes the effective green time for the lane group (s); and $X$ denotes the v/c ratio or degree of saturation for the lane group.

***Incremental Delay***

Equation (3.13) is used to estimate the incremental delay due to non-uniform arrivals and temporary cycle failures (random delay) as well as delay caused by sustained periods of oversaturation (oversaturation delay).

$$d_2 = 900T\left[(X-1)+\sqrt{(X-1)^2+\frac{8klX}{cT}}\right] \tag{3.13}$$

where $d_2$ denotes the incremental delay (s/veh); $T$ denotes the duration of analysis period (h); $k$ denotes the incremental delay factor that is dependent on controller settings; $l$ denotes the upstream filtering/metering adjustment factor; $c$ denotes the lane group capacity (veh/h); and $X$ denotes the lane group v/c ratio or degree of saturation.

***Aggregated Delay Estimates***

Based on the delay calculation for each lane group, the aggregated delay for an approach is computed by Equation (3.14):

$$d_A = \frac{\sum d_i v_i}{\sum v_i} \tag{3.14}$$

where $d_A$ denotes the delay for approach A (s/veh); $d_i$ denotes the delay for lane group $i$ (on approach A) (s/veh); and $v_i$ denotes the adjusted flow for lane group $i$ (veh/h).

3.4.8 Web-Based Application for Intersection Delay Calculation

With the parameter calibrations, the HCM delay equations can help estimate the intersection delay under different signal control settings. Hence, the increased intersection delay caused by changing the left-turn phasing from permitted-only to protected-only can be easily obtained. Because parameter calibration and delay equation implementation are often time-consuming, this research first gathers real-world traffic information from the UDOT Automated Traffic Signal Performance Measures (ATSPM) to calibrate delay equation parameters and further develops a web-based application (http://10.0.0.166:8080/) to compute the intersection delay.

Appendix A shows the configuration guide for the web-based application and Appendix B presents all source codes. Figure 3.3 demonstrates the designed application interface. Once all information (e.g., number of lanes, lane width, traffic volume, signal timing plan, etc.) is input into the application, the average delay for all vehicles can be computed under the control of “permitted” and “protected” left-turn signals. Then, the increased delay by changing permitted-only to protected-only can be generated to further estimate the delay cost.

### 3.5 Left-Turn Phasing Design Recommendation

Based on the previous discussion, changing left-turn signals to protected-only would be an effective countermeasure to reduce left-turn crashes. However, it will bring an increase in operational cost, which includes the signal replacement cost and the increased delay cost. The replacement cost can be computed by counting the labor and material fees. The increased delay cost can be calculated based on the estimation of the before-after intersection delays. After obtaining the total operational and crash cost, this research proposes a method to decide whether changing the left-turn signal to protected-only is recommended. Figure 3.4 illustrates the workflow of the proposed method.


**Figure 3.3 Interface of the web-based application for delay calculation**

### 3.6 Summary

This chapter focused on presenting the proposed methods for computing safety benefits, equipment replacement costs, and intersection delay costs. First, a method based on the historical crash records, crash cost values, and CRFs was introduced to compute the safety benefit (i.e., reduced crash cost). Second, the method for calculating the operational cost was presented, which consists of signal replacement cost and increased delay cost. Lastly, the procedure of making recommendations on left-turn phasing design was introduced. Notably, the method developed by this research can only help make the initial phasing design decisions. In practice, if any intersections are suggested to implement protected-only left-turn phasing, more investigation into the resulting costs and delay costs should be conducted. For example, more detailed analyses of CMFs and traffic simulation could be used to fine tune estimates of safety benefits and operational costs.

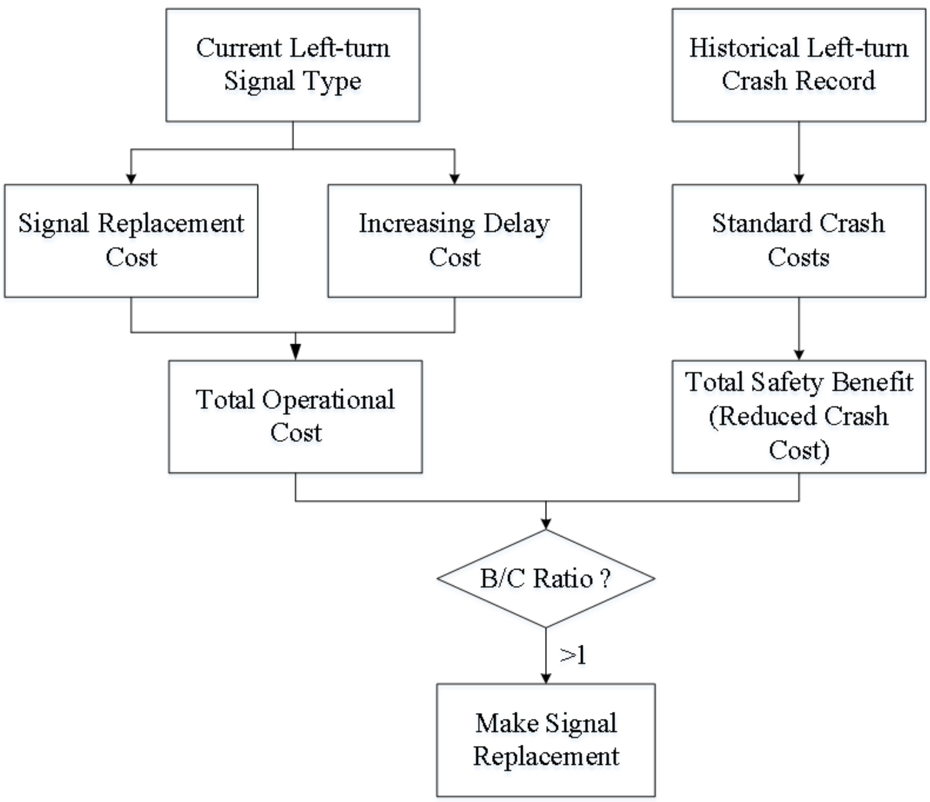



**Figure 3.4 Proposed method for comparing the operational and crash cost**

# 4.0 EXPERIMENTAL ANALYSIS

## 4.1 Overview

To test the developed tool in analyzing the benefit and cost of changing left-turn phasing plans, this research selects four signalized intersections in Utah for experimental analysis. According to the historical crash data, three of the study intersections (Signals 7104, 7198, and 7162) have high crash risks (i.e., large number of crash records) and one (Signal 6434) has medium crash risks. Also, the HCM equations for computing intersection delay are evaluated at two intersections (signal 7104 and signal 7107) by comparing the estimated delay with ATSPM data. Based on the collected information, the safety benefit (reduced crash cost) and operational cost associated with replacing permitted left-turn phasing with protected phasing are computed for each intersection. Finally, phasing design recommendations are made based on the B/C ratio.

## 4.2 Case Studies

### 4.2.1 Case Setting

This research selects two peak periods (7:00am – 9:00am, and 4:00pm – 7:00pm) and two off-peak periods (9:00am – 4:00pm, and 7:00pm – 10:00pm) for case studies. To estimate the intersection delays with different left-turn phasing treatments, calibrated parameters that are used in the HCM delay equations are listed in Table 4.1.

**Table 4.1 Parameters for computing intersection delay**

| Parameters | Value |
|---|---|
| Percent grade adjustment factor ($f_g$) | 1 for all movements |
| Parking adjustment factor ($f_p$) | 1 for all movements |
| Bus blockage adjustment factor ($f_{bb}$) | 1 for all movements |
| Area type adjustment factor ($f_a$) | 1 for all movements |
| Lane utilization adjustment factor ($f_{LU}$) | 0.952 for all through movements; 1 for all other movements. |

| | |
|---|---|
| Left-turn adjustment factor ($f_{LT}$) | 0.95 for protected exclusive left-turn lane; needs to be calculated for permitted left-turn; 1 for all other movements. |
| Right-turn adjustment factor ($f_{RT}$) | 0.85 for exclusive right-turn lane; 1 for all other movements. |
| Left-turn ped/bicycle adjustment factor ($f_{Lpb}$) | 1 for all movements |
| Right-turn ped/bicycle adjustment factor ($f_{Rpb}$) | 1 for all movements |
| Base saturation ($S_o$) | 1,900 veh/h |

### 4.2.2 Delay Estimation Method Evaluation

As the developed tool adopts calibrated HCM delay models to estimate intersection delays, it is critical to examine whether the estimated delays can truly reflect the reality. Hence, this research selects two intersections (Signals 7104 and 7017) to conduct the model evaluation. For comparison, approach delays are retrieved from UDOT ASTPM. Table 4.2 shows the difference between the model estimated delays and ATSPM approach delays. The minimal differences, 3.5% for Signal 7104 and 6.4% for Signal 7107, indicate that the calibrated HCM delay models can yield acceptable estimates.

**Table 4.2 Evaluation results**

| Case | Computed delay | Approach delay from ATSPM | Difference |
|---|---|---|---|
| Signal 7104 | 33.18 s | 34.38 s | 3.5% |
| Signal 7107 | 30.04 s | 28.10 s | 6.4% |

### 4.2.3 Delay Calculations

***Case 1: Signal 7104 (Redwood Road @ 4100 South)***

Figure 4.1 shows the geometric features of the signal 7104 intersection, which include one left-turn lane for each direction. Table 4.3 lists all collected information for computing the intersection delay.

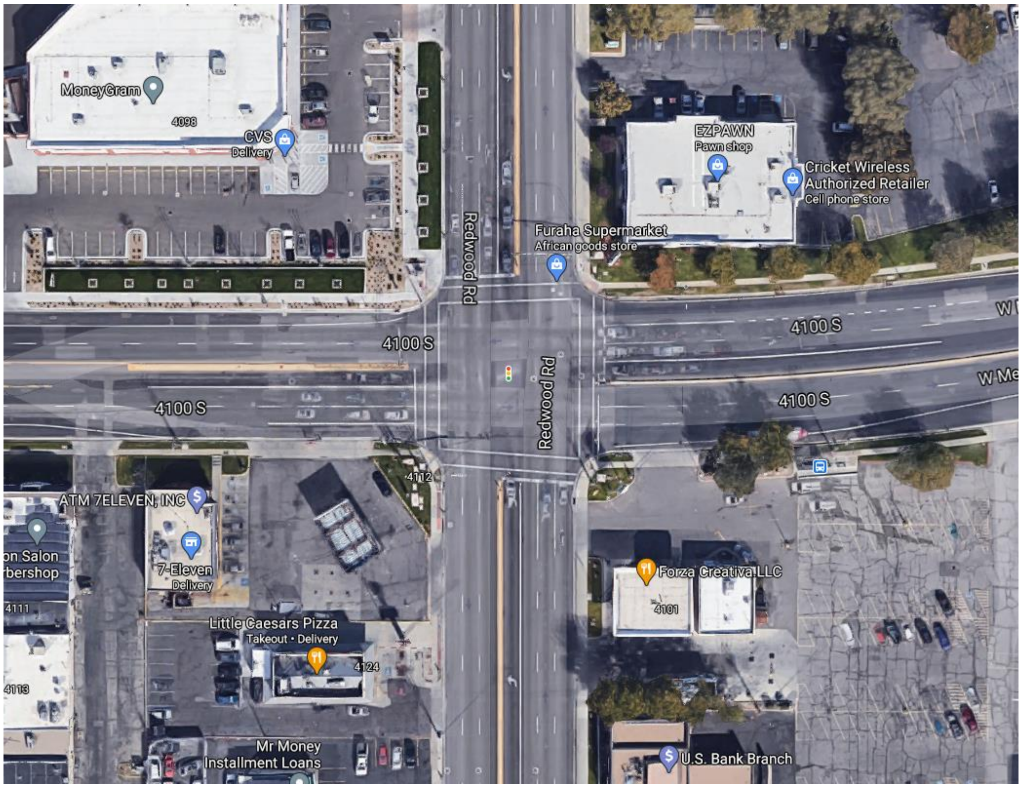



**Figure 4.1 Geometric features of Signal 7104**

**Table 4.3 Collected data for Signal 7104**

| Parameters | NBL | NBT | NBT/R | SBL | SBT | SBT/R | WBL | WBT | WBR | EBL | EBT | EBR |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Lane width (ft) | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 |
| Number of lanes | 1 | 2 | 1 | 1 | 2 | 1 | 1 | 2 | 1 | 1 | 2 | 1 |
| Peak-hour volume | 101 | 401 | 219 | 86 | 488 | 230 | 130 | 450 | 405 | 116 | 405 | 103 |
| Off peak-hour volume | 65 | 248 | 153 | 74 | 298 | 133 | 100 | 236 | 64 | 71 | 211 | 63 |
| Green time (s) | 23 | 31 | 31 | 23 | 31 | 31 | 20 | 34 | 34 | 20 | 34 | 34 |

Based on the collected data listed in Table 4.3, this research implements the calibrated HCM models to estimate intersection delays at the Signal 7104 intersection. Moreover, to calculate the increased delay resulting from replacing permitted left turns with protected left turns, this research further compares the resulting delay with these two phasing plans. Table 4.4 shows the obtained average delay (in seconds) during both peak and off-peak hours.

**Table 4.4 Estimated delays for Signal 7104**

| Time | Signal type | Average delay (s) |
|---|---|---|
| Peak hours | Permitted | 8.19 |
| | Protected | 34.27 |
| Off-peak hours | Permitted | 6.09 |
| | Protected | 29.23 |

***Case 2: Signal 7198 (700 East @ 9400 South)***

Figure 4.2 shows the geometric features of the signal 7198 intersection, which include one left-turn lane for each direction. Table 4.5 lists all collected information for computing the intersection delay.

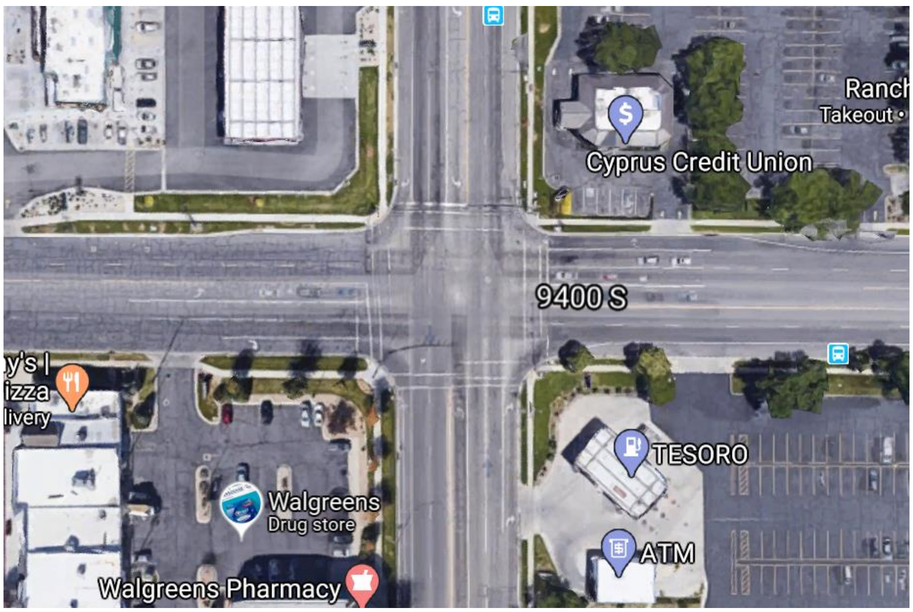



**Figure 4.2 Geometric features of Signal 7198**

**Table 4.5 Collected data for Signal 7198**

| Parameters | NBL | NBT | NBR | SBL | SBT | SBR | WBL | WBT | WBR | EBL | EBT | EBR |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Lane width (ft) | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 |
| Number of lanes | 1 | 2 | 1 | 1 | 2 | 1 | 1 | 2 | 1 | 1 | 2 | 1 |
| Peak-hour volume | 172 | 886 | 225 | 145 | 810 | 120 | 208 | 316 | 89 | 152 | 297 | 109 |
| Off peak-hour volume | 112 | 173 | 225 | 148 | 532 | 105 | 176 | 316 | 89 | 113 | 242 | 84 |

Table 4.6 summarizes the obtained average delay (in seconds) during both peak and off-peak hours, under the control of both permitted and protected left-turn phasing plans.

**Table 4.6 Estimated delays for Signal 7198**

| Time | Signal type | Average delay (s) |
|---|---|---|
| Peak hours | Permitted | 11.52 |
| | Protected | 27.14 |
| Off-peak hours | Permitted | 11.24 |
| | Protected | 24.30 |

***Case 3 - Signal 7162 (State Street @ 5900 South)***

Figure 4.3 shows the geometric features of the signal 7162 intersection, which include one left-turn lane for each direction. Table 4.7 lists all collected information for computing the intersection delay.

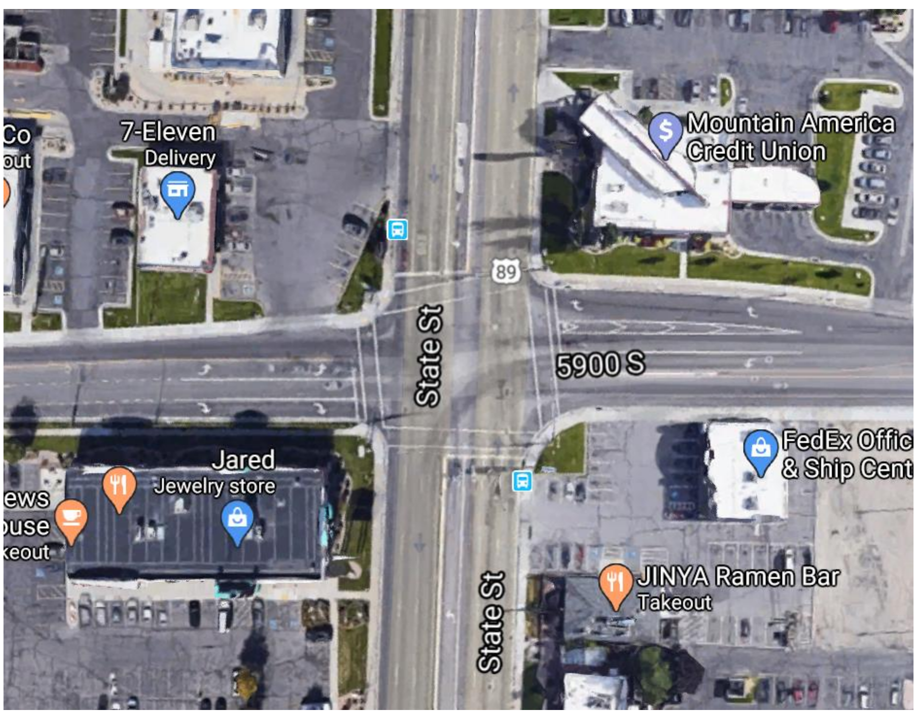



**Figure 4.3 Geometric features of Signal 7162**

**Table 4.7 Collected data for Signal 7162**

| Parameters | NBL | NBT | NBT/R | SBL | SBT | SBT/R | WBL | WBT | WBR | EBL | EBT | EBR |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Lane width (ft) | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 |
| Number of lanes | 1 | 2 | 1 | 1 | 2 | 1 | 1 | 2 | 1 | 1 | 2 | 1 |
| Peak-hour volume | 83 | 761 | 327 | 171 | 856 | 359 | 100 | 121 | 125 | 131 | 169 | 76 |
| Off peak-hour volume | 67 | 585 | 280 | 173 | 658 | 297 | 103 | 136 | 168 | 98 | 113 | 55 |

Table 4.8 summarizes the obtained average delay (in seconds) during both peak and off-peak hours, under the control of both permitted and protected left-turn phasing plans.

**Table 4.8 Estimated delays for Signal 7162**

| Time | Signal type | Average delay (s) |
|---|---|---|
| Peak hours | Permitted | 9.79 |
| | Protected | 23.93 |
| Off-peak hours | Permitted | 10.20 |
| | Protected | 22.96 |

***Case 4 - Signal 6434 (Freedom Boulevard @ 300 South)***

Figure 4.4 shows the geometric features of the signal 6434 intersection, which include one left-turn lane for each direction. Table 4.9 lists all collected information for computing the intersection delay, including lane width (in feet), number of lanes, peak-hour volume (in vehicles per hour), and off-peak hour volume (in vehicles per hour).

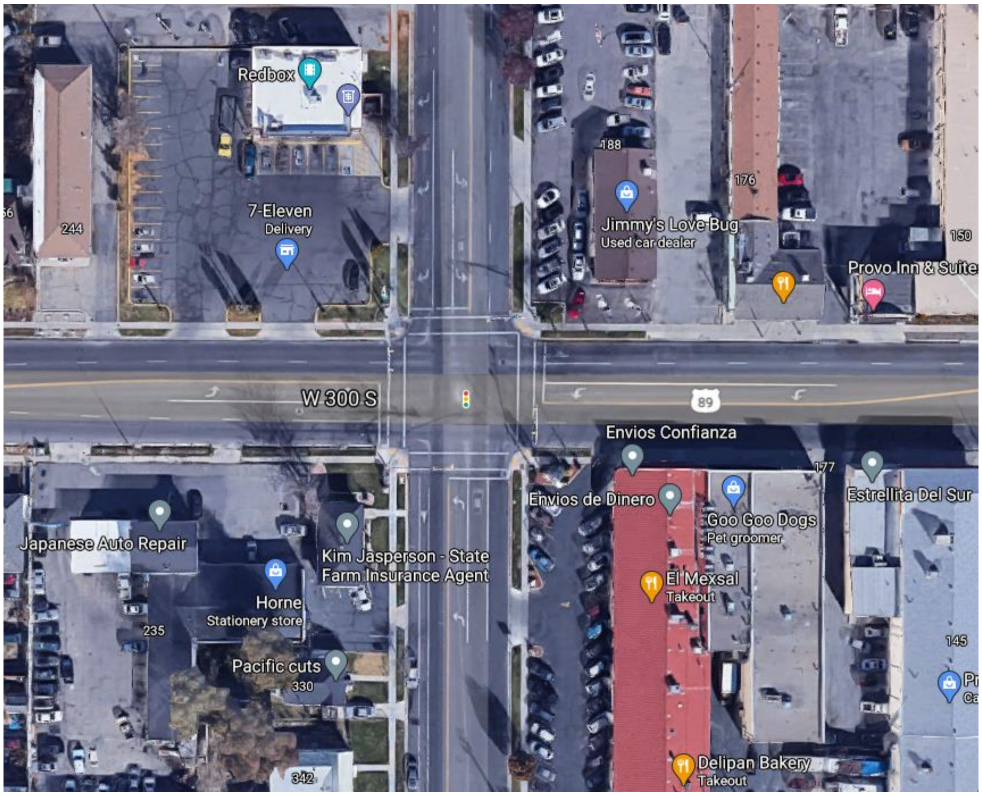



**Figure 4.4 Geometric features of Signal 6434**

**Table 4.9 Collected data for Signal 6434**

| Parameters | NBL | NBT | NBT/R | SBL | SBT | SBT/R | WBL | WBT | WBR | EBL | EBT | EBR |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Lane width (ft) | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 |
| Number of lanes | 1 | 2 | 1 | 1 | 2 | 1 | 1 | 2 | 1 | 1 | 2 | 1 |
| Peak-hour volume | 83 | 761 | 327 | 171 | 856 | 359 | 100 | 121 | 125 | 131 | 169 | 76 |
| Off peak-hour volume | 67 | 585 | 280 | 173 | 658 | 297 | 103 | 136 | 168 | 98 | 113 | 55 |

Table 4.10 summarizes the obtained average delay (in seconds) during both peak and off-peak hours, under the control of both permitted and protected left-turn phasing plans.

**Table 4.10 Estimated delays for Signal 7162**

| Time | Signal type | Average delay (s) |
|---|---|---|
| Peak hours | Permitted | 9.83 |
| | Protected | 22.91 |
| Off-peak hours | Permitted | 9.62 |
| | Protected | 20.95 |

### 4.2.4 Truck Percentage and Delay Cost Rate

In addition to intersection delays, truck percentages in the traffic and delay cost rates for both trucks and passenger cars are also needed to estimate the delay costs. Truck percentage information was collected from the UDOT Data Portal (2021) and the delay cost rates were provided by the UDOT TOC. Table 4.11 lists the truck percentage at each of the four case study intersections and Table 4.12 summarizes the delay cost rates.

**Table 4.11 Truck percentage**

| Signal | Truck percentage |
|---|---|
| 7104 | 8.7% |
| 7198 | 6.8% |
| 7162 | 7.3% |
| 6434 | 5.6% |

**Table 4.12 Cost rate for truck and passenger vehicles**

| Vehicle Type | Cost Rate |
|---|---|
| Truck | $94.04/h |
| Passenger vehicle | $17.67/h |

### 4.2.5 Cost Analysis

The total intersection delays with different signal phasing plans are computed by Equation (4.1). With the estimated delays, the total delay cost can be calculated by Equation (4.2). Then, the increased delay cost for each intersection after the left-turn phasing replacement can be obtained by examining the resulting cost difference between permitted and protected phasing plans. Tables 4.13 – 4.16 present the total delay and total delay cost for all studied cases.

Total delay = (peak hour delay * volume1 * hours1 + off-peak delay * volume2 * hours2) * yearly number of weekdays * number of years (4.1)

Total cost = Total delay * cost rate of passenger car * percentage of passenger car + Total delay * cost rate of trucks * percentage of trucks (4.2)

**Table 4.13 Delay cost for Signal 7104**

| Signal type | Weekday daily delay (h) | Yearly number of workdays | Total delay from 2010-2017 (h) | Total delay cost |
|---|---|---|---|---|
| Permitted | 88 | 247 | 173,888 | $4,227,946 |
| Protected | 288 | 247 | 569,088 | $13,836,914 |
| Increased delay | 200 | 247 | 395,200 | $9,608,968 |

**Table 4.14 Delay cost for Signal 7198**

| Signal type | Weekday daily delay (h) | Yearly number of workdays | Total delay from 2010-2017 (h) | Total delay cost |
|---|---|---|---|---|
| Permitted | 129 | 247 | 254,904 | $5,827,911 |
| Protected | 289 | 247 | 571,064 | $13,056,327 |
| Increased delay | 160 | 247 | 316,160 | $7,228,416 |

**Table 4.15 Delay cost for Signal 7162**

| Signal type | Weekday daily delay (h) | Yearly number of workdays | Total delay from 2010-2017 (h) | Total delay cost) |
|---|---|---|---|---|
| Permitted | 122 | 247 | 241,072 | $5,603,721 |
| Protected | 283 | 247 | 559,208 | $12,998,795 |
| Increased delay | 161 | 247 | 318,136 | $7,395,074 |

**Table 4.16 Delay cost for Signal 6434**

| Signal type | Weekday daily delay (h) | Yearly number of workdays | Total delay from 2010-2017 (h) | Total delay cost |
|---|---|---|---|---|
| Permitted | 61 | 247 | 120,536 | $2,645,370 |
| Protected | 137 | 247 | 270,712 | $5,941,240 |
| Increased delay | 76 | 247 | 150,176 | $3,295,870 |

The operational cost consists of replacement cost and increasing delay cost. By adding the replacement cost information in Table 3.7, Table 4.17 illustrates the increased delay cost, replacement cost, and total operational cost for all cases.

**Table 4.17 The results of operational cost**

| Signal ID | Increased delay cost | Replacement cost | Operational cost |
|---|---|---|---|
| 7104 | $9,608,968 | $0 | $9,608,968 |
| 7198 | $7,228,416 | $38,000 | $7,266,416 |
| 7162 | $7,395,074 | $38,000 | $7,433,074 |
| 6364 | $3,295,870 | $67,000 | $3,362,870 |

Notably, the proposed evaluation tool assumes that 90% of left-turn crashes can be avoided after implementing the protected left-turn phasing. Hence, the safety benefit after the phasing replacement can be estimated by the crash costs during the operation of permitted phasing. Table 4.18 shows the operational cost, estimated safety benefits (reduced crash costs) and B/C ratio. If the B/C ratio is greater than one, a replacement of permitted phasing with protected phasing is recommended. Based on the results in Table 4.18, it can be observed that three intersections (Signals 7104, 7198, and 7162) have a B/C ratio higher than one while the other (Signal 6364) has a ratio lower than one. By examining the historical crash records of the four intersections, it is found that the three intersections with higher B/C ratios have experienced much larger numbers of left-turn crashes during the past eight years, compared with the Signal 6364 intersection. Hence, the calculation results are consistent with the intuition that intersections with more left-turn crashes should implement protected left-turn phasing.

**Table 4.18 Comparison between operational cost and crash cost for studied intersections**

| Signal | Operational cost | Safety Benefit | B/C ratio | Recommendation |
|---|---|---|---|---|
| 7104 | $9,608,968 | $12,613,950 | **1.31** | Replace with protected phasing |
| 7198 | $7,266,416 | $11,665,080 | **1.61** | Replace with protected phasing |
| 7162 | $7,433,074 | $12,249,990 | **1.65** | Replace with protected phasing |
| 6364 | $3,362,870 | $496,170 | **0.15** | Keep the permitted phasing |

## 4.3 Summary

This chapter conducted case studies of four signalized intersections. Based on the collected geometric data, traffic volumes, truck percentages, signal replacement costs, and historical crash records, the safety benefits and operational costs of replacing the current permitted left-turn phasing with protected phasing were calculated. The resulting B/C ratios indicated that three intersections with higher crash frequency and severity could benefit from protected phasing. In conclusion, the developed evaluation tool in this research can offer a convenient way to help UDOT make decisions on left-turn phasing plan selections.

# 5.0 CONCLUSIONS

## 5.1 Summary

This research develops a convenient tool to evaluate the corresponding safety benefit and operational cost when replacing permitted left-turn phasing with protected phasing. In the first step, the left-turn crash cost is estimated based on the historical left-turn crash data at the intersection and rate of crash cost. Then, the safety benefit (i.e., reduced crash cost) is calculated by assuming 90% of crashes are avoidable after protected phasing is implemented. In the second step, the operational cost, which consists of signal replacement cost and increased delay cost, is computed. The increased delay cost is obtained by multiplying the increased delay after phasing replacement by the delay cost rates. The required intersection delays with different phasing plans are estimated by the calibrated HCM delay models. Then, in the last step, the B/C ratio is obtained by dividing the calculated safety benefit by the operational cost. If the resulting B/C ratio is greater than one, it is suggested that the current permitted left-turn phasing be replaced by protected phasing. For numerical testing, the proposed tool is implemented at four signalized intersections in Utah. The testing results indicate that three of them should consider adopting the protected left-turn phasing for reducing crash costs.

## 5.2 Findings

Based on the study results, several findings can be reached:

- Delay costs and safety costs can be quantified and compared to help select appropriate left-turn treatments to reduce the crash rate.
- Replacing permitted left-turn phasing with protected phasing is expected to bring some safety benefits by reducing crash rates but produce increased intersection delays especially during peak hours.
- For intersections with high crash rates and crash costs, it is suggested to change the permitted left-turn signal to protected only.

### 5.3 Limitations and Challenges

The study results rely on the quality of collected historical information. The developed tool can help the decision-making process in selecting proper left-turn phasing plans, but more detailed investigations of intersection safety and operational performance should be conducted prior to implementing left-turn phasing changes. The developed tool should only be used to conduct the first round of screening to identify the intersections that should be studied further (e.g., the ones with higher B/C ratios).

## REFERENCES


Agent, Kenneth R., and Robert C. Deen. “Warrants for left-turn signal phasing.” (1978).

Agent, Kenneth R. "An Evaluation of Permissive Left-Turn Phasing." (1979).

Agent, Kenneth R. *Guidelines for the Use of Protected/permissive Left-turn Phasing. Final Report.* No. UKTRP-85-19. 1985.

Agent, Kenneth R., Nikiforos Stamatiadis, and Bryan Dyer. "Guidelines for the Installation of Left-turn Phasing." (1995).

Al-Kaisy, A. F., and J. A. Stewart. "New approach for developing warrants of protected left-turn phase at signalized intersections." *Transportation Research Part A: Policy and Practice* 35, no. 6 (2001): 561-574.

Asante, Seth A., Siamak Ardekani, and James C. Williams. "Selection criteria for left-turn phasing and indication sequence." *Transportation research record* 1421 (1993).

Chen, Li, Cynthia Chen, and Reid Ewing. "Left-turn phase: Permissive, protected, or both? A quasi-experimental design in New York City." *Accident Analysis & Prevention* 76 (2015): 102-109.

Medina, Juan C., M. Scott Shea, and Nuzhat Azra. *Safety Effects of Protected and Protected/Permissive Left-Turn Phases*. No. UT-19.04. Utah. Dept. of Transportation, 2018.

AASHTO. "Highway Safety Manual. " *Amercian Association of State Transportation Officials*: *Washington, D.C. USA* (2010).

Cottrell, Benjamin H. *Guidelines for Protected Permissive Left Turn Signal Phasing*. Virginia Highway & Transportation Research Council, 1985.

CMF Clearinghouse. < http://www.cmfclearinghouse.org/index.cfm> (accessed Jan 28, 2021).

Davis, Gary A., and Nathan Aul. *Safety effects of left-turn phasing schemes at high-speed intersections*. No. MN/RC-2007-03. 2007.

Federal Highway Administration. "Permissive/Protected Left-Turn Phasing." *Intersection Safety Case Study* (2010).

Golias, John, and Violette PORIKOU. "Investigating the adjustment factor for protected left turns from a shared lane." *IATSS research* 28, no. 1 (2004): 95-100.

Harkey, David L. *Accident modification factors for traffic engineering and ITS improvements*. Vol. 617. Transportation Research Board, 2008.

Hedges, Adam J. " Signalized left-turn capacity–model estimation through micro-simulation." *University of Kentucky, Lexington, KY* (2014).

Elefteriadou, Lily A. " The Highway Capacity Manual: A Guide for Multimodal Mobility Analysis." *TR News* 306 (2016).

ITE Florida Section. “Left turn phase design in Florida.” *ITE journal* 9 (1982): 28–35.

Lalani, Nazir, Daniel Cronin, David Hattan, and Terence Scads. "A Summary of the Use of Warrants for the Installation of Left-Turn Phasing at Signalized Intersections." *ITE journal* (1986): 57-59.

McShane, William R., and Roger P. Roess. "*Traffic engineering*."1990.

National Highway Traffic Safety Administration. "Traffic Safety Facts: 2016." *U.S. Department of Transportation* (2018).

National Highway Traffic Safety Administration. "Traffic Safety Facts 2014." *U.S. Department of Transportation* (2016).

National Highway Traffic Safety Administration. "Traffic Safety Facts: 2015." *U.S. Department of Transportation* (2015).

Ozmen, Ozlem, Zong Z. Tian, and A. Reed Gibby. "Safety of the Las Vegas left-turn display." *Accident Analysis & Prevention* 62 (2014): 95-101.

Srinivasan, Raghavan, Forrest Council, Craig Lyon, Frank Gross, Nancy Lefler, and Bhagwant

Persaud. "Safety effectiveness of selected treatments at urban signalized intersections." *Transportation Research Record* 2056, no. 1 (2008): 70-76.

Stamatiadis, Nikiforos, Kenneth R. Agent, and Apostolos Bizakis. "Guidelines for left-turn phasing treatment." *Transportation Research Record* 1605, no. 1 (1997): 1-7.

Stamatiadis, Nikiforos, Adam Hedges, and Adam Kirk. "A simulation-based approach in determining permitted left-turn capacities." *Transportation Research Part C: Emerging Technologies* 55 (2015): 486-495.

Stamatiadis, Nikiforos, Salee Tate, and Adam Kirk. "Left-turn phasing decisions based on conflict analysis." *Transportation research procedia* 14 (2016): 3390-3398.

UDOT ATSPM. < https://udottraffic.utah.gov/ATSPM> (accessed Feb 18, 2020).

UDOT DATA PORTAL. < https://data-uplan.opendata.arcgis.com/datasets/aadt-open-data> (accessed Jan 15, 2021).

Upchurch, Jonathan E. "Guidelines for selecting type of left-turn phasing." *Transportation Research Record* 1069 (1986): 30-38.

Wang, Xuesong, and Mohamed Abdel-Aty. "Modeling left-turn crash occurrence at signalized intersections by conflicting patterns." *Accident Analysis & Prevention* 40, no. 1 (2008): 76-88.

Yang, Qiaoli, Zhongke Shi, Shaowei Yu, and Jie Zhou. "Analytical evaluation of the use of left-turn phasing for single left-turn lane only." *Transportation Research Part B: Methodological* 111 (2018): 266-303.

Zhang, Zhao, Runan Yang, Yun Yuan, Glenn Blackwelder, and Xianfeng Yang. "Examining driver injury severity in left-turn crashes using hierarchical ordered probit models." *Traffic injury prevention* (2020): 1-6.

# APPENDIX A: PROGRAM SETUP

## A.1 Reference hyperlinks for used framework/packages:

a) Front end framework

- vue https://vuejs.org/v2/guide/
- vue cli https://cli.vuejs.org/guide/cli-service.html
- vuetify https://vuetifyjs.com/zh-Hans/getting-started/quick-start/

b) Back end framework

- flask https://flask.palletsprojects.com/en/1.1.x/api/
- flask-cors https://flask-cors.readthedocs.io/en/latest/ (CORS setting)

Setting-up development/production environment:

## A.2 Set up Vue development environment for first time:

```
conda install -c conda-forge nodejs
npm install vue
npm install -g @vue/cli
vue create tsppm
cd tsppm
vue add vuetify
npm install axios --save
```

## A.3 Run the following code for setting up in two command prompts

```
npm run serve
python ./pyservice/pyservice.py
```

```
(js) PS D:\GoogleDriveUtah\signalcal\code\signalcal> python ./pyservice/pyservice.py
 * Serving Flask app "pyservice" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 179-937-040
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

```
 DONE  Compiled successfully in 14463ms


  App running at:
  - Local:   http://localhost:8080/
  - Network: http://192.168.0.243:8080/

  Note that the development build is not optimized.
  To create a production build, run npm run build.
```

**A.4 Set up py/flask environment**

https://programminghistorian.org/en/lessons/creating-apis-with-python-and-flask#creating-a-basic-flask-application

## APPENDIX B:  PROGRAM SOURCE CODE

```
import numpy as np
# Lane utilization adjustment factor
def through_lane(lane_number):
    if lane_number == 1:
        f_lu = 1
    elif lane_number == 2:
        f_lu = 0.952
    else:
        f_lu = 0.908
    return f_lu

def left_lane(lane_number):
    if lane_number == 1:
        f_lu = 1
    else:
        f_lu = 0.971
    return f_lu

def right_lane(lane_number):
    if lane_number == 1:
        f_lu = 1
    else:
        f_lu = 0.885
    return f_lu

# right turn adjustment factor
def right_adjustment_factor(lane_condition, p_rt):
    if lane_condition == "Exclusive lane":
        f_rt = 0.85
    elif lane_condition == "Shared lane":
        f_rt = 1 - 0.15 * p_rt
    else:
        f_rt = 1 - 0.135 * p_rt
    return f_rt
# left turn adjustment factor
# def left_adjustment_factor(lane_condition):
#     if lane_condition == 'Exclusive lane':
#         f_lt=0.95
#     else:
#         f_lt=1/(1+0.05*p_lt)
#     return f_lt
def progression_factor(gc):
    if gc <= 0.2:
        pf = 1
```

```
elif 0.2 < gc <= 0.3:
    pf = 0.986
elif 0.3 < gc <= 0.4:
    pf = 0.895
elif 0.4 < gc <= 0.5:
    pf = 0.767
elif 0.5 < gc <= 0.6:
    pf = 0.576
else:
    pf = 0.256
return pf

def EL1(calculated_oppsing_flow):
if calculated_oppsing_flow <= 1:
el = 1.3
elif 1 < calculated_oppsing_flow <= 200:
el = 1.6
elif 200 < calculated_oppsing_flow <= 400:
el = 1.9
elif 400 < calculated_oppsing_flow <= 600:
el = 2.3
elif 600 < calculated_oppsing_flow <= 800:
el = 2.8
elif 800 < calculated_oppsing_flow <= 1000:
el = 3.3
else:
el = 4.0
return el

def Delay1(green, cycle, vc, lane_group_num):
delay1 = []
for i in range(lane_group_num):
numerator = 0.5 * cycle * (1 − green[i] / cycle) * (1 − green[i] / cycle)
denominator = 1 − min(1, vc[i]) * green[i] / cycle
pf = progression_factor(green[i] / cycle)
delay = numerator / denominator * pf
delay1.append(delay)
return delay1

def saturation_flow(
s0,
n,
f_w,
f_hv,
f_g,
f_p,
```

```
    f_bb,
    f_a,
    f_lu,
    f_lt,
    f_rt,
    f_lpb,
    f_rpb,
    lane_group_num,
):
    saturation = []
    for i in range(lane_group_num):
        s = (
            s0[i]
            * n[i]
            * f_w[i]
            * f_hv[i]
            * f_g[i]
            * f_p[i]
            * f_bb[i]
            * f_a[i]
            * f_lu[i]
            * f_lt[i]
            * f_rt[i]
            * f_lpb[i]
            * f_rpb[i]
        )
        saturation.append(s)
    return saturation

def cal_permitted_coordinated(
    Lane_width,
    Lane_num,
    definded_saturation,
    Lane_group_volume,
    pr,
    green_time_lane_group,
    cycle,
    defined_PHF,
    LR_condition,
    Heavy_ratio,
    defined_area_type_adjustment,
    LT_flow,
    lane_num_in_LT_group,
    lane_num_in_opposing_approach,
    opposing_flow,
    LT_green_time,
```

```
    opposing_green_time,
):
    lane_group_num = 4
    base_saturation = [definded_saturation for i in range(lane_group_num)]
    PHF = [defined_PHF for i in range(lane_group_num)]
    LT_number = 4
    PHF_L = [defined_PHF for i in range(LT_number)]  # Input
    adjusted_LT = [LT_flow[i] / PHF_L[i] for i in range(LT_number)]
    adjusted_opposing = [opposing_flow[i] / PHF_L[i] for i in range(LT_number)]
    defined_lost_time = 4  # Input
    lost_time = [defined_lost_time for i in range(LT_number)]
    LT_flow_per_cycle = [adjusted_LT[i] * cycle / 3600 for i in range(LT_number)]
    # opposing_lu_factor=[through_lane(lane_num_in_opposing_approach[i]) for i in range(LT_n
umber)]
    opposing_lu_factor = [0.95, 0.95, 0.95, 0.95]
    opposing_flow_per_lane = [
        adjusted_opposing[i]
        * cycle
        / (3600 * lane_num_in_opposing_approach[i] * opposing_lu_factor[i])
        for i in range(LT_number)
    ]
    gf = [0 for i in range(LT_number)]
    rpo = [1.333 for i in range(LT_number)]
    qro = [
        max(1 - rpo[i] * opposing_green_time[i] / cycle, 0) for i in range(LT_number)
    ]
    gq = [
        opposing_flow_per_lane[i]
        * qro[i]
        / (0.5 - opposing_flow_per_lane[i] * (1 - qro[i]) / opposing_green_time[i])
        for i in range(LT_number)
    ]
    gu = [LT_green_time[i] - gq[i] for i in range(LT_number)]
    calculated_opposing = [
        opposing_flow[i] / opposing_lu_factor[i] for i in range(LT_number)
    ]
    el = [EL1(calculated_opposing[i]) for i in range(LT_number)]
    pl = [1 for i in range(LT_number)]
    f_min = [2 * (1 + pl[i]) / LT_green_time[i] for i in range(LT_number)]
    fm = [
        gf[i] / LT_green_time[i]
        + (gu[i] / LT_green_time[i]) * (1 / (1 + pl[i] * (el[i] - 1)))
        for i in range(LT_number)
    ]
    flt_list = [
        (fm[i] + 0.91 * (lane_num_in_LT_group[i] - 1)) / lane_num_in_LT_group[i]
```

```
    for i in range(LT_number)
]

# Lane width adjustment factor
f_w = [1 + (i - 12) / 30 for i in Lane_width]
ET = 2  # passenger-car equivalent
f_hv = [100 / (100 + Heavy_ratio * (ET - 1)) for i in range(lane_group_num)]
# Percentage grade adjustment factor
f_g = [1 for i in range(lane_group_num)]
# parking adjustment factor
f_p = [1 for i in range(lane_group_num)]
# bus blockage adjustment factor
f_bb = [1 for i in range(lane_group_num)]
f_a = [
    defined_area_type_adjustment for i in range(lane_group_num)
]  # 0.9for CBD and 1 for others
# Lane utilization adjustment factor

f_lu = [
    through_lane(Lane_num[0]),
    through_lane(Lane_num[1]),
    through_lane(Lane_num[2]),
    through_lane(Lane_num[3])
]
# left-right lane condition
LR_condition = [
    "Exclusive lane",
    "Exclusive lane",
    "Exclusive lane",
    "Exclusive lane",
]
# lef_turn adjust factor
f_lt = [flt_list[0],flt_list[1],flt_list[2],flt_list[3]]
# right turn adjustment factor
f_rt = [
    right_adjustment_factor(LR_condition[0], pr[0]),
    right_adjustment_factor(LR_condition[1], pr[1]),
    right_adjustment_factor(LR_condition[2], pr[2]),
    right_adjustment_factor(LR_condition[3], pr[3])
]
# left turn pedestrain/bicycle adjustment factor
f_lpb = [1 for i in range(lane_group_num)]
# right turn pedestrain/bicycle adjustment factor
f_rpb = [1 for i in range(lane_group_num)]

saturation = saturation_flow(
```

```
    base_saturation,
    Lane_num,
    f_w,
    f_hv,
    f_g,
    f_p,
    f_bb,
    f_a,
    f_lu,
    f_lt,
    f_rt,
    f_lpb,
    f_rpb,
    lane_group_num,
)

gc_ratio = [green_time_lane_group[i] / cycle for i in range(lane_group_num)]
lane_group_capacity = [
    green_time_lane_group[i] / cycle * saturation[i] for i in range(lane_group_num)
]
# lane_group_capacity
adjusted_demand = [Lane_group_volume[i] / PHF[i] for i in range(lane_group_num)]
vc_ratio = [
    adjusted_demand[i] / lane_group_capacity[i] for i in range(lane_group_num)
]
# progression factor
PF = [progression_factor(gc_ratio[i]) for i in range(lane_group_num)]

delay1 = Delay1(green_time_lane_group, cycle, vc_ratio, lane_group_num)

T = [0.25 for i in range(lane_group_num)]
I = [1 for i in range(lane_group_num)]
k = [0.5 for i in range(lane_group_num)]

delay2 = [
    900
    * T[i]
    * (
        (vc_ratio[i] - 1)
        + np.sqrt(
            (vc_ratio[i] - 1) * (vc_ratio[i] - 1)
            + 8 * k[i] * vc_ratio[i] * I[i] / (lane_group_capacity[i] * T[i])
        )
    )
    for i in range(lane_group_num)
]
```

```
    delay3 = [0 for i in range(lane_group_num)]

    delay = [delay1[i] + delay2[i] * PF[i] + delay3[i] for i in range(lane_group_num)]

    approach_delay = delay

    intersection_delay = (approach_delay[0] * adjusted_demand[0] +
                          approach_delay[1] * adjusted_demand[1] +
                          approach_delay[2] * adjusted_demand[2] +
                          approach_delay[3] * adjusted_demand[3]) / sum(adjusted_demand)

    return intersection_delay

if __name__ == "__main__":
    # lane width
    # Input
    EB_L_W = 12
    EB_T_W = 12
    EB_R_W = 12
    WB_L_W = 12
    WB_T_W = 12
    WB_R_W = 12
    NB_L_W = 12
    NB_T_W = 12
    NB_R_W = 12
    SB_L_W = 12
    SB_T_W = 12
    SB_R_W = 12
    Lane_width = [EB_L_W, WB_T_W, NB_L_W, SB_T_W]
    # lane number
    # Input
    EB_L_N = 1
    EB_T_N = 2
    EB_R_N = 1
    WB_L_N = 1
    WB_T_N = 2
    WB_R_N = 1
    NB_L_N = 1
    NB_T_N = 2
    NB_R_N = 1
    SB_L_N = 1
    SB_T_N = 2
    SB_R_N = 1
    Lane_num = [
        EB_L_N + EB_T_N + EB_R_N,
```

```
    WB_L_N + WB_T_N + WB_R_N,
    NB_L_N + NB_T_N + NB_R_N,
    SB_L_N + SB_T_N + SB_R_N,
]
# base saturation
# Input
definded_saturation = 1900

# Volume
# Input
EB_L_V = 116
EB_T_V = 405
EB_R_V = 0
WB_L_V = 130
WB_T_V = 450
WB_R_V = 0
NB_L_V = 101
NB_T_V = 620
NB_R_V = 0
SB_L_V = 86
SB_T_V = 718
SB_R_V = 0
Lane_group_volume = [
    EB_L_V + EB_T_V + EB_R_V,
    WB_L_V + WB_T_V + WB_R_V,
    NB_L_V + NB_T_V + NB_R_V,
    SB_L_V + SB_T_V + SB_R_V,
]
pr = [
    EB_R_V / EB_L_V + EB_T_V + EB_R_V,
    WB_R_V / WB_L_V + WB_T_V + WB_R_V,
    NB_R_V / NB_L_V + NB_T_V + NB_R_V,
    SB_R_V / SB_L_V + SB_T_V + SB_R_V,
]
# Green time
# Input
EB_L_G = 31
EB_T_G = 31
WB_L_G = 31
WB_T_G = 31
NB_L_G = 34
NB_T_G = 34
SB_L_G = 34
SB_T_G = 34
green_time_lane_group = [
    EB_T_G,
```

```
    WB_T_G,
    NB_T_G,
    SB_T_G,
  ]
  # Lane condition of left and right lanes
  # Input
  EB_L_C = "Exclusive lane"
  EB_R_C = "Exclusive lane"
  WB_L_C = "Exclusive lane"
  WB_R_C = "Exclusive lane"
  NB_L_C = "Exclusive lane"
  NB_R_C = "Exclusive lane"
  SB_L_C = "Exclusive lane"
  SB_R_C = "Exclusive lane"
  LR_condition = [EB_L_C, EB_R_C, WB_L_C, WB_R_C, NB_L_C, NB_R_C, SB_L_C, SB
_R_C]
  # Cycle
  # Input
  cycle = EB_T_G + NB_T_G

  # peak hour factor
  # Input
  defined_PHF = 0.95

  # Heavy-vehicle adjustment factor
  Heavy_ratio = 3  # Input
  defined_area_type_adjustment = 1

  # lane_num_in_LT_group = [EB_L_N, WB_L_N, NB_L_N, SB_L_N]

  lane_num_in_LT_group = [EB_L_N + EB_T_N + EB_R_N,
                          WB_L_N + WB_T_N + WB_R_N,
                          NB_L_N + NB_T_N + NB_R_N,
                          SB_L_N + SB_T_N + SB_R_N]
  LT_flow = [WB_T_V, EB_T_V, SB_T_V, NB_T_V]
  opposing_flow = [
    WB_L_V + WB_T_V + WB_R_V,
    EB_L_V + EB_T_V + EB_R_V,
    SB_L_V + SB_T_V + SB_R_V,
    NB_L_V + NB_T_V + NB_R_V,
  ]
  lane_num_in_opposing_approach = [
    WB_L_N + WB_T_N + WB_R_N,
    EB_L_N + EB_T_N + EB_R_N,
    SB_L_N + SB_T_N + SB_R_N,
    NB_L_N + NB_T_N + NB_R_N,
```

```
    ]
    LT_green_time = [EB_T_G, WB_T_G, NB_T_G, SB_T_G]

    opposing_green_time = [WB_T_G, EB_T_G, SB_T_G, NB_T_G]

    res = cal_permitted_coordinated(
        Lane_width,
        Lane_num,
        definded_saturation,
        Lane_group_volume,
        pr,
        green_time_lane_group,
        cycle,
        defined_PHF,
        LR_condition,
        Heavy_ratio,
        defined_area_type_adjustment,
        LT_flow,
        lane_num_in_LT_group,
        lane_num_in_opposing_approach,
        opposing_flow,
        LT_green_time,
        opposing_green_time,
    )
    print(res)

import numpy as np
# Lane utilization adjustment factor
def through_lane(lane_number):
    if lane_number == 1:
        f_lu = 1
    elif lane_number == 2:
        f_lu = 0.952
    else:
        f_lu = 0.908
    return f_lu

def left_lane(lane_number):
    if lane_number == 1:
        f_lu = 1
    else:
        f_lu = 0.971
    return f_lu

def right_lane(lane_number):
    if lane_number == 1:
```

```
        f_lu = 1
    else:
        f_lu = 0.885
    return f_lu

# right turn adjustment factor
def right_adjustment_factor(lane_condition, p_rt):
    if lane_condition == "Exclusive lane":
        f_rt = 0.85
    elif lane_condition == "Shared lane":
        f_rt = 1 - 0.15 * p_rt
    else:
        f_rt = 1 - 0.135 * p_rt
    return f_rt

def progression_factor(gc):
    if gc <= 0.2:
        pf = 1
    elif 0.2 < gc <= 0.3:
        pf = 0.986
    elif 0.3 < gc <= 0.4:
        pf = 0.895
    elif 0.4 < gc <= 0.5:
        pf = 0.767
    elif 0.5 < gc <= 0.6:
        pf = 0.576
    else:
        pf = 0.256
    return pf

def cal_protected_isolated(
    Lane_width,
    Lane_num,
    definded_saturation,
    defined_PHF,
    Lane_group_volume,
    pr,
    LR_condition,
    Heavy_ratio,
    defined_area_type_adjustment,
    opposing_flow,
    LT_flow,
    lane_num_in_LT_group,
    lane_num_in_opposing_approach,
):
    lane_group_num = 8
```

```
base_saturation = [definded_saturation for i in range(lane_group_num)]

# peak hour factor
# Input
#defined_PHF = 0.95
PHF = [defined_PHF for i in range(lane_group_num)]

# Lane width adjustment factor
f_w = [1 + (i - 12) / 30 for i in Lane_width]

ET = 2  # passenger-car equivalent
f_hv = [100 / (100 + Heavy_ratio * (ET - 1)) for i in range(lane_group_num)]
# Percentage grade adjustment factor
f_g = [1 for i in range(lane_group_num)]
# parking adjustment factor
f_p = [1 for i in range(lane_group_num)]
# bus blockage adjustment factor
f_bb = [1 for i in range(lane_group_num)]

f_a = [
    defined_area_type_adjustment for i in range(lane_group_num)
]  # 0.9for CBD and 1 for others
# Lane utilization adjustment factor

f_lu = [
    left_lane(Lane_num[0]),
    through_lane(Lane_num[1]),
    left_lane(Lane_num[2]),
    through_lane(Lane_num[3]),
    left_lane(Lane_num[4]),
    through_lane(Lane_num[5]),
    left_lane(Lane_num[6]),
    through_lane(Lane_num[7]),
]
# left-right lane condition
# LR_condition=['Exclusive lane','Exclusive lane','Exclusive lane','Exclusive lane',
#               'Exclusive lane','Exclusive lane','Exclusive lane','Exclusive lane']
# lef_turn adjust factor
f_lt = [0.95, 1, 0.95, 1, 0.95, 1, 0.95, 1]
# right turn adjustment factor
f_rt = [
    1,
    right_adjustment_factor(LR_condition[1], pr[1]),
    1,
    right_adjustment_factor(LR_condition[3], pr[3]),
    1,
```

```
    right_adjustment_factor(LR_condition[5], pr[5]),
    1,
    right_adjustment_factor(LR_condition[7], pr[7]),
]
# left turn pedestrain/bicycle adjustment factor
f_lpb = [1 for i in range(lane_group_num)]
# right turn pedestrain/bicycle adjustment factor
f_rpb = [1 for i in range(lane_group_num)]

def saturation_flow(
    s0, n, f_w, f_hv, f_g, f_p, f_bb, f_a, f_lu, f_lt, f_rt, f_lpb, f_rpb
):
    saturation = []
    for i in range(lane_group_num):
        s = (
            s0[i]
            * n[i]
            * f_w[i]
            * f_hv[i]
            * f_g[i]
            * f_p[i]
            * f_bb[i]
            * f_a[i]
            * f_lu[i]
            * f_lt[i]
            * f_rt[i]
            * f_lpb[i]
            * f_rpb[i]
        )
        saturation.append(s)
    return saturation

saturation = saturation_flow(
    base_saturation,
    Lane_num,
    f_w,
    f_hv,
    f_g,
    f_p,
    f_bb,
    f_a,
    f_lu,
    f_lt,
    f_rt,
    f_lpb,
    f_rpb,
```

```
)

adjusted_demand = [Lane_group_volume[i] / PHF[i] for i in range(lane_group_num)]
vs_ratio = [adjusted_demand[i] / saturation[i] for i in range(lane_group_num)]
# max(vs_ratio[0],vs_ratio[2])
critical_flow_ratio = [
    max(vs_ratio[0], vs_ratio[2]),
    max(vs_ratio[1], vs_ratio[3]),
    max(vs_ratio[4], vs_ratio[6]),
    max(vs_ratio[5], vs_ratio[7]),
]
yc = sum(critical_flow_ratio)
lost_time = 4
phase_number = 4
cycle = 1.5 * lost_time * phase_number / (1 - yc)
if cycle < 60:
    cycle = 60
xc = yc * cycle / (cycle - lost_time * phase_number)
g1 = np.ceil(critical_flow_ratio[0] * cycle / xc)
g2 = np.ceil(critical_flow_ratio[1] * cycle / xc)
g3 = np.ceil(critical_flow_ratio[2] * cycle / xc)
g4 = np.ceil(critical_flow_ratio[3] * cycle / xc)
green_time_lane_group = [g1, g2, g1, g2, g3, g4, g3, g4]

gc_ratio = [green_time_lane_group[i] / cycle for i in range(lane_group_num)]
lane_group_capacity = [
    green_time_lane_group[i] / cycle * saturation[i] for i in range(lane_group_num)
]
# lane_group_capacity
# adjusted_demand=[Lane_group_volume[i]/PHF[i] for i in range(lane_group_num)]
vc_ratio = [
    adjusted_demand[i] / lane_group_capacity[i] for i in range(lane_group_num)
]
# progression factor
PF = [progression_factor(gc_ratio[i]) for i in range(lane_group_num)]

def Delay1(green, cycle, vc):
    numerator = [
        0.5 * cycle * (1 - green[i] / cycle) * (1 - green[i] / cycle)
        for i in range(lane_group_num)
    ]
    denominator = [
        1 - min(1, vc[i]) * green[i] / cycle for i in range(lane_group_num)
    ]
    delay1 = [numerator[i] / denominator[i] for i in range(lane_group_num)]
    return delay1
```

```python
    delay1 = Delay1(green_time_lane_group, cycle, vc_ratio)

    T = [0.25 for i in range(lane_group_num)]
    I = [1 for i in range(lane_group_num)]
    k = [0.5 for i in range(lane_group_num)]

    delay2 = [
        900
        * T[i]
        * (
            (vc_ratio[i] - 1)
            + np.sqrt(
                (vc_ratio[i] - 1) * (vc_ratio[i] - 1)
                + 8 * k[i] * vc_ratio[i] * I[i] / (lane_group_capacity[i] * T[i])
            )
        )
        for i in range(lane_group_num)
    ]

    delay3 = [0 for i in range(lane_group_num)]

    delay = [delay1[i] + delay2[i] * PF[i] + delay3[i] for i in range(lane_group_num)]

    approach_delay = [
        (delay[0] * adjusted_demand[0] + delay[1] * adjusted_demand[1])
        / (adjusted_demand[0] + adjusted_demand[1]),
        (delay[2] * adjusted_demand[2] + delay[3] * adjusted_demand[3])
        / (adjusted_demand[2] + adjusted_demand[3]),
        (delay[4] * adjusted_demand[4] + delay[5] * adjusted_demand[5])
        / (adjusted_demand[4] + adjusted_demand[5]),
        (delay[6] * adjusted_demand[6] + delay[7] * adjusted_demand[7])
        / (adjusted_demand[6] + adjusted_demand[7]),
    ]

    intersection_delay = (
        approach_delay[0] * (adjusted_demand[0] + adjusted_demand[1])
        + approach_delay[1] * (adjusted_demand[2] + adjusted_demand[3])
        + approach_delay[2] * (adjusted_demand[4] + adjusted_demand[5])
        + approach_delay[3] * (adjusted_demand[6] + adjusted_demand[7])
    ) / sum(adjusted_demand)

    return intersection_delay

if __name__ == "__main__":
    # lane width
```

```
# Input
EB_L_W = 12
EB_T_W = 12
EB_R_W = 12
WB_L_W = 12
WB_T_W = 12
WB_R_W = 12
NB_L_W = 12
NB_T_W = 12
NB_R_W = 12
SB_L_W = 12
SB_T_W = 12
SB_R_W = 12
Lane_width = [EB_L_W, EB_T_W, WB_L_W, WB_T_W, NB_L_W, NB_T_W, SB_L_W, S
B_T_W]

# lane number
# Input
EB_L_N = 2
EB_T_N = 2
EB_R_N = 1
WB_L_N = 2
WB_T_N = 2
WB_R_N = 1
NB_L_N = 2
NB_T_N = 3
NB_R_N = 1
SB_L_N = 2
SB_T_N = 3
SB_R_N = 1
Lane_num = [
    EB_L_N,
    EB_T_N + EB_R_N,
    WB_L_N,
    WB_T_N + WB_R_N,
    NB_L_N,
    NB_T_N + NB_R_N,
    SB_L_N,
    SB_T_N + SB_R_N,
]
# base saturation
# Input
definded_saturation = 1900
defined_PHF = 0.95
# Volume
# Input
```

```
EB_L_V = 327
EB_T_V = 546
EB_R_V = 0
WB_L_V = 158
WB_T_V = 466
WB_R_V = 0
NB_L_V = 175
NB_T_V = 739
NB_R_V = 0
SB_L_V = 254
SB_T_V = 640
SB_R_V = 0

# Lane_group_volume=[116,405,130,450,101,620,86,718]
# Input
Lane_group_volume = [
    EB_L_V,
    EB_T_V + EB_R_V,
    WB_L_V,
    WB_T_V + WB_R_V,
    NB_L_V,
    NB_T_V + NB_R_V,
    SB_L_V,
    SB_T_V + SB_R_V,
]
pr = [
    0,
    EB_R_V / EB_T_V + EB_R_V,
    0,
    WB_R_V / WB_T_V + WB_R_V,
    0,
    NB_R_V / NB_T_V + NB_R_V,
    0,
    SB_R_V / SB_T_V + SB_R_V,
]

# Lane condition of left and right lanes
# Input
EB_L_C = "Exclusive lane"
EB_R_C = "Exclusive lane"
WB_L_C = "Exclusive lane"
WB_R_C = "Exclusive lane"
NB_L_C = "Exclusive lane"
NB_R_C = "Exclusive lane"
SB_L_C = "Exclusive lane"
SB_R_C = "Exclusive lane"
```

```
    LR_condition = [EB_L_C, EB_R_C, WB_L_C, WB_R_C, NB_L_C, NB_R_C, SB_L_C, SB
_R_C]

    # Heavy-vehicle adjustment factor
    Heavy_ratio = 3
    # area type adjustment factor
    defined_area_type_adjustment = 1

    lane_num_in_LT_group = [EB_L_N, WB_L_N, NB_L_N, SB_L_N]

    lane_num_in_opposing_approach = [
        WB_L_N + WB_T_N + WB_R_N,
        EB_L_N + EB_T_N + EB_R_N,
        SB_L_N + SB_T_N + SB_R_N,
        NB_L_N + NB_T_N + NB_R_N,
    ]
    LT_flow = [WB_T_V, EB_T_V, SB_T_V, NB_T_V]
    opposing_flow = [
        WB_L_V + WB_T_V + WB_R_V,
        EB_L_V + EB_T_V + EB_R_V,
        SB_L_V + SB_T_V + SB_R_V,
        NB_L_V + NB_T_V + NB_R_V,
    ]

    res = cal_protected_isolated(
        Lane_width,
        Lane_num,
        definded_saturation,
        defined_PHF,
        Lane_group_volume,
        pr,
        LR_condition,
        Heavy_ratio,
        defined_area_type_adjustment,
        opposing_flow,
        LT_flow,
        lane_num_in_LT_group,
        lane_num_in_opposing_approach,
    )

    print(res)

import numpy as np
# Lane utilization adjustment factor
def through_lane(lane_number):
    if lane_number == 1:
```

```
        f_lu = 1
    elif lane_number == 2:
        f_lu = 0.952
    else:
        f_lu = 0.908
    return f_lu

def left_lane(lane_number):
    if lane_number == 1:
        f_lu = 1
    else:
        f_lu = 0.971
    return f_lu

def right_lane(lane_number):
    if lane_number == 1:
        f_lu = 1
    else:
        f_lu = 0.885
    return f_lu

# right turn adjustment factor
def right_adjustment_factor(lane_condition, p_rt):
    if lane_condition == "Exclusive lane":
        f_rt = 0.85
    elif lane_condition == "Shared lane":
        f_rt = 1 - 0.15 * p_rt
    else:
        f_rt = 1 - 0.135 * p_rt
    return f_rt

def progression_factor(gc):
    if gc <= 0.2:
        pf = 1
    elif 0.2 < gc <= 0.3:
        pf = 0.986
    elif 0.3 < gc <= 0.4:
        pf = 0.895
    elif 0.4 < gc <= 0.5:
        pf = 0.767
    elif 0.5 < gc <= 0.6:
        pf = 0.576
    else:
        pf = 0.256
    return pf
```

```
def Delay1(green, cycle, vc, lane_group_num):
    delay1 = []
    for i in range(lane_group_num):
        numerator = 0.5 * cycle * (1 - green[i] / cycle) * (1 - green[i] / cycle)
        denominator = 1 - min(1, vc[i]) * green[i] / cycle
        pf = progression_factor(green[i] / cycle)
        delay = numerator / denominator * pf
        delay1.append(delay)
    return delay1

def cal_protected_coordinated(
    Lane_width,
    Lane_num,
    definded_saturation,
    Lane_group_volume,
    pr,
    green_time_lane_group,
    cycle,
    defined_PHF,
    LR_condition,
    Heavy_ratio,
    defined_area_type_adjustment,
):
    lane_group_num = 8
    base_saturation = [definded_saturation for i in range(lane_group_num)]
    PHF = [defined_PHF for i in range(lane_group_num)]
    # Lane width adjustment factor
    f_w = [1 + (i - 12) / 30 for i in Lane_width]

    ET = 2  # passenger-car equivalent
    f_hv = [100 / (100 + Heavy_ratio * (ET - 1)) for i in range(lane_group_num)]
    # Percentage grade adjustment factor
    f_g = [1 for i in range(lane_group_num)]
    # parking adjustment factor
    f_p = [1 for i in range(lane_group_num)]
    # bus blockage adjustment factor
    f_bb = [1 for i in range(lane_group_num)]
    # area type adjustment factor

    f_a = [
        defined_area_type_adjustment for i in range(lane_group_num)
    ]  # 0.9for CBD and 1 for others
    # Lane utilization adjustment factor

    f_lu = [
        left_lane(Lane_num[0]),
```

```
    through_lane(Lane_num[1]),
    left_lane(Lane_num[2]),
    through_lane(Lane_num[3]),
    left_lane(Lane_num[4]),
    through_lane(Lane_num[5]),
    left_lane(Lane_num[6]),
    through_lane(Lane_num[7]),
]
# left-right lane condition
# LR_condition=['Exclusive lane','Exclusive lane','Exclusive lane','Exclusive lane',
#                'Exclusive lane','Exclusive lane','Exclusive lane','Exclusive lane']
# lef_turn adjust factor
f_lt = [0.95, 1, 0.95, 1, 0.95, 1, 0.95, 1]
# right turn adjustment factor
f_rt = [
    1,
    right_adjustment_factor(LR_condition[1], pr[1]),
    1,
    right_adjustment_factor(LR_condition[3], pr[3]),
    1,
    right_adjustment_factor(LR_condition[5], pr[5]),
    1,
    right_adjustment_factor(LR_condition[7], pr[7]),
]
# left turn pedestrain/bicycle adjustment factor
f_lpb = [1 for i in range(lane_group_num)]
# right turn pedestrain/bicycle adjustment factor
f_rpb = [1 for i in range(lane_group_num)]

def saturation_flow(
    s0, n, f_w, f_hv, f_g, f_p, f_bb, f_a, f_lu, f_lt, f_rt, f_lpb, f_rpb
):
    saturation = []
    for i in range(lane_group_num):
        s = (
            s0[i]
            * n[i]
            * f_w[i]
            * f_hv[i]
            * f_g[i]
            * f_p[i]
            * f_bb[i]
            * f_a[i]
            * f_lu[i]
            * f_lt[i]
            * f_rt[i]
```

```
            * f_lpb[i]
            * f_rpb[i]
        )
        saturation.append(s)
    return saturation

saturation = saturation_flow(
    base_saturation,
    Lane_num,
    f_w,
    f_hv,
    f_g,
    f_p,
    f_bb,
    f_a,
    f_lu,
    f_lt,
    f_rt,
    f_lpb,
    f_rpb,
)

gc_ratio = [green_time_lane_group[i] / cycle for i in range(lane_group_num)]
lane_group_capacity = [
    green_time_lane_group[i] / cycle * saturation[i] for i in range(lane_group_num)
]
# lane_group_capacity
adjusted_demand = [Lane_group_volume[i] / PHF[i] for i in range(lane_group_num)]
vc_ratio = [
    adjusted_demand[i] / lane_group_capacity[i] for i in range(lane_group_num)
]
# progression factor
PF = [progression_factor(gc_ratio[i]) for i in range(lane_group_num)]
delay1 = Delay1(green_time_lane_group, cycle, vc_ratio, lane_group_num)
T = [0.25 for i in range(lane_group_num)]
I = [1 for i in range(lane_group_num)]
k = [0.5 for i in range(lane_group_num)]

delay2 = [
    900
    * T[i]
    * (
        (vc_ratio[i] - 1)
        + np.sqrt(
            (vc_ratio[i] - 1) * (vc_ratio[i] - 1)
            + 8 * k[i] * vc_ratio[i] * I[i] / (lane_group_capacity[i] * T[i])
```

```
            )
        )
        for i in range(lane_group_num)
    ]

    delay3 = [0 for i in range(lane_group_num)]

    delay = [delay1[i] + delay2[i] * PF[i] + delay3[i] for i in range(lane_group_num)]

    approach_delay = [
        (delay[0] * adjusted_demand[0] + delay[1] * adjusted_demand[1])
        / (adjusted_demand[0] + adjusted_demand[1]),
        (delay[2] * adjusted_demand[2] + delay[3] * adjusted_demand[3])
        / (adjusted_demand[2] + adjusted_demand[3]),
        (delay[4] * adjusted_demand[4] + delay[5] * adjusted_demand[5])
        / (adjusted_demand[4] + adjusted_demand[5]),
        (delay[6] * adjusted_demand[6] + delay[7] * adjusted_demand[7])
        / (adjusted_demand[6] + adjusted_demand[7]),
    ]

    intersection_delay = (
        approach_delay[0] * (adjusted_demand[0] + adjusted_demand[1])
        + approach_delay[1] * (adjusted_demand[2] + adjusted_demand[3])
        + approach_delay[2] * (adjusted_demand[4] + adjusted_demand[5])
        + approach_delay[3] * (adjusted_demand[6] + adjusted_demand[7])
    ) / sum(adjusted_demand)

    return intersection_delay

if __name__ == "__main__":
    # lane width
    # Input
    EB_L_W = 12
    EB_T_W = 12
    EB_R_W = 12
    WB_L_W = 12
    WB_T_W = 12
    WB_R_W = 12
    NB_L_W = 12
    NB_T_W = 12
    NB_R_W = 12
    SB_L_W = 12
    SB_T_W = 12
    SB_R_W = 12
```

```
    Lane_width = [EB_L_W, EB_T_W, WB_L_W, WB_T_W, NB_L_W, NB_T_W, SB_L_W, S
B_T_W]

    # lane number
    # Input
    EB_L_N = 1
    EB_T_N = 2
    EB_R_N = 1
    WB_L_N = 1
    WB_T_N = 2
    WB_R_N = 1
    NB_L_N = 1
    NB_T_N = 2
    NB_R_N = 1
    SB_L_N = 1
    SB_T_N = 2
    SB_R_N = 1

    Lane_num = [
        EB_L_N,
        EB_T_N + EB_R_N,
        WB_L_N,
        WB_T_N + WB_R_N,
        NB_L_N,
        NB_T_N + NB_R_N,
        SB_L_N,
        SB_T_N + SB_R_N,
    ]

    # base saturation
    # Input
    definded_saturation = 1900

    # Volume
    # Input
    EB_L_V = 116
    EB_T_V = 405
    EB_R_V = 0
    WB_L_V = 130
    WB_T_V = 450
    WB_R_V = 0
    NB_L_V = 101
    NB_T_V = 620
    NB_R_V = 0
    SB_L_V = 86
    SB_T_V = 718
```

```
SB_R_V = 0

# Lane_group_volume=[116,405,130,450,101,620,86,718]
Lane_group_volume = [
    EB_L_V,
    EB_T_V + EB_R_V,
    WB_L_V,
    WB_T_V + WB_R_V,
    NB_L_V,
    NB_T_V + NB_R_V,
    SB_L_V,
    SB_T_V + SB_R_V,
]
pr = [
    0,
    EB_R_V / EB_T_V + EB_R_V,
    0,
    WB_R_V / WB_T_V + WB_R_V,
    0,
    NB_R_V / NB_T_V + NB_R_V,
    0,
    SB_R_V / SB_T_V + SB_R_V,
]
# Green time
# Input
EB_L_G = 23
EB_T_G = 31
WB_L_G = 23
WB_T_G = 31
NB_L_G = 20
NB_T_G = 34
SB_L_G = 20
SB_T_G = 34
green_time_lane_group = [
    EB_L_G,
    EB_T_G,
    WB_L_G,
    WB_T_G,
    NB_L_G,
    NB_T_G,
    SB_L_G,
    SB_T_G,
]
# Cycle
# Input
cycle = 108
```

```
    # peak hour factor
    # Input
    defined_PHF = 0.92

    # Lane condition of left and right lanes
    # Input
    EB_L_C = "Exclusive lane"
    EB_R_C = "Exclusive lane"
    WB_L_C = "Exclusive lane"
    WB_R_C = "Exclusive lane"
    NB_L_C = "Exclusive lane"
    NB_R_C = "Exclusive lane"
    SB_L_C = "Exclusive lane"
    SB_R_C = "Exclusive lane"
    LR_condition = [EB_L_C, EB_R_C, WB_L_C, WB_R_C, NB_L_C, NB_R_C, SB_L_C, SB
_R_C]

    # Heavy-vehicle adjustment factor
    Heavy_ratio = 3  # Input
    defined_area_type_adjustment = 1

    res = cal_protected_coordinated(
        Lane_width,
        Lane_num,
        definded_saturation,
        Lane_group_volume,
        pr,
        green_time_lane_group,
        cycle,
        defined_PHF,
        LR_condition,
        Heavy_ratio,
        defined_area_type_adjustment,
    )
    print(res)
import numpy as np
# Lane utilization adjustment factor
def through_lane(lane_number):
    if lane_number == 1:
        f_lu = 1
    elif lane_number == 2:
        f_lu = 0.952
    else:
        f_lu = 0.908
    return f_lu
```

```
def left_lane(lane_number):
    if lane_number == 1:
        f_lu = 1
    else:
        f_lu = 0.971
    return f_lu

def right_lane(lane_number):
    if lane_number == 1:
        f_lu = 1
    else:
        f_lu = 0.885
    return f_lu

# right turn adjustment factor
def right_adjustment_factor(lane_condition, p_rt):
    if lane_condition == "Exclusive lane":
        f_rt = 0.85
    elif lane_condition == "Shared lane":
        f_rt = 1 - 0.15 * p_rt
    else:
        f_rt = 1 - 0.135 * p_rt
    return f_rt

# progression adjustment factor
def progression_factor(gc):
    if gc <= 0.2:
        pf = 1
    elif 0.2 < gc <= 0.3:
        pf = 0.986
    elif 0.3 < gc <= 0.4:
        pf = 0.895
    elif 0.4 < gc <= 0.5:
        pf = 0.767
    elif 0.5 < gc <= 0.6:
        pf = 0.576
    else:
        pf = 0.256
    return pf

def EL1(calculated_oppsing_flow):
    if calculated_oppsing_flow <= 1:
        el = 1.3
    elif 1 < calculated_oppsing_flow <= 200:
        el = 1.6
```

```
    elif 200 < calculated_oppsing_flow <= 400:
        el = 1.9
    elif 400 < calculated_oppsing_flow <= 600:
        el = 2.3
    elif 600 < calculated_oppsing_flow <= 800:
        el = 2.8
    elif 800 < calculated_oppsing_flow <= 1000:
        el = 3.3
    else:
        el = 4.0
    return el

def cal_permitted_isolated(
    Lane_width,
    Lane_num,
    Lane_num_1,
    definded_saturation,
    defined_PHF,
    Lane_group_volume,
    Lane_group_volume_1,
    pr,
    LR_condition,
    Heavy_ratio,
    defined_area_type_adjustment,
    LT_flow,
    lane_num_in_LT_group,
    lane_num_in_opposing_approach,
    opposing_flow,
):
    lane_group_num = 4
    lane_group_num_1 = 8
    # peak hour factor
    # Input
    #defined_PHF = 0.95
    PHF = [defined_PHF for i in range(lane_group_num)]
    PHF_1 = [defined_PHF for i in range(lane_group_num_1)]
    base_saturation = [definded_saturation for i in range(lane_group_num)]
    base_saturation_1 = [definded_saturation for i in range(lane_group_num_1)]
    initial_adjusted_demand = [
        Lane_group_volume_1[i] / PHF_1[i] for i in range(lane_group_num_1)
    ]
    initial_vs_ratio = [
        initial_adjusted_demand[i] / (base_saturation_1[i] * Lane_num_1[i]) for i in range(lane_gro
up_num_1)
    ]
    # max(vs_ratio[0],vs_ratio[2])
```

```
    initial_critical_flow_ratio = [
        max(initial_vs_ratio[0], initial_vs_ratio[3]),
        max(initial_vs_ratio[1], initial_vs_ratio[2]),
        max(initial_vs_ratio[4], initial_vs_ratio[7]),
        max(initial_vs_ratio[5], initial_vs_ratio[6]),
    ]
    yc = sum(initial_critical_flow_ratio)
    lost_time = 4
    phase_number = 4
    cycle = 1.5 * lost_time * phase_number / (1 - yc)
    if cycle < 60:
        cycle = 60
    xc = yc * cycle / (cycle - lost_time * phase_number)
    g1 = np.ceil(initial_critical_flow_ratio[0] * cycle / xc)
    g2 = np.ceil(initial_critical_flow_ratio[1] * cycle / xc)
    g3 = np.ceil(initial_critical_flow_ratio[2] * cycle / xc)
    g4 = np.ceil(initial_critical_flow_ratio[3] * cycle / xc)
    green_time_lane_group = [g2, g2, g4, g4]
    if g2+g4+lost_time*4<60:
        per_initial_adjusted_demand = [
            Lane_group_volume[i] / PHF[i] for i in range(lane_group_num)
        ]
        initial_vs_ratio = [
            per_initial_adjusted_demand[i] / (base_saturation[i]*Lane_num[i]) for i in range(lane_gr
oup_num)
        ]
        per_initial_critical_flow_ratio = [
            max(initial_vs_ratio[0], initial_vs_ratio[1]),
            max(initial_vs_ratio[2], initial_vs_ratio[3]),
        ]
        per_yc = sum(per_initial_critical_flow_ratio)
        per_lost_time = 4
        per_phase_number = 2
        per_cycle = 1.5 * per_lost_time * per_phase_number / (1 - per_yc)
        if per_cycle < 60:
            per_cycle = 60
        per_xc = per_yc * per_cycle / (per_cycle - per_lost_time * per_phase_number)
        per_g1 = np.ceil(per_initial_critical_flow_ratio[0] * per_cycle / per_xc)
        per_g2 = np.ceil(per_initial_critical_flow_ratio[1] * per_cycle / per_xc)
        green_time_lane_group = [per_g1, per_g1, per_g2, per_g2]

    LT_number = 4
    LT_green_time = [g2, g2, g4, g4]
    opposing_green_time = [g2, g2, g4, g4]

    PHF_L = [defined_PHF for i in range(LT_number)]  # Input
```

```
    adjusted_LT = [LT_flow[i] / PHF_L[i] for i in range(LT_number)]
    adjusted_opposing = [opposing_flow[i] / PHF_L[i] for i in range(LT_number)]
    defined_lost_time = 4  # Input
    lost_time = [defined_lost_time for i in range(LT_number)]
    LT_flow_per_cycle = [adjusted_LT[i] * cycle / 3600 for i in range(LT_number)]
    # opposing_lu_factor=[through_lane(lane_num_in_opposing_approach[i]) for i in range(LT_n
umber)]
    opposing_lu_factor = [0.95, 0.95, 0.95, 0.95]
    opposing_flow_per_lane = [
        adjusted_opposing[i]
        * cycle
        / (3600 * lane_num_in_opposing_approach[i] * opposing_lu_factor[i])
        for i in range(LT_number)
    ]
    gf = [0 for i in range(LT_number)]
    rpo = [1.333 for i in range(LT_number)]
    qro = [
        max(1 - rpo[i] * opposing_green_time[i] / cycle, 0) for i in range(LT_number)
    ]
    gq = [
        opposing_flow_per_lane[i]
        * qro[i]
        / (0.5 - opposing_flow_per_lane[i] * (1 - qro[i]) / opposing_green_time[i])
        for i in range(LT_number)
    ]
    gu = [LT_green_time[i] - gq[i] for i in range(LT_number)]
    calculated_opposing = [
        opposing_flow[i] / opposing_lu_factor[i] for i in range(LT_number)
    ]
    el = [EL1(calculated_opposing[i]) for i in range(LT_number)]
    pl = [1 for i in range(LT_number)]
    f_min = [2 * (1 + pl[i]) / LT_green_time[i] for i in range(LT_number)]
    fm = [
        gf[i] / LT_green_time[i]
        + (gu[i] / LT_green_time[i]) * (1 / (1 + pl[i] * (el[i] - 1)))
        for i in range(LT_number)
    ]
    flt_list = [
        (fm[i] + 0.91 * (lane_num_in_LT_group[i] - 1)) / lane_num_in_LT_group[i]
        for i in range(LT_number)
    ]

    # Lane width adjustment factor
    f_w = [1 + (i - 12) / 30 for i in Lane_width]

    ET = 2  # passenger-car equivalent
```

```
f_hv = [100 / (100 + Heavy_ratio * (ET - 1)) for i in range(lane_group_num)]
# Percentage grade adjustment factor
f_g = [1 for i in range(lane_group_num)]
# parking adjustment factor
f_p = [1 for i in range(lane_group_num)]
# bus blockage adjustment factor
f_bb = [1 for i in range(lane_group_num)]

f_a = [
    defined_area_type_adjustment for i in range(lane_group_num)
]  # 0.9for CBD and 1 for others
# Lane utilization adjustment factor

f_lu = [
    through_lane(Lane_num[0]),
    through_lane(Lane_num[1]),
    through_lane(Lane_num[2]),
    through_lane(Lane_num[3])
]
# left-right lane condition
# LR_condition=['Exclusive lane','Exclusive lane','Exclusive lane','Exclusive lane',
#               'Exclusive lane','Exclusive lane','Exclusive lane','Exclusive lane']
# lef_turn adjust factor
f_lt = [flt_list[0], flt_list[1], flt_list[2], flt_list[3]]
# right turn adjustment factor
f_rt = [
    right_adjustment_factor(LR_condition[0], pr[0]),
    right_adjustment_factor(LR_condition[1], pr[1]),
    right_adjustment_factor(LR_condition[2], pr[2]),
    right_adjustment_factor(LR_condition[3], pr[3])
]
# left turn pedestrain/bicycle adjustment factor
f_lpb = [1 for i in range(lane_group_num)]
# right turn pedestrain/bicycle adjustment factor
f_rpb = [1 for i in range(lane_group_num)]

def saturation_flow(
    s0, n, f_w, f_hv, f_g, f_p, f_bb, f_a, f_lu, f_lt, f_rt, f_lpb, f_rpb
):
    saturation = []
    for i in range(lane_group_num):
        s = (
            s0[i]
            * n[i]
            * f_w[i]
            * f_hv[i]
```

```
            * f_g[i]
            * f_p[i]
            * f_bb[i]
            * f_a[i]
            * f_lu[i]
            * f_lt[i]
            * f_rt[i]
            * f_lpb[i]
            * f_rpb[i]
        )
        saturation.append(s)
    return saturation

saturation = saturation_flow(
    base_saturation,
    Lane_num,
    f_w,
    f_hv,
    f_g,
    f_p,
    f_bb,
    f_a,
    f_lu,
    f_lt,
    f_rt,
    f_lpb,
    f_rpb,
)

gc_ratio = [green_time_lane_group[i] / cycle for i in range(lane_group_num)]
lane_group_capacity = [
    green_time_lane_group[i] / cycle * saturation[i] for i in range(lane_group_num)
]
# lane_group_capacity
adjusted_demand = [Lane_group_volume[i] / PHF[i] for i in range(lane_group_num)]
vc_ratio = [
    adjusted_demand[i] / lane_group_capacity[i] for i in range(lane_group_num)
]
# progression factor
PF = [progression_factor(gc_ratio[i]) for i in range(lane_group_num)]

def Delay1(green, cycle, vc):
    numerator = [
        0.5 * cycle * (1 - green[i] / cycle) * (1 - green[i] / cycle)
        for i in range(lane_group_num)
    ]
```

```
        denominator = [
            1 - min(1, vc[i]) * green[i] / cycle for i in range(lane_group_num)
        ]
        delay1 = [numerator[i] / denominator[i] for i in range(lane_group_num)]
        return delay1

    delay1 = Delay1(green_time_lane_group, cycle, vc_ratio)

    T = [0.25 for i in range(lane_group_num)]
    I = [1 for i in range(lane_group_num)]
    k = [0.5 for i in range(lane_group_num)]

    delay2 = [
        900
        * T[i]
        * (
            (vc_ratio[i] - 1)
            + np.sqrt(
                (vc_ratio[i] - 1) * (vc_ratio[i] - 1)
                + 8 * k[i] * vc_ratio[i] * I[i] / (lane_group_capacity[i] * T[i])
            )
        )
        for i in range(lane_group_num)
    ]

    delay3 = [0 for i in range(lane_group_num)]

    delay = [delay1[i] + delay2[i] * PF[i] + delay3[i] for i in range(lane_group_num)]

    approach_delay = delay

    intersection_delay = (approach_delay[0] * adjusted_demand[0] +
                          approach_delay[1] * adjusted_demand[1] +
                          approach_delay[2] * adjusted_demand[2] +
                          approach_delay[3] * adjusted_demand[3]) / sum(adjusted_demand)

    return intersection_delay

if __name__ == "__main__":
    lane_group_num = 8
    # lane width
    # Input
    EB_L_W = 12
    EB_T_W = 12
    EB_R_W = 12
    WB_L_W = 12
```

```
WB_T_W = 12
WB_R_W = 12
NB_L_W = 12
NB_T_W = 12
NB_R_W = 12
SB_L_W = 12
SB_T_W = 12
SB_R_W = 12
Lane_width = [EB_L_W, WB_T_W, NB_L_W, SB_T_W]
# lane number
# Input
EB_L_N = 2
EB_T_N = 2
EB_R_N = 1
WB_L_N = 2
WB_T_N = 2
WB_R_N = 1
NB_L_N = 2
NB_T_N = 3
NB_R_N = 1
SB_L_N = 2
SB_T_N = 3
SB_R_N = 1
Lane_num = [
    EB_L_N + EB_T_N + EB_R_N,
    WB_L_N + WB_T_N + WB_R_N,
    NB_L_N + NB_T_N + NB_R_N,
    SB_L_N + SB_T_N + SB_R_N
]
Lane_num_1 = [
    EB_L_N, EB_T_N + EB_R_N,
    WB_L_N, WB_T_N + WB_R_N,
    NB_L_N, NB_T_N + NB_R_N,
    SB_L_N, SB_T_N + SB_R_N
]
# base saturation
# base saturation
# Input
definded_saturation = 1900
defined_PHF = 0.95
# Volume
# Input
EB_L_V = 327
EB_T_V = 546
EB_R_V = 0
WB_L_V = 158
```

```
    WB_T_V = 466
    WB_R_V = 0
    NB_L_V = 175
    NB_T_V = 739
    NB_R_V = 0
    SB_L_V = 254
    SB_T_V = 640
    SB_R_V = 0
    # Lane_group_volume=[116,405,130,450,101,620,86,718]
    # Input
    Lane_group_volume = [
        EB_L_V + EB_T_V + EB_R_V,
        WB_L_V + WB_T_V + WB_R_V,
        NB_L_V + NB_T_V + NB_R_V,
        SB_L_V + SB_T_V + SB_R_V
    ]
    Lane_group_volume_1 = [
        EB_L_V, EB_T_V + EB_R_V,
        WB_L_V, WB_T_V + WB_R_V,
        NB_L_V, NB_T_V + NB_R_V,
        SB_L_V, SB_T_V + SB_R_V
    ]
    pr = [
        EB_R_V / EB_L_V + EB_T_V + EB_R_V,
        WB_R_V / WB_L_V + WB_T_V + WB_R_V,
        NB_R_V / NB_L_V + NB_T_V + NB_R_V,
        SB_R_V / SB_L_V + SB_T_V + SB_R_V
    ]

    # Lane condition of left and right lanes
    # Input
    EB_L_C = "Exclusive lane"
    EB_R_C = "Exclusive lane"
    WB_L_C = "Exclusive lane"
    WB_R_C = "Exclusive lane"
    NB_L_C = "Exclusive lane"
    NB_R_C = "Exclusive lane"
    SB_L_C = "Exclusive lane"
    SB_R_C = "Exclusive lane"
    LR_condition = [EB_L_C, EB_R_C, WB_L_C, WB_R_C, NB_L_C, NB_R_C, SB_L_C, SB
_R_C]

    # Heavy-vehicle adjustment factor
    Heavy_ratio = 3
    # area type adjustment factor
    defined_area_type_adjustment = 1
```

```
lane_num_in_LT_group = [EB_L_N + EB_T_N + EB_R_N,
                        WB_L_N + WB_T_N + WB_R_N,
                        NB_L_N + NB_T_N + NB_R_N,
                        SB_L_N + SB_T_N + SB_R_N]

lane_num_in_opposing_approach = [
    WB_L_N + WB_T_N + WB_R_N,
    EB_L_N + EB_T_N + EB_R_N,
    SB_L_N + SB_T_N + SB_R_N,
    NB_L_N + NB_T_N + NB_R_N,
]
LT_flow = [WB_T_V, EB_T_V, SB_T_V, NB_T_V]
opposing_flow = [
    WB_L_V + WB_T_V + WB_R_V,
    EB_L_V + EB_T_V + EB_R_V,
    SB_L_V + SB_T_V + SB_R_V,
    NB_L_V + NB_T_V + NB_R_V,
]
res = cal_permitted_isolated(
    Lane_width,
    Lane_num,
    Lane_num_1,
    definded_saturation,
    defined_PHF,
    Lane_group_volume,
    Lane_group_volume_1,
    pr,
    LR_condition,
    Heavy_ratio,
    defined_area_type_adjustment,
    LT_flow,
    lane_num_in_LT_group,
    lane_num_in_opposing_approach,
    opposing_flow,
)
print(res)
```