Turkish Journal of Electrical Engineering and Computer Sciences

Volume 32 | Number 1 Article 1

2-7-2024

Traffic signal optimization using multiobjective linear programming for oversaturated traffic conditions

Mustafa Murat COŞKUN
e1881143@ceng.metu.edu.tr

Cevat ŞENER
sener@ceng.metu.edu.tr

İsmail Hakkı TOROSLU
toroslu@ceng.metu.edu.tr

Recommended Citation

COŞKUN, Mustafa Murat; ŞENER, Cevat; and TOROSLU, İsmail Hakkı (2024) "Traffic signal optimization using multiobjective linear programming for oversaturated traffic conditions," *Turkish Journal of Electrical Engineering and Computer Sciences*: Vol. 32: No. 1, Article 1. https://doi.org/10.55730/1300-0632.4052
Available at: https://journals.tubitak.gov.tr/elektrik/vol32/iss1/1

Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2024) 32: 1 – 20
© TÜBİTAK
doi:10.55730/1300-0632.4052

Traffic signal optimization using multiobjective linear programming for oversaturated traffic conditions

Mustafa Murat COŞKUN[1], **Cevat ŞENER**[1,2], **İsmail Hakkı TOROSLU**[1,3]*
[1]Department of Computer Engineering, Faculty of Engineering, Middle East Technical University, Ankara, Turkiye

Received: 20.02.2023 • **Accepted/Published Online:** 28.11.2023 • **Final Version:** 07.02.2024

Abstract: In this study, we present a framework designed to optimize signals at intersections experiencing oversaturated traffic conditions, utilizing mixed-integer linear programming (MILP) techniques. The proposed MILP solutions were developed with different objective functions, namely a reduction in the total remaining queue and fair distribution of the remaining queue after each signal cycle. Our framework contains two distinct stages. The initial stage applies two distinct MILP methodologies, while the subsequent stage employs a neighborhood search method to further reduce the delays associated with the green signal timings derived from the first stage. Ultimately, to evaluate their effectiveness across various intersections, we employed the HCM 2000 delay model for all the models we developed. Our experimental results show that the proposed approach reduces the delay significantly for various intersection designs.

Key words: Mixed-integer linear programming, traffic signal optimization, signalized intersections, oversaturated conditions, deterministic queuing

1. Introduction

In most metropolitan areas, traffic congestion has escalated to significant levels due primarily to population growth and urbanization, emerging as a pressing issue for both residents and decision-makers [1]. Therefore, the most important objective of traffic engineering is to create more habitable cities by reducing traffic congestion within urban traffic networks.

Although there is no universally agreed definition of traffic congestion, it can be described as a situation that arises when traffic demands on road networks or the volume of vehicles seeking access to these networks surpass the network's capacity. In simpler terms, urban traffic encounters congestion when the road networks' capacity falls below the traffic demands [2].

Traffic signal control strategies are primarily designed to optimize traffic flows at intersections and in urban road networks with the most efficiency. Researchers have focused on various goals in their efforts to formulate traffic control strategies for efficient traffic management. These objectives can be in forms such as minimizing the total number of stops, maximizing traffic flow, reducing total queue lengths, and reducing fuel consumption. Furthermore, traffic control systems must consider the safety of both vehicles and pedestrians at intersections.

To develop effective traffic signal control management, a variety of optimization techniques and artificial intelligence-based algorithms have been employed. An effective traffic signal control system should be cost-effective, easy to implement, and applicable to real-world conditions. It should aim to maximize traffic flows at

*Correspondence: toroslu@ceng.metu.edu.tr

intersections, considering the traffic volumes in each direction. In the present study, we use mixed-integer linear programming (MILP) approaches because of their clarity and cost efficiency in solving traffic signal control optimization problems.

Vehicles are required to come to a stop and wait at intersections when they approach while the traffic signal displays a red light. In instances in which the traffic demand is relatively low (i.e. undersaturated), it is possible to design the signal cycle in such a way that vehicles that have come to a stop at the preceding red light can go through the intersection during the next green phase. However, in cases of oversaturation, no matter how the cycle length is configured, there will still be some vehicles left at the intersection after the green phase finishes.

If oversaturated traffic conditions are not handled at intersections, they can get worse at each signal cycle. Therefore, until the traffic flow rate drops below the capacity of the road, the traffic signal cycle settings of the intersections must be adjusted to reduce the stress of the traffic. The aim of the present work was to reduce the queue left at intersections after each cycle in order to reduce the delays of vehicles due to oversaturated conditions.

This paper introduces a novel framework that includes MILP approaches, each with different objective functions aimed at optimizing fixed-time traffic signal control for isolated signalized intersections facing oversaturated conditions. The objective functions were designed to address the limitations of MILP models previously explored in the literature. They include minimizing the total residual queue length and fairly distributing residual queues among lane groups under oversaturated conditions. Our models assume a fixed and uniform vehicle arrival pattern, as seen in linear programming methods in the literature [3, 4]. We evaluate the developed models based on delay performance criteria and queue length and analyze the impact of different phase designs on delays.

Within our proposed framework, a new MILP model, abbreviated as MMQLM, was developed to be able to handle the fairness issue regarding residual queues, which is observed in the MILP model inspired by Liu (2008). We demonstrate that this new approach produces better solutions in most cases. Additionally, the framework includes a novel neighborhood search phase to enhance the solution quality obtained from the MILP models. The main contributions of our paper can be summarized as follows:

- We reformulated the maximization of departure rate formula from Liu (2008) by minimizing the residual queue, simplifying the model.
- We utilized the HCM 2000 delay model to assess the actual impact of the proposed queue-based solutions and to evaluate the quality of these solutions.
- We introduced an alternative optimization function that fairly distributes the residual queues across all lanes.
- We developed an end-to-end framework for implementing the above-mentioned MILP solutions, followed by a novel neighborhood search algorithm to further enhance the solution's quality.
- We validated the success of the proposed framework across various intersection architectures using simulation tools.

This paper is organized as follows: Section 2 introduces related work and compares it with our proposed model. Section 3 introduces and defines the core concepts in traffic engineering. Section 4 describes the details of the linear programming-based approaches proposed for traffic signal optimization. Section 5 presents the results of further experiments and, finally, section 6 contains our conclusions.

2. Related work

The main objective of traffic signal optimization is to minimize vehicle delays at intersections. Estimating delays is not a straightforward task, especially when considering the nonlinearity of the HCM 2000 delay estimation formula. Consequently, classical linear programming-based optimization methods are not always directly applicable. To deal with this complexity, a variety of approaches have been employed, including neural networks [5, 6], fuzzy logic [8, 24], reinforcement learning [9–13], and genetic algorithms [1, 14–17].

Since the signal optimization problem is inherently very complex, linear programming-based methods should have certain assumptions, like the constant speed of vehicles [3, 4, 18]. In the present work, we made similar assumptions, and we showed that the method proposed for isolated oversaturated intersections in [4] can be further improved. In addition to optimizing the signals for isolated intersection, recent research has focused on optimizing signals of multiple intersections.

The signal optimization for arterial roads is an important variant. The approach used in [4] has been extended in [19] as a two-stage arterial signal coordination model. [19] presents a two-way signal coordinated control method aimed at enhancing the traffic efficiency of oversaturated arterial traffic. The approach involves establishing a traffic flow model based on LWR theory, utilizing vehicle trajectory data for model verification, and introducing a coordinated control model that optimizes cycle length, green time, and signal coordination at consecutive intersections through a multiobjective algorithm to maximize throughput and minimize delay simultaneously.

The complexity of the signal optimization problem increases when attempting to optimize signals at multiple intersections simultaneously. Most recent work, such as [20] and [21], attacks different versions of this problem. Due to the complexities of these optimization problem variants, mainly heuristic-based approaches are employed.

[20] proposes a region-based evaluation particle swarm optimization algorithm, REPSO, with solution libraries to provide a signal timing scheme for undersaturated and oversaturated traffic flow states in real time and a region-based evaluation strategy to reduce the number of fitness evaluations.

A two-stage game perimeter control, TSGPC, model is proposed in [21] that integrates macro and micro control based on game theory to allocate regional regulation quantity to each intersection reasonably and to optimize signal timing schemes, respectively, where the queue length predicted by the Kalman filter is used as the basis of games.

3. Traffic engineering

There are various types of intersections and the present study focuses on signalized intersections. In urban areas, signalized intersections typically fall into three categories: isolated intersections, arterial networks, and general (grid) networks [22]; herein we focus on isolated intersections.

Traffic signal control systems can be classified into three main categories. Fixed-time traffic signal control aims to maximize traffic flow at intersections, with signal settings based on historical data and minimal changes throughout the day. As a result, fixed-time traffic signal control strategies struggle to adapt to fluctuations in traffic demands. The other two methods, namely traffic-actuated and adaptive traffic control strategies, adjust signal settings in real time based on detected traffic demands through sensors or technologies like GPS. However, these methods are costlier than fixed-time strategies.

Traffic flow in a road network can be expressed as the total number of vehicles that pass through a point in a certain time interval. The total number of vehicles is generally expressed as the hourly traffic volume (demand) or simply the flow rate. Traffic flows can also be expressed in lanes at signalized intersections.

In traffic engineering, the first step is phase design. Using the structures of phases, the number of vehicles from different directions that can pass through an intersection is determined. This design process involves assessing traffic movements and flow directions while considering intersection geometry to prevent potential conflicts. The number of phases may vary depending on intersection types and geometries.

Traffic signals generally have three signal durations: actual green time, yellow time, and all-red time. Actual green time allows vehicles with the right of way in a phase to proceed through the intersection.

The traffic signal cycle, also known simply as the cycle, comprises a sequence of signals in which all phases gain the right of way through the intersection. Once the cycle is completed, the starting phase regains the right of way. Cycle length refers to the time it takes to complete a cycle.

A lane group consists of one or more lanes, and this information is used for different purposes in traffic engineering, such as determining the level of service, queuing, capacity, and delay analysis [23]. Among the lane groups with green time in a phase, the one(s) with the highest flow ratio is called the critical lane group(s). Since noncritical lane groups have lower traffic demand, their demand is considered satisfied when the demand of the critical lane group is met. Thus, conditions for critical lane groups are specifically defined.

One critical parameter in effectively designing a traffic signal control system at an intersection is the saturation flow rate. The Highway Capacity Manual (HCM) defines the saturation flow rate as "the equivalent hourly rate at which previously queued vehicles can traverse an intersection approach under prevailing conditions, assuming the green signal is available at all times and no lost times are experienced, in vehicles per hour or vehicles per hour per lane" [23, p. 61].

The saturation flow rate at intersections is affected by several factors, making it nearly impossible to obtain an exact value. In many studies, a lane-based saturation flow rate of 1800 veh/h/lane has been assumed [24**?**]. HCM also recommends this value when vehicle speeds are below 50 km/h [23, p. 172]. Hence, in the present study, we assume a lane-based saturation flow rate of 1800 veh/h/lane.

Given a saturation flow rate (s), the effective green time (g) of a lane group, and the cycle length (C), the capacity (c) of the lane group can be calculated as $c = s * (g/C)$.

Another important parameter for understanding queue formation at signalized intersections and accurately measuring vehicle delays is the q/c (volume–capacity) ratio. Traffic congestion becomes significant as the hourly traffic volume (q) approaches the lane group's capacity. Therefore, determining the q/c ratio for each lane group at intersections is essential before calculating delays and queuing. The q/c ratio is computed by dividing the hourly traffic volume (q) by the lane group's capacity, yielding the degree of saturation (X_c) or the volume-to-capacity ratio: $X_c = q/c$.

If the degree of saturation of a lane group exceeds 1, assuming a uniform arrival pattern, queues form at every signal cycle, continuously increasing in size. Conversely, if the degree of saturation of a lane group is less than or equal to 1, the queue accumulated during the cycle is cleared, leaving no residual queue for the next cycle. These conditions are known as oversaturated and undersaturated conditions, respectively.

4. Linear programming-based solutions to the traffic signal control problem

The delay calculation formula in the HCM 2000 delay model (given in Appendix B) is nonlinear, as outlined by HCM [23]. Additionally, for intersections, optimizing the cycle length is very important in minimizing average control delay. The total cycle length should balance between not too short and not too long, as highlighted by Papageorgiou [25]. If the cycle length is too short, the effective green times become insufficient to clear the queue of waiting vehicles. Consequently, the proportion of total lost time in a phase over the total cycle length rises, potentially leading to an increase in average control delay, which might seem contradictory to expectations. Conversely, if the cycle length is too long, vehicles in the queue are forced to wait longer for the next green phase, which can also result in a higher average control delay at the intersection. Therefore, optimizing the total cycle length can also be a reasonable objective for an optimization program. However, it is essential to note that this objective introduces nonlinearity into the problem. Further, for oversaturated cases, regardless of the length of the cycle, there will always be a residual queue left after the effective green time. Thus, there is a large amount of research focused on optimizing the queue length for fixed-time traffic signal strategies [22]. For these reasons, our paper also focuses on optimizing queuing length using linear programming models.

4.1. Notations and problem specification

Table 1 lists the notations used throughout the paper.

Table 1. Nomenclature.

Notations	
P:	Set of signal phases indexed by p
N:	Number of phases
CG:	Set of critical lane groups
G^p:	Set of all allowable lane groups in the pth phase
n_i :	Number of lanes in the lane group i
ω_i :	Demand ratio of a lane group i
Ω :	Total demand ratio of an intersection
R :	Conversion coefficient
a_i :	Allocation ratio of a lane group i
λ_i :	Arrival rate per second for lane group i
θ_i :	Lane-based saturation low rate per second (veh/s/lane) for lane group i
C:	Total cycle length
L:	Total lost time per cycle
x^p :	Effective green time for the phase p (decision variable)
g^p_{min} :	Minimum effective green time for the phase p
g^p_{max} :	Maximum effective green time for the phase p

In the present study, three methods were proposed to handle oversaturated conditions in signalized isolated intersections. These methods, MTQLM, MMQLM, and NSM, which we will introduce shortly, are specifically intended for applications in which the traffic volume exceeds the capacity, indicating oversaturated conditions. On the other hand, if the intersection has a higher capacity than traffic, signifying undersaturated conditions, strategies to minimize the residual queue length are no longer needed. The last method, namely NSM, is designed to be applied as a follow-up phase of the previous two methods. To assess and compare these methods, we employed the HCM 2000 delay model to calculate delays corresponding to different green light settings.

The entire framework is presented in Figure 1. As depicted in the figure, it first gets the parameters of the intersection and scenario to optimize: lane-based saturation flow rates, hourly traffic volumes, phase design, lost time, cycle length, yellow time, and all red time. Then it determines the critical lane group(s) as the degree of the saturation of a critical lane group defines the saturated condition of an intersection. Thus, if **Xc** is less than or equal to 1, the algorithm directly stops as the intersection is found to be undersaturated. On the other hand, once we determine that the intersection and its associated traffic fall into the oversaturated category by having **Xc** greater than 1, the two-stage process is followed. Initially, we apply the MTQLM and MMQLM methods, and then we select the one yielding the superior delay performance. Subsequently, the chosen method's results are further enhanced through the application of the NSM method and used as the effective green times for the intersection.

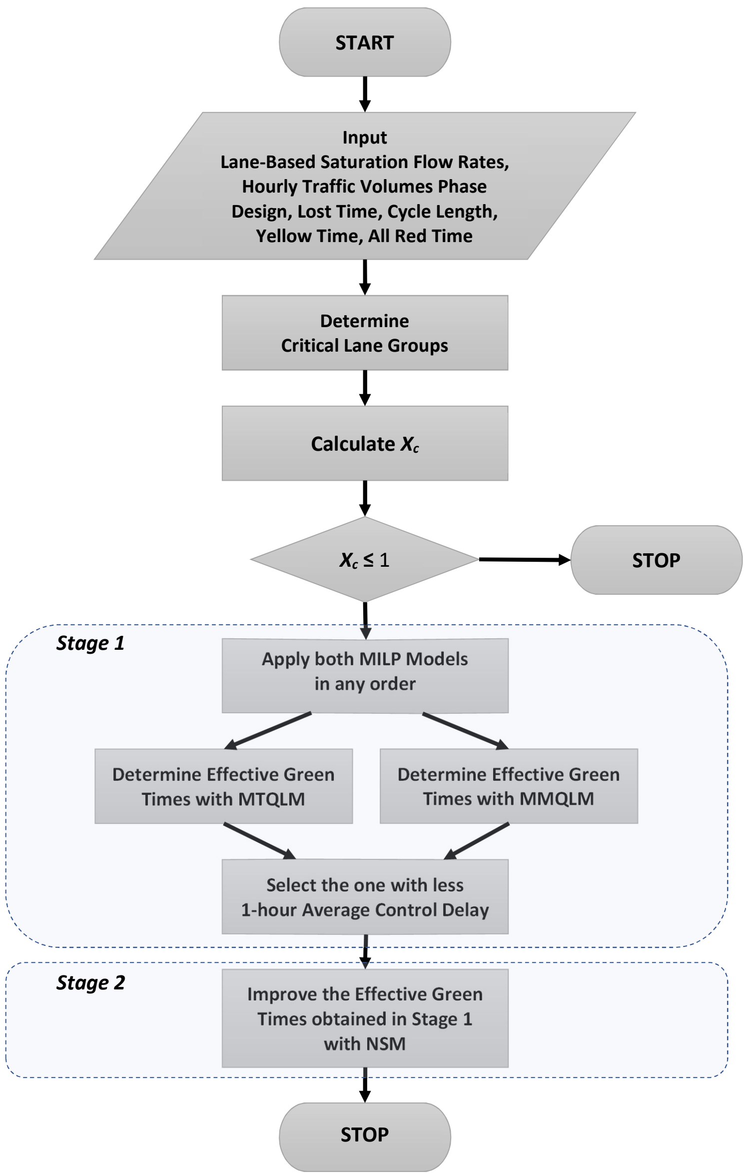

Figure 1. The flow diagram of our framework.

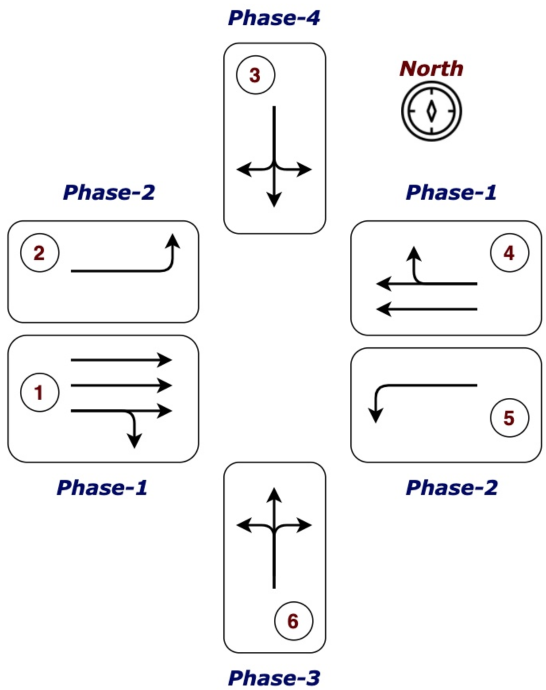

Figure 2. Intersection 1 phases.

To explain the linear programming-based approaches, we designed a four-legged intersection with four distinct phases, specifying its parameters as presented in Table 2. Given these parameters, with a cycle time (C) of 135 s and a total loss time for this cycle (L) calculated as $2\ (lost\ time\ per\ phase) \times 4\ (number\ of\ phases) + 1\ (all - red\ time\ per\ phase) \times 4\ (number\ of\ phases) = 12\ s$, the total assignable effective green time $(C - L)$ amounts to 123 s. In Figure 2, we provide an illustration of a sample intersection. To demonstrate the developed approaches, we will utilize scenario 1.1, as outlined in Table 2.

Table 2. Parameters (left) and scenario (right) of intersection 1.

Parameters	
Initial Cycle Length	**135 s**
Minimum Effective Green Time per Phase	**9 s**
Lost Time per Phase	**2 s**
Yellow Time	**3 s**
All Red Time per Phase	**1 s**
Number of Phases	**4**
Number of Lane Groups	**6**
Preassumed Range Value (δ) for NSM	**5**

	Left (veh/h)	Through (veh/h)	Right (veh/h)
West	300	1800	144
North	50	350	50
East	156	550	100
South	75	400	75

4.2. Solutions for oversaturated scenarios

We present two mixed-integer linear programming (MILP)-based solutions for addressing oversaturated traffic scenarios. The first approach is from [4] and focuses on minimizing the residual queue length after a traffic cycle is completed. However, this method, in certain cases, unfairly extends queues in some lane groups, resulting in increased delays. To overcome this issue, we adapted the approach to minimize the maximum queue length within lane groups following the cycle. Although this adjustment may lead to a slight increase in the total queue length, it often results in reduced overall delays for specific scenarios. Additionally, changing the distribution of green times to various phases can further reduce total delay. Therefore, we introduce a novel neighborhood search method to identify solutions with improved delay metrics. This approach utilizes solutions obtained from MILP methods and, instead of conducting an exhaustive search to determine optimal green times, explores the neighborhood of values derived from the MILP solutions, enhancing their quality. We demonstrate the high effectiveness of this approach, as it rapidly identifies either the best solution or a solution very close to the optimal one.

4.2.1. Minimize total queue length method (MTQLM)

MTQLM is a MILP approach designed to minimize the total remaining queue length after each cycle. By reducing the total remaining queue length, the approach simultaneously maximizes the number of vehicles that can pass through the intersection, ultimately boosting throughput.

This algorithm is proposed in the work of [4], where the proposed solution assumes for expected oversaturation of traffic over a defined time period. It aims to determine the optimal traffic light durations for each phase to maximize total throughput throughout the oversaturated period. In contrast, we operate on the assumption that we only have knowledge of the traffic volume for the upcoming time period, which is expected to be oversaturated, and our objective is to determine traffic light durations to minimize the queue length after a single cycle. Therefore, we modified their approach slightly for this version.

As mentioned previously, when Xc is greater than 1, specific critical lane groups will consistently have a residual queue after each cycle. This indicates that the effective green time is insufficient to clear these queues within the effective green period. Consequently, the residual queue continues to grow during the analysis period, leading to persistent oversaturation in these lane groups.

In this method, while MTQLM attempts to clear the queues for some lane groups, it may leave residual queues for others. Since MTQLM aims to minimize the total remaining queue length, it allocates the appropriate effective time to clear queues in different critical lane groups as soon as it finishes clearing a queue in any specific group. For each critical lane group i within the set of critical lane groups CG ($\boldsymbol{i} \in \boldsymbol{CG}$), and the corresponding phase p within the set of phases $\boldsymbol{p} \in \boldsymbol{P}$, the following constraint applies for oversaturated traffic volume (Total discharging $\leq$ Cumulative arriving): $\mathbf{x^p}\theta_\mathbf{i}\mathbf{n_i} \leq \lambda_\mathbf{i}\mathbf{C}$.

Another constraint in this method involves determining minimum effective green times for phases. A traffic signal setting must ensure pedestrian safety to allow pedestrians to traverse the intersection securely. Although the present study did not specifically address pedestrians at intersections, minimum effective green times were still determined for the phases.

For each phase p within the set of phases P, the constraints for minimum effective green times can be expressed as $x^p \geq g^p_{min}$. The final constraint in this approach guarantees that the total assignable effective green time is equal to the sum of the effective green times for each phase. The total assignable effective green times are also equal to the difference between the cycle length and the total lost time, represented as follows:

$$\sum_{p \in P} \mathbf{x^P} = \mathbf{C} - \mathbf{L} \tag{1}$$

$$
\begin{array}{lr}
\text{minimize} \sum_{\mathbf{p}\in\mathbf{P}} \sum_{\mathbf{i}\in\mathbf{G^p}} (\lambda_\mathbf{i}\,\mathbf{C} - \mathbf{n_i}\,\mathbf{x^p}\,\theta_\mathbf{i}) \quad (\mathit{Minimize\,Total\,Res.\,Queue}) & \\
\text{subject to} & \\
x^p\theta_i n_i \leq \lambda_i C \ , \ \forall i \in CG \qquad (\mathit{Discharge\,Constraint}) & (2)\\
x^p \geq g^p_{min} \ , \ \forall p \in P \qquad (\mathit{Min.\,Green\,Constraint}) & \\
\sum_{p\in P} x^p = C - L \ , \ \forall p \in P \qquad (\mathit{Cycle\,Constraint}) &
\end{array}
$$

In scenario 1.1, the critical lane groups identified are 1, 2, 3, and 6, while lane groups 4 and 5 are classified as noncritical in this context. Thus, MTQLM calculates the effective green times for each phase by considering these critical lane groups and their associated constraints. For this scenario, MTQLM determines effective green times of 48, 22, 20, and 33 s for phases 1 through 4, respectively.

Using the effective green times generated by MTQLM for scenario 1, after 30 cycles, the total remaining queue (residual queue) lengths were calculated as 364.5. A detailed breakdown of the residual queue and its percentages for each lane group is presented in Table 3.

In oversaturated conditions, the HCM 2000 delay analysis gives similar results for both 15-min and 1-h durations. Thus, we conducted delay analysis for a 1-h duration in these cases. Employing the HCM 2000 delay analysis for scenario 1.1 over a 1-h analysis period, we determined the intersection's average control delay as 134.30 s. Additionally, the average control delays for each lane group were calculated as 67.17, 115.00, 99.80, 35.65, 58.53, and 548.39 s, respectively, for lane groups 1 to 6.

Table 3. Residue queue results for scenario 1.1 determined by MTQLM (left) and by MMQLM (right).

Lane Groups	Total Arriving (veh)	Total Residue Queue (veh)	Residue Queue Percentage (%)
1	2187	27	1
2	337.5	7.5	2
3	506.25	11.4	2
4	731.25	0	0
5	175.5	0	0
6	618.75	318.6	51

Lane Groups	Total Arriving (veh)	Total Residue Queue (veh)	Residue Queue Percentage (%)
1	2187	342	15
2	337.5	52.5	15
3	506.25	86.4	17
4	731.25	0	0
5	175.50	0	0
6	618.75	93.6	15

Analyzing the residual queue results for each lane group in scenario 1.1 reveals that MTQLM leaves disproportionately long residual queues in certain lane groups. Notably, almost all of the total remaining queues accumulate in lane group 6. In this scenario, 50% of the vehicles arriving at lane group 6 remain in the queue and, as the analysis period progresses, these vehicles are forced to wait for one or more green signal cycles to pass through the intersection. This condition is mainly due to the remarkably high degree of saturation in lane group 6, which is calculated as 2.06.

Although MTQLM works well in optimizing the total residual queue, it exhibits certain limitations as it may result in exceptionally lengthy residual queues in specific lane groups. Consequently, this leads to persistent oversaturation in these lane groups as the analysis period extends, subsequently contributing to significantly higher average control delays in these lanes when compared to others that experience no residual queues after each cycle. In simple terms, MTQLM appears to penalize certain lane groups while minimizing the total remaining queue, thus causing an imbalance in the allocation of effective green times to different phases.

4.2.2. Minimize maximum queue length method (MMQLM)

MMQLM is an alternative MILP approach aimed at addressing the issue of fairness related to residual queue length and average control delay in the context of the MTQLM approach. Similar to MTQLM, MMQLM employs the same set of constraints for allocating effective green times to the phases. The key distinction between MMQLM and MTQLM is in their respective objective functions.

While MTQLM focuses on minimizing the overall residual queue length at the intersection, MMQLM's objective is to minimize the maximum remaining queue length in any lane group after each cycle. In other words, it aims to distribute the remaining total queue among critical lane groups following each cycle. To achieve a fair allocation of the remaining queue length to each critical lane group, this method employs the concept of the demand ratio, which differs slightly from the flow ratio. As mentioned previously, the flow ratio of a critical lane group is determined by dividing the total traffic volume by the total saturation flow rate. The total saturation flow rate is computed as the sum of the saturation flow rates for each lane within a lane group. In the present study, the saturation flow rate per lane was assumed to be 1800 vehicles per hour per lane. Consequently, the total saturation flow rate is equal to the product of the number of lanes in a critical lane group and the saturation flow rate for a single lane.

On the other hand, the demand ratio for a critical lane group is calculated by dividing the total traffic volume by the saturation flow rate per lane. The formula for calculating the demand ratio of a critical lane group is as follows: $\omega_\mathbf{i} = \mathbf{q_i}/\mathbf{s_i}$. Once the demand ratios for each critical lane group have been calculated, the total demand ratio is determined by summing these individual demand ratios. To obtain the total demand ratio for an intersection, the following formula is used:

$$\boldsymbol{\Omega} = \sum_{\mathbf{i}\in\mathbf{CG}} \omega_\mathbf{i} \tag{3}$$

Next, the allocation ratios for each critical lane group are determined by dividing each individual demand ratio by the total demand ratio of the intersection, represented as $\mathbf{a_i} = \omega_\mathbf{i}/\boldsymbol{\Omega}$.

Unlike the MTQLM approach, MMQLM aims to minimize the maximum possible residual queue in any critical lane group by adjusting its allocation ratio in the residual queue formula. Consequently, when the saturation flow rate per lane is the same for all lanes, each lane within a critical lane group will have a residual queue after each cycle with approximately the same percentage. The MILP formulation is illustrated in the following equation:

$$\begin{array}{cc} \text{minimize } \boldsymbol{max_{i\,\in\,CG}}((\lambda_i\, C - n_i\, x^p\, \theta_i)/a_i) & (Min.\, Max\, Res.\, Queue) \\ \text{subject to} & \\ x^p\theta_i n_i \leq \lambda_i C \ \ , \ \ \forall i \in CG & (Discharge\, Constraint) \\ x^p \geq g^p_{min} \ \ , \ \ \forall p \in P & (Min.\, Green\, Constraint) \\ \sum_{p\in P} x^p = C - L \ \ , \ \ \forall p \in P & (Cycle\, Constraint) \end{array} \tag{4}$$

In scenario 1.1, MMQLM computes the effective green times for each phase as 41, 19, 35, and 28 s, respectively. Using the effective green times provided by MMQLM, we determined a residual queue analysis for scenario 1.1 to determine the total queue length after 30 cycles, which has been found as 574.5. A detailed breakdown of the residual queues and their percentages for each lane group can be found in Table 3.

By employing HCM 2000 delay analysis over a 1-h period, the intersection's average control delay is determined to be 127.09 s for scenario 1.1. Additionally, the average control delays for each lane group are determined as 136.93, 173.64, 168.03, 42.32, 65.29, and 150.68 s for each lane group. In scenario 1.1, MMQLM generates a total residual queue after 30 cycles of 574.5, which corresponds to a 57.6% increase compared to MTQLM. However, it is worth noting that MMQLM allocates residual queues to all critical lane groups with nearly equal percentages. This suggests that the MMQLM approach provides a fairer treatment of critical lane groups than MTQLM. Furthermore, for scenario 1.1, the MMQLM approach reduces the average vehicle delay at the intersection by 5.4% compared to MTQLM, resulting in an average delay of 127.09 s per vehicle. Consequently, it is safe to conclude that, for scenario 1.1, selecting the MMQLM approach is more suitable in terms of minimizing the average vehicle delay at the intersection and fairly distributing residual queues.

Finally, it is important to note that minimizing the total residual queue at an intersection does not always correspond to a reduction in the average control delay in certain traffic scenarios. Better delay outcomes are typically achieved by setting values closer to those that minimize queue lengths. Therefore, we propose a simple heuristic to search for improved settings that can enhance delay performance.

4.2.3. Neighborhood search method (NSM)

The neighborhood search method (NSM) corresponds to the second stage of our framework for oversaturated cases, as depicted in Figure 1. In some oversaturated traffic scenarios, the choice between the MTQLM and MMQLM approaches depends on which one yields better results concerning the HCM 2000 delay model. Consequently, for a given traffic scenario, these two methods are initially executed in the first stage and compared based on their average control delay.

During the second stage of the workflow, NSM is employed to search for more optimal effective green times with a focus on reducing delay. NSM utilizes the effective green times obtained in the first stage as a starting point to enhance them. This search for improvements is limited to a predefined range, which extends from $-\delta$ to $+\delta$ around the best solution from the first stage. The overall flow, incorporating NSM, follows these steps:

1. Obtain effective green times from the first stage using both MTQLM and MMQLM.

2. Determine the minimum and maximum limits for each effective green time using a predefined range constant (δ).

3. Within the specified limits for each phase, search for an improved green time that reduces the average control delay based on the HCM 2000 delay formula.

It is important to note that NSM does not guarantee the discovery of the global optimum average control delay within the predefined range at all times.

In contrast, a simple exhaustive search (brute search) algorithm can potentially identify the global minimum delay by evaluating all possible combinations of green time settings. However, as the number of phases and the total cycle length increase for an intersection, the exhaustive search algorithm may require an extensive amount of time to complete.

For scenario 1.1, NSM enhances the effective green times by exploring within a range of δ (set to 5 s) around the effective green times obtained in the first stage. This results in new effective green times of 46, 18, 33, and 26 s for phases 1 to 4, with an average control delay of 110.74 s. This represents a 5.18% improvement compared to the delay obtained by MMQLM, which was 127.09 s.

The number of iterations for NSM can be expressed as $(2\delta)^N$, while the number of iterations for the exhaustive search algorithm can be expressed as $(g^p_{max})^N$.

For this intersection, g^p_{min} is determined as 9 s, which sets the minimum effective green time for a phase. The maximum effective green times of a phase are computed as $g^p_{max} = C - L - (N-1) * g^p_{min}$.

As this equation illustrates, the number of iterations for the exhaustive search algorithm increases as the cycle length and the number of phases increase. On the other hand, NSM has only a small increase in iterations, dependent solely on the number of phases.

For the present study, both NSM and the exhaustive search algorithm were executed 10 times, and the average execution time was computed to compare the efficiency of these two algorithms. In the case of an intersection with 4 phases, a minimum effective green time of 9 s, and a range value δ of 5 s, the mean runtime for the NSM approach was calculated at 0.13 s, whereas it was 35.34 s for the exhaustive search. NSM was set to perform 10,000 iterations for this experiment, while the exhaustive search executed more than 84 million iterations. For this case the NSM algorithm is approximately 270 times faster than the exhaustive search algorithm, with the number of iterations reflecting their respective execution times.

As mentioned earlier, NSM does not guarantee the discovery of the global minimum delay every time. When NSM and the exhaustive search algorithms were tested using scenario 1.1, the average control delays for the intersection were determined as 110.74 s with the NSM method versus 107.53 s with the exhaustive search method. Although the difference is minimal, NSM's execution time is significantly faster than that of the exhaustive algorithm.

5. Experiments

Intersection 1 with four legs was introduced and examined in the previous section, while the methods MTQLM, MMQLM, and NSM were explained and applied to this sample intersection under a single scenario. This section will analyze two more intersections with different characteristics using the same three methods: intersection 2 with three legs and intersection 3 with four legs but with more lanes than the first. Here, these two intersections are tested under 4 and 5 scenarios, respectively. Moreover, the last one was studied with two phase design alternatives to show their effect on the saturation condition.

All these 45 experiments (3 MILP methods applied to 2 intersections under a total of 5 scenarios plus 1 intersection under 5 scenarios with 2 phase designs) were implemented using Pulp, an open-source Python library for modeling and solving linear programming problems. It provides a high-level interface for defining linear programs in which you need to define decision variables you want to optimize, an objective function to minimize or maximize, and constraints on these decision variables. Its default solver, CBC or COIN-OR Branch and Cut, was used during these experiments. It is a general-purpose, open-source solver and is available for various platforms. Appendix A exemplifies the implementation details by applying MTQLM to intersection 1 under scenario 1.1.

5.1. Analyzing intersection 2

Intersection 2, illustrated in Figure 3, is designed as a T-shaped intersection and has a total of 11 lanes. These lanes form 6 distinct lane groups, depending on the various movements within the intersection. Figure 3 shows that the phase design includes several instances of repeated lane groups. For instance, lane group 9 is granted the right of way in more than one phase. In contrast to the first intersection, all movements at this location are separated from each other, and there are additional lanes allocated for each movement. To assess the linear programming approaches introduced above, 4 distinct traffic scenarios, labeled 2.1 to 2.4, were generated for this particular intersection. The parameters and hourly traffic volumes for each lane group in every scenario for intersection 2 are given in Table 4.

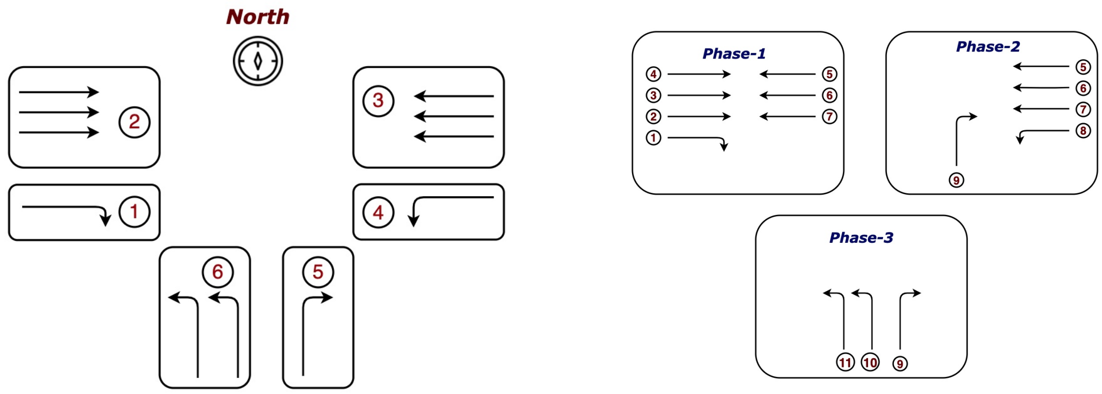

Figure 3. Lane groups (left) and phase design (right) of intersection 2.

Table 4. Parameters (left) and scenarios (right) of intersection 2.

Parameters	
Initial Cycle Length	**90 s**
Minimum Effective Green Time per Phase	**8 s**
Lost Time per Phase	**2 s**
Yellow Time	**3 s**
All Red Time	**1 s**
Number of Phases	**3**
Preassumed Range Value (δ) for NSM	**5**

	Lane Groups					
Scenario No	**1 (w-r)**	**2 (w-s)**	**3 (e-s)**	**4 (e-l)**	**5 (s-r)**	**6 (s-l)**
2.1	150	1872	990	550	110	990
2.2	180	1872	1080	600	120	1080
2.3	165	2028	990	600	110	1170
2.4	165	2028	990	700	170	1125

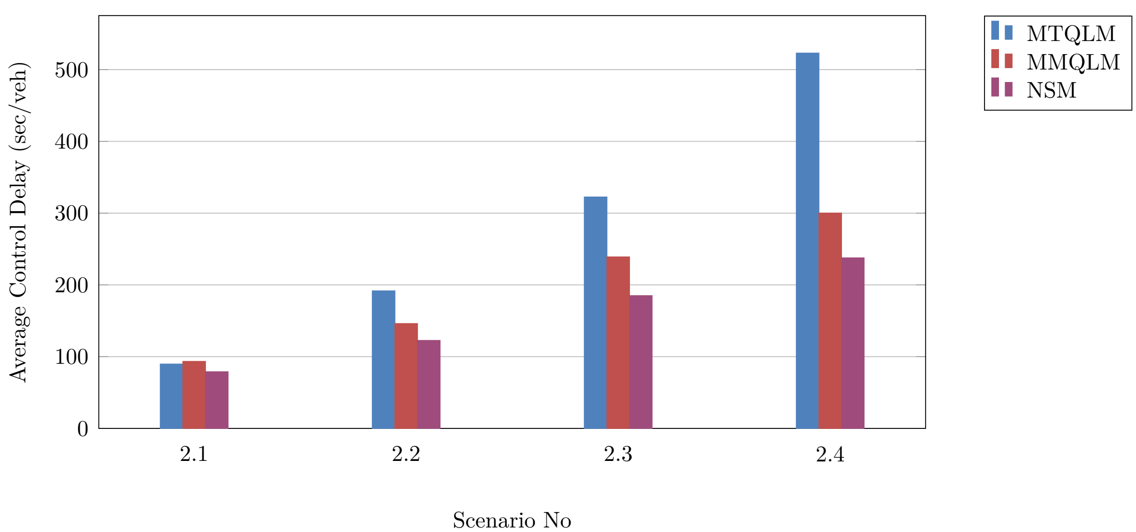

Figure 4. Delay comparison for intersection 2.

In this intersection, all scenarios produce critical lane groups with a degree of saturation values exceeding 1, resulting in oversaturated conditions. Delays are then compared between MTQLM, MMQLM, and NSM. The 1-h HCM 2000 average control delay results for these scenarios are depicted in the bar plot in Figure 4. In particular, NSM consistently produces significantly superior delay results in all scenarios, outperforming the other two methods. Additionally, MTQLM consistently suffers higher delays than MMQLM and NSM, particularly as the total hourly traffic volume of the intersection rises. This can be attributed to an increase in the unbalanced allocation of residual queues in MTQLM as traffic volumes at the intersection increase.

Next, the delay comparison results for the exhaustive search algorithm and NSM are presented in Table 5. According to these results, NSM managed to identify the global minimum delay in scenarios 2.1, 2.2, and 2.3, but not in scenario 2.4. Finally, the effective green times and delay found for each scenario are summarized in Table 6.

Table 5. Delay comparison between NSM and exhaustive search for intersection 2.

Scenario No	***1-Hour Delay (s/veh) NSM***	***1-Hour Delay (s/veh) Exhaustive Search***
2.1	***78.88***	***78.88***
2.2	***122.55***	***122.55***
2.3	***184.97***	***184.97***
2.4	***237.62***	***235.29***

Table 6. Effective green times and delays determined for intersection 2.

Scenario No	**Phase-1**	**Phase-2**	**Phase-3**	**1-Hour Delay (s/veh)**
2.1	**32**	**24**	**25**	**78.88**
2.2	**31**	**24**	**26**	**122.55**
2.3	**33**	**21**	**27**	**184.97**
2.4	**33**	**24**	**24**	**237.62**

5.2. Analyzing intersection 3

Intersection 3 is a four-legged intersection, as depicted in Figure 5. It has additional lanes for all left, right, and straight movements. In particular, all right turns are granted the right of way in two consecutive phases. In total, intersection 3 contains 14 lanes, which collectively form 12 distinct lane groups corresponding to various movements.

Unlike intersection 2, two distinct phase designs were created for this intersection to illustrate the impact of phase design on the intersection's average control delay. These phase designs, labeled phase design 1 and phase design 2, along with the movements associated with each phase, are detailed in Figure 6. It is important to note that even under the same traffic scenario, one phase design may lead to traffic conditions that are undersaturated, while another may result in oversaturation, as exemplified below.

For this intersection, five different traffic scenarios, denoted as 3.1 to 3.5, were generated. The parameters and hourly traffic volumes for each lane group in each scenario of intersection 3 are provided in Table 7.

For each scenario at this intersection, the delays experienced by each method were thoroughly examined and compared. Subsequently, the effective green times and delays identified for phase design 1 & 2 are presented in Table 8. Here "n/a" in the Delay column indicates that the corresponding scenario and phase design produce an unsaturated condition and, hence, the models were not applied.

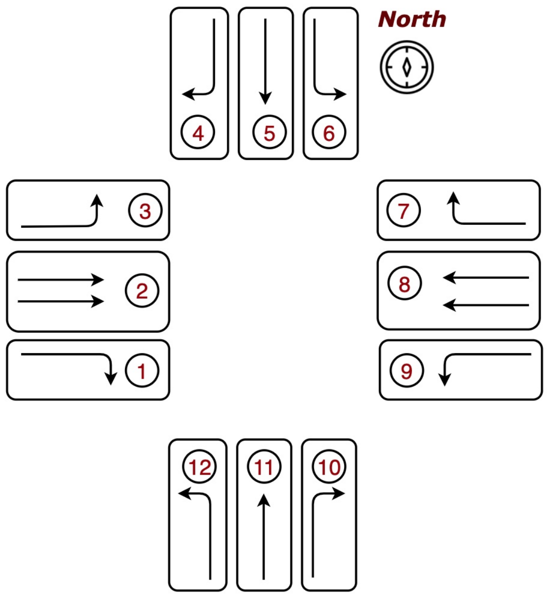

Figure 5. Intersection 3 - lane groups.

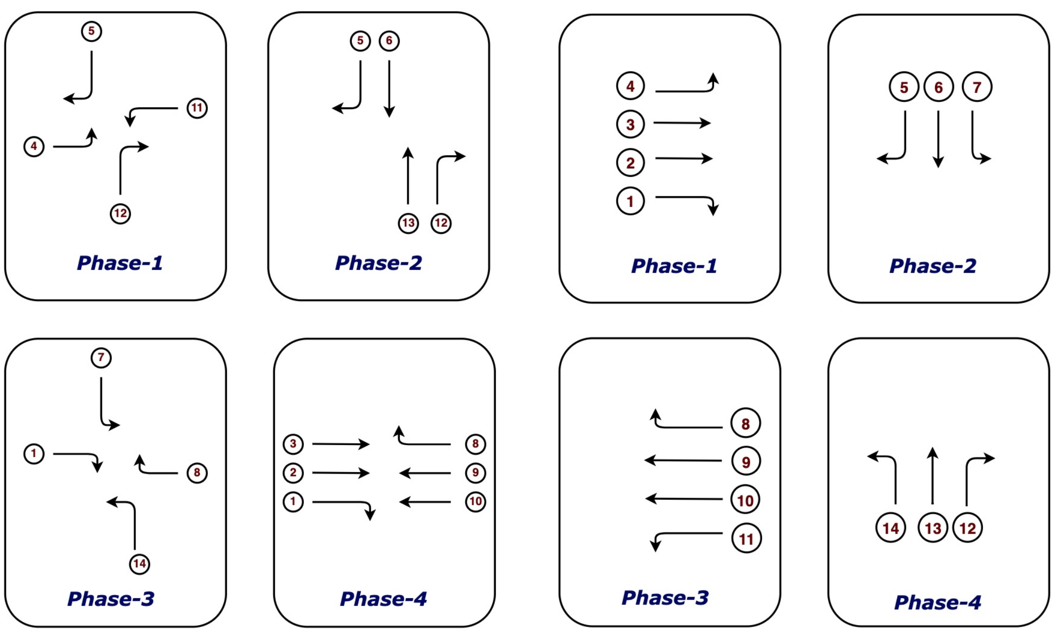

Figure 6. Intersection 3 - phase design-1 (left) and phase design-2 (right).

Table 7. Parameters (left) and scenarios (right) of intersection 3.

Parameters	
Initial Cycle Length	**135 s**
Minimum Effective Green Time Per Phase	**9 s**
Lost Time Per Phase	**2 s**
Yellow Time	**3 s**
All Red Time	**1 s**
Number of Phases	**4**
Preassumed Range Value (δ) for NSM	**5**

	Lane Groups											
Scenario No	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
3.1	110	550	165	165	495	275	138	1040	385	110	165	165
3.2	110	550	165	165	585	288	138	1040	420	110	165	165
3.3	130	650	195	195	585	325	163	1040	455	130	195	195
3.4	130	650	195	195	585	325	163	1040	455	130	300	195
3.5	130	720	195	195	585	325	163	1040	455	130	300	195

Table 8. Effective green times and delays determined for phase design 1 (left) and 2 (right) of intersection 3.

Scenario No	Phase-1	Phase-2	Phase-3	Phase-4	1-Hour Delay (s/veh)
3.1	**28**	**35**	**40**	**20**	**99.37**
3.2	**27**	**38**	**39**	**19**	**163.14**
3.3	**28**	**37**	**38**	**20**	**193.96**
3.4	**28**	**37**	**38**	**20**	**190.46**
3.5	**28**	**37**	**38**	**20**	**190.72**

Scenario No	Phase-1	Phase-2	Phase-3	Phase-4	1-Hour Delay (s/veh)
3.1	**24**	**41**	**44**	**14**	**n/a**
3.2	**22**	**45**	**42**	**14**	**n/a**
3.3	**25**	**44**	**39**	**15**	**n/a**
3.4	**24**	**40**	**39**	**20**	**125.90**
3.5	**27**	**38**	**39**	**19**	**142.36**

As depicted in Figure 7, the delays attributed to phase design 1 and phase design 2 can exhibit significant differences, as apparent in scenarios 3.4 and 3.5. Moreover, phase design can have the effect of changing the traffic conditions to undersaturated. Specifically, for this intersection, selecting phase design 2 results in scenarios 3.1, 3.2, and 3.3 being categorized as undersaturated, consequently leading to unreported delay values.

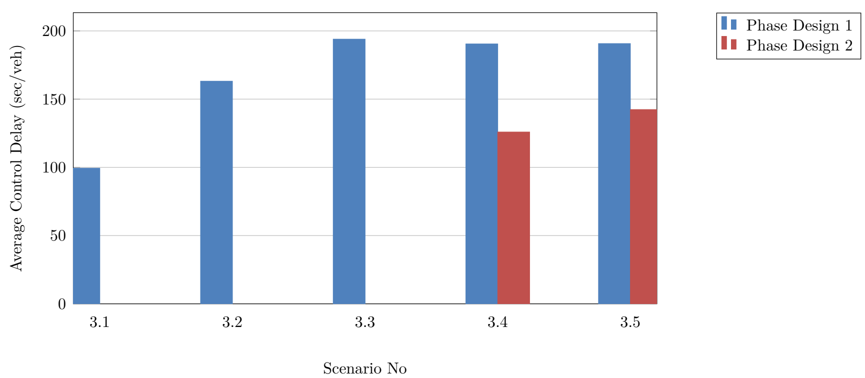

Figure 7. Delay comparison between phase design 1 and 2 of intersection 3.

5.3. Delay comparison with PTV VISSIM

PTV VISSIM is behavior-based and commercial microscopic simulation software that analyzes and optimizes traffic flows and tests real-life models. It provides detailed reports (vehicle delay, fuel consumption, average queue length, etc.) by analyzing models and traffic flows and can visualize models in detail [26].

In this section, intersection 3, previously evaluated with the HCM 2000 delay calculation, will be tested in the PTV VISSIM simulation environment. The simulation aims to validate our framework by obtaining delay results comparable to those obtained by the analytical method HCM 2000.

PTV VISSIM has a stochastic nature and uses the Poisson distribution when vehicles enter the network. Therefore, it can be predicted that there will be a certain difference between the average vehicle delay measured in PTV VISSIM and the delay obtained analytically with HCM 2000. Furthermore, saturation flow rates cannot be directly entered as a value for simulation in PTV VISSIM. Instead, they are determined by driver behavior and car-following logic. In the present study, we attempted to approach the predetermined lane-based saturation flow rate of 1800 veh/h/lane by adjusting driver behavior parameters. However, it should be noted that it is nearly impossible to achieve an exact saturation flow rate of 1800 veh/h/lane in the simulation.

In order to simulate intersection 3 and its scenarios, the lane group, their signal heads, and a four-phase signal controller were created, and all the necessary parameters were supplied. The 3D view of this simulation setup can be seen in Figure 8. This PTV VISSIM simulation was conducted 10 times for each scenario, with different random seed values, and each time for an hour. The results of the 1-h control delay of HCM 2000 and the average vehicle delay of PTV VISSIM can be found in Table 9. Although the analytical and simulation delays are not the same, this is a predicted result, as explained above. However, the delay variances are minor except for scenario 3.2. Based on these findings, it can be concluded that the simulations confirm our analytical results to some extent.

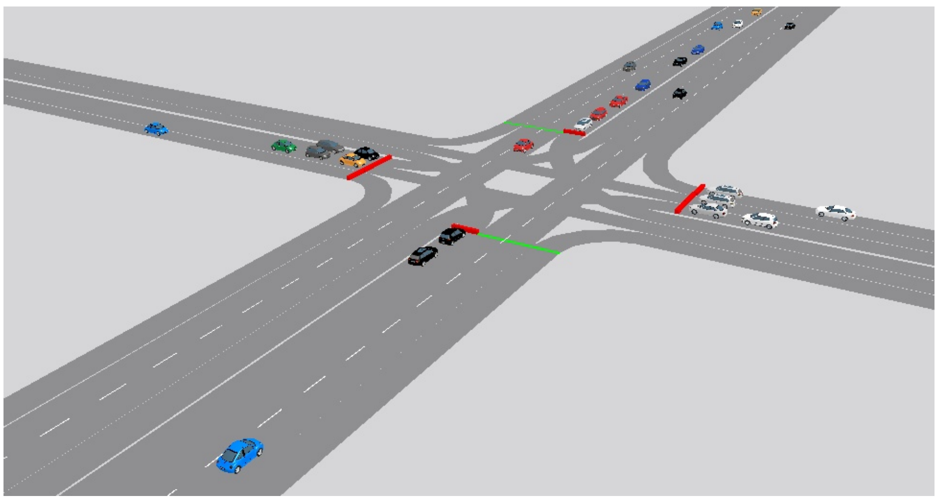

Figure 8. 3D view of intersection 3 simulation in PTV VISSIM.

Table 9. HCM 2000 delay vs. VISSIM delay for each scenario of intersection 3.

Scenario No	1-Hour HCM 2000 Average Vehicle Delay (s/veh)	1-Hour PTV VISSIM Average Vehicle Delay (s/veh)	Delay Difference Percentage (%)
1	61.69	64.54	+%4.6
2	71.44	78.53	+%9.9
3	92.54	95.26	+%2.9
4	125.90	129.12	+%2.6
5	142.36	141.96	-%0.3

5.4. Discussion of experiment results

In our experiments, we designed various signalized intersection configurations and different phase designs to demonstrate the flexibility of our framework, regardless of the intersection type. Furthermore, we tested our approach across a range of oversaturated scenarios. Our observations revealed that, in most instances, our optimization approach, specifically aimed at minimizing the maximum queue length, outperformed a simplistic approach of total queue length minimization.

While MILP models are oriented towards optimizing queue lengths, the actual cost to vehicles is the delay suffered while waiting at the intersection. Consequently, there is potential for further improvement by refining signal timings. To address this, we applied our novel neighborhood search heuristic to the solutions derived from MILP. Notably, we found that in almost all cases these proposed solutions could be further improved with this more cost-effective heuristic.

6. Conclusions

Isolated signalized intersections are very common on roads. Designing roads to accommodate the demands of rush hour traffic is often impractical. Consequently, when the traffic demand exceeds the capacity of these intersections, oversaturation becomes inevitable. This study presents a framework that includes various objective functions for addressing the traffic signal control problem through the application of linear programming optimization techniques. Our approach is specifically designed for determining signal settings for oversaturated conditions for isolated signalized intersections.

We employ a fixed-time signal control strategy with the objective of optimizing traffic control at isolated signalized intersections. While traffic optimization is often framed as the maximization of traffic flow through the intersection, we choose to model our problem in terms of the residual queue left after each traffic cycle. Since, for individual vehicles, the actual impact of any traffic issue is the delay experienced by the vehicle, the quality of our solution is best assessed by considering the average delay that a vehicle experiences when passing through the intersection. This is why we utilize the HCM 2000 delay model to evaluate the MILP models that we explored.

In oversaturation situations, the MTQLM approach, aimed at minimizing overall queue lengths, can lead to uneven queues in different lane groups. This problem becomes more obvious, particularly in high-traffic scenarios. As a response, a new method, MMQLM, has been developed. The aim of MMQLM is to achieve a fairer treatment of lane groups by considering vehicle arrival rates when distributing the remaining queue length.

Furthermore, it has been observed that the NSM approach, employed to further reduce average vehicle delays at intersections using the green times derived from the initial stage, yields even lower delay results for scenarios characterized by oversaturated conditions. Especially since NSM operates as a neighborhood search heuristic, it works with shorter execution times in comparison to exhaustive search methods. Consequently, this two-step approach proposed for addressing oversaturated conditions can serve as a feasible alternative to other linear programming models featured in the literature.

Our experiments with various intersection architectures show that our approach is quite effective. We also verified the models through a simulation tool.

References

[1] Singh L, Tripathi S, Arora H. Time optimization for traffic signal control using genetic algorithm. International Journal of Recent Trends in Engineering 2009; 2 (2): 4.

[2] Weisbrod G, Vary D, Treyz G. Economic Implications of Congestion. Technical report, NCHRP REPORT 463, National Cooperative Highway Research Program, Transportation Research Board, 2001.

[3] Srisurin P, Singh A. Optimal signal plan for minimizing queue lengths at a congested intersection. International Journal of Traffic and Transport Engineering 2017; 6 (3): 53-63.

[4] Liu H, Balke KN, Lin WH. A reverse causal-effect modeling approach for signal control of an oversaturated intersection. Transportation Research Part C: Emerging Technology 2008; 16 (6): 742-754.

[5] Srinivasan D, Choy MC, Cheu RL. Neural networks for real-time traffic signal control. IEEE Transactions on Intelligent Transportation Systems 2006; 7 (3): 261-272.

[6] Castro GB, Hirakawa AR, Martini JSC. Adaptive traffic signal control based on bio-neural network. Procedia Computer Science 2017; 109: 1182-1187.

[7] Trabia MB, Kaseko MS, Ande M. A two-stage fuzzy logic controller for traffic signals. Transportation Research Part C: Emerging Technology 1999; 7 (6): 353-367.

[8] Chiu S. Adaptive traffic signal control using fuzzy logic. In: Proceedings of the IEEE Intelligent Vehicles Symposium; Westlake Village, CA, USA; 1992. pp. 98-1007.

[9] Zeng J, Hu J, Zhang Y. Adaptive traffic signal control with deep recurrent Q-learning. In: IEEE Intelligent Vehicles Symposium (IV), Proceedings; Changshu, China; 2018. pp. 1215-1220.

[10] Liang X, Du X, Wang G, Han Z. Deep reinforcement learning for traffic light control in vehicular networks. IEEE Transactions on Vehicular Technology 2018; 1-11.

[11] Shoufeng L, Ximin L, Shiqiang D. Q-learning for adaptive traffic signal control based on delay minimization strategy. In: IEEE International Conference on Networking, Sensing and Control; Sanya, China; 2008. pp. 687-691.

[12] Van der Pol E, Oliehoek FA. Coordinated deep reinforcement learners for traffic light control. In: 30th Conference on Neural Information Processing Systems; Barcelona, Spain; 2016.

[13] Xiao N, Yu L, Yu J, Chen P, Liu Y. A cold-start-free reinforcement learning approach for traffic signal control. Journal of Intelligent Transportation Systems 2021; 26 (4): 1-10.

[14] Park B, Messer CJ, Urbanik T II. Traffic signal optimization program for oversaturated conditions: genetic algorithm approach. Transportation Research Record 1999; 1683 (1): 133-142.

[15] Chin YK, Yong KC, Bolong N, Yang SS, Teo KTK. Multiple intersections traffic signal timing optimization with genetic algorithm. In: Proceedings of International Conference on Control System, Computing and Engineering, ICCSCE 2011; Penang, Malaysia; 2011. pp. 454-459.

[16] Stevanovic A, Martin PT, Stevanovic J. VisSim-based genetic algorithm optimization of signal timings. Transportation Research Record 2007; 2035 (1): 59-68.

[17] Lee J, Abdulhai B, Shalaby A, Chung EH. Real-time optimization for adaptive traffic signal control using genetic algorithms. Journal of Intelligent Transportation Systems 2005; 9 (3): 111-122.

[18] Coll P, Factorovich P, Loiseau I, Gomez R. A linear programming approach for adaptive synchronization of traffic signals. International Transactions in Operational Research 2013; 20 (5): 667-679.

[19] Wang H, Peng X. Coordinated control model for oversaturated arterial intersections. IEEE Transactions on Intelligent Transportation Systems 2022; 23 (12): 24157-24175.

[20] Zhang C, Li JY, Chen CH, Li Y, Zhan ZH. Region-based evaluation particle swarm optimization with dual solution libraries for real-time traffic signal timing optimization. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '23; 2023. New York, NY, USA: Association for Computing Machinery. p. 111-118.

[21] Li S, Kang L, Huang H, Liu L. A perimeter control model of urban road network based on cooperative-noncooperative two-stage game. Physica A: Statistical Mechanics and its Applications 2023; 626: 129081.

[22] Eom M, Kim BI. The traffic signal control problem for intersections: a review. European Transport Research Review 2020; 12 (1): 50.

[23] HCM 2000 Highway Capacity Manual. Technical Report. Washington DC, USA: Transport Research Board, 2000.

[24] Shao CQ, Rong J, Liu XM. Study on the saturation flow rate and its influence factors at signalized intersections in China. Procedia - Social and Behavioral Sciences 2011; 16: 504-514.

[25] Papageorgiou M, Diakaki C, Dinopoulou V, Kotsialos A, Wang Y. Review of road traffic control strategies. Proceedings of the IEEE 2003; 91 (12): 2043-2065.

[26] Fellendorf M, Vortisch P. Microscopic traffic flow simulator VISSIM. In: Barceló J (editor). Fundamentals of Traffic Simulation. International Series in Operations Research & Management Science, vol 145. New York, NY, USA: Springer, 2010, pp. 63-93.

Appendix A: Applying MTQLM to scenario 1.1

To apply MTQLM to intersection 1 under scenario 1.1, we need to calculate the parameters in the method formulation.

First, based on Figure 2 and Table 3, the per-second arrival rate for each lane group is calculated as follows, abbreviating Arrival Rate to AR:

$\lambda_1 = AR_{\text{West-to-Through}} + AR_{\text{West-to-Right}} = 1800 + 144 \text{ (per h)} = 0.54 \text{ (per s)}$
$\lambda_2 = AR_{\text{West-to-Left}} = 300 \text{ (per h)} = 0.083 \text{ (per s)}$
$\lambda_3 = AR_{\text{North-to-Right}} + AR_{\text{North-to-Through}} + AR_{\text{North-to-Left}} = 50 + 350 + 50 \text{ (per h)} = 0.125 \text{ (per s)}$
$\lambda_4 = AR_{\text{East-to-Right}} + AR_{\text{East-to-Through}} = 100 + 550 \text{ (per h)} = 0.18 \text{ (per s)}$
$\lambda_5 = AR_{\text{East-to-Left}} = 156 \text{ (per h)} = 0.043 \text{ (per s)}$
$\lambda_6 = AR_{\text{South-to-Right}} + AR_{\text{South-to-Through}} + AR_{\text{South-to-Left}} = 75 + 400 + 75 \text{ (per h)} = 0.152 \text{ (per s)}$

The number of lanes in each lane group is apparent in Figure 2:

$n_1 = 3, n_2 = 1, n_3 = 1, n_4 = 2, n_5 = 1, n_6 = 1$

As mentioned in Section 2, the saturation flow rate is assumed to be 1800 veh/h/lane; hence:

$\theta = 1800 \text{ veh/h} = 0.5 \text{ veh/s for each lane}$

Moreover, Section 3.1 and Table 2 contain the following parameters:

Total cycle length, $C = 135$ s
Total lost time per cycle, $L = 12$ s
Minimum effective green time per phase, $g_{\min} = 9$ s

Lastly, the decision variables can be expressed as below:

X_p : Effective green time for the phase p

Based on the parameters explained above, the MTQLM code in Python and its output are given in Figure 9 and Figure 10, respectively.

```
# Apply MTQLM to Scenario 1.1 of Intersection 1
import pulp
# per-second arrival rate for each lane group
lambda1 = 0.54    # lane group 1 in phase 1
lambda2 = 0.083   # lane group 2 in phase 2
lambda3 = 0.125   # lane group 3 in phase 4
lambda4 = 0.18    # lane group 4 in phase 1
lambda5 = 0.043   # lane group 5 in phase 2
lambda6 = 0.152   # lane group 6 in phase 3
# number of lanes in each lane group
n1 = 3
n2 = n3 = n5 = n6 = 1
n4 = 2
# other parameters
C = 135      # total cycle length
L = 12       # total lost time per cycle
theta = 0.5  # saturation flow rate for each lane
g_min = 9    # minimum effective green time for each phase
# minimization problem to solve with default solver
problem = pulp.LpProblem("Traffic_Light_Optimization", pulp.LpMinimize)
# decision variables
x1 = pulp.LpVariable("Phase1", cat='Integer')
x2 = pulp.LpVariable("Phase2", cat='Integer')
x3 = pulp.LpVariable("Phase3", cat='Integer')
x4 = pulp.LpVariable("Phase4", cat='Integer')
# objective: to minimize total residual queue length
problem += ((lambda1 * C) - (theta * n1 * x1))  # lane group 1 in phase 1
        + ((lambda2 * C) - (theta * n2 * x2))  # lane group 2 in phase 2
        + ((lambda3 * C) - (theta * n3 * x4))  # lane group 3 in phase 4
        + ((lambda4 * C) - (theta * n4 * x1))  # lane group 4 in phase 1
        + ((lambda5 * C) - (theta * n5 * x2))  # lane group 5 in phase 2
        + ((lambda6 * C) - (theta * n6 * x3))  # lane group 6 in phase 3
# cycle constraint
problem += x1 + x2 + x3 + x4 == C - L
# minimum green time constraint
problem += x1 >= g_min
problem += x2 >= g_min
problem += x3 >= g_min
problem += x4 >= g_min
# discharge constraint
problem += x1 * n1 * theta <= C * lambda1      # lane group 1 in phase 1
problem += x2 * n2 * theta <= C * lambda2      # lane group 2 in phase 2
problem += x3 * n6 * theta <= C * lambda6      # lane group 6 in phase 3
problem += x4 * n3 * theta <= C * lambda3      # lane group 3 in phase 4
# solve the problem
problem.solve()
# print the results
print(pulp.LpStatus[problem.status])
print("Phase1: " + str(pulp.value(x1)) + " Phase2: " + str(pulp.value(x2))
   + " Phase3: " + str(pulp.value(x3)) + " Phase4: " + str(pulp.value(x4)))
```

Figure 9. MTQLM code in Python for scenario 1.1.

```
Result - Optimal solution found

Objective value:                -168.50000000
Enumerated nodes:               0
Total iterations:               0
Time (CPU seconds):             0.00
Time (Wallclock seconds):       0.01

Option for printingOptions changed from normal to all
Total time (CPU seconds):       0.01   (Wallclock seconds):       0.01

Optimal
Phase1: 48.0 Phase2: 22.0 Phase3: 20.0 Phase4: 33.0
```

Figure 10. Output of the MTQLM code for scenario 1.1.

Appendix B: HCM 2000 analytical delay model

According to the HCM 2000 delay model [23, p. 317], the average control delay in a lane group can be calculated by the following equations:

$$\begin{aligned}
\mathbf{d} &= \mathbf{d_1(PF) + d_2 + d_3} \\
\mathbf{d_1} &= \frac{\mathbf{0.5C(1-\frac{g}{C})^2}}{\mathbf{1-[min(1,X)\frac{g}{C}]}} \\
\mathbf{d_2} &= \mathbf{900T[\ (X-1) + \sqrt{(X-1)^2 + \frac{8kIX}{cT}}\ ]}
\end{aligned} \tag{5}$$

Here
d = control delay (s/veh)
d_1 = uniform delay (s/veh)
d_2 = incremental delay (s/veh)
d_3 = initial queue delay (s/veh) (is chosen as 0 assuming no initial queue)
PF = progression adjustment factor (is chosen as 1 for isolated intersections)
X = volume-to-capacity ratio (q/c or q/c) or degree of saturation for lane group
C = cycle length (s)
c = capacity of the lane group (s)
g = effective green time for the lane group (s)
T = duration of analysis period (h)
k = incremental delay adjustment for actuated control (is chosen as 0.5 for fixed-time signal designs)
I = incremental delay adjustment for filtering and metering by upstream signals (1 for isolated intersections)

We studied **isolated signalized intersections**, which are at least 1.6 km from any other upstream signalized intersections [23]. For this reason, I, the upstream filtering and metering adjustment factor, is included in the formula as 1, as HCM suggested. We also studied optimizing fixed-time signal design in the present study. Therefore, the value k in the formula is taken as 0.5 for the fixed-time signal control. The progression adjustment factor is calculated and included to explain the effect of coordinated traffic signal control. In isolated intersections, this value is taken as 1. Finally, we assume no initial queue is formed before the analysis period, so the control delay $d3$ was taken as 0 for all computations.

According to the HCM 2000 delay model, in cases in which the degree of saturation is relatively low, i.e. not too close to 1, the delay is usually caused by the uniform delay $d1$ part of the formula. In such cases, the effect of the incremental delay $d2$ is low as well. However, in cases in which the degree of saturation is very close to 1 or greater than 1, the effect of $d2$ is relatively high, as well as the uniform delay. Moreover, if there is persistent saturation in a lane group, $d2$ will continue to increase as the analysis period gets longer. It is crucial to know this relationship to understand the effect of delays while analyzing intersection performance.