


Article

Adaptive Approaches for Tidal-Flow Lanes in Urban-Road Networks

Sören Striewski *, Ingo Thomsen and Sven Tomforde 

Intelligent Systems Group, Christian-Albrechts-Universität zu Kiel, 24118 Kiel, Germany; int@informatik.uni-kiel.de (I.T.); st@informatik.uni-kiel.de (S.T.)

Abstract: Every year, traffic congestion costs the global economy billions of dollars in lost productivity, particularly in urban areas. Traffic congestion is a complex problem, as traffic conditions may change at any time. Tidal-flow lanes can be utilised as a feasible traffic-congestion-mitigation strategy to balance the fluctuating traffic demands throughout the day. This paper proposes an adaptive-lane-reversal approach for tidal-flow lanes, to decrease the impact of traffic congestion in urban areas. In order to evaluate the adaptive approach under various traffic conditions, several algorithms and parameter sets are examined, using various network models and traffic demands. As a result, the total travel time of the vehicles in the various networks was decreased by up to 81%.

Keywords: tidal-flow lanes; traffic-congestion mitigation; traffic management

1. Introduction

1.1. Motivation

Traffic congestion is a serious problem in many parts of the world. According to the Urban Mobility Report 2019 [1], the annual cost of traffic congestion in the United States in 2017 was projected to be USD 179 billion, which included 8.8 billion lost working hours and 3.3 billion gallons (about 12.5 billion litres) of wasted fuel. It was, also, revealed that travellers in locations with populations exceeding one million spent an additional 71 h of travel time each year, which corresponds to nearly two weeks of vacation. The reason for these enormous costs is that traffic congestion is a highly complex problem to solve. Additionally, there rarely is a single root cause to address. Growing population and urbanisation as well as road incidents, extreme weather, and ongoing repairs all play a role in traffic congestion.

Tidal-flow-lane control, also known as reversible-lane control or contraflow-lane control, is an established traffic-management strategy that helps to mitigate traffic congestion, by reversing the direction of a so-called buffer lane [2] (see Figure 1). As a result, the capacity of a congested road in one direction is increased, while the capacity of the opposite direction is decreased, allowing the available capacity in each direction to vary in response to directional traffic flows [2]. This makes tidal-flow lanes ideal, in locations where the direction of heavy traffic demand switches throughout the day. As a result, they are an excellent instrument for intelligent control, to lessen the consequences of traffic congestion.

1.2. Current Research

Currently, there is no dynamic tidal-flow lane-control system in use anywhere in the world. According to [3,4], the British Columbia Ministry of Transportation launched a study in 1992, to refine a static-control system and investigate other approaches, such as adaptive-tidal flow-lane control, in a tunnel. However, they uncovered a number of flaws in the research, which include the failure to account for errors caused by traffic monitors as well as algorithms that were too primitive and inefficient. After several years of using the program, it was discovered that traffic patterns did not vary enough to justify the



Citation: Striewski, S.; Thomsen, I.; Tomforde, S. Adaptive Approaches for Tidal-Flow Lanes in Urban-Road Networks. *Future Transp.* **2022**, *2*, 567–588. <https://doi.org/10.3390/futuretransp2030031>

Academic Editor: Armando Carteni

Received: 6 May 2022

Accepted: 6 June 2022

Published: 27 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

continued expense of system operations and maintenance. As a result, the system was decommissioned [4].

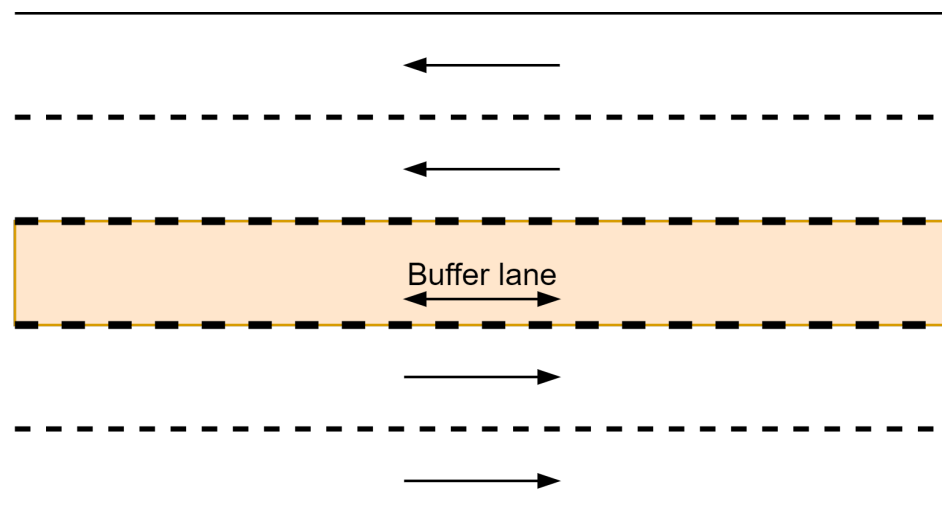


Figure 1. Bi-directional roadway with a central Buffer lane (orange) (inspired by [5]).

This is why, to avoid repeating the same mistake, an accurate and cost-effective algorithm should be employed. Furthermore, technology has progressed, in the years since 1992, and errors from induction loops are not as prevalent anymore [3]. This means that it is easier than ever to gather real-time traffic data, analyse it, and then decide on buffer-lane reversal [3,4]. As the aim of this work is to show the effectiveness of adaptive-tidal-flow lanes in urban environments, traffic patterns will differ significantly from the location chosen by the British Columbia Ministry of Transportation.

Since then, there has been little research on the real-time reversal of tidal-flow lanes to mitigate traffic congestion, in response to dynamically changing traffic demands. Often, the research in this context uses a bi-level optimisation strategy. This kind of approach is a mathematical program, with optimisation problems in their constraints. However, as a road network grows in scale, this is no longer feasible, since the problem grows dramatically [6]. This is due to the fact that bi-level programming is an NP-hard problem.

Additionally, there currently is a lack of research on tidal-flow lanes in combination with urban-road networks that feature traffic-light-controlled junctions. The main reason for this is the increased difficulty of this problem compared to motorways, especially at crossroads, where turning traffic, competing cross streets, and even pedestrians have to be considered.

1.3. Contribution

Currently, the only two techniques for managing tidal-flow lanes throughout the world are based on manual operation, where a person decides on the reversal of a buffer lane, and on fixed-schedule (static) management, both of which have flaws (see Section 2.2).

This work proposes an adaptive approach to real-time tidal-flow-lane regulation in urban environments. Such an adaptive control automates the reversal as an immediate response to certain conditions being met. It, also, captures rush hours and can time reversals more efficiently than the manual and static approaches (see Section 7).

The primary goal of this paper is to illustrate the usefulness of adaptive-tidal-flow lanes in urban-road networks, particularly in terms of reduced travel time for vehicles. For this, a number of traffic models, each with a set of traffic demands, will be tested in a number of traffic scenarios. These will each be evaluated without tidal-flow-lane management, with a static (fixed) schedule, and with an adaptive-management approach. This is done so that the existing standard and our method may be compared subsequently.

Four distinct reversal algorithms for the adaptive-lane reversal will be developed and compared, to discover the best in terms of travel-time improvement. The key difference between all of those algorithms will be the traffic parameter (e.g., speed, density) that is used for the condition of the reversal (see Section 3). The implementation and deployment of these reversal algorithms should be kept as easy and inexpensive as possible, to avoid the mistakes made by the British Columbia Ministry of Transportation. Moreover, they should not be affected by the size of the network, like the bi-level programming approach. For the evaluation, the traffic simulator Aimsun Next 2020 is used, which is a versatile simulation environment with a wide range of capabilities to model transportation projects [7]. It is, typically, accepted within the community as a close-to-reality simulation environment.

1.4. Organisation

The remainder of this paper is organised as follows: Section 2 provides further information about tidal-flow lanes and the requirements for efficient tidal-flow-lane usage. Section 3 then explains the algorithms used in this paper, followed by the description of the traffic models in Section 4. The evaluation process and the results are presented in Sections 6 and 7. Finally, the outlook on future work and the conclusion are given in Section 8.

2. Tidal-Flow Lanes

This section discusses tidal-flow lanes and their present use, as well as guidelines that have an effect on this work and, as a result, minimise the amount of trial and error in determining parameters for models and algorithms. The expressions buffer lane and tidal-flow lanes will be used interchangeably, while, strictly speaking, tidal-flow lanes refer to the overall idea and buffer lane describes the actual lane that is reversible.

2.1. Tidal-Flow-Lane Control

Tidal-flow-lane control is a traffic-management strategy for mitigating traffic congestion, by reversing the direction of a variable number of so-called “buffer lanes”. This increases the vehicle capacity significantly, without adding extra lanes, as described by [2] and illustrated in Figure 1. Therefore, already existing infrastructure can be used more efficiently, while postponing or removing the need to use more land and other resources for additional roads [8]. This is the reason for the majority of tidal-flow-lane installations to be employed in critical bottleneck locations, where expanding the road is simply not feasible, as in the cases of tunnels and bridges [4]. According to the Institute of Transportation Engineers, tidal-flow lanes are, also, one of the most efficient ways to increase road capacity during rush hours and decrease traffic congestion [9].

2.2. Background

The most prevalent methods of tidal-flow-lane operation are manual or fixed-schedule control [10]. However, both have their drawbacks. In the first case, a team of operators monitors the state of traffic and makes decisions about the reversal of a buffer lane, based on branch-specific guidelines or on their personal experience. The most prevalent issue with manual-schedule management is that a human operator cannot estimate the impact of a lane reversal on the global travel time in the network [5]. On the other hand, fixed-schedule control simply reverses the buffer lane at predetermined times. To find the appropriate reversal schedule, a large amount of data is required. This method is, also, incapable of dealing with non-recurring traffic congestion. Furthermore, both methods frequently tend to result in a direction reversal that is either too early or too late, thereby considerably reducing overall system performance [11].

An alternative would be an adaptive-lane-reversal algorithm that can be used to reverse the direction of a buffer lane in response to real-time traffic demand. Moreover, compared to the manual tidal-flow-lane approach, there is far less maintenance necessary once the initial setup is complete.

2.3. Requirements

Tidal-flow lanes are not suitable for all road types, and certain requirements must be satisfied before considering tidal-flow lanes for traffic mitigation. The requirements presented in this section will be used later, to develop the models in Section 4.

According to the American Association of State Highway and Transportation Officials (AASHTO) [12], tidal-flow lanes have advantages and are applicable only on bi-directional undivided streets, when 65% or more of the traffic moves in one direction during peak periods. In addition, at least five lanes are required for the most efficient use of tidal-flow operations. In case of only three or four lanes in total, just one lane is available for the direction of lighter flow. A turning vehicle in that direction, then, might completely obstruct traffic, which could cause more congestion than if no reversal approach had been used in the first place. To avoid this from happening and to keep emergency vehicles from being obstructed, at least two lanes in each direction must be maintained. The AASHTO, also, advises closely monitoring left turns, as they can clog traffic and negate the benefits of tidal-flow lanes. This was later confirmed in Section 4. Introducing tidal-flow lanes in urban road networks is, therefore, only feasible for continuous street segments with a high number of lanes but without a median strip. It should, also, be possible to restrict left turns and on-street parking [12].

2.4. Reversal Management

A recurring statement in the literature is that traffic participants must be informed of the reversal, without any delay. They must, also, be given sufficient time to depart the buffer lane safely and orderly.

There are several methods for notifying vehicles in both directions, to minimise confusion and/or head-on collisions. Variable-message signs (VMS) are a more advanced version of the analogue street sign. They mark accessible lanes with a red cross and non-accessible lanes with a green arrow (see Figure 2). There is usually no physical separation of traffic in both directions, but a VMS allows lanes to be closed adaptively, making it the most popular kind of lane separation for tidal-flow lanes. This is, also, the type of traffic management assumed in this paper.

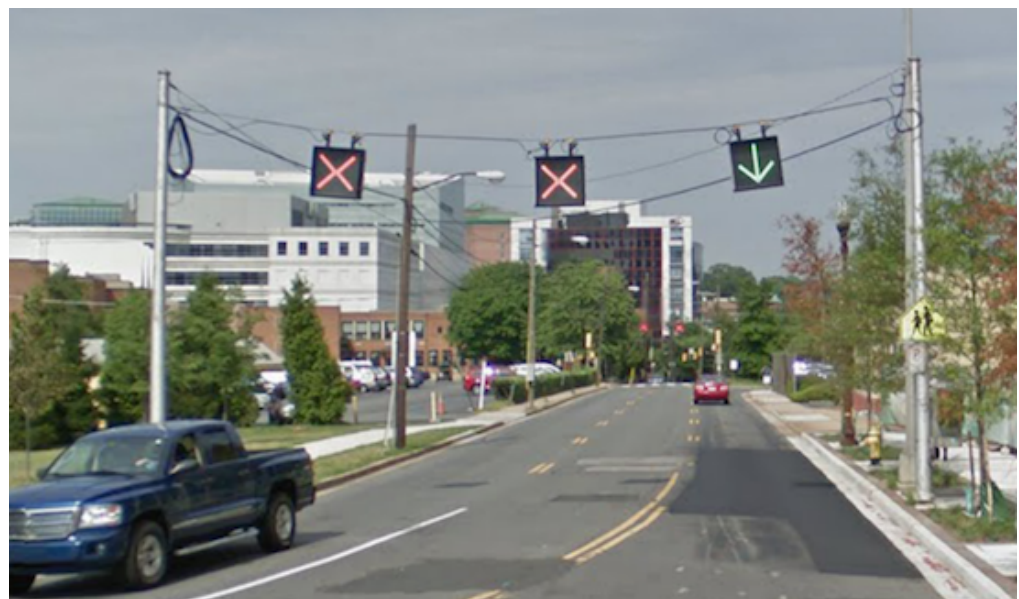


Figure 2. Overhead digital signals from Washington Blvd., Arlington, VA, USA [13].

2.5. Transition Time

According to [4], the transition time of the buffer lane from one direction to the other is the most critical time period for the safety of traffic participants. This is similar to the intersection-clearance times in urban-traffic control. It was mentioned that the transition period should be as short as feasible, to avoid losing the intended benefits that the tidal-flow lane is meant to provide. At the same time, it must be long enough for other traffic participants to observe, understand, and respond to the lane change. Each driver must be allowed to safely depart the buffer lane.

An example, provided by [4], is the city of Edmonton, UK, where the transition time is defined as the time necessary to traverse the whole buffer lane plus the complete cycle duration of traffic signals within the road segment. In this work, the transition periods will all be based on this example, as this is a tested and used standard in real-world circumstances.

3. Algorithms

In this paper, four algorithms are presented, which all operate similarly. The applied decision logic is based on a window of the last x minutes, commonly known as the aggregation time. Section 6.2 describes additional factors, such as the transition time. To obtain reliable traffic information, the models described in Section 4 will include simulated detectors that represent, for instance, induction loops within the road. When the traffic in one direction reaches a specified critical value and, while the opposite direction is below the same threshold, the buffer lane is reversed in favour of the first direction. The main distinction between the algorithms is the traffic parameter used for this comparison, such as speed or density. In the following, each algorithm will be explained in detail.

3.1. Flow Density

The first algorithm used in this work is an adapted version of the one proposed by [2], which uses the flow-density graph to find the optimal reversal time (see Figure 3). The original algorithm was intended to reverse several lanes, which is not necessary for this context, as any improvements on just one lane translate to multiple lanes. In [2], a set of pre-specified critical densities is employed. These are derived from traffic data collected from a highway over the course of one month in 2016, where the algorithm was only tested on a continuous route. To deal with the ambiguity in the data, a set of variables was created. As a result, the technique depends on an approximation for finding the set of critical densities. This is unnecessary for this work because the theoretical optimal critical density was calculated using the fixed formula in Equation (1). This adapted algorithm was later improved using a “flattening” technique, which favours more recent data over older data (see Section 6.2.3).

Based on the flow-density diagram in Figure 3, some observations regarding the connection between traffic flow and density can be made.

A first, trivial insight is that there is no flow while no vehicles are on the road. Second, the density and flow of vehicles will progressively increase as cars enter the road segment, and vehicles will eventually begin to travel at a slower pace to maintain a safe distance from the car in front of them.

This suggests that there is a critical point k_{crit} , where more vehicles are added to the road section than it can handle, and the flow decreases due to a lack of room for vehicles to travel at free-flow speed. The critical point k_{crit} is located at the maximum flow Q_{max} of the road section. Finally, as more vehicles enter the road segment beyond the point k_{crit} , there will be a period when no vehicle is able to move, defined as k_{jam} in Figure 3.

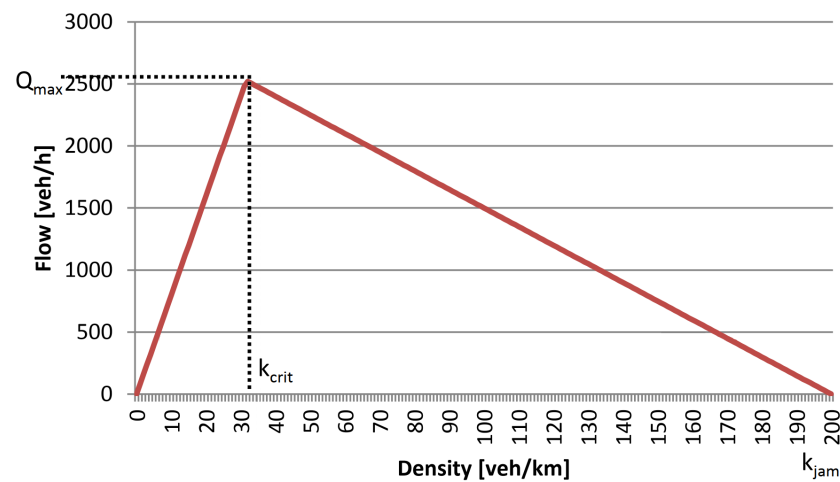


Figure 3. The flow-density curve [14]. k_{crit} depicts the turning point from normal traffic flow to congested traffic flow.

The average length of the standard car inside Aimsun Next is 4 m. Cars, also, leave an average clearance of 1 m even amid traffic congestions. The maximum density (k_{jam}) of a road segment can, thus, be computed as:

$$\frac{1}{4 + 1} \frac{veh}{m} = 200 \frac{veh}{km} \tag{1}$$

According to [7], k_{crit} for vehicles of known length can be calculated as follows:

$$k_{crit} = \frac{1}{VR + \frac{1}{k_{jam}}} \tag{2}$$

V represents the free-flow speed in km/h, R represents the vehicle’s reaction time (0.8 s), and k_{jam} represents the maximum-density parameter in veh/km [14]. V and R must use the same time unit.

3.2. Speed

The speed algorithm will assess the free-flow speed in each direction, reversing the buffer lane if the cars’ average speed falls below a certain level (e.g., 40 km/h). This algorithm implements the basic concept of reversible lanes, which asserts that buffer lanes make sense only if the operating speed is reduced by more than 25%, over an extended time period [4]. To determine the point of reversal, the main assumptions of the speed-density diagram are used (see Figure 4). As more cars are added to the road segment, speed decreases proportionally. Monitoring the present speed provides a good indication of the road’s condition. The reversal for this algorithm is meant to occur when the speed falls below a critical threshold, rather than when it climbs over a critical threshold.

3.3. Ratio

The ratio algorithm counts the cars travelling in each direction and adds them up to be the total number of vehicles travelling in both directions during the aggregation time frame. The ratio (percentage) of cars travelling in each direction may be determined using this sum. A reversal occurs when a directions ratio surpasses a specific threshold (e.g., 70%). This algorithm implements the basic concept of reversible lanes, as described in Section 2.3: For a reversible lane to be effective, directional traffic should be at least 65% in the direction of the heavier flow over an extended length of time.

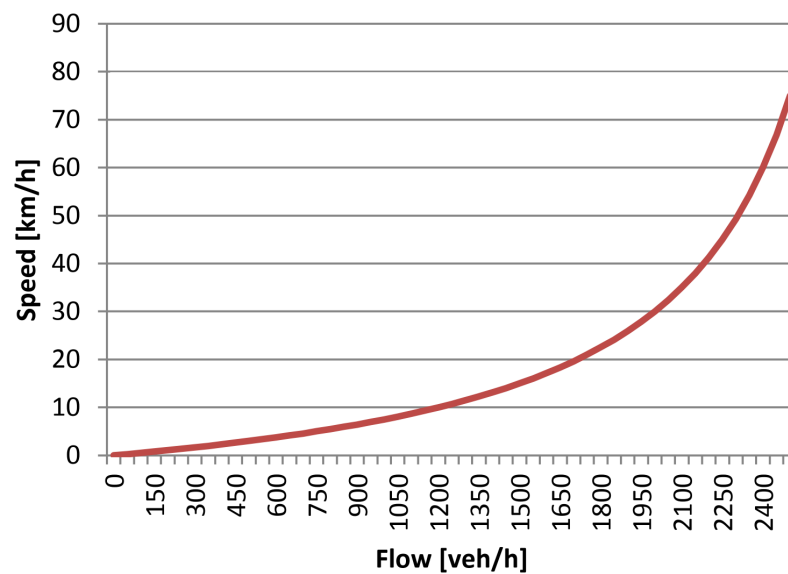


Figure 4. An uncongested road section’s speed-flow curve. The flow (veh/h) of the road segment grows when the speed (km/h) increases, as long as all vehicles can travel at free-flow speed. This is a fundamental diagram of traffic flow that is widely acknowledged in the literature [14].

3.4. Occupancy

Finally, the occupancy algorithm uses the percentage of time a detector was active during the aggregation time. When the total “duration of activation” exceeds a critical threshold (e.g., 70% activation in the previous 15 min), a reversal is initiated, unless the opposite direction is running under the same critical occupancy.

4. Models

Three models were created, based on the requirements in Section 2.3, to test the algorithms under various traffic conditions. In all cases, the primary road segment is one kilometre long and features a variable number of intersections. In these models, the free-flow speed is set to 50 km/h, since this offers good comparability between the models.

In the following, we will use the terms simulation and replications, as used in the context of Aimsun Next. In simple terms, a simulation is a whole test case with multiple traffic demands, whereas a single replication is a randomised traffic scenario produced from a single traffic demand. Each simulation includes three demands and five replications, which are based on random seeds to generate averages. This is done to obtain more consistent findings, which are not simply outliers owing to a poor seed selection, to gain more reliable results. The five replications are then averaged internally in Aimsun Next, by summing up the results and dividing by the number of replications. The resulting average for a single-parameter set is then saved in an external database for further analysis.

As mentioned in Section 3, the models feature induction loops (“detectors”) to obtain information on the current traffic. For the following model descriptions, motorways are called an uninterrupted-flow infrastructure, whereas urban arterial routes are termed interrupted-flow infrastructures, since cars come to a complete stop at traffic signals [15].

4.1. Turning Traffic

The amount of turning traffic in the models is kept to a minimum, to maximise the efficiency of the tidal flow lanes without imposing left-turn restrictions. The sum of left- and right-turning traffic is set to 15% (each turning direction features 7.5%). Those numbers are derived from the fact that, at least for the models in this work, the algorithms ceased to work as the amount of left-turning traffic rose above 15%. This is because, as the share of turning traffic rises, potentially more cars will remain in the turning lane and block the buffer lane or the detectors, which prevents or delays reversals. Moreover, stop lines are

put in place to prevent cars from pulling into the junction while waiting for oncoming traffic to stop, as this reduces the advantage of the buffer lanes.

4.2. Signalisation

The traffic lights in Figure 5 have a fixed signalisation cycle of 90 s, with a 3 s yellow phase and a 2 s red phase between the signalisation groups. Horizontal traffic and cars exiting a horizontal section through a right turn are served for 25 s, while left-turning traffic is served for 10 s. These four signal groups then add up to the signalisation cycle of 90 s.

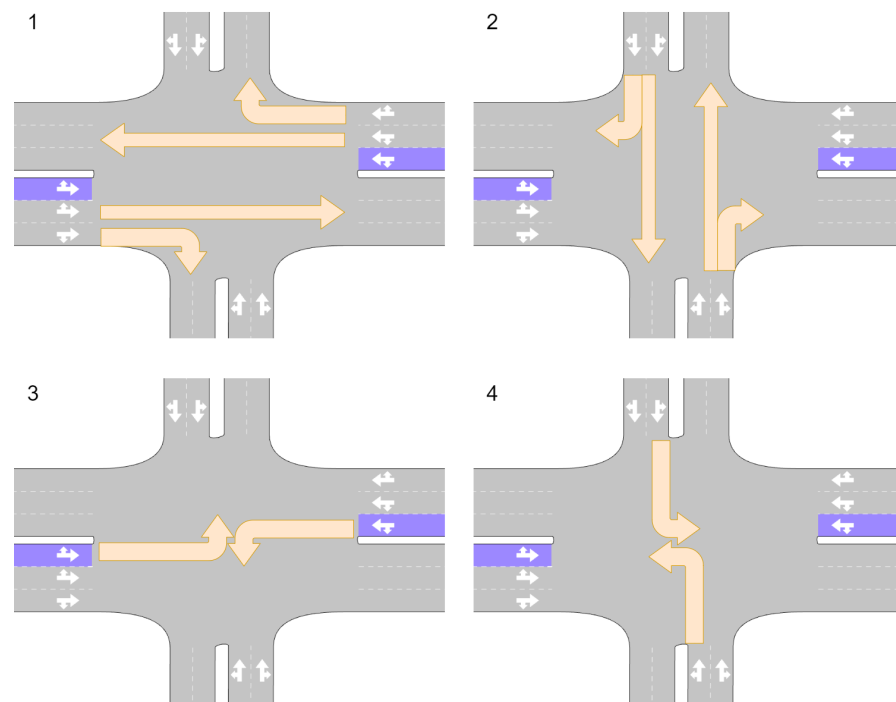


Figure 5. Demonstration of the signal groups. The signal groups for turning traffic are shown by orange arrows. Horizontal traffic and cars exiting a horizontal section through a right turn are served in signal group 1. Then, signal group 2 allows all left turns from the horizontal portions through. Vertical traffic is handled in the same way in signal groups 3 and 4. The signal group can be switched and is not crucial for the subsequent analysis.

4.3. Aimsun Next

Aimsun Next features a multitude of settings that may be tweaked. In this work, the default parameters for an urban-street section and a standard car were used, which are outlined in the extensive documentation [7]. Aimsun Next is a traffic simulation for perfect situations in which drivers are not distracted, and each traffic participant is fully aware of the intentions of the other traffic participants as well as the distances between them. However, this is all within the scope of expectations. It is professional simulation software that is used by experts all around the world for the construction of real-world roadways.

4.4. 0-Intersections

The model in Figure 6 depicts a motorway environment with no intersections or traffic signals, implying that the traffic is uninterrupted and the only way to cause delays is to significantly increase the number of cars attempting to enter the network. This model was created to assess the difference of the adaptive approach on a motorway to that in an urban area with intersections. It features three lanes in each direction, with the innermost lane in each direction acting as a buffer lane for one direction of traffic at a time. This is the only feasible workaround to implement buffer lanes in Aimsun Next.

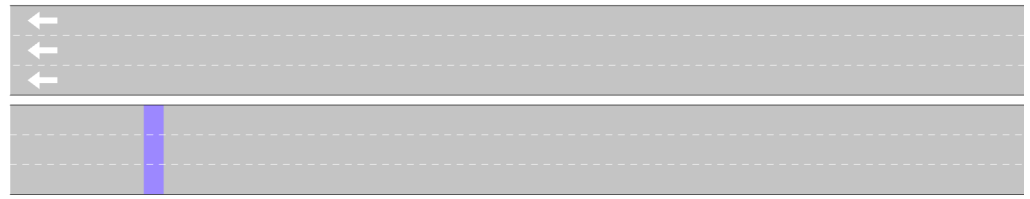


Figure 6. An enlarged version of the 0-Intersection model, representing an uninterrupted motorway with a detector spanning all lanes (purple).

The traffic in this section traverses in a west–east or an east–west direction. The detectors are placed 10 m after the start of a street segment in each direction (the purple line in Figure 6). Other models, such as the one represented in [2], were also explored, but in order to have comparable findings for models with and without intersections, the models' schemas were kept as basic and identical as feasible.

4.5. 1-Intersections

The network model in Figure 7 represents an urban environment with an interrupted traffic flow. It is similar to the 0-Intersection model, as it contains three lanes in both directions, with the innermost lane acting as a buffer lane for one direction of traffic at a time as well as an intersection with an arterial-road section, with two lanes in each direction of traffic. Each section is 482 m long: one kilometre minus the width of the junction, which is placed directly in the middle. Again, the detectors were placed 10 m after the beginning of each horizontal section. It was later discovered that the distance between the detectors and the traffic lights influences how well the algorithms perform in this model. This will be discussed in Section 7.3.

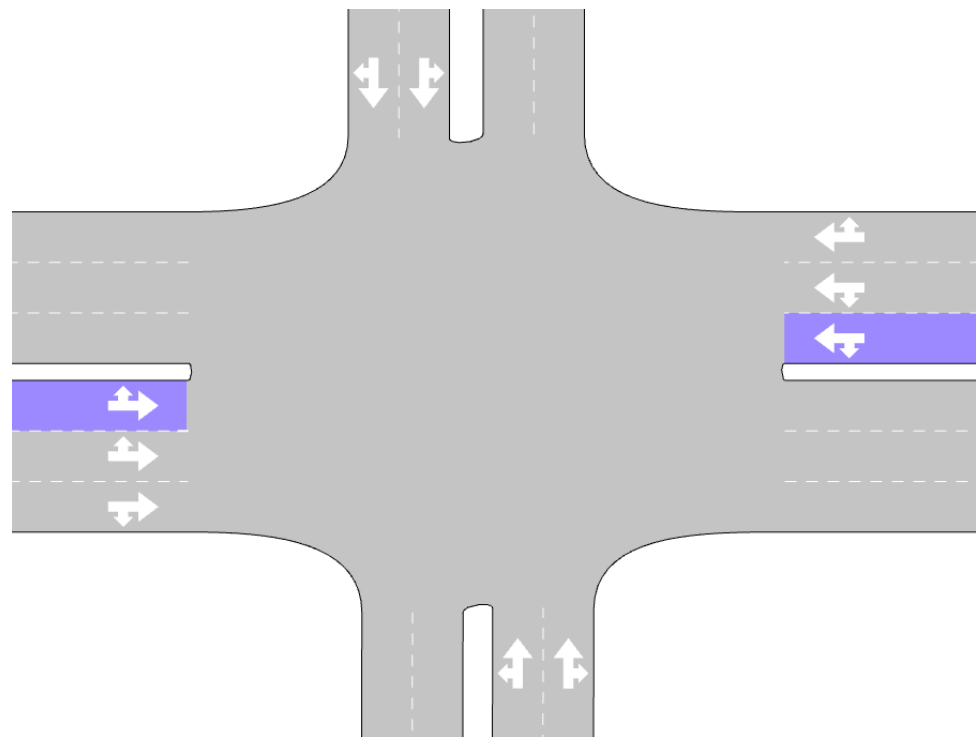


Figure 7. An enlarged version of the 1-Intersection model, representing an interrupted urban setting with a detector spanning one whole lane (purple).

4.6. 2-Intersections

The significant difference from Figure 8 to the 1-Intersection model is that there are two intersections evenly spaced along a road of one kilometre (310 m for each section). Each

junction features two arterial roads with two lanes in each direction, as in the 1-Intersection model. The detectors were placed 10 m after the start of each horizontal section. However, unlike the 1-Intersection model, moving the detectors closer to the traffic signals had no discernible benefit.

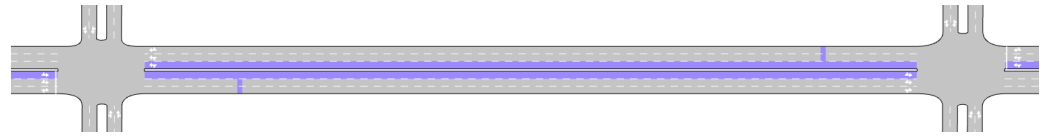


Figure 8. The 2-Intersection model, which represents an interrupted urban setting with a detector spanning one whole lane (purple).

4.7. Demands

Traffic demands can be represented as an origin–destination matrix in Aimsun Next, which defines how many cars travel from each origin to each destination on an hourly basis. In order to create traffic congestion, the capacity of a road section was manually determined, by evaluating various increased traffic demands. The road capacity, then, was periodically exceeded, as it is normal during rush hours in cities (see Table 1 for the amount of vehicles to exceed the capacity of the roads). Minor flow refers to typical traffic flow that does not exceed the capacity of the road segment, whereas major flow refers to rush-hour traffic that exceeds the capacity of the road section. The terms upper and lower will be used to address the two ways in which the buffer lane might shift. The upper one represents the traffic direction of the primary street segment from east to west, while the bottom one represents the traffic direction from west to east (see Figure 6).

A rush hours will always start with more traffic in the upper section, followed by the next rush hour in the lower section, and so on. The beginning and ending timings of rush hours for each demand may be found in Table 2. The difference in the simulated minor and major flow between the 0-Intersection model and the other models is due to the traffic characteristics in those models. The motorway setting provides for uninterrupted traffic flow, whereas the urban setting is an interrupted traffic flow. Both types of models have been pushed to the point that a virtual queue, as part of the simulation in Aimsun Next, has formed outside of the simulated network, causing traffic congestion. This virtual queue is intended (see Section 6 for a rationale).

- Demand 1** reflects a normal workday, covering 10 h, with major rush hours in the morning and evening, and two shorter rush hours in between. This was set up, as the algorithms were designed to primarily enhance travel times during daily commuting.
- Demand 2** is a compressed version of Demand 1, as it no longer simulates a regular workday and instead seeks to stress test the algorithms over the period of four hours, with repeated rush hours in opposing directions. This is done to emphasise the advantages of an adaptable approach in the face of rapidly changing traffic demands.
- Demand 3** adds one rush hour to Demand 2 because adaptive-lane reversal should yield greater efficiency, with more directional shifts than the static one, for example.

Table 1. Amount of minor and major flow in veh/h for each model.

Model	Minor Flow	Major Flow
0-Intersections	2000	6000
1-Intersections	850	1500
2-Intersections	850	1500

Table 2. Start and end times of the rush hours for each demand. Demand 1 lasted 10 h, Demand 2 lasted 4 h, and Demand 3 lasted 5 h.

Demand 1	Demand 2	Demand 3
1:00–3:00	0:30–1:00	0:30–1:00
3:30–4:30	1:20–2:20	1:20–2:20
6:00–6:30	2:30–2:50	2:30–2:50
7:00–9:00	3:00–3:30	3:00–3:30
-	-	4:00–4:30

4.8. Static

A static-reversal schedule was established for subsequent comparison, which misses part of the rush hours, demonstrating the static-reversal management's shortcomings. The reversal times for Demand 1 are as follows: the upper half uses the buffer lane from 0:00–7:00, while the lower half uses the buffer lane from 7:00–10:00, clearly missing two rush hours, which represent shorter rush hours but are not periodic in the training dataset. As these rush hours are not periodic, the static-traffic-control method cannot handle them. For Demand 2 and 3, the reversal times are as follows: the upper section uses the buffer lane from 0:00 to 3:00, while the bottom section uses the buffer lane from 3:00 to 4:00, or rather 5:00, in the case of Demand 3. Again, some rush hours will be missed.

4.9. False Negatives

False negatives in this work indicate that the algorithm will initiate a reversal, even if there is no rush hour in the now-reversed direction. This can occur when there is a brief increase in traffic in one direction that lasts just for the duration of the aggregation time period and then fades. The traffic demands do not reflect false negatives, as they would have minimal influence on the results, since the buffer lane can only be positioned in one of two directions. A false negative shifts the lane away from the direction that will be under rush-hour conditions next, forcing the algorithm to shift the lane again, but this already occurs in the models because the rush hours are always in the opposite direction of the current buffer-lane direction. Alternatively, the false negative leads the buffer lane to point in the direction of the next rush hour, which even improves network performance.

A significant number of false negatives in rapid succession may decrease the overall network performance, since the buffer lane is continually blocked for a given time period, due to the transition time, leaving it unavailable in both directions. However, this may be fine-tuned using the aggregation time frame. During the evaluation phase, it was discovered that the algorithm reversed too frequently, resulting in little performance loss, and it was decided that as a result, false negatives are not relevant enough to be considered.

5. Client-Server Architecture

The lane-reversal algorithm is implemented as a client-server application, and the code can only be made available for academic research. For this, the Aimsun Next model is provided with an application programming interface (API) module, written in Python 3, which acts as the client side (see Figure 9). The API module is used to receive network calls in this case. The client sends relevant data to a Java server, where the data are processed and assessed as well as decisions are made. Since the client and server will communicate via rest endpoints, it is possible to replace either of them while maintaining a viable program, as long as all of the endpoints stay the same. This allows for the programming language to be changed in the future and makes it possible to integrate it into the Organic Traffic-Control System (OTC). This traffic-management system proposes an Observer/Controller approach

that optimises intersection signalization and a self-organizing coordination mechanism that enables the design of traffic-responsive progressive signal systems (“green waves”) [16].

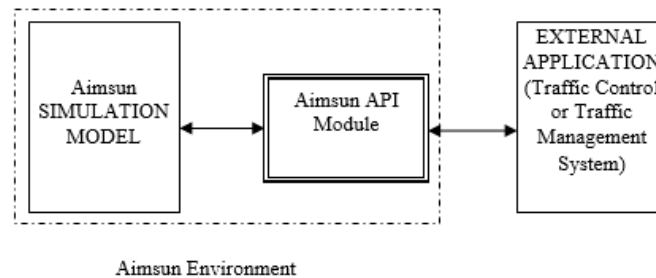


Figure 9. Aimsun Next API Implementation [7].

5.1. Client

The initial client-side code was generated using the quicktype code generator [17], which takes specifications in a JSON file to create classes in a wide range of languages. The resulting client-side code then was populated with logic that interacts with the servers’ rest endpoints. The client implements the 13 high-level Aimsun Next API functions (see Figure 10), but only *AAPInit*, *AAPIManage*, and *AAPIFinish* are used. The induction-loop-detector characteristics are sent to the server-side in *AAPInit* at the start of the simulation. This message also informs the server that a new simulation has begun.

Since the buffer lane has to be attributed to one direction of traffic, *AAPIManage* performs the first policy activation at time step 0.0, assigning the buffer lane to the lower direction of traffic in this work. Since the initial rush hour always occurs in the top part, the algorithms are forced to reverse the buffer lane. This was done to ensure that the algorithms receive no unintended benefits. However, it is irrelevant which direction the buffer lane is allocated to in the initial time step, as long as the direction is consistent throughout all test cases. Since *AAPInit* is unable to handle policies owing to separation of concerns, *AAPIManage* is used to activate the first policy, instead of *AAPInit*.

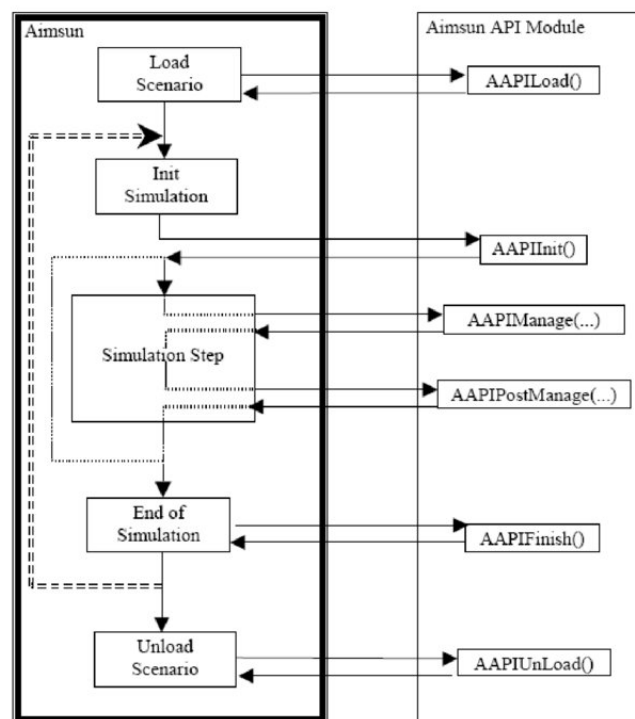


Figure 10. Aimsun Next and API Module interaction [7].

In addition to the current time and direction of the buffer lane, the following measurements are transferred to the server every 0.8 s (the Aimsun default) via *AAPIManage*: number of cars passing a detector, occupancy, speed, and current density, as measured by the detectors in the previous time step.

Following the data transfer, the client awaits a response from the server, containing the decision as to whether or not the buffer lane is supposed to be reversed. After this, the client side either reverses the buffer lane or does nothing, and the simulation proceeds as usual. Eventually, *AAPIFinish* informs the server about the ending of the simulation.

5.2. Server

Java 11 was chosen as the programming language, to simplify the future integration into the Organic Traffic-Control System [16]. The Java server analyses the simulation data and makes real-time decisions about the reversal, based on a variety of variables, depending on which algorithm is utilised in the current simulation. It is responsible for the decision logic, while the client only passes the data through and potentially reverses the buffer lane.

6. Evaluation

This section analyses the presented algorithms to investigate which comes with the highest benefit for traffic-flow management. The evaluation consists of several phases, which are addressed in this section along with the evaluation criteria and parameters.

6.1. Criteria

It was found that different algorithms affect the frequency of lane changes, the number of stops, and the duration of stops. These changes were not evaluated here, as there is no way to account for waiting vehicles outside of the simulated-road network, since Aimsun does not track that data for vehicles in virtual queues. The models were designed to be as compact as possible and, by design, will allow a virtual queue to form outside the network. Larger networks would provide a different set of challenges, such as significantly longer simulation periods. In this work, the travel time and number of reversals were determined to be the best indicator for algorithm performance, as outlined in the following.

6.1.1. Travel Time

The most important factor is the vehicle's travel time. As no travel-time statistics are available for vehicles in virtual queues, the average time spent in the queue (essentially, waiting in traffic congestion) and the average travel time of cars from entering to departing the model are summed up. This sum then results in the total travel time and indicates how much time a car may spend in traffic on average.

6.1.2. Number of Reversals

Evidently, an algorithm should only reverse as many times as necessary. For example, if a model contains four rush hours, with the subsequent rush hours always modelled to be in the opposite direction, a good algorithm will reverse three or four times depending on the direction of the buffer lane in comparison to the direction of the first rush hour. However, fewer reversals indicate that the algorithm did not accurately anticipate a rush hour, whereas more reversals lead to increased driver confusion and possibly increased safety concerns.

6.2. Parameters

This section further describes the transition times as well as the aggregation time frame, in addition to the explanation for the flattening technique employed.

6.2.1. Transition Times

Based on the previously mentioned example of the city of Edmonton [4], three possible transition times were generated for each model. Since the 0-Intersection model does not represent an urban area and, hence, lacks signalised junctions, transition times are based (1) on the time it takes a vehicle to travel the complete length of the motorway, (2) one and a half that time, and (3) twice that time.

In contrast, the transition time in the other two models was mostly determined by the dynamic scenario: (1) waiting for the buffer lane to be totally empty in both directions. In addition to this case, (2) the time it takes a vehicle to clear the whole length until it hits a traffic light, and then a full traffic-light cycle of 90 s was added, as well as (3) the time it takes a car to clear the section twice plus a full traffic-light cycle.

6.2.2. Aggregation Time Frame

The approach in this work is based on simulated-traffic data, such as flow and speed, which are internally represented as time series. The adaptive reversal algorithms each apply their logic on a certain window (number of preceding time steps), in which the data are summed up and aggregated into a single value using the aggregation time frame. The time intervals 5 min, 10 min, and 15 min were chosen, as 15 min is half the duration of the shortest rush hour. Depending on this rush hour, other durations are possible. However, shorter ones may result in more frequent oscillation, whereas longer ones may miss a rush hour entirely.

6.2.3. Flattening

Up to now, the means of the aggregated values are used. However, with flattening, the last x percent of values are processed twice and counted as separate entries. Each data point in the prioritised time frame appears twice, suggesting that values outside of this region have less effect on the final result. Several values for x were evaluated: 50%, 40%, 30%, 20%, 10%, and 5%. The most consistently performing value for x turned out to be 30%.

6.3. Evaluation Process

To assess the effectiveness of adaptable tidal-flow-lane solutions, the algorithms have to be compared with regard to the criteria in Section 6.1. This was done in three phases:

6.3.1. First Phase

A set of algorithm parameters was generated by examining server log data shortly before and after the occurrence of a rush hour. This yielded upper and lower bounds for the values that may occur around potential reversals. Based on this, other parameter sets were derived by increasing as well as decreasing these bounds and evaluating the effect on the final result. These values were then used to test the overall approach and models. Finally, the values were used to assess the best values for the aggregation time frame, transition time, and detector location in the final models.

6.3.2. Second Phase

The second step was used to determine the optimal critical values for each algorithm, in terms of trip-time improvement and number of reversals, using the best transition time and detector placement from the previous phase. In terms of the aggregation time frame, all three potential values were tested throughout all simulations because each has advantages and disadvantages that are not only reflected in the travel time. The best values were found by iteratively evaluating each potential algorithm parameter in small increments. This revealed parameters that perform well across all models and demands.

6.3.3. Final Phase

Finally, once all of the best parameter combinations for each algorithm were identified, the flattening approach was applied to all these parameter sets to assess the possibility for further improvement (see Section 6.2.3). Based on this the best algorithm was selected.

7. Results

The findings obtained during the evaluation phase Section 6 are provided here, together with the evaluation process, parameter combinations, and evaluation criteria. More than 1300 simulations were conducted and evaluated to arrive at these conclusions.

7.1. Base Cases

To assess the overall approaches, two cases must be compared: the 'static base', as described in Section 4.8, with its static reversal timings was conducted for each model. In comparison, for the 'no-strategy base', the buffer lane was assigned to one of the directions during a simulation. This was done for each direction, to allow for a fair comparison, without having to completely remove the lane. Otherwise, introducing a lane would obviously result in better travel times.

7.2. Turning Traffic

At least for the models shown in this work, the algorithms ceased to work as the amount of left-turning traffic increased too much: Potentially more cars will remain in the turning lane and block the buffer lane, which prevents or delays reversals. Furthermore, all algorithms rely on precise detector readings, but as the queue in the left lane grows, cars will come to a halt on the induction loop, rendering the acquired data useless.

More than 7.5% of left-turning traffic during the rush-hour phase resulted in significantly longer waiting times, until the reversal was completed. Once 15% left-turning traffic was achieved, the models failed totally because both ways were congested the whole time after the first few rush hours. This was due to a lack of space for left-turning traffic to relocate to, after a reversal commenced, as well as the detectors being occupied the entire time, due to the long line of turning vehicles.

This effect even occurred when there is no right-turning traffic. A possible solution for future work could be contraflow left-turn lanes, which allow for more free flow of left-turning traffic [18]. As a result, it is advised to either limit the amount of left-turning traffic to a maximum of 10% or restrict it entirely.

7.3. Detector Placement

The distance between the detectors and the traffic lights affects how well each algorithm performs: the optimal for each method, however, was not identified in this work, since this would need significantly more experiments. For the 1-Intersection model, distances of 370 m and 470 m from the traffic light were tested to confirm the assumption that detector placement is more important for some algorithms than others. Generally, by moving the detector further away by 100 m, the ratio algorithm saw an improvement of up to 78%, while all the other algorithms performed worse, by up to 13%.

Based on this, the detectors should not be placed too close to the traffic lights. Cars stopping on the detector potentially trigger a reversal, even if the road segment is not congested. It is recommended that the appropriate distance is calculated by assessing the usual queue length at the traffic signal and then adding a few metres for safety.

7.4. Aggregation Time

While the 5 min aggregation time performed the best in all simulations, it was discovered to cause more reversals than the other aggregation times, perhaps contributing to driver confusion. It is better to choose a higher aggregation time based on the average length of a rush hour in the area, sacrificing some performance for the safety of all traffic participants. In the actual world, there may be false negatives, which are unlikely to be extremely long, since otherwise they would be considered actual rush hours.

7.5. Transition Time

When comparing the dynamic and fixed transition times for the 1-Intersection and 2-Intersection models, the dynamic-reversal-transition time, which waits for all vehicles to leave the buffer lane, performed best in most situations. It is also the safest option, as the entire lane in both directions is clear. Some of the static transition periods suffered performance losses when transitioning to the dynamic transition time. This suggests that the buffer lane was reversed while cars were still in the lane. This is unsafe in any situation.

If buffer-lane-clearance technology is not practicable and/or specific safety precautions are in place, static transition periods may be used in real-life situations. Instead of one traffic-signal cycle, the static transition time should then be based on two traffic-signal cycles, as this is the safer option.

7.6. Flattening

The flattening approach from the third evaluation phase only improved the performance of the flow-density algorithm, while all the other algorithms were unaffected.

7.7. Ratio Algorithm

The ratio algorithm beats any other algorithm presented in this work; with the proper parameter settings, this results in the algorithm regularly triggering up to 30 reversals on a four-rush-hour model. It is so sensitive that it can identify rush hours sooner than any other algorithm, but it will also trigger reversals when there is little to no traffic at all. As stated in Section 6, the number of reversals must be kept to a minimum. Therefore, the parameter sets for the ratio algorithm were restricted to sets that resulted in an amount of reversals not more than twice the number of rush hours in the given model. With this constraints in place, the ratio method produced results equivalent to, if not outperforming, the other algorithms, at least on the 0-Intersection model. In the other scenarios, the ratio method consistently outperforms all other algorithms for Demands 2 and 3, while falling short for Demand 1 (the 10 h workday). This suggests a better performance, in case of quick reversals, that results in calm traffic rather than long periods of traffic surges.

However, due to the nature of the data acquired via induction loops, the ratio method has a severe flaw, making it the worst for Demand 1. Since vehicles travelling through the induction loop are counted, the number of vehicles passing through the induction loop falls when there is a traffic jam. This means that when fewer vehicles travel through the induction loop on a congested road, the ratio decreases. In other words, the buffer lane might be 'reversed away' from a busy road since it is no longer over the critical value.

Furthermore, in the 1-Intersection and 2-Intersection models, the ratio algorithm was the most sensitive to transition and aggregation times: In an urban traffic scenario, a little change might result in big performance increases but potentially massive performance losses. This implies a large effort in fine-tuning this algorithm to the specific location where it is being implemented. It was also established that detector positions have a substantial impact on this algorithm. Therefore, it might not be deployable at all street sections.

All of these points render this algorithm unsuitable for adaptive-lane reversal in urban areas.

7.8. Speed Algorithm

The speed algorithm failed completely on the uninterrupted-traffic model because the change in overall speed was not significant enough, making it unsuitable for such traffic situations. However, it performed well on the 1-Intersection and 2-Intersection models, almost always being second-best after the occupancy algorithm.

This algorithm, however, is quite sensitive to the critical value chosen. A change in value commonly causes additional reversals and dramatic variations in travel time.

7.9. Flow-Density Algorithm

With the exception of the aggregation time frame and transition time, which other algorithms must also deal with, there is no need to modify any critical value with this algorithm because it is calculated using a formula. This algorithm can be easily implemented in many locations with minimal adjustments. Regarding the aggregation time frame and the transition time, the algorithm is also highly robust: any changes cause only small fluctuations in the results. The most notable advantage of this approach is its improvability by flattening. This shrinks the gap between the best-performing algorithm while keeping its robustness.

However, the flattening approach, along with the 5 min aggregation time period, resulted in more reversals, while still being inferior to the occupancy algorithm.

7.10. Occupancy Algorithm

This algorithm performed the best in practically all of the examined cases (excluding the ratio algorithm), but only if very low values were used, which might lead to it being susceptible to false negatives. However, there is some leeway in adjusting the algorithm parameters and still having it be the best-performing algorithm. This algorithm also resulted in an almost ideal number of reversals in proportion to the rush hours.

Choosing a lower aggregation time frame nearly always had a positive effect, most commonly in the range of 1% to 13%, with each step. The algorithm gains more from a lower aggregation time, whereas the flow density, for example, only exhibits minor effects with the same steps.

7.11. Overall Results

All presented algorithms improved the performance of the travel time which can be seen in Figures 11–13. For the final evaluation, the occupancy algorithm will be used in comparison to the base cases, since it was consistently the best-performing algorithm: it resulted in an almost ideal number of reversals in proportion to the number of rush hours. The occupancy algorithm also performed better for Demand 1, which represents a normal working day. The final improvements obtained by the occupancy algorithm, in comparison to the base cases, are outlined in Table 3, together with a conclusion of the results in Section 8.

There is no mechanism to simulate driver urgency in Aimsun Next, making it difficult to compel all cars to exit the buffer lane right after a reversal. Even if the lane is closed for vehicles, according to an Aimsun policy, cars will remain in the lane and complete their intended current turn, despite the fact that other turning possibilities are available. In a real-world scenario, all of the results provided may be better, since human drivers might feel a sense of urgency when the buffer lane is about to reverse.

By setting the simulated cooperation level to 100%, this shortcoming was partly compensated for: other cars always let turning vehicles enter their lane without being selfish, but as certain cars have no desire to leave the lane, this does not solve the problem entirely.

Table 3. The occupancy algorithm’s overall travel-time decrease (in percent), when compared to the base cases with 15 min, 10 min, and 5 min as aggregation times and the optimal transition times for each model. This means that the lowest decrease in travel time comes from the 15 min aggregation time, and the largest decrease in travel times comes from the 5 min aggregation time.

	Static	Upper Closed	Lower Closed
0-Intersections:			
Demand 1	24.84–46.73	69.47–78.36	73.58–81.28
Demand 2	46.13–64.93	0.00–34.90	54.00–70.05
Demand 3	44.18–65.12	11.43–44.64	43.85–64.91
1-Intersections:			
Demand 1	35.20–45.93	65.34–71.08	40.58–50.42
Demand 2	35.56–48.68	35.54–48.67	36.52–49.45
Demand 3	36.72–49.60	40.85–52.89	48.65–59.10
2-Intersections:			
Demand 1	56.03–63.55	61.97–68.47	43.37–53.06
Demand 2	36.72–45.40	20.51–31.41	25.45–35.68
Demand 3	40.01–49.48	20.62–33.15	36.20–46.27

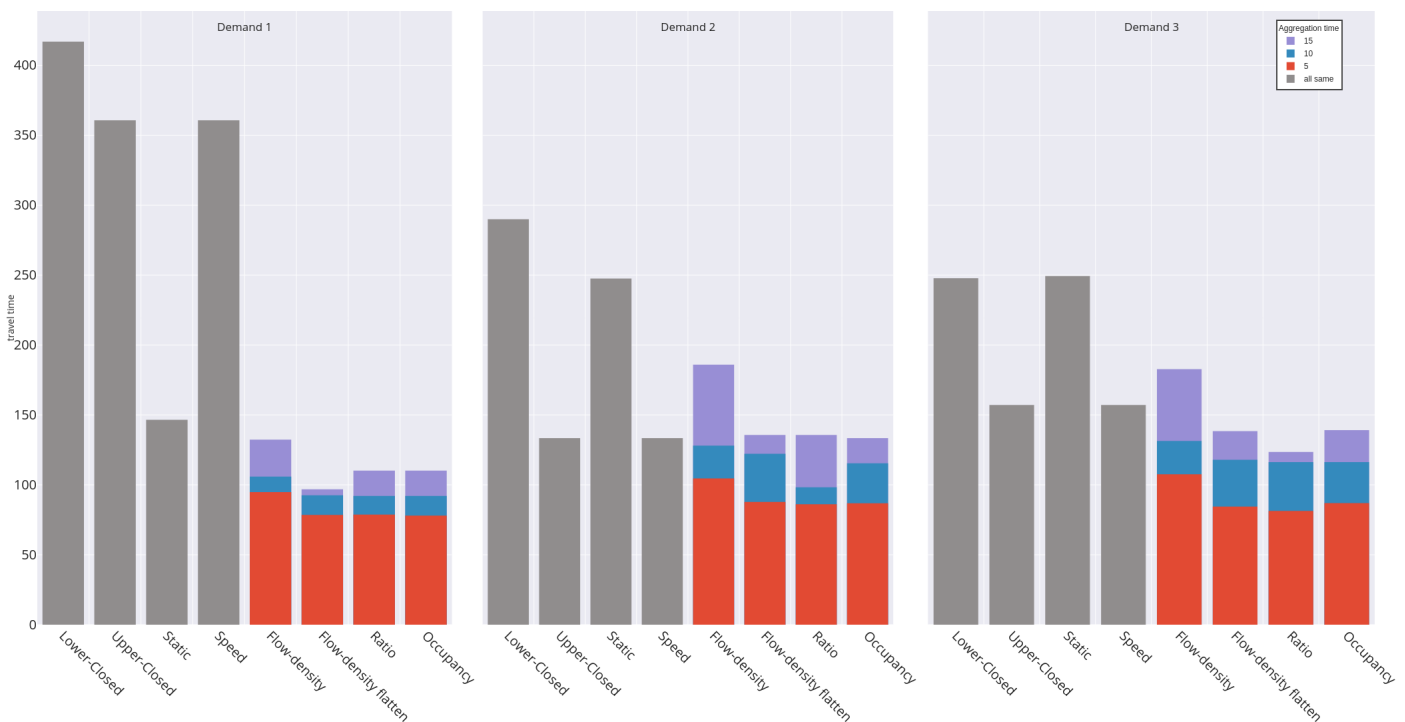


Figure 11. Travel times in seconds for the three demands for the 0-Intersection model. The red bar chart represents the 5 min aggregation time, the blue one represents the 10 min aggregation time, and the purple one represents the 15 min aggregation time. They are placed on top of each other to demonstrate the difference in aggregation times. The grey bars represent the base cases (no algorithm applied) as well as the speed algorithm, since it performed consistently across all test situations. Each demand is shown in its own bar chart, allowing them to be compared.

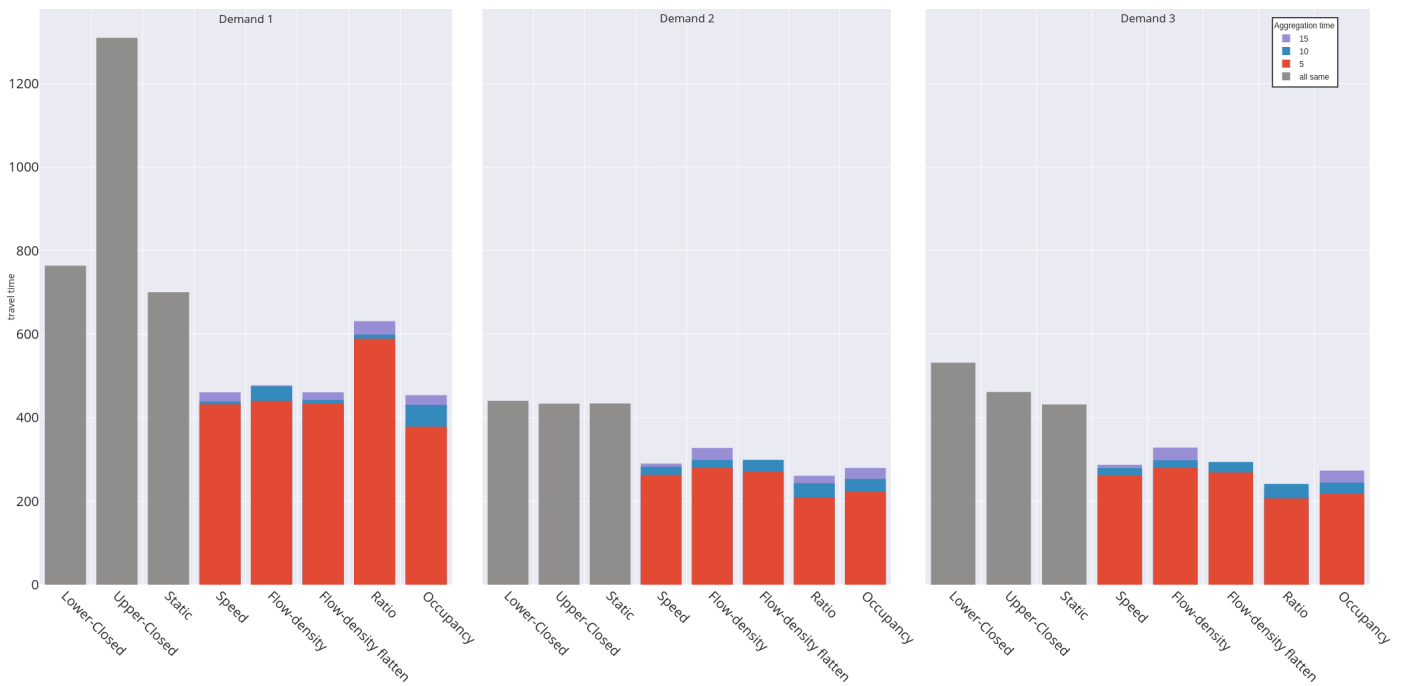


Figure 12. Travel times in seconds for the three demands for the 1-Intersection model. The red bar chart represents the 5 min aggregation time, the blue one represents the 10 min aggregation time, and the purple one represents the 15 min aggregation time. They are placed on top of each other to demonstrate the difference in aggregation times. The grey bars represent the base cases, where no algorithm is applied. Each demand is shown in its own bar chart, allowing them to be compared.

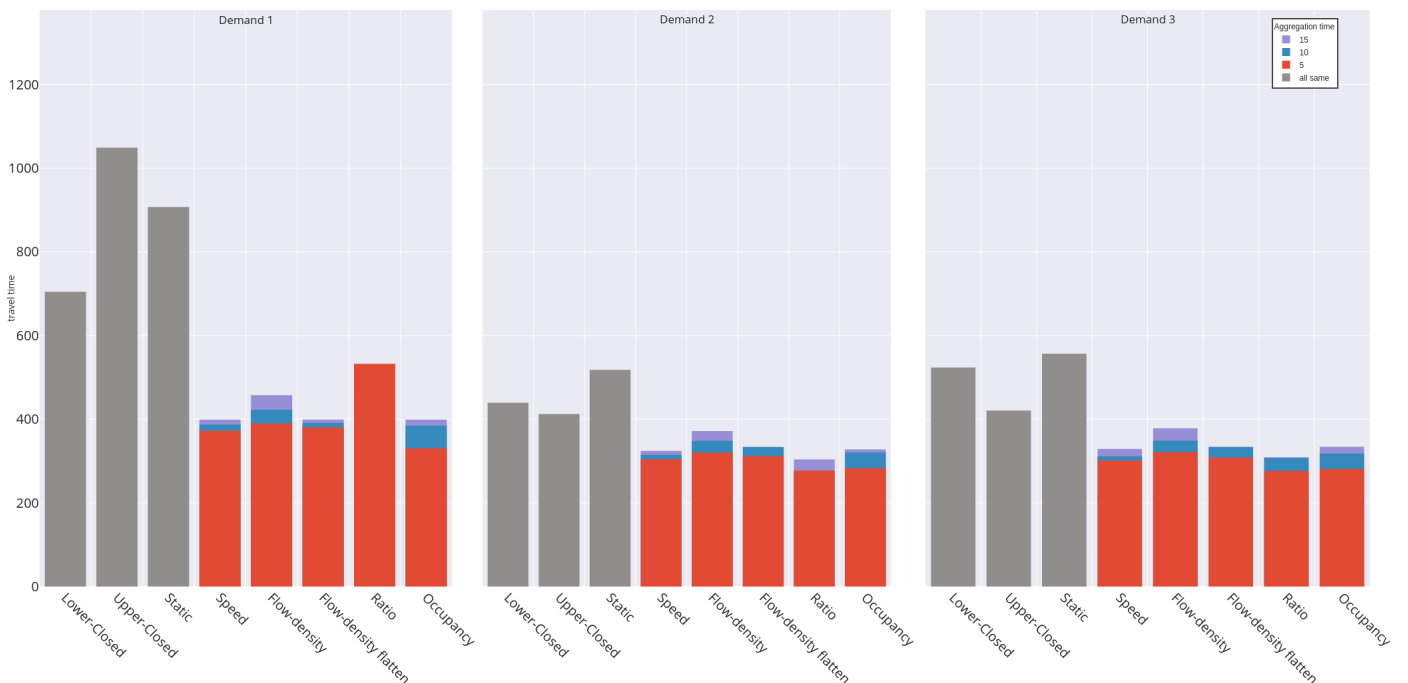


Figure 13. Travel times in seconds for the three demands for the 2-Intersection model. The red bar chart represents the 5 min aggregation time, the blue one represents the 10 min aggregation time, and the purple one represents the 15 min aggregation time. They are placed on top of each other to demonstrate the difference in aggregation times. The grey bars represent the base cases, where no algorithm is applied. Each demand is shown in its own bar chart, allowing them to be compared.

8. Conclusions

Traffic congestion is a serious problem everywhere around the world. The alleviation of congestion would lead to a reduction in commuting stress, saving money and increasing productivity [19]. To address this, four different tidal-flow-management algorithms were developed and evaluated using the traffic-simulation software Aimsun Next.

Three network situations were constructed to evaluate the performance of the algorithms: an extra lane for the upper direction, an additional lane for the lower direction, and a static approach. The total travel time, which includes waiting time outside of the network and driving time inside the network, as well as the number of reversals, was taken into account for the evaluation. To fine-tune the algorithm's performance, three parameters were used: critical value, transition time, and aggregation time. The placement of a detector was identified as the fourth parameter. However, the potential of each algorithm having a perfect location was just noted and not investigated further.

Finally, after determining the best possible parameter combinations for each algorithm, the occupancy algorithm was identified as the best with regard to the criteria (travel time and number reversals), regardless of the fact that all algorithms improved travel time. Using the occupancy algorithm as the evaluation algorithm and depending on the compared base case, total travel time may be reduced in the 0-Intersection model by 0–81%, in the 1-Intersection model by 35–71%, and in the 2-Intersection model by 20–68%. With each additional simulated rush hour, the algorithms' performance compared to the base cases increased.

8.1. Limitations

Although the algorithm performs well in a simulation, there are concerns regarding practical field implementation: an algorithm that repeatedly reverts the buffer lane, as the ratio method does, may cause driver confusion and impair the efficacy of the suggested solution. However, this issue could be mitigated by adjusting the aggregate timeframe as well as the transition time.

As previously noted, a longer aggregation period may result in minor performance losses of the suggested algorithms, which will, in turn, prevent the algorithms from triggering frequent oscillations. The transition time has little influence on total trip-time improvements and may, as indicated in this work, be changed within a reasonable range. A longer transition period will benefit traffic participants' safety, since the danger of perplexing drivers will be minimised if adequate time is allowed for them to assess the new scenario.

These adjustments must all be taken into account during field implementation, which is why we have provided lower and upper constraints for algorithm improvements. We also created demand 2 and demand 3 to test how the algorithms react in case of frequent traffic directional changes. Since the safety of all traffic participants is essential, it is preferable to utilise the highest acceptable values for the aggregate timeframe as well as the transition time.

8.2. Future Work

Future plans include integrating the concept of adaptive-lane reversal in the Organic Traffic-Control (OTC) system—a self-adaptive and self-organised urban-traffic-management system that controls the green duration of traffic lights [20], establishes progressive signal systems [16] (“green waves”) and guides road users through the network using variable message signs [21]. Possible challenges include devising a full decentralised extension of flow-lane reversal as well as using the OTC's ability to adapt its control strategy for the improvements of such an extension.

Potentially, the results from this paper could be improved with some adjustments to the algorithms and the models chosen. Some of the improvements are described below.

8.2.1. Simulation Models

The models used in this work were as compact as possible, but larger models as well as real-life location models with real-life traffic data are required, to further investigate the influence of tidal-flow lanes in urban contexts. In this work, only five lanes in total were modelled. This number could be expanded to six or seven lanes, resulting in an algorithm that could reverse not one but several buffer lanes in each direction.

8.2.2. Algorithms

Ensemble techniques could be applied to combine the algorithms and mitigate their individual flaws. It could also be possible to learn the best set of parameters for each road segment automatically and individually. An algorithm that can decide on the reversal of several separate tidal-flow lanes that may interact with each other might be created to improve overall network performance.

8.2.3. Others

Additional studies in the field of reversible lanes could be conducted, as this work primarily focused on the everyday commute and associated traffic congestion. In addition, with usage of larger models, left turns and potential restrictions could be investigated. Furthermore, employing the algorithm from this paper, the topic of exclusive lanes, especially exclusive bus lanes, might be researched.

The premise of this work was that reversible lanes would be beneficial. Based on this, an additional procedure could prove for a given traffic flow that one party may receive higher benefits from a reversible lane than another.

Author Contributions: Conceptualisation, S.S.; methodology, S.S. and S.T.; software, S.S.; validation, S.S.; formal analysis, S.S.; investigation, S.S.; resources, S.T.; data curation, S.S.; writing—original draft preparation, S.S. and I.T.; writing—review and editing, S.S., I.T. and S.T.; visualisation, S.S.; supervision, I.T. and S.T.; project administration, S.S. and I.T.; funding acquisition, S.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Schrank, D.; Eisele, B.; Lomax, T. *Urban Mobility Report 2019*; The Texas A&M Transportation Institute with Cooperation from INRIX. 2019. Available online: <https://static.tti.tamu.edu/tti.tamu.edu/documents/umr/archive/mobility-report-2019.pdf> (accessed on 6 May 2022).
2. Ampountolas, K.; Furtado, H.S.; Carlson, R.C. Motorway Tidal Flow Lane Control. *IFAC-PapersOnLine* **2018**, *51*, 279–284. [[CrossRef](#)]
3. Xue, D.; Dong, Z. An intelligent contraflow control method for real-time optimal traffic scheduling using artificial neural network, fuzzy pattern recognition, and optimization. *IEEE Trans. Control. Syst. Technol.* **2000**, *8*, 183–191. [[CrossRef](#)]
4. Guebert, A. *Guidelines for the Planning, Design, Operation and Evaluation of Reversible Lane Systems*; Transportation Association of Canada: Ottawa, ON, Canada, 2010.
5. Hausknecht, M.; Au, T.; Stone, P.; Fajardo, D.; Waller, T. Dynamic lane reversal in traffic management. In Proceedings of the 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), Washington, DC, USA, 5–7 October 2011; pp. 1929–1934. [[CrossRef](#)]
6. You, R.; Chen, W.; Gong, Y.; Lin, Y.; Zhang, J. A Histogram Estimation of Distribution Algorithm for Reversible Lanes Optimization Problems. In Proceedings of the 2019 IEEE Congress on Evolutionary Computation (CEC), Wellington, New Zealand, 10–13 June 2019; pp. 1960–1966. [[CrossRef](#)]
7. Aimsun Next. *Aimsun Next Documentation*; Aimsun Next: London, UK, 2021.

8. Transportation Research Board; National Academies of Sciences, Engineering, and Medicine. *Convertible Roadways and Lanes*; Transportation Research Board: Washington, DC, USA, 2004. [CrossRef]
9. Institute of Transportation Engineers. *Traffic Engineering Handbook*, 5th ed.; Institute of Transportation Engineers: Washington, DC, USA, 1999.
10. Wolshon, B.; Lambert, L. Reversible Lane Systems: Synthesis of Practice. *J. Transp. Eng.* **2006**, *132*, 933–944. [CrossRef]
11. Glickman, T.S. Heuristic Decision Policies for the Control of Reversible Traffic Links. *Transp. Sci.* **1973**, *7*, 362–376. [CrossRef]
12. AASHTO. *A Policy on Geometric Design of Highways and Streets*; American Association of State Highway and Transportation Officials: Washington, DC, USA, 2011.
13. Google. Google Maps Image from the 3251 Washington Blvd, Arlington, VA 22201, USA. Retrieved 5 November 2014. Available online: <https://goo.gl/maps/u6RVoQzDkzTDXf9JA> (accessed on 6 May 2022).
14. Transport Simulation Systems. *Derivation of the Fundamental Diagrams of Traffic Flow from the Mesocar-Following Model*; Aimsun SLU, 2010.
15. Roess, R.P.; Prassas, E.S. *Overview of Uninterrupted Flow Methodologies of the Highway Capacity Manual*; Springer Tracts on Transportation and Traffic; Springer International Publishing: Cham, Switzerland, 2014; pp. 119–131. [CrossRef]
16. Tomforde, S.; Prothmann, H.; Rochner, F.; Branke, J.; Hähner, J.; Müller-Schloer, C.; Schmeck, H. Decentralised Progressive Signal Systems for Organic Traffic Control. In Proceedings of the 2008 Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems, Venezia, Italy, 20–24 October 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 413–422. [CrossRef]
17. Quicktype GitHub. 2021. Available online: <https://github.com/quicktype/quicktype> (accessed on 6 May 2022).
18. Wu, J.; Liu, P.; Zhou, Y.; Yu, H. Stationary condition based performance analysis of the contraflow left-turn lane design considering the influence of the upstream intersection. *Transp. Res. Part C Emerg. Technol.* **2021**, *122*, 102919. [CrossRef]
19. Seidel, T.; Wickerath, J. *Rush Hours and Urbanization*; Center for Economic Studies: Munich, Germany, 2019.
20. Prothmann, H.; Branke, J.; Schmeck, H.; Tomforde, S.; Rochner, F.; Hähner, J.; Müller-Schloer, C. Organic traffic light control for urban road networks. *Int. J. Auton. Adapt. Commun. Syst.* **2009**, *2*, 203–225. [CrossRef]
21. Sommer, M.; Tomforde, S.; Hähner, J. An Organic Computing Approach to Resilient Traffic Management. In *Autonomic Road Transport Support Systems*; Birkhäuser: Basel, Switzerland, 2016; pp. 113–130.