

Cellular Automata Intersection Model

Tim Vranken · Michael Schreckenberg

University of Duisburg-Essen, Physics of Transport and Traffic, Duisburg, Germany,
E-mail: tim.vranken@uni-due.de

Received: 16 January 2020 / Last revision received: 10 March 2020 / Accepted: 19 March 2020
DOI: [10.17815/CD.2020.80](https://doi.org/10.17815/CD.2020.80)

Abstract This paper introduces a cellular automaton design of intersections and defines rules to model traffic flow through them so that urban traffic can be simulated. The model is able to simulate an intersection of up to four streets crossing. Each street can have a variable number of lanes. Furthermore, each lane can serve multiple purposes at the same time, like allowing vehicles to keep going straight or turn left and/or right. The model also allows the simulation of intersections with or without traffic lights and slip lanes. A comparison to multiple empirical intersection traffic data shows that the model is able to realistically reproduce traffic flow through an intersection. In particular, car following times in free flow and the required time value for drivers that turn within the intersection or go straight through it are reproduced. At the same time, important empirical jam characteristics are retained.

Keywords Celular automata · urban traffic · vehicles · microscopic · models

1 Introduction and related Work

Vehicular traffic in urban networks is known to be more complex than highway traffic because of additional influences like traffic lights, stopped vehicles, pedestrians or other influences [1–3]. The situation is further complicated by the possibility of vehicles not just to change lanes, accelerate or decelerate, but also turn directions at intersections. This creates an interaction with the vehicles of different driving directions. Drivers that intend to turn can be forced to decelerate because of crossing traffic which, combined with the intentional interruption through traffic lights, creates vehicle queues [4]. These queues, in turn, result in an uneven distribution of traffic over the system. Traffic can be saturated (or even over-saturated) in front of intersections and sparse behind them.

In the past years, many models have been presented with the aim to describe and analyse traffic in urban networks. Some of these models are highly complex and can simulate

whole cities like the Manhattan model [5], or Transims [6]. The downside of these models is that they are created and calibrated specifically for the cities they simulate and an adaption to different scenarios can be difficult. A more general model is called ARTIST [7], which allows to use different car-following models, analyse and compare them. The traffic model TRANSYT [8] can be used to analyse and then determine the optimum fixed-time traffic signal settings. The model by Zhao et al. [9] is specialised on the simulation and analysis of pedestrians crossing the street under different circumstances without a traffic light and through ongoing vehicular traffic flow. Lastly, the model INTERGRATION [10] can be used to simulate specific intersections based upon a general configuration, which is also the goal of this work but its calculation of velocities is only based upon a macroscopic statistic. The focus in the following work will be placed on cellular automata (CA) models like Brockfeld et al. inner-city model [11]. This model is based on the Nagel-Schreckenberg-model (NaSch-model) [12] which is known as one of the most simple CA-models for traffic flow. CA-models have the advantage that they can realistically describe traffic flow statistics on a microscopic level, while they only require the use of a small set of rules which facilitates an algorithmic implementation. Since the publication of Brockfeld et al. inner-city model [11], multiple other models have been presented. The most relevant of these models implement realistic rules for the motion of turning for either the NaSch-model [12] or the brake light model [13]. Examples of these urban models were presented in [14] and [3] where intersections are represented as additional cells after every street section. An agent is only allowed to drive into these additional cells if those are free of crossing flows. These models come with severe restrictions for network designs. The model in [3] for example is only able to reproduce one-lane traffic in front of intersections. In [14] multi-lane traffic can be reproduced but every lane leading into an intersection only serves one purpose (left-/right-turning or straight going traffic). In reality, multiple lanes with multiple functions are possible. A more complex simulation of intersections was presented in [15] or [16] which are able to reproduce traffic on multiple lanes where agents can turn or go straight. The downside to both models is the use of unrealistic high (infinite) braking capacities and turning conflict resolving through "first come first serve" or the introduction of virtual impingements between the conflict cells. Lastly, the model presented in [17] is able to reproduce multiple lanes from which agents can turn or go straight. Furthermore, in this model agents can only turn left if the next straight going vehicle is further away than 3.0 seconds based on its current velocity. Through this, the influence of left-turning agents on multi-function lanes can be analysed but the velocity of the turning agent itself is still not considered in the safety rule due to an infinite braking capacity.

Different from these models, this work aims to reproduce multi-lane traffic with a variable number of lanes going in and out. It further presents a left-turning rule, where the agent only turns if it does not interrupt the crossing straight going traffic flow. Lastly, the model aims to simulate realistic single-vehicle dynamics through a limited deceleration capacity.

Recently, a new CA-model by Lee et al. [18] (Lee-model) was introduced, in which agents can at most decelerate with a maximum of D cells per time step instead of an infinite deceleration potential like in the previous models. The model also introduces driver

behaviour through which agents are able to drive closer to their leading vehicle than the minimum safety distance, which can result in accidents. Through these two changes, the model is able to reproduce car following times and traffic behaviour in synchronized traffic [19] comparable to empirical findings. In [20] Pottmeier et al. (Pottmeier-model or Pottmeiers extension) made small adjustments to the Lee-model through which accidents are prevented, while still keeping the synchronized flow and realistic car following times. Pottmeiers extension also introduced lane-changing rules to simulate two-lane traffic flow. This paper now aims to adapt the Lee-model (with Pottmeiers adjustments) to urban traffic, in order to create an urban city model with a realistic car following times and possible synchronized traffic flow. Because of the finite braking capability, it is not always possible for agents to stop in front of the additional intersection cells if other agents just entered them. To adapt the Lee-model for urban traffic and to incorporate multiple lanes in front of an intersection — with possible multiple purposes—, a new intersection design is to be presented and additional rules are implemented.

To this end, first, we introduce the Lee-model and the modification made by Pottmeier are explained in the following Sec. 2. Afterwards, we expand the CA by a two-dimensional array in Sec. 3 to represent intersections and cells with important functions which are specified. Based on this expansion, new rules are introduced in Sec. 4, which guide vehicles past traffic lights and intersections in accordance with empirical data, so that urban traffic can be simulated. In Sec. 5, we analyse deadlock situations based on these rules and methods to prevent them are inserted in the model. Lastly in Sec. 6 we draw a conclusion and an outlook on future work is given.

2 Lee et al. model and Pottmeiers extension

Before we introduce the new intersection design and the rules to simulate agents driving through it, we will first shortly summarize the model by Lee et al. followed by Pottmeiers extension.

2.1 The model by Lee et al.

The CA-model introduced by Lee et. al. in 2003 [18] allows the reproduction of realistic car following times while also reproducing empirically found behaviour in synchronized flow. Furthermore, the model introduced a maximum deceleration of D because of which collision-free driving has to be realised through moderate driving instead of infinite braking capabilities. In order to prevent accidents, every agent has to calculate a safe speed c_n^{t+1} where the agent n stops behind the vehicle in front even if this vehicle suddenly starts braking. This is achieved by calculating c_n^{t+1} so that the following inequality is satisfied.

$$x_n^t + \Delta + \sum_{i=0}^{\tau_f(c_n^{t+1})} (c_n^{t+1} - D \cdot i) \leq x_{n+1}^t + \sum_{i=1}^{\tau_l(v_{n+1}^t)} (v_{n+1}^t - D \cdot i) \quad (1)$$

This inequality can be solved for c_n^{t+1} :

$$c_n^{t+1} \leq \frac{x_{n+1}^t - x_n^t - \Delta + v_{n+1}^t \cdot \tau_l(v_{n+1}^t)}{\tau_f(c_n^{t+1}) + 1} + \frac{\frac{D}{2}[\tau_f^2(c_n^{t+1}) + \tau_f(c_n^{t+1}) - \tau_l^2(v_{n+1}^t) - \tau_l(v_{n+1}^t)]}{\tau_f(c_n^{t+1}) + 1} \quad (2)$$

Here, x_n^t is the position of the n th vehicle at time step t and vehicle $n+1$ represents the vehicle ahead of the n th one. v_{n+1}^t is the velocity of the vehicle in front and $\tau_l(v_{n+1}^t)$, $\tau_f(c_n^{t+1})$ describe the number of time steps the respective vehicles need to decelerate until they stop. Δ is the minimal coordinate difference required between the two vehicles, so that no accidents occur (i.e. at least the size of the length of the vehicle L). Because drivers desire to drive as fast as possible, the maximum c_n^{t+1} ($\tilde{c}_n^t + 1$), which still satisfies equation Eq. 2, will be used. To reproduce synchronised traffic and low car following times in free flow, agents need different behaviour depending on the local traffic situation. To judge the situation, the agent not only looks at the preceding vehicle $n+1$ but also at $n+2$ and calculates a two-state variable γ_n^t based on their velocities. This way the agent judges the situation optimistically ($\gamma_n^t = 0$) when it is not to be expected that the vehicle in front will have to decelerate for any reason within the next time-step or defensive ($\gamma_n^t = 1$) if the agent assumes that the $n+1$ vehicle could decelerate. This judgement procedure is reflected by:

$$\gamma_n^t = \begin{cases} 0, & \text{for } v_n^t \leq v_{n+1}^t \leq v_{n+2}^t \text{ or } v_{n+2}^t \geq v_{\max}^S - 1 \\ 1, & \text{else} \end{cases} \quad (3)$$

Now, τ_f , τ_l , and Δ have to change depending on γ_n^t . If the driver judges the situation optimistically, Δ will only be the length of the preceding vehicle L while τ_f , τ_l will be lower than the complete time each agent needs to come to a stop. This means, that the agent does not look if it is possible to stop in time before creating an accident but only if an accident would be created in the next t_{safe} time steps. On the other hand, if the agent does judge the situation defensive ($\gamma_n^t = 1$), then τ_f , τ_l will be the complete time frame the vehicle needs to come to a standstill, while Δ will be increased with an additional safety gap g_{add} . The three variables will be calculated according to:

$$\Delta = L + \gamma_n^t \max(0, \min(g_{\text{safe}}, v_n^t - g_{\text{safe}})) \quad (4)$$

$$\tau_f(v) = \gamma_n^t \frac{v}{D} + (1 - \gamma_n^t) \max(0, \min(\frac{v}{D}, t_{\text{safe}}) - 1) \quad (5)$$

$$\tau_l(v) = \gamma_n^t \frac{v}{D} + (1 - \gamma_n^t) \min(\frac{v}{D}, t_{\text{safe}}) \quad (6)$$

The -1 in Eq. 5 the calculation for τ_f accounts for the one second reaction time a driver needs on average. $\tilde{c}_n^t + 1$ has to be calculated numerically every time step in such a way that it also satisfies these three equations. The velocity calculation and movement of one

agent can then be calculated according to:

$$p = \max(p_d, p_0 - \frac{v_n^t(p_0 - p_d)}{v_{\text{slow}}}) \quad (7)$$

$$\tilde{c}_n^t + 1 = \max(c_n^{t+1} \mid c_n^{t+1} \text{ satisfies Eq. 1-6}) \quad (8)$$

$$\tilde{v}_n^{t+1} = \min(v_{\text{min}}, v_n^t + a, \max(0, v_n^t - D, \tilde{c}_n^t + 1)) \quad (9)$$

$$v_n^{t+1} = \max(0, v_n^t - D, \tilde{v}_n^{t+1} - \eta) \quad (10)$$

$$x_n^{t+1} = x_n^t + v_n^{t+1} \quad (11)$$

with a random delay according to

$$\eta = \begin{cases} 1, & \text{if } q < p \\ 0, & \text{else} \end{cases} \quad (12)$$

Here, q is a random value between zero and one. Eq. 7 ensures that stopped vehicles dawdle with a higher probability than moving vehicle (slow to start rule [21]). The deceleration restriction is ensured in Eq. 9 and Eq. 10. This means that changes done to c_n^{t+1} later on in Sec. 4 and Sec. 5 will not break the deceleration restriction. Lee et. al. found the following parameter choices to be the best fit on empirical data: $a = 1, D = 2, L = 5, v_{\text{fast}} = 19, t_{\text{safe}} = 3, g_{\text{add}} = 4, p_0 = 0.32, p_d = 0.11, v_{\text{slow}} = 5, v_{\text{max}} = 20$ as well as the length of a time step $\Delta t = 1$ s and the cell length of $\Delta x = 1.5$ m.

2.2 Pottmeiers extension

Pottmeier et al. [20] provide an extension to the Lee model that implements accident-free multi-lane traffic into the model.

2.2.1 Accident-free traffic

To prevent accidents from occurring, Pottmeier et al. [20] introduced the brake light and slightly changed Eq. 3. The brake light d_n^t is a boolean variable which shows whether the leading vehicle decelerated because of traffic restrictions or it just dawdled. It is determined by

$$b_n^t = \begin{cases} 1, & \text{for } \tilde{v}_n^{t+1} < v_n^t, \\ 0, & \text{else} \end{cases} \quad (13)$$

A check for the brake light of the leading $n + 2$ vehicle, as well as harder restrictions for the optimistic state, change Eq. 3 to the following equation.

$$\gamma_n^t = \begin{cases} 0, & \text{for } b_{n+2}^t = 0 \text{ and } (v_n^t \leq v_{n+1}^t < v_{n+2}^t \text{ or} \\ & (v_{n+2}^t \geq v_{\text{max}}^S - 1 \text{ and } v_n^t - v_{n+1}^t \leq D)) \\ 1, & \text{else} \end{cases} \quad (14)$$

Due to the brake light, the change from $v_{n+1}^t \leq v_{n+2}^t$ to $v_{n+1}^t < v_{n+2}^t$, and the addition of $v_n^t - v_{n+1}^t \leq D$ the behaviour was changed enough, so that no traffic accidents were registered in 20 simulations over 60,000 time steps for different densities between 10 and $100 \frac{\text{cars}}{\text{km}}$. This means that even if accidents are possible, they are so unlikely that they are statically not relevant.

2.2.2 Multi lane traffic model

The next step to simulate traffic is to add lane-changing rules, so that multi lane traffic can be added. Traffic rules for asymmetrical lane changes were implemented and analysed [22] for highway traffic. Because in urban traffic lane changes are often dominated by restrictions from turning or stopping vehicles, the following work will only use symmetrical lane changing rules as implemented in [23]. Originally, lane changes have to be desired and possible for $t_{lc} = 3$ seconds before a lane is changed, to ensure save lane changes and adapt the empirical finding of an average lane changing time of 3 seconds [24]. In the following, roads with more than two lanes will be simulated. On these roads, it can happen that two agents want to change lane into the same gap. For example on a three-lane street, one agent from the most inner lane (right in driving direction) can change to the lane in the middle at the same time as one from the outermost lane (left in driving direction). Because even with $t_{lc} = 3$ the lane change only takes one second at the last time step, this can not be prevented. Hence, $t_{lc} = 1$ will be used in the following simulations and agents are only allowed to change to a lane to their right (in driving direction) on even ($t \bmod 2 = 0$) and to their left during uneven time steps ($t \bmod 2 \neq 0$). Since this change is corresponding to changing $t_{lc} = 2$, it is to be expected that this results in a slightly higher traffic flow and number of lane changes as shown in [23].

Before this is considered, it is tested whether the agent wants to change the lane. If a lane change would not allow the agent to accelerate more than on his current lane, or if the agent currently judges the situation positive, a lane change is not initiated. To formulate the conditions for this, first, l is defined as an index to indicate the lane of the agent and $1 - l$ indicates the lane the agent wants to change to. The second condition can then be written as $c_{n,l-1}^{t+1} > c_{n,l}^{t+1}$. If the three conditions are met, the agent tests whether a lane change is possible. A lane change is safe, if the vehicle on the targeted lane in front $x_{F(1-l,n)}^t$ as well as behind $x_{B(1-l,n)}^t$ is further away than g_{safe} . Furthermore, the velocity that corresponds to the respective safe velocity on the targeted lane $c_{n,l-1}^{t+1}$ should not be smaller than $v_n^t - D$ and the velocity of the vehicle behind the agent $v_{B(1-l,n)}^t - \beta$ should be lower than $c_{n,l-1}^{t+1}$, so that both vehicles can decelerate enough. Here β is a constant between 0 and D which controls how strong an agent is allowed to thwart the agent behind him for a lane change. According to [24] thwarting the agent behind for a lane change is fine, as long as the agent behind has to decelerate not more than around $-1.25 \frac{\text{m}}{\text{s}^2}$ (which is roughly the engine brake). Because of this, we use $\beta = 1 \frac{\text{cells}}{\text{time step}^2} = 1.5 \frac{\text{m}}{\text{s}^2}$ for the following work. Summarized, an agent changes its lane if all of the following condition

hold true at once:

$$(t \bmod 2 = 0 \text{ \& lane change to the right}) \text{ or } (t \bmod 2 \neq 0 \text{ \& lane change to the left}) \quad (15)$$

$$v_{n,l}^t < v_{\max} \quad (16)$$

$$c_{n,l-1}^{t+1} > c_{n,l}^{t+1} \quad (17)$$

$$\gamma_n^t = 1 \quad (18)$$

$$x_n^t - x_{B(1-l(n))}^t > L + g_{\text{safe}} \quad (19)$$

$$x_{F(1-l,n)}^t - x_n^t > L + g_{\text{safe}} \quad (20)$$

$$v_n^t - D \leq \tilde{c}_{n,1-l_n}^{t+1} \quad (21)$$

$$v_{B(1-l,n)}^t - \beta < c_{n,l-1}^{t+1} \quad (22)$$

With these rules, accident-free multi-lane traffic can be simulated. To also simulate urban traffic, intersections have to be introduced and necessary rules have to be added to the model.

3 Intersection design

In urban networks, more complex approaches are required to implement an intersection design that reflects real traffic dynamics. While the traffic flow of a road mostly depends on the speed limit and number of lanes, traffic in and in front of intersections depends on a number of different conditions like, how many streets are crossing, if it is allowed to turn left/right, or if pedestrians cross the street while vehicles try to turn. The high variety in designs of intersections makes their adaptation into CA-models challenging. For this reason, simplifications have been identified, that aim to describe the traffic behaviour in intersections without the need to restrict the design to a specific topology. The most common example of this is found in the previously mentioned [14] and [3], where intersections were replaced by an extension of every street. Vehicles are then not allowed to enter these extensions when a vehicle is currently driving on a corresponding extension of a crossing street. A problem of these designs is that the model presented in [3] only works with one lane, while [14] may use multiple lanes, but each lane only allows for one purpose (e.g. only turning left/right or going straight ahead). It is also not possible for a vehicle to stop before the extension, if the crossing extension is occupied, because of the maximum deceleration $D = 2$ in the Lee-model. Due to these limitations, a two-dimensional intersection is presented in Fig. 1 which aims to overcome both problems. Note that this is a rather complex example intersection to show as many functions as possible at once. A vehicle that leaves a street simply moves according to its previous direction through this array. When they meet a crossing point (marked with a 1 for left- or 2 for right turning) and they desire to turn, they check whether turning is possible for them at this cell, or if the crossing point is for a different direction (for eastbound traffic while the agent is driving north for example). If it is desired and possible, it is tested if

the turn can be done without forcing crossing vehicles to decelerate, according to the rule explained in the following Sec. 4. When all conditions are met, the vehicle will move to the next cell in turning direction (inertia will be considered separately in section Sec. 4.2).

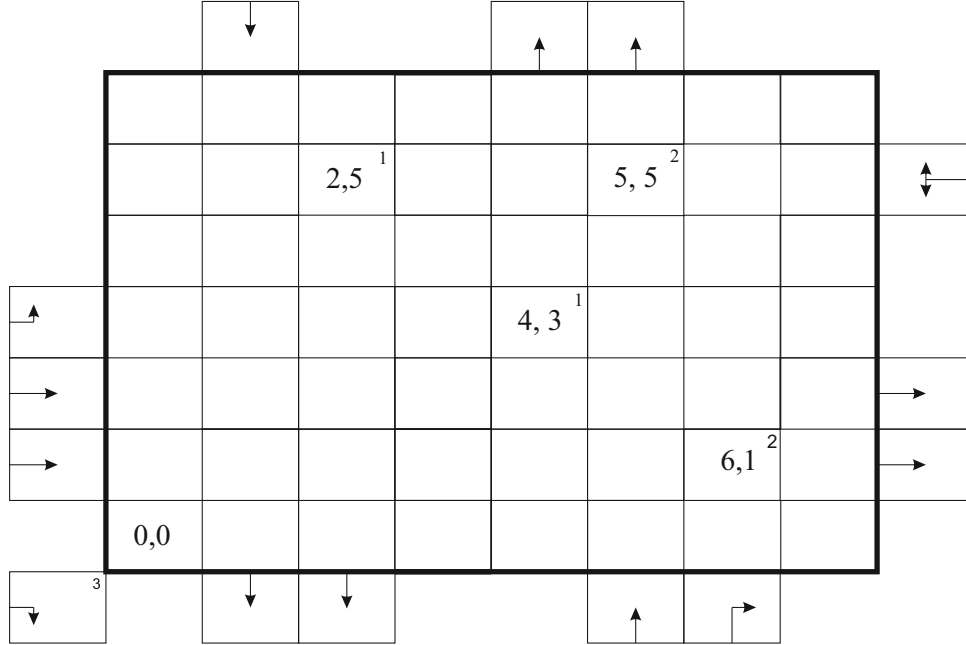


Figure 1 A two dimensional intersection with multiple roads connecting to it. The arrows show the functions of each lane while the bigger numbers represent the x, y- coordinate of the specific cell. The smaller numbers indicate specific turning function t_c of the cell c . Cells with $t_c = 1$ allow vehicles to turn to their left (in driving direction), while $t_c = 2$ does the same for the right. $t_c = 3$ allow vehicles to turn right outside of the intersection. All other cells have the value $t_c = 0$ which is not shown.

The intersection in Fig. 1 shows multiple different functions of lanes in front of it. The vehicle, which comes from the west (so east-bound traffic), have four lanes to use. The most inner (right in driving direction) lane is a representation of a slip lane, where vehicles can turn right without driving through the intersection (often also without a traffic light). The next two lanes only allow for straight going traffic flow and vehicles on the last lane have to turn left at the cell (4, 3). On the other hand, the inner lane on the north-bound traffic allows vehicles to only turn right at the cell (6, 1). The west-bound vehicles can turn either right or left depending on their route and vehicles driving south can only keep driving south even though turning left would be possible at the cell (1, 1) or (1, 2) but they both have $t_c = 0$ so turning is forbidden there. Like this, an intersection with up to four roads can be modelled. Agents now only need to know if they are going east, west, north, or south. If they are going north, their next cell is in decreasing y-direction, south-bound traffic is going in increasing y-direction and west/east going traffic is going in the increasing/decreasing x-direction respectively.

Each intersection can now be designed with the knowledge of the number of lanes n_{in}^d coming from $d \in [e, w, n, s]$ (east, west, north or south) as well as the number of lanes

Table 1 (x , y) coordinates of the cells where each direction connects with the intersection.

•	ingoing roads	outgoing roads
east	(0 , y)	($x_{\max}$, $y + r^e$)
west	(0 , $y_{\max} - y$)	($x_{\max}$, $y_{\max} - (y + r^w)$)
north	($x_{\max} - y$, 0)	($x_{\max} - (y + r^n)$, $y_{\max}$)
south	(y , $y_{\max}$)	($y + r^s$, 0)

leaving in each direction n_{out}^d in three steps. Note that slip lanes are not counted here. First necessary length in the x-direction is then calculated by:

$$x_{\max} = \max(n_{\text{in}}^n, n_{\text{out}}^n + r^n) + \max(n_{\text{in}}^s, n_{\text{out}}^s + r^s) + 3 \quad (23)$$

First, the higher number of lanes from north- and south-bound traffic, which lead into the intersection or out of it, is found. Depending on the higher number of lanes, the intersection is expanded. The parameter r^d counts the number of lanes of ingoing roads, which are no slip lanes, from which agents are only allowed to turn right. An example of this is the first lane on the ingoing road for north-bound traffic. Because agents are only allowed to turn right from the first lane, the outgoing lanes have to start one cell earlier (so with a lower x coordinate). This leads to a larger array, even though the ingoing and outgoing roads for north-bound traffic both only have two lanes. Finally, 3 additional cells in length are added. One cell is placed between the two opposite going directions (in this case the cells with an x-coordinate of 3), one at the beginning (x-coordinate of 0) and one at the end (x-coordinate of 6). The central cells are used to physically separate the opposite going flows. The additional cells at the beginning and end of the array are implemented because in reality most of the times there are also pedestrians crossings between the stopping point for vehicles and the intersection. In the same way, the y-length can be calculated by:

$$y_{\max} = \max(n_{\text{in}}^e, n_{\text{out}}^e + r^e) + \max(n_{\text{in}}^w, n_{\text{out}}^w + r^w) + 3 \quad (24)$$

In the second step, the last cell of each lane y of an ingoing or outgoing road will be connected to the respective cell in the intersection array. In Tab. 1 the (x , y) coordinates of each cell for ingoing and outgoing roads in each direction is defined, in dependence on the previous calculated values ($x_{\max}$, $y_{\max}$) and the current y-coordinate of the lane. Because slip lanes are still not taken into account here, we only count from the first lane leading into the intersection.

Finally, in the third step, the t_c values for turning points have to be identified. For that, the purposes of each lane in front of an intersection (e.g does it allow turning in one or multiple directions and if yes in which) has to be specified as a characteristic of the lane. Then the t_c of the last cell of every slip lane is changed to 3. To calculate the left- and right-turning cells within the intersection, for each direction d the maximal number of lanes, from which a turn right $t_{r,\max}^d$ or left $t_{l,\max}^d$ is allowed, has to be identified. On a

Table 2 (x, y) coordinates (written in vector notation) of the turning cells for left turning and right turning in dependence of the original driving direction.

•	left turning ($t_c = 1$)	right turning ($t_c = 2$)
east	$\begin{pmatrix} 2 + \max(n_{\text{in}}^s, n_{\text{out}}^s + r^s) + t_{l,\text{max}}^e - t_l^e \\ n_{\text{in}}^e + t_l^e - t_{l,\text{max}}^e \end{pmatrix}$	$\begin{pmatrix} t_r^e + r^s \\ t_r^e \end{pmatrix}$
west	$\begin{pmatrix} n_{\text{out}}^s + r^s + t_l^w - t_{l,\text{max}}^w \\ 2 + \max(n_{\text{in}}^e, n_{\text{out}}^e + r^e) - t_l^w + t_{l,\text{max}}^w \end{pmatrix}$	$\begin{pmatrix} x_{\text{max}} - (t_l^w + r^n) \\ y_{\text{max}} - t_l^w \end{pmatrix}$
north	$\begin{pmatrix} x_{\text{max}} - n_{\text{in}}^n + t_{l,\text{max}}^n - t_l^n \\ \max(n_{\text{in}}^e, n_{\text{out}}^e + r^e) + 2 - t_{l,\text{max}}^n - t_l^n \end{pmatrix}$	$\begin{pmatrix} x_{\text{max}} - t_r^n \\ t_r^n + r^e \end{pmatrix}$
south	$\begin{pmatrix} n_{\text{in}}^s + t_{l,\text{max}}^s - t_l^s \\ n_{\text{out}}^s + r^s + t_l^s - t_{l,\text{max}}^s \end{pmatrix}$	$\begin{pmatrix} t_r^s \\ y_{\text{max}} - (t_r^s + r^w) \end{pmatrix}$

road where multiple lanes allow turning into the same direction, each lane gets a number depending on its order in the driving direction, where we assign a 1 to the most inner (right) lane and the maximum occurring value to the outermost (left) lane respectively. For example, we consider a road that has three lanes from which agents are allowed to turn left. The most inner lane of these left turning lanes corresponds to $t_l^d = 1$ while the next one is $t_l^d = 2$ and the last one $t_l^d = 3$. The (x, y)-coordinates for each turning position can then be calculated according to Tab. 2. Because of the rather complex form of the left turning coordinates, they are written in $\vec{r} = \begin{pmatrix} x \\ y \end{pmatrix}$ notation. After explaining how vehicles move through an intersection, in the next section, the rules for the behaviour of agents will be defined, so that they move without accidents and according to empirical values.

In an additional step, the last cell of all slip lanes are connected to the first cell of the respective outgoing road to their right and the t_c value is changed to 3.

4 Intersection behaviour

Before the characteristic traffic behaviour within intersections is described, we model the traffic in front of it. It should be noted that traffic in urban areas, in general, is slower than on highways. For highway traffic, the Lee-model is implemented with a standard value of $v_{\text{max}} = 20 \frac{\text{cells}}{\text{time step}} \approx 100 \frac{\text{km}}{\text{h}}$ while in urban areas $v_{\text{max}} \approx 10 \frac{\text{cells}}{\text{time step}} \approx 50 \frac{\text{km}}{\text{h}}$ is more appropriate and will be used throughout the rest of this work.

4.1 Behaviour in front of traffic lights

To avoid accidents in front of a traffic light, the situation assessment of agents has to be modified first. Agents close to a traffic light $T_{S,y}^t$ on their street S and lane y , which is red ($T_{S,y}^t = R$) or yellow ($T_{S,y}^t = Y$) in this time step, should not judge a situation optimistic. In these cases, it is to be expected that the vehicles in front on their lane y will stop

soon. As long as the traffic light is not green ($T_{S,y}^t \neq G$), the vehicle should judge the situation defensively $\gamma_n^t = 1$. This only applies if a vehicle is close enough to a red light to see it. According to [13] a driver only considers a temporal headway h to judge the situation. The agent will only consider the vehicle in front if the distance to the respective vehicle is smaller than the distance it will pass in the next h seconds, based on its current speed. Because most traffic lights are not only fixed to the side of the road but also raised above the height of a normal vehicle, agents are able to perceive red traffic lights which are further away. A traffic light is taken into account as long as the agent needs less than $2 \cdot h$ time steps to reach it. As pointed out in [13], this temporal headway is judged to be $6 \leq h \leq 11$ s depending on the velocity of the vehicle, the driver type, and the surrounding situation. Because vision is usually strongly restricted in urban areas, we consider the most limited case of $h = 6$ s which means only traffic lights with a distance of $d_{n,tl}^t = 2 \cdot 6 = 120$ cells or less are considered. This changes the behaviour of agents to

$$\gamma_n^t = \begin{cases} 0, & \text{for } (b_{n+2}^t = 0 \text{ or } (T_{S,y}^t \neq G \text{ and } d_{n,tl}^t \leq d_{n,tl}^t)) \text{ and} \\ & (v_n^t \leq v_{n+1}^t < v_{n+2}^t \text{ or } (v_{n+2}^2 \geq v_{\max}^S - 1 \text{ and } v_n^t - v_{n+1}^t \leq D)) \\ 1, & \text{else} \end{cases} \quad (25)$$

Next, it has to be ensured that the vehicle is able to stop before the traffic light if necessary. For that, it is not enough to check the $n + 1$ st vehicle and keep a safe distance to it because the vehicle ahead could still be able to pass the traffic light while the n th agent has to stop. Instead, the brake distance also has to be calculated. As stated in [25] the stopping distance $\delta(c_n^{t+1})$ for a discrete system can be calculated according to

$$\delta(c_n^{t+1}) = D \left[\alpha_n \beta_n + \frac{\alpha_n(\alpha_n - 1)}{2} \right] \quad (26)$$

with α_n, β_n defined as the integer and fractional part of c_n^{t+1}/D . If this stopping distance together with the position the vehicle will have in the next time step x_n^{t+1} is greater than the length L_S of the street S the vehicle is currently driving, the vehicle would not be able to stop before it reaches the red light from the next time step on:

$$\delta(v_n) + x_n^{t+1} > L_S \quad (27)$$

To prevent this, c_n^{t+1} has to be reduced according to:

$$c_n^{t+1} = \max(v_n^t - \beta_2, 0) \quad (28)$$

where

$$\beta_2 = \begin{cases} 2, & \text{if } x_n^{t+1} \leq L_S - 2 \\ 1, & \text{else} \end{cases} \quad (29)$$

so that the agent drives up to L_S and stops there. If the traffic light is not red but yellow and Eq. 27 does not hold true, the vehicle keeps driving without braking. When the traffic light is green, the agent can drive freely without additional restraints. In this case, [14]

showed that the average vehicle should pass a traffic light $t_B = 2 \pm 0.5$ s (time requirement) after the previous vehicle. In [26] Krbálek et al. presented empirical data for the time requirement (clearance time) of multi-lane traffic with no influence from other cars moving in different lanes or arms. There, they observed a time requirement of around 1.6 ± 0.6 s. To be able to replicate both these values, the model will aim to produce a value $t_B \in [1.5, 2.2]$ s. In fact, because urban traffic is often dominated by traffic light controlled intersections, the reproduction of this time requirement takes priority over the highway traffic jam values. It was found, that the outflow of a jam is on average around $F_{\text{out}} \approx 1800 \frac{\text{vehicles}}{\text{h}}$ [27] and a jam propagates with a velocity around $v_{\text{jam}} \approx -15 \frac{\text{km}}{\text{h}}$ [28, 29] against the traffic flow. To calibrate the traffic in such a way, p_0 , p_d and v_{slow} have to be adjusted accordingly. It usually takes around $\delta t \approx 3 - 4$ [30] seconds before this time requirement is fulfilled. As stated in [1], this δt can be calculated according to:

$$\delta t = \Delta t - n_{\Delta t} t_B \quad (30)$$

Here, $n_{\Delta t}$ denotes the number of agents which pass the traffic light within Δt . To calculate δt in dependence of p_0 , p_d , and v_{slow} traffic flow interrupted by a traffic light was simulated. In this simulation, it was ensured that the number of vehicles queuing in front of the traffic light is always higher than the number of vehicles that pass it during one green phase.

Note that in case of periodic boundary conditions (as used for free flow simulations) the traffic light would result in a high density of vehicles before the traffic light. This, in turn, would result in a very low density behind the traffic light which would reduce the inflow into the traffic jam before the traffic light after a few circles. Due to this, simulations in this work that include traffic lights were done with open boundary conditions. There, the traffic inflow was set to a value significantly greater than the capacity of the traffic light and vehicles that passed the traffic light or intersection were taken out of the system after it is ensured that they are not relevant for the vehicle-vehicle interaction in or before the intersection.

For each vehicle, we then calculated the required time t_B it needs to pass the traffic light after the vehicle in front of it has already passed it. We obtain ten million time values by running a simulation for 100 hours, corresponding to 360,000 time steps. This results in significant average values for δt as shown in Fig. 2. It becomes apparent that δt is always greater than 4 time steps, independent of the choice of p_0 , p_d , v_{slow} . This can be explained by the empirical findings presented in [31]. The authors found that drivers do not always start driving after the vehicle in front accelerated but sometimes they accelerate at the same time as the vehicle in front (or even before them and topping again afterwards). Here, the driver close to the front of the queue tends to rather react to the traffic light change than to the vehicle in front. This results in a shorter acceleration time and has to be taken into account in this model. In real intersection traffic, a vehicle that accelerates earlier than the preceding vehicle (because it is dawdling) can simply stop again after a few centimetres. Because of the discretization of time and space, this is not possible within simulations and each vehicle can only accelerate after the vehicle ahead started. Since taking this into account would require an unreasonably fine discretization

of space, this is not feasible with the priority of computational speed and instead the maximal acceleration a has to be increased temporarily in urban traffic.

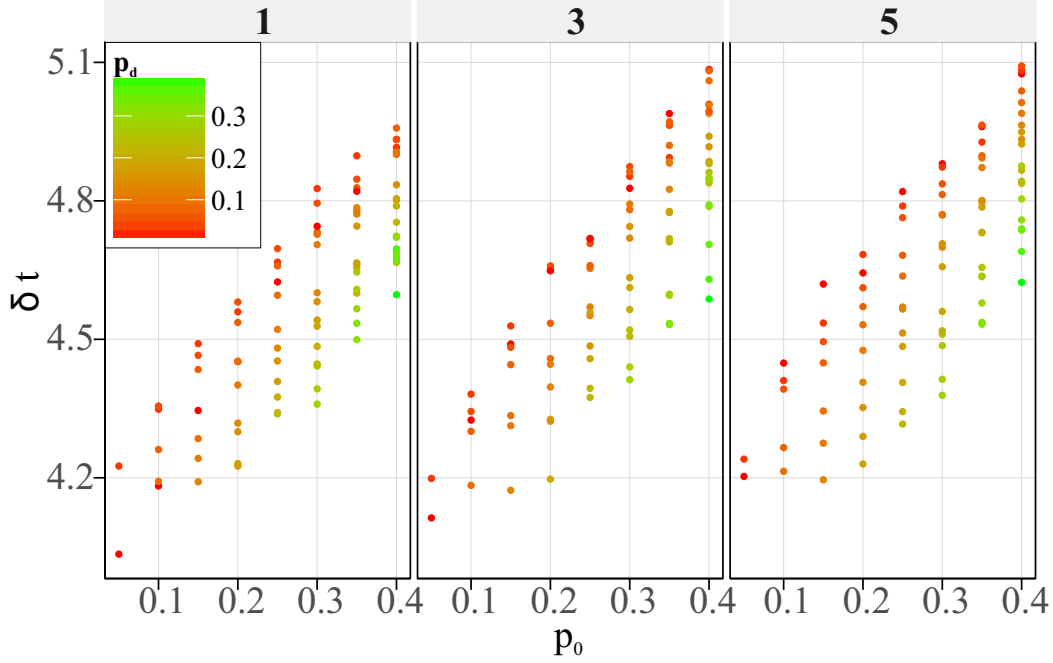


Figure 2 The time it takes for traffic flow to become stable after the traffic light turns green δt . The three figures correspond to the different v_{slow} 1, 2, and 3. For each v_{slow} p_0 and p_d were changed in different simulation runs. Note that it was ensured that $p_0 > p_d$ allways holds true.

In [19] Kerner et. al. faced the same problem and also concluded that a has to be temporally increased. In accordance with their work, an additional step will be added each time step to calculate the acceleration as following:

$$a_n^{t+1} = \begin{cases} k \cdot a, & \text{if } v_{n+1}^t - v_n^t + a_{n+1}^t \tau \geq \Delta v_a \\ a, & \text{else} \end{cases} \quad (31)$$

Here, k , τ , Δv_a are parameters that have to be calibrated. The parameter k defines how many times faster the agent is allowed to accelerate. A greater τ considers longer times the agent in front accelerated before agent n and Δv_a is a safety parameter. This means that to reproduce the two empirical values $\delta t \approx 3 - 4$ s and $t_B \in [1.5, 2.2]$ s, six parameters k , τ , Δv_a , p_0 , p_d , v_{slow} have to be defined. This can be reduced by [31], where the authors find an empirical maximal acceleration in front of a traffic light at around $6 \frac{\text{miles}}{\text{h} \cdot \text{s}} \approx 2,7 \frac{\text{m}}{\text{s}^2} \approx 2 \frac{\text{cells}}{\text{time step}}$ so that $k = \Delta v_a = 2 \frac{\text{cells}}{\text{time step}}$ can be applied in accordance with empirical findings. The increased a is supposed to reproduce faster reaction times and $\tau > 1$ would be relevant if the agent dawdled the time step instead of reacting fast. Together with the fact that there are more parameters to calibrate than empirical values that can be used to estimate them by empirical fits (which means there is a freedom of choice), we use $\tau = 1$, so that only agents that drive directly behind the vehicle ahead are effected. To calibrate

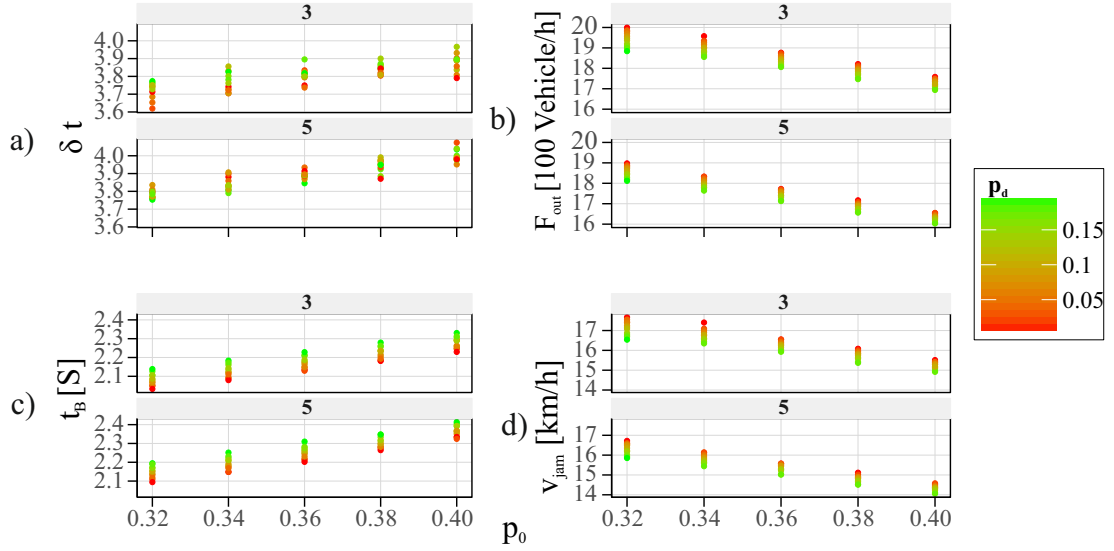


Figure 3 Different δt , t_B , v_{jam} , F_{out} in dependence of p_0 , p_d for the two different $v_{\text{slow}} = 3$ and $v_{\text{slow}} = 5$ in figure a) - d) respectively.

the model in such a way that the four values t_B , Δt , v_{jam} , and F_{out} are in accordance with empirical findings, 100 hours (or 360000 time steps) were calculated in a simulation for each set of parameters (p_0 , p_d , v_{slow}). In this simulation, a street is interrupted by a traffic light (with $G = 30$, $Y = 5$, $R = 65$ time steps) and the traffic flow was set high enough so that the traffic is permanently in over-saturated flow [1]. In each simulation for every agent that passed the green light, the time step in which it passed the traffic light was noted. From the time step difference and the knowledge, when a traffic light turns green and red, t_B can be calculated. Furthermore, the number of cars that pass in each green phase can be counted and δt calculated based on these two values. The corresponding results are displayed in Fig. 3 a) and c). To create b) and d) in Fig. 3 vehicles were simulated, driving within a 15 km long circle for 60,000 time steps with a set density. This simulation was repeated using different densities and the average velocity for all vehicles in the system was taken in the second half. Out of the density and velocity, the fundamental diagram was reproduced. The value for v_{jam} can then be obtained by calculating the average gradient of the fundamental diagram in the jammed traffic phase. Retracting the graph with this gradient up to the point of the highest flow results in the value for F_{out} .

As we can see, the values for t_B , v_{jam} , F_{out} follow the expected curve, where higher p_0 , p_d result in lower v_{jam} , F_{out} or higher t_B . δt also increases with higher p_0 as it is to be expected because a higher probability to dawdle, if a vehicle stands still, means it takes longer for a sufficient number of vehicles to start so that the flow gets stable. This also means that vehicles that already started to drive do not have as much influence on δt because if the vehicle $n - 1$ behind the agent n dawdled (i. e. does not start) it does not matter whether agent n dawns or not. It only matters, if $n - 1$ does not dally. This can also be seen in Fig. 3 a) where δt does not depend on p_d . All values for δt and t_B are

within the empirically found intervals, while they vary strongly for F_{out} and v_{jam} where no combination of values satisfies both perfectly. For $p_0 = 0.34$, $p_d = 0.14$, $v_{\text{slow}} = 5$ we found $v_{\text{jam}} \approx -15.5 \frac{\text{km}}{\text{h}}$ and $F_{\text{out}} \approx 1770 \frac{\text{vehicles}}{\text{h}}$, such that both values are in good agreement with the empirical values. They are even closer than in the original configuration of the Lee-model where $p_0 = 0.32$, $p_d = 0.11$, $v_{\text{slow}} = 5$ are used, which resulted in $v_{\text{jam}} \approx -14 \frac{\text{km}}{\text{h}}$ and $F_{\text{out}} = 1900 \frac{\text{vehicles}}{\text{h}}$ [23]. $t_B = 2.23 \pm 0.04 \text{ s}$ and $\delta t = 3.7 \pm 0.1 \text{ s}$ are (including their error) also in accordance with the empirical found values of $t_B \in [1.5, 2.2]$ and $\delta t = 3 - 4 \text{ seconds}$. [26] not only presented a mean empirical value for the time requirement but also a distribution of the scaled time requirement. Due to the discretization of time into 1 s intervals, the model is only able to reproduce time requirement values of 1, 2, or 3 s and so can not recreate the distribution realistically but only the mean value. Looking at t_B one can note that we choose values for p_0 , p_d , and v_{slow} which correspond to the upper limit of the empirical values of t_B . Since lower values of p_0 , p_d , and v_{slow} also reflect realistic values of t_B , it would be conceivable to choose these values according to a distribution. Every vehicle which is placed into the system would then receive a certain value in the first time step according to these distributions with a certain probability. The influence of different distributions on the vehicular traffic and the required time value at traffic lights will be studied in a future work.

The additional values for g_{safe} , t_{safe} also have to be calibrated. t_{safe} controls how many seconds ahead in the calculation of c_n^{t+1} are considered. A higher value of t_{safe} results in safer driving but also in greater distances between the vehicles in free flow and so in higher car following times. Because the car following times should remain below one second in free flow [32], $t_{\text{safe}} = 3$ is kept in accordance with the Lee-model. In [23] the influence of g_{safe} was analysed and it was shown that a higher value reduces the influence of dawdling and wrong driver behaviour. This results in a more stable synchronized traffic phase, but also reduces the maximal possible traffic flow.

The influence and existence of the synchronized traffic phase for urban traffic were analyzed in [1]. It was shown that the synchronized phase is still important in urban traffic in the case of green waves (when the traffic light at the next traffic light is set in such a way that it is green when a vehicle arrives). In these cases, it can happen that the first vehicle is within the interaction range of the next traffic light before it turns green. Then it still has to decelerate for several time steps before the traffic light turns green. This creates a local transition from free flow into the synchronized flow. The moment a traffic light turns green and the vehicle starts to drive again, the traffic phase changes from wide moving jam to free flow ($J \rightarrow F$) without a transition into the synchronized phase. This means that neither g_{safe} nor t_{safe} have a significant influence on δt and t_B . Because of this, $g_{\text{safe}} = 4$ can also be retained.

Lastly, agents should only be allowed to drive into intersections when they would leave from the right lane. As shown in figure Fig. 1, different lanes in front of intersections can have different functions and so agents have to enter the intersection from the correct lane in accordance with their intention in the intersection (left / right turning or driving straight). If the agent leaves from the wrong lane, it has to be prevented just like with a red light by rules Eq. 26 - Eq. 28. Furthermore, it has to be ensured that the agent also changes to the correct lane, at best before he reaches the intersection and has to brake.

This will be ensured by additional lane changing rules.

4.1.1 Lane changes before traffic lights

The agents should aim to be on the correct lane so that they can pass into the intersection without braking when the traffic light is green. This means that once the traffic light comes into vision (120 cells) the rules to check if the agent wants to change lanes (equations Eq. 16 and Eq. 17) will be replaced by a check whether the agent is on the correct lane and/or if the lane change would be in the direction of the right lane. To this extent, each lane gets one or multiple purposes selected from either straight, left or right and every agent is assigned a route which contains one of these functions for the specific street. Then, the agent can turn into the direction he intends to by making use of the order of the roads. To get towards right-turning lanes, one always has to do lane changes towards a more inner lane (to their right in driving direction) and to get towards left turning lanes they have to do a lane change towards a more outer lane (to their left in driving direction). To get to a straight going lane, the agent has to change towards a more outer lane if he is on a right turning lane and towards a more inner lane if he is on a left turning lane. In [33] driver change lanes more aggressively in front of intersections to still get to their desired lanes. This means that if equations Eq. 19 - Eq. 20 do not hold true but

$$x_n^t - x_{B(1-l(n))}^t > L \ \& \ v_{B(1-l(n))}^t - 2 < v_n^t \quad (32)$$

$$x_{F(1-l,n)}^t - x_n^t > L \ \& \ v_n^t - 2 < v_{F(1-l(n))}^t \quad (33)$$

does, the agent still changes the lane. Furthermore, β in Eq. 22 is changed to 2, to allow the agent to slow down the following vehicle stronger.

With this, the agent will change to the right lane before leaving the street if possible and if not hold. In reality, it is often observed that drivers who can not change to the correct lane in front of a traffic light do not hold at the holding line but earlier. We implement this empirical observation in our model by the vehicles that can not change to the correct lane starting to brake from the next time step on until the vehicle stops. Then, the agent does not accelerate until the lane was changed. Furthermore, in [34] it is described that drivers in the targeted lane can act with a so-called "courtesy", which means that they will actively change their driving behaviour in order to create a gap in front of them, in which the agent who intends to change lane can merge. To this end, we introduce a turning signal. As long as the turning signal $B_n^t = 1$ is active, the agent brakes and informs the subsequently following agent $n_{B(1-l(n))} - 1$ on the targeted lane, which is more than L cells on its targeted lane behind. This agent can then decide, based on the velocity difference to agent n and the driving behaviour, to decelerate to create a gap. If the agent is currently standing still (waiting in queue) it is also possible for the agent not to accelerate when the leading vehicle started, to let the agent n merge in front. For all following simulations in this paper, all agents will make use of this courtesy. In future simulations, only parts of agents could make use of it while others don't allow agents to merge before them. By drawing such exceptions from a suitable probability distribution one could test the impact of partial courtesy on traffic flow and travel time.

With this, vehicles going straight through an intersection without turning can now be fully simulated but for agents who attempt to turn, additional rules have to be defined.

4.2 Turning rules

For vehicles that want to turn, two points have to be considered. First, in reality, vehicles that are turning or will turn soon drive slower than vehicles that just keep driving straight. With lower speed, the drivers can better react to intercepting traffic and pedestrians that are also crossing the street. It also makes driving more comfortable because turning with higher speed results in higher forces affecting all passengers. Furthermore, drivers who intend to turn left have to ensure that they do not interrupt crossing traffic. In the study of Li et al. [2] left turning traffic in an intersection with an arrow traffic signal was analyzed. There it was shown that even when there is no other interaction (no pedestrians or crossing traffic) turning vehicles on average only drive at most $v_{\max, \text{turner}} \approx 40 \frac{\text{km}}{\text{h}} \approx 7 \frac{\text{Cells}}{\text{time steps}}$. In case of crossing pedestrians, this can even go down to $20 \frac{\text{km}}{\text{h}} \approx 3 \frac{\text{Cells}}{\text{time steps}}$ depending on the density of crossing pedestrians. Because pedestrians will not be simulated individually and left turning with a normal (non-arrow) traffic signal should also be included in the model, we will choose a value closer to the minimum value. The empirical time requirement value for turning traffic was calculated in [31] and found to be significantly higher than for straight going traffic. The final $v_{\max, \text{turner}}$ will be calculated later after the rules for turning were introduced. Then the model will be calibrated in such a way that the required time is about 1.3 seconds higher than for straight going traffic, in accordance with the empirical findings from [31]. The decreased maximal velocity increases the time requirement t_B to approximate 3.01 time-steps.

Next, the turning agent n has to test if turning without disturbing the traffic flow it would cross is possible. For that, the next agent n^* that is still in front of the cell where the agent n would cross its path has to be identified. The equation

$$\Delta x_{n^*}(\Delta t_n) < \tilde{d}_{n^*} + g_{\text{safe}} \quad (34)$$

has to stay true for the respective agent. $\Delta x_{n^*}(\Delta t_n)$ is the expected distance the vehicle n^* passes in the time Δt_n , the agent n needs to reach the turning point and $\tilde{d}_{n^*}$ is the current number of cells the agent n^* is away from the intersection cell. If Eq. 34 is true, the agent n^* is expected to be still before the intersection cell when the turning agent n arrives. Otherwise, the agent will have already passed the cell and the next agent behind $n^* - 1$ will be tested until the first agent $\tilde{n}^*$ for which Eq. 34 is true is identified or no other agent is within interaction range.

It has to be noted that $c_{\tilde{n}^*}^f$ will be used to anticipate acceleration or deceleration of the agent for the calculation of $\Delta x_{\tilde{n}^*}(\Delta t_n)$. In the real world, the turning driver observes the traffic flow the vehicle would cross while driving up to the turning point, so the driver has gathered past information about its behaviour. The driver then, for example, could notice that the vehicles are all decelerating which would imply that the traffic light turned yellow or even red and then could use this information to judge the situation. The simulation model is a Markov process of the first order, which means that to calculate the next time

step and $\Delta x_{\tilde{n}^*}(\Delta t_n)$ only the current situation is considered and so $c_{\tilde{n}^*}^t$ reflects necessary information for the agent n to make a realistic judgement of the situation in real-time. To prevent the agent from being all-knowing it is only checked if $c_{\tilde{n}^*}^t > v_{\tilde{n}^*}^t$ where $\Delta x_{\tilde{n}^*}(\Delta t_n)$ is calculated expecting that the agent $\tilde{n}^*$ will accelerate to $v_{\max}^S$, or $c_{\tilde{n}^*}^t < v_{\tilde{n}^*}^t$ where it is calculated expecting that the agent will decelerate to 0.

If no agent $\tilde{n}^*$ can be identified, then agent n is expected to be able to turn freely and the agents drive accordingly. Else it has to be tested, if agent $\tilde{n}^*$ would have to decelerate because of the agent n . Agent n is only allowed to turn if

$$\tilde{d}_{\tilde{n}^*} > \Delta x_{\tilde{n}^*}(\Delta t_n) + g_{\text{safe}} + \Delta x_{\text{safe}}(t_+) + \Delta x_L(\tilde{\tau}) \quad (35)$$

is true. Else the agent has to stop before reaching the intersection cell in accordance with Eq. 26 to Eq. 28. $\tilde{\tau} = \lceil \frac{L+1-v_{n,\text{anti}}^{\tilde{\tau}}}{a_n^t} \rceil$ is the number of time steps the agent n needs to finish the turning manoeuvre and leave the cell where he crosses the flow. It depends on the anticipated velocity of agent n at the time step it probably reaches the intersection cell $\tilde{\tau}$. In this time the agent $\tilde{n}^*$ will cross the distance $\Delta x_L(\tilde{\tau})$. Because multiple of these values can only be guessed with more or less high accuracy, the agent $\tilde{n}^*$ has to be placed an additional distance $\Delta x_{\text{safe}}(t_+)$ away from the crossing point, depending on his speed. For $t_+ \geq 3$ no accidents or forced deceleration occur, which is in accordance with the 3 s time differentials in [17]. In case of multiple lanes the agent n has to cross, one agent $\tilde{n}^*$ has to be identified for each of these lanes and Eq. 35 has to be true for each of them.

This restriction now increases the required time value depending on the crossing flow. In [31] it was found that the required time value should increase about 1.3 seconds for a low crossing flow but it also fluctuates highly. To test the influence of crossing flow on the required time value, an intersection where a left turning lane crosses straight going traffic is simulated. Then multiple simulations were run, where the traffic flow on the turning lane is higher than can pass the traffic light, while each time the traffic flow of the crossing traffic is increased. Each simulation runs for 360,000 time steps (100 hours) and the average required time value, as well as its standard derivation, are shown in Fig. 4. One can see the significant increase in the required time value even for low and a higher fluctuation for higher crossing flows. The required time value more than doubles for a crossing flow greater than $400 \frac{\text{vehicles}}{\text{h}}$, which directly translates to fewer vehicles passing the traffic light within the same time frame. For very high crossing traffic flows only the first turning vehicle—which enters the intersection and waits for an opportunity to cross turn—can pass the intersection within one green phase. To prevent such little flows in city traffic, most larger intersections have separate turning cycles where only the two left-turning flows are allowed to drive (often with arrow traffic lights). For smaller intersections without left turning only lanes, the two opposite streets often do not have the same green cycle. There, one of the two lanes gets green for a few seconds to allow turning first. Then both simultaneously have a green phase for some time and at the end the other street has additional green time to allow turning.

Lastly for right turning traffic which is changing lanes over slip lanes, Eq. 35 has to be slightly adapted. Slip lanes allow traffic to turn right without driving through the intersections. The last section of the lane is separated by other lanes of the street and it is often

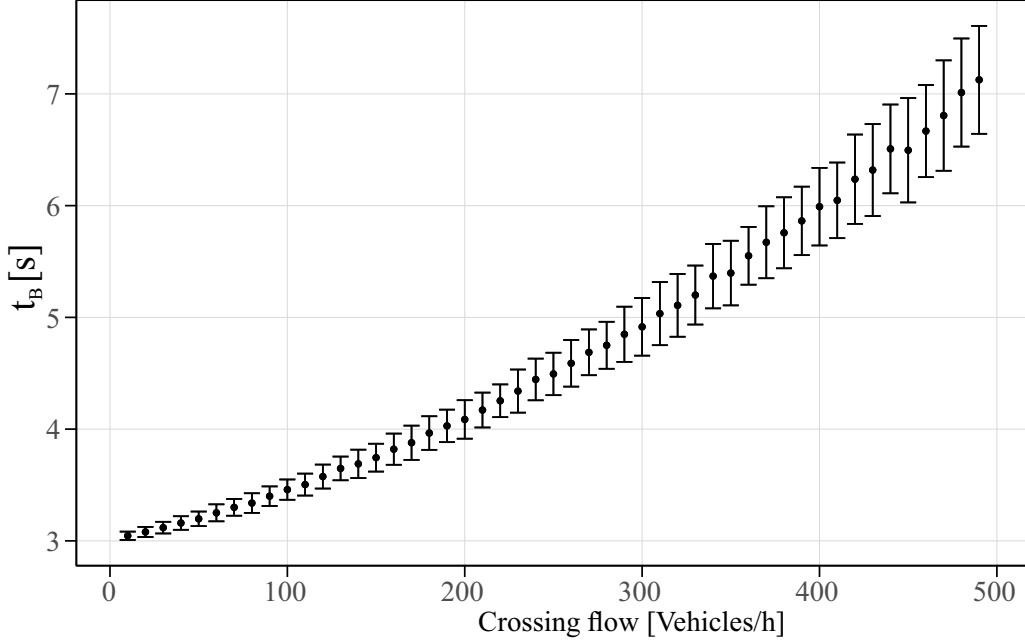


Figure 4 Required time value in dependency of the incoming crossing flow.

not controlled by traffic lights or only has simplified yellow-red phased ones. Instead, the vehicle have to ensure that they only leave their lane when they are not disturbing the on-going traffic flow they want to merge with. To prevent a distribution of the ongoing flow, the vehicles are only allowed to merge if they do not force following vehicles to actively brake. To ensure this, an additional distance $\Delta x_v(v_{\text{diff}})$ has to be added. This extension depends on the expected velocity difference of the agent n^* that intends to merge and the vehicle that will follow the agent after the merging $n^* - 1$:

$$\Delta x_v(v_{\text{diff}}) = \begin{cases} \sum_{i=0}^{t_{\text{diff}}} \max(v_{n^*-1}^{\tilde{t}} + i \cdot a_{n^*-1}^t, v_{\text{max}}), & \text{for } v_{n^*-1}^{\tilde{t}} < c_{n^*-1}^t \\ \sum_{i=0}^{t_{\text{diff}}} v_{n^*-1}^{\tilde{t}}, & \text{for } v_{n^*-1}^{\tilde{t}} = c_{n^*-1}^t \\ \sum_{i=0}^{t_{\text{diff}}} \max(v_{n^*-1}^{\tilde{t}} - i \cdot D, 0), & \text{for } v_{n^*-1}^{\tilde{t}} > c_{n^*-1}^t \end{cases} \quad (36)$$

with

$$t_{\text{diff}} = \begin{cases} \max(\frac{v_{n^*}^{\tilde{t}} - v_{n^*-1}^{\tilde{t}}}{a_{n^*}^t}, 0), & \text{for } v_{n^*-1}^{\tilde{t}} < c_{n^*-1}^t \text{ \& } v_{n^*-1}^{\tilde{t}} > v_{n^*}^{\tilde{t}} \\ \max(\frac{v_{n^*}^{\tilde{t}} - v_{n^*-1}^{\tilde{t}}}{a_{n^*}^t}, 0), & \text{for } v_{n^*-1}^{\tilde{t}} = c_{n^*-1}^t \text{ \& } v_{n^*-1}^{\tilde{t}} > v_{n^*}^{\tilde{t}} \\ \max(\frac{v_{n^*}^{\tilde{t}} - v_{n^*-1}^{\tilde{t}}}{D + a_{n^*}^t}, 0), & \text{for } v_{n^*-1}^{\tilde{t}} > c_{n^*-1}^t \text{ \& } v_{n^*-1}^{\tilde{t}} > v_{n^*}^{\tilde{t}} \\ 0, & \text{for } v_{n^*-1}^{\tilde{t}} \leq v_{n^*}^{\tilde{t}} \end{cases} \quad (37)$$

equation Eq. 35 changes to

$$\tilde{d}_{n^*} > \Delta x_{n^*}(\Delta t_n) + g_{\text{safe}} + \Delta x_{\text{safe}}(t_+) + \Delta x_L(\tilde{t}) + \Delta x_v(v_{\text{diff}}). \quad (38)$$

This has to be checked for all traffic the agent might merge with straight going, crossing, or left turning opposite flow.

If Eq. 38 is also checked for the traffic flow left-turning agents want to merge with and if Eq. 35 is checked for the crossing flow of straight going vehicles, then intersections can be simulated without traffic lights. In such a scenario, instead of traffic light checks, it has to be tested whether the vehicle is allowed to interrupt all crossing flows (e.g. with respect to agents having to give way to other agents to their right or priority assignments of some lanes). Furthermore, without traffic lights to separate traffic and pedestrian flows, the influence of pedestrian crossing the streets has to be considered [9].

With these additional rules, agents driving through an intersection can be fully simulated in accordance with empirical data. Due to general limitations of cellular automata simulations of agents, problems can arise where no agent can move any more, so-called "deadlocks". The following section will explain where deadlocks can occur and how to prevent them.

5 Deadlock prevention

There are three possible deadlock situations within the proposed model. First, there are left-turning vehicles, interrupting each other and thus stopping traffic in this intersection completely. Secondly, traffic jams in front of an intersection that extend into the preceding intersection and prevent other traffic flows from passing through it. Lastly, the third situation results from synchronized left-turning behaviour of multiple agents in a relatively small area of multiple intersections.

The first deadlock situation is the easiest to understand and fix. When two agents with opposite flow directions would want to turn in an intersection, they each prevent the other one from turning by stopping within the crossing cell. In most intersections in real urban traffic, two left-turning vehicles of opposite flows turn in front of each other instead of behind or next to them, like in the model. To simulate this, the agents will act as if they pass behind each other, even though they are not. Effectively this means that they ignore other left-turning agents of the opposite flow while calculating if they can pass and while crossing.

The second deadlock is a specific problem of urban traffic where the length of streets between two intersections is often very short, sometimes even only 100 to 200 m. A street length of 150 m means that only around 20 vehicles (with an average occupation space of 7.5 m in a jam [12]) can queue in front of an intersection before the next vehicle will stop within the following intersection. To prevent a queue from one traffic light to interrupt traffic in the previous one, agents are only allowed to drive into the intersection when it is to be expected that they can leave it. If there is an agent $n + 2$ (predecessor of the predecessor of agent n) within the interaction range of agent n and all three Eq. 39-Eq. 41 is true, then it is expected that the agent $n + 1$ will also brake and stop shortly after leaving the intersection. This, in turn, creates a high risk of the agent n coming to a stop within the intersection, which is to be prevented. So agent n is only allowed to enter the intersection if one or more of these equations is not true or if no agent $n + 2$ is within the interaction range. Agent $n + 2$ is considered here instead of $n + 1$ because agent n has to decelerate ahead of time to be able to stop before entering the intersection. This is not assured if

only the $(n + 1)$ st agent is considered. In equation form, this means that agents are only allowed to enter intersections if one of the three following equations does not hold true:

$$x_{n+2}^f + \delta(v_{n+2}^f) < 14 \quad (39)$$

$$a_{n+2} = 1 \quad (40)$$

$$v_{n+2} < v_{\text{fast}} \quad (41)$$

The last deadlock situation is visualized in Fig. 5. It becomes apparent that no vehicle can move because the first vehicle in every queue is an agent that intends to turn left but is not able to because they would have to stop in the intersection. This is forbidden because of the previous deadlock situation. Because the same holds true for all four intersections no vehicle can move anymore. In reality, the drivers would simply change their direction and drive straight or right to get to their intended destination over a slightly different route. Agents, on the other hand, would wait an infinite amount of time for a space in the queue to free themselves. This deadlock situation will be solved in the same way as in [14], where the vehicle that stands in front of a green traffic light but does not move for a long time is simply removed. This can be done because, as shown in [14] this deadlock situation happens rarely enough that removing single agents does not significantly affect the statistics. For this reason, agents that stand still more than two green phases are removed from the simulation.

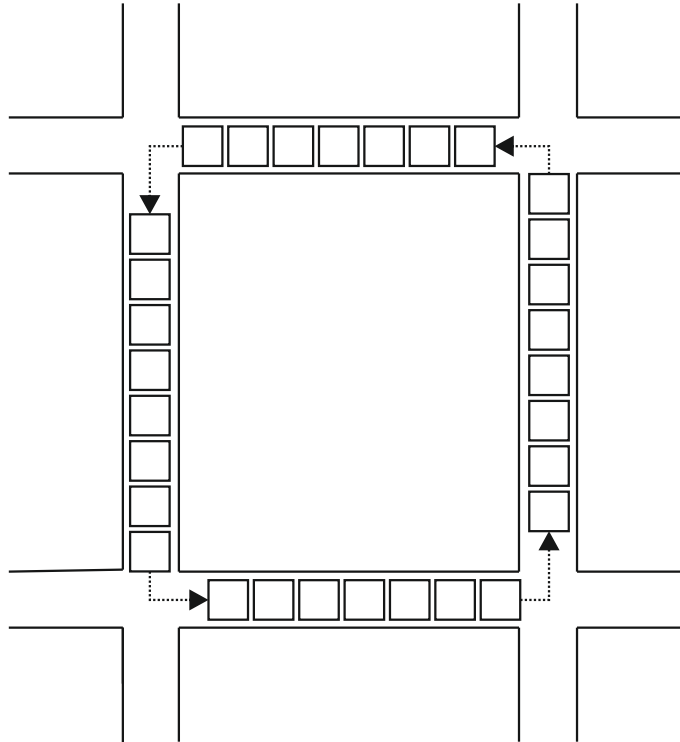


Figure 5 Deadlock situation three: One can see four intersections where in every intersection the first vehicle wants to turn left but can't because he then would have to stop in the intersection.

6 Conclusion

This paper proposes a model to simulate the behaviour of drivers before and through an intersection in accordance with empirical data while also incorporating realistic car following times t_B , jam propagation velocity v_{jam} , and traffic outflow from a jam F_{out} . The Lee et. al. model [18] was chosen as a basis for the proposed intersection model. Most intersection models use an intersection area that can't be entered if opposite flows have entered their respective intersection area to recreate traffic crossing. Because of the finite deceleration implemented in the Lee-model and the inability of these models to simulate multiple lanes with multiple purposes (left-/right-turning or going straight ahead), a new design for intersections is introduced. Afterwards, additional rules based on the Lee-model were defined so that the agents could pass through these newly designed intersections accident-free and in accordance with empirical traffic data.

In future works, based on the intersection model, urban networks will be created and different behaviour based influences will be tested. For example, a driver who never grants the courtesy of decelerating to make space for a lane changer or always judge their situation pessimistically (never drive closer than the minimum safety distance) could be implemented and their impact on the traffic flow analysed. Furthermore, p_0 , p_d and v_{slow} will be chosen in a certain interval according to different distributions and the influence of these distributions will be analysed.

Acknowledgements Part of the work on this paper has been supported by Deutsche Forschungsgemeinschaft (DFG) within the Collaborative Research Center SFB 876 “Providing Information by Resource-Constrained Analysis”, project B4.

References

- [1] Kerner, B.S., Klenov, S.L., Schreckenberg, M.: Traffic breakdown at a signal: classical theory versus the three-phase theory of city traffic. *Journal of Statistical Mechanics: Theory and Experiment* **2014**(3), P03001 (2014). doi:[10.1088/1742-5468/2014/03/p03001](https://doi.org/10.1088/1742-5468/2014/03/p03001)
- [2] Li, H., Li, S., Li, H., Qin, L., Li, S., Zhang, Z.: Modeling left-turn driving behavior at signalized intersections with mixed traffic conditions. *Mathematical Problems in Engineering* **2016**(1), 1–11 (2016). doi:[10.1155/2016/4015271](https://doi.org/10.1155/2016/4015271)
- [3] Tonguz, O.K., Viriyasitavat, W., Bai, F.: Modeling urban traffic: A cellular automata approach. *IEEE Communications Magazine* **47**(5), 142–150 (2009). doi:[10.1109/MCOM.2009.4939290](https://doi.org/10.1109/MCOM.2009.4939290)
- [4] Chopard, B., Luthi, P.O., Quelo, P.A.: Cellular automata model of car traffic in a two-dimensional street network. *Journal of Physics A: Mathematical and General* **29**(10), 2325–2336 (1996). doi:[10.1088/0305-4470/29/10/012](https://doi.org/10.1088/0305-4470/29/10/012)

- [5] Bai, F., Sadagopan, N., Helmy, A.: The important framework for analyzing the impact of mobility on performance of routing protocols for adhoc networks. *Ad Hoc Networks* **1**(4), 383–403 (2003). doi:[10.1016/S1570-8705\(03\)00040-4](https://doi.org/10.1016/S1570-8705(03)00040-4)
- [6] Nagel, K., Stretz, P., Pieck, M., Donnelly, R., Barrett, C.L.: Transims traffic flow characteristics (1997). URL <https://arxiv.org/pdf/adap-org/9710003>
- [7] Krautter, W., Bleile, T., Manstetten, D., Schwab, T.: Traffic simulation with artist. In: *Proceedings of Conference on Intelligent Transportation Systems*, pp. 472–477 (1997). doi:[10.1109/ITSC.1997.660520](https://doi.org/10.1109/ITSC.1997.660520)
- [8] Robertson, I.: TRANSYT: a traffic network study tool. Road Research Laboratory Report LR 253 (1969)
- [9] Zhao, H.T., Yang, S., Chen, X.X.: Cellular automata model for urban road traffic flow considering pedestrian crossing street. *Physica A: Statistical Mechanics and its Applications* **462**, 1301–1313 (2016). doi:[10.1016/j.physa.2016.06.146](https://doi.org/10.1016/j.physa.2016.06.146)
- [10] van Aerde, M., Hellenga, B.R., Baker, M.R., Rakha, H.A.: Integration : An overview of traffic simulation features. In: *Transportation Research Board Annual Meeting*, pp. 1–14 (1996). URL <https://pdfs.semanticscholar.org/5ba0/8ab26cdaa90ad0eb7c37a71b7146622096c6.pdf>
- [11] Brockfeld, E., Barlovic, R., Schadschneider, A., Schreckenberg, M.: Optimizing traffic lights in a cellular automaton model for city traffic. *Physical Review E* **64**(5 Pt 2), 056132 (2001). doi:[10.1103/PhysRevE.64.056132](https://doi.org/10.1103/PhysRevE.64.056132)
- [12] Nagel, K., Schreckenberg, M.: A cellular automaton model for freeway traffic. *Journal de Physique I* **2**(12), 2221–2229 (1992). doi:[10.1051/jp1:1992277](https://doi.org/10.1051/jp1:1992277)
- [13] Knospe, W., Santen, L., Schadschneider, A., Schreckenberg, M.: Towards a realistic microscopic description of highway traffic. *Journal of Physics A: Mathematical and General* **33**(48), L477–L485 (2000). doi:[10.1088/0305-4470/33/48/103](https://doi.org/10.1088/0305-4470/33/48/103)
- [14] Esser, J.: Simulation von Stadtverkehr auf der Basis zellularer Automaten. Dissertation, Gerhard-Mercator-Universität-Gesamthochschule Duisburg, Duisburg (1997)
- [15] Vasic, J., Ruskin, H.J.: A cellular automata-based network model for heterogeneous traffic: Intersections, turns and their connection. In: Sirakoulis, G.C., Bandini, S. (eds.) *Cellular automata, Lecture Notes in Computer Science*, vol. 7495, pp. 835–844. Springer, Berlin (2012). doi:[10.1007/978-3-642-33350-7_87](https://doi.org/10.1007/978-3-642-33350-7_87)
- [16] Zhao, H.T., Liu, X.R., Chen, X.X., Lu, J.C.: Cellular automata model for traffic flow at intersections in internet of vehicles. *Physica A: Statistical Mechanics and its Applications* **494**, 40–51 (2018). doi:[10.1016/j.physa.2017.11.152](https://doi.org/10.1016/j.physa.2017.11.152)

- [17] Jin, C.J., Wang, W., Jiang, R., Wang, H.: Cellular automaton simulations of a four-leg intersection with two-phase signalization. *International Journal of Modern Physics C* **25**(03), 1350099 (2014). doi:[10.1142/S012918311350099X](https://doi.org/10.1142/S012918311350099X)
- [18] Lee, H.K., Barlovic, R., Schreckenberg, M., Kim, D.: Mechanical restriction versus human overreaction triggering congested traffic states. *Physical Review Letters* **92**(23), 238702 (2004). doi:[10.1103/PhysRevLett.92.238702](https://doi.org/10.1103/PhysRevLett.92.238702)
- [19] Kerner, B.S.: *Breakdown in Traffic Networks*. Springer Berlin Heidelberg, Berlin, Heidelberg (2017). doi:[10.1007/978-3-662-54473-0](https://doi.org/10.1007/978-3-662-54473-0)
- [20] Pottmeier, A., Thiemann, C., Schadschneider, A., Schreckenberg, M.: Mechanical restriction versus human overreaction: Accident avoidance and two-lane traffic simulations. In: *Traffic And Granular Flow*, pp. 503–508. Springer (2005). doi:[10.1007/978-3-540-47641-2_46](https://doi.org/10.1007/978-3-540-47641-2_46)
- [21] Barlovic, R., Santen, L., Schadschneider, A., Schreckenberg, M.: Metastable states in cellular automata for traffic flow. *The European Physical Journal B* **5**(3), 793–800 (1998). doi:[10.1007/s100510050504](https://doi.org/10.1007/s100510050504)
- [22] Habel, L., Schreckenberg, M.: Asymmetric lane change rules for a microscopic highway traffic model. In: *Cellular automata, Lecture Notes in Computer Science / Theoretical Computer Science and General Issues*, vol. 8751, pp. 620–629. Springer (2014). doi:[10.1007/978-3-319-11520-7_66](https://doi.org/10.1007/978-3-319-11520-7_66)
- [23] Pottmeier, A.: *Realistic cellular automaton model for synchronized two-lane traffic: Simulation, validation, and applications*. Dissertation, Universität Duisburg-Essen, Duisburg (18.12.2007)
- [24] Sparmann, U.: *Spurwechsellvorgänge auf zweispurigen BAB-Richtungsfahrbahnen*. In: *Forschung Straßenbau und Straßenverkehrstechnik* 263 (1978)
- [25] Krauss, S., Wagner, P., Gawron, C.: Metastable states in a microscopic model of traffic flow. *Physical Review E* **55**(5), 5597–5602 (1997). doi:[10.1103/PhysRevE.55.5597](https://doi.org/10.1103/PhysRevE.55.5597)
- [26] Krbálek, M., Šleis, J.: Vehicular headways on signalized intersections: theory, models, and reality. *Journal of Physics A: Mathematical and General* **48**(1), 015101 (2015). doi:[10.1088/1751-8113/48/1/015101](https://doi.org/10.1088/1751-8113/48/1/015101)
- [27] S. Kerner, B., Rehborn, H.: Experimental features and characteristics of traffic jams. *Physical Review E* **53**(2), R1297–R1300 (1996). doi:[10.1103/PhysRevE.53.R1297](https://doi.org/10.1103/PhysRevE.53.R1297)
- [28] Chiabaut, N., Buisson, C., Leclercq, L.: Fundamental diagram estimation through passing rate measurements in congestion. *IEEE Transactions on Intelligent Transportation Systems* **10**(2), 355–359 (2009). doi:[10.1109/TITS.2009.2018963](https://doi.org/10.1109/TITS.2009.2018963)

- [29] Dervisoglu, G., Gomes, G., Kwon, J., Muralidharan, A., Varaiya, P., Horowitz, R.: Automatic calibration of the fundamental diagram and empirical observations on capacity. Transportation Research Board 88th Annual Meeting p. 15 (2009). URL https://people.eecs.berkeley.edu/~varaiya/papers_ps.dir/trbParameter_Calibration.pdf
- [30] Dion, F., Rakha, H., Kang, Y.S.: Comparison of delay estimates at under-saturated and over-saturated pre-timed signalized intersections. Transportation Research Part B: Methodological **38**(2), 99–122 (2004). doi:10.1016/S0191-2615(03)00003-1
- [31] Greenshields, B.D., Schapiro, D., Ericksen Elroy L.: Traffic Performace At Urban Street Intersections. Bureau Highway Traffic, Yale University, Washington, DC United States (1947)
- [32] Knospe, W., Santen, L., Schadschneider, A., Schreckenberg, M.: Single-vehicle data of highway traffic: microscopic description of traffic phases. Physical Review E **65**(5 Pt 2), 056133 (2002). doi:10.1103/PhysRevE.65.056133
- [33] Gipps, P.G.: A model for the structure of lane-changing decisions. Transportation Research Part B: Methodological **20**(5), 403–414 (1986). doi:10.1016/0191-2615(86)90012-3
- [34] Hidas, P.: Modelling lane changing and merging in microscopic traffic simulation. Transportation Research Part C: Emerging Technologies **10**(5), 351–371 (2002). doi:10.1016/S0968-090X(02)00026-8