Modeling of Transport Flows in Critical Situations Using Cellular Automata in MATLAB

Andriy Shatyrko[1], Volodymyr Kovtun[1,2], Bedrich Puza[3] and Veronika Novotna[3]

[1]Taras Shevchenko National University of Kyiv, 64, Volodymyrska str., Kyiv, 01033, Ukraine
[2]University of L'Aquila, Piazzale Ernesto Pontieri, 67100 L'Aquila AQ, Italy
[3]Brno University of Technology, Kolejní 2906/4, 612 00 Brno-Královo Pole, Brno, Czech Republic

Abstract

Cellular automata can be used to model traffic flow by applying a simple algorithm that regulates the acceleration and deceleration of vehicles. Despite its simplicity, this model allows for the observation of large-scale phenomena, such as traffic jams that propagate backward. These backward-propagating congestion patterns emerge naturally from the interactions between vehicles without the need for centralized control.

A variation of this model, implemented in MATLAB, was used to study how different parameters, such as vehicle density on the road or the number of lanes, affect traffic flow intensity. The simulation approach leverages a one-dimensional array of discrete cells, where each cell represents a possible vehicle position and its velocity.

Experiments revealed that in a system with a single lane and a maximum speed of 1, a phase transition occurs at a density of 0.08 cars per cell. This phase transition marks a critical shift in traffic dynamics: below the threshold, traffic flows freely, while above it, jams persist and propagate. Moreover, this transition was observed only in cases where the maximum speed exceeded one, confirming the non-linear nature of traffic flow dynamics even under simple rule sets.

Keywords

Cellular automata, Nagel–Schreckenberg (NS) model, numerical simulation, MATLAB, animated visualization

1. Introduction

A cellular automaton (CA) consists of a structured grid where each cell possesses a limited number of possible states. These cells evolve over time based on a predefined set of rules. Despite their simplicity, such rules enable the cellular automaton to exhibit complex emergent behaviors that would not typically be expected in such a basic system. Cellular automata are widely utilized for analyzing and simulating systems like traffic flow and load-bearing structures.

CA can be regarded as a form of computational intelligence particularly suited for modeling large-scale, decentralized systems. Their local interaction rules enable the emergence of complex global patterns, such as traffic waves or jams, without requiring global coordination. Compared to other CI approaches, such as reinforcement learning, CA models offer greater scalability and computational simplicity, making them ideal for simulating traffic behavior across thousands of vehicles. However, unlike learning-based approaches, CA rules are hand-crafted, which may limit adaptability in highly dynamic scenarios [1]. They provide an effective modeling framework for physical systems since these systems are inherently discrete and governed by localized interactions. In the context of traffic flow, these interactions manifest as vehicles braking or changing lanes in response to the movements of the car directly ahead.

Nagel and Schreckenberg introduced a CA model for traffic flow in their work *A cellular automaton model for freeway traffic* [2]. This model, which will be referred to as the NS model, is structured as a one-dimensional array of L sites with either open or periodic boundary conditions. Open boundary

International Workshop on Computational Intelligence, co-located with IV International Scientific Symposium "Intelligent Solutions" (IntSol-2025), May 1–5, 2025, Kyiv–Uzhhorod, Ukraine
shatyrko.a@knu.ua (A. Shatyrko); vpkovtun111@gmail.com (V. Kovtun); puza@vutbr.cz (B. Puza); novotna@fbm.vutbr.cz (V. Novotna)
0000-0002-5648-2999 (A. Shatyrko); 0009-0000-5798-5718 (V. Kovtun); 0000-0002-2949-4708 (B. Puza); 0000-0001-9360-3035 (V. Novotna)

conditions mean that the road has an end, allowing cars to exit while new ones enter at a designated entry point. In contrast, periodic boundary conditions create a loop where cars continuously circulate.

As previously mentioned, CA models operate with a finite number of states. In the NS model, a cell's state indicates whether it is occupied by a car and, if so, the vehicle's velocity. To represent this, empty sites are assigned a value of zero, while occupied sites are assigned a numerical value corresponding to the vehicle's velocity, ranging from one to $v_{\max} + 1$ for vehicles moving at $v_{\max}$. In computational implementations, this numerical representation is used for processing. However, in a visual representation, empty sites may appear as dots, while occupied sites display their respective velocities.

A simple one-lane system, where the maximum velocity is one, can be visualized as an array consisting solely of zeros and ones. Similarities between the NS model and the Ising spin model arise due to the binary nature of the site states. In the Ising model, spins depend on neighboring interactions, and likewise, in traffic simulations, the velocity of a car depends solely on the immediate forward environment, with no influence from cars behind. This local-only interaction framework is what leads to globally emergent traffic patterns. This configuration follows the fundamental CA structure, where each cell has only two possible states, forming a one-dimensional site array.

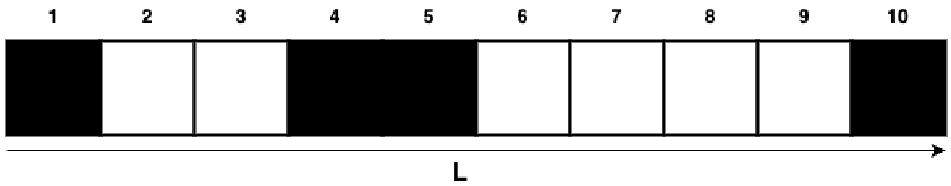

Figure 1: An example of the simple one-lane array with a road length of 10 and 4 cars.

The configuration evolves according to the following set of rules for each car:

1. **Acceleration**: if the velocity v of a vehicle is lower than $v_{\max}$ and if the distance to the next car ahead is larger than $v + 1$, the speed is increased by 1 ($v = v + 1$).
2. **Deceleration**: if a vehicle at site i sees the next vehicle at site $i + j$ (with $j \leq v$), it reduces its speed to $j - 1$ ($v = j - 1$).
3. **Randomisation**: with probability p, the velocity of each vehicle (if greater than zero) is decreased by one ($v = v - 1$).
4. **Car motion**: each vehicle is advanced v sites.

Where:

$$\begin{aligned} v &= \text{velocity of car} \\ i &= \text{position of site} \\ j &= \text{position of comparison site} \\ v_{\max} &= \text{maximum velocity} \end{aligned}$$

Among the four update rules, the randomisation step plays a pivotal role in reproducing realistic stop-and-go traffic. Without it, the model would quickly stabilize into a steady state with uniform spacing, lacking the variability observed in actual traffic. It is this randomness that triggers instability, which then propagates and creates dynamic congestion.

In this case, the third stage, the randomized braking, plays a crucial role in triggering emergent behavior in the system when the velocities exceed one.

Typically, when vehicles are spaced within approximately $v_{\max}$ sites, the leading car may unexpectedly slow down. This sudden deceleration prompts the following cars to react by braking in sequence, leading to the formation of a traffic jam.

As congestion builds up, cars approaching the jam continue moving at nearly maximum velocity but are forced to come to an abrupt halt upon reaching the stationary traffic ahead. Meanwhile, if there is sufficient space, the first car at the front of the jam gradually accelerates away. This process repeats cyclically, resulting in the backward propagation of the traffic jam.

Traffic jams tend to dissipate in low-density scenarios since vehicles exiting the jam do so at a faster rate than those joining it. However, at higher densities, the jam becomes a persistent feature of the periodic road, continuously propagating backward and sustaining itself indefinitely.

Thus, the CA-based traffic model provides a minimal yet powerful framework for exploring a wide range of traffic phenomena, from stable flow to jam propagation, under varying densities and road configurations.

2. The Model

The initial configuration of the road is generated by randomly shuffling a list of numbers ranging from 1 to L, where L represents the total road length. Each vehicle on the road is sequentially assigned a number from this list, creating a $1 \times N$ array. This array is then expanded by adding a $1 \times N$ array for initial velocities, which are typically all set to zero, and a $1 \times N$ array for lane assignments. If the system operates with a single lane, the lane array is omitted, leaving a $2 \times N$ representation, where N denotes the total number of vehicles on the road.

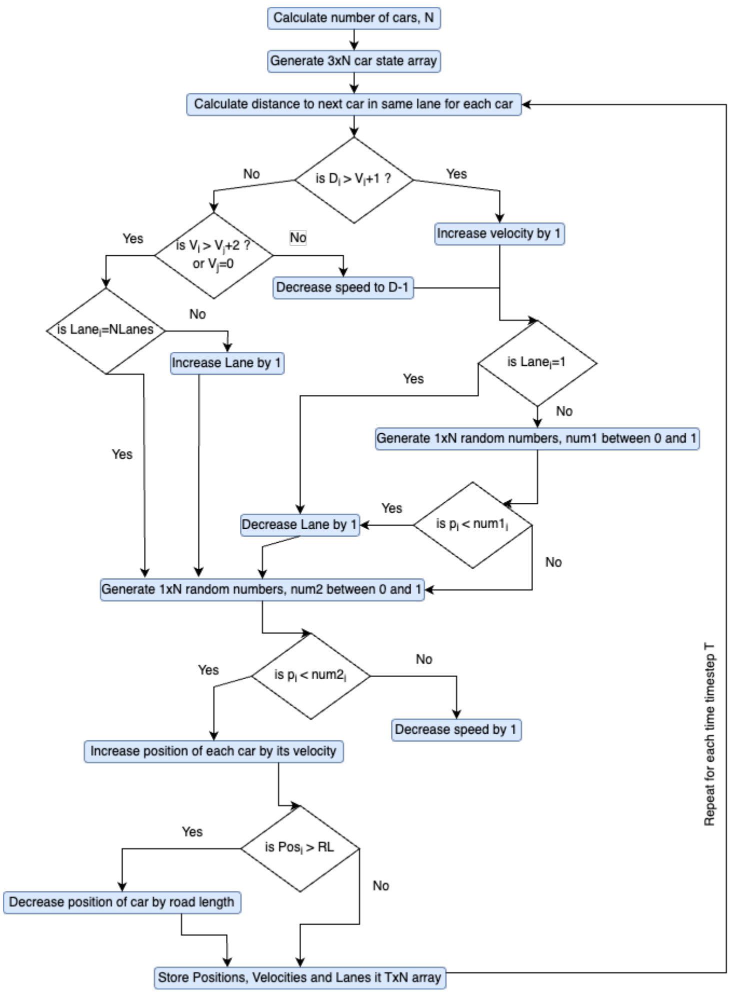

Figure 2: Flowchart of the traffic flow simulation process.

Instead of relying on the NS model, the program employs a more compact data structure that retains only the essential details: position, velocity, and lane assignment for each vehicle within a $3 \times N$ matrix (or $2 \times N$ for single-lane cases). The program applies Boolean logic to analyze column data, computing vehicle spacing and allowing the simulation to process the four defined stages in parallel.

In multilane scenarios, lane-switching decisions are integrated with the braking mechanism. Vehicles initiate an overtaking maneuver when they encounter a slower vehicle with a speed differential of at least two units or if the leading car is stationary. While this approach to lane-changing might seem simplistic, it effectively reproduces emergent traffic behavior that closely aligns with real-world observations.

For a lane change to be performed, the target lane must have sufficient free space both ahead and behind the vehicle. Specifically, the distance to the next vehicle in the adjacent lane must be greater than the current speed, and the gap behind must exceed a minimum threshold to avoid rear-end collisions. Additionally, the decision to change lanes is probabilistic and depends on a pre-defined overtaking probability factor. This allows the simulation to capture a variety of driver behaviors, from conservative to aggressive, and ensures that lane changes occur only under safe and justifiable conditions.

Since vehicle placement is initially randomized, some configurations may lead to congestion where traffic would not naturally emerge. This effect is particularly noticeable at lower densities. To mitigate this, the simulation runs for a number of timesteps equal to the road length, allowing the system to stabilize. For road lengths exceeding 500 units, no significant stabilization occurs beyond half the road length. As a result, an initialization phase of 1000 timesteps is introduced to enhance computational efficiency.

The second phase of the simulation continues from the final state of the first phase, running for T timesteps. During each step, the system updates stored arrays containing vehicle positions, velocities, and lanes. At the end of the simulation, the average velocity is computed and multiplied by the density to determine the primary performance metric—the traffic flow rate.

A key aspect of the model is the periodic boundary condition, ensuring that vehicles at the start and end of the road maintain accurate distance calculations. This is achieved by sorting vehicles based on their position.

For lane changes, vehicles in the target lane (either overtaking or returning to a previous lane) are selected, and their positions and velocities are adjusted accordingly. Vehicles with extreme positions in the lane undergo boundary modifications, allowing smooth interaction with other cars on the road.

Several visualization techniques were employed to analyze the simulation results. One of the most effective methods is the space-time plot, which clearly illustrates the backward propagation of traffic jams. However, this approach relies on color coding to differentiate vehicle speeds and struggles to represent multiple lanes without excessive data overlap. A more intuitive representation is the animated road plot, where individual vehicle speeds and jam formations are visually distinct. This animation directly corresponds to space-time plot data at any given moment.

3. Model Innovations and Implementation Specifics

The model introduced in this study represents a modification of the classic Nagel–Schreckenberg (NS) cellular automaton. In contrast to traditional multi-lane CA implementations, our approach introduces a refined overtaking logic, wherein vehicles decide to change lanes only if there is a speed differential of at least two units or the leading vehicle is completely halted. This rule-based yet lightweight strategy was observed to produce traffic dynamics that align more realistically with real-world observations, even under dense traffic conditions.

Another novel aspect of this model lies in the handling of smart vehicles, defined as those that do not perform random braking. The simulation introduces a customizable proportion of such smart cars, allowing for mixed-autonomy traffic scenarios. The quantitative impact of their inclusion—such as the increase in flow rate under various densities—was systematically measured, demonstrating the strong influence of partial automation on traffic efficiency. These features distinguish this work from other implementations of the NS model and contribute to the current research on hybrid traffic systems.

4. Implementation Details of the Traffic Flow Simulation Algorithm

To build the traffic flow simulation model, we implemented a custom algorithm in **MATLAB** based on the NS cellular automaton model, with several modifications to support multi-lane behavior and overtaking logic. This algorithm simulates the behavior of vehicles over discrete time steps and space segments. Each car on the road is represented by its position, velocity, and current lane, all of which are stored in a structured matrix for efficient processing.

The implemented model extends the classical NS CA by introducing a modular structure that supports multi-lane traffic with dynamic lane-changing logic. A unique feature of this implementation is the overtaking mechanism, which not only checks safety criteria but also considers speed differentials, enhancing realism. Moreover, we quantitatively assess the influence of "smart vehicles"—agents immune to random braking—on flow efficiency. Unlike traditional multi-lane CA models, our system captures the emergence of hybrid traffic phenomena in mixed-autonomy environments through parameter-controlled driver behavior.

The simulation proceeds in iterations. During each time step, the algorithm applies a series of logical rules to update the velocity and position of each car, taking into account nearby vehicles and random braking probabilities. The core logic is based on four main rules: acceleration, deceleration, random braking, and movement.

To represent the road and cars, we use a $3 \times N$ matrix, where N is the number of vehicles. The rows represent positions, velocities, and lanes. For a single-lane road, only the first two rows are needed. Initially, all cars are assigned random positions and zero velocities. This setup avoids artificial congestion due to clustered placement.

The simulation proceeds as follows:

1. **Acceleration**: Cars increase their velocity by one unit if the distance to the next vehicle is sufficient and the velocity is below the allowed maximum.
2. **Deceleration**: If a car is too close to the vehicle ahead, it reduces its speed to avoid a collision.
3. **Random braking**: With a fixed probability, vehicles may randomly reduce their speed. This simulates human reaction time, distractions, or unexpected slowdowns.
4. **Lane changing**: If the car is blocked and sees a chance to overtake, it will change to a neighboring lane, provided that the move is safe.
5. **Movement**: Each car moves forward by a number of sites equal to its current velocity. Periodic boundary conditions are applied.

Each step is vectorized to improve computation speed, using MATLAB's logical indexing and element-wise operations. We also implemented plotting functions to visualize the road dynamically in the form of animated plots or space-time diagrams.

Below is a simplified version of the MATLAB code for a single update iteration:

```
for t = 1:T
    % Step 1: Calculate distance to next car
    sorted_pos = sort(positions);
    distances = [diff(sorted_pos), L - sorted_pos(end) + sorted_pos
        (1)];

    % Step 2: Acceleration
    velocities = min(velocities + 1, vmax);

    % Step 3: Deceleration
    for i = 1:N
        if distances(i) <= velocities(i)
            velocities(i) = distances(i) - 1;
        end
```

```
    end

    % Step 4: Random braking
    braking = rand(1, N) < p;
    velocities(braking & velocities > 0) = velocities(braking &
        velocities > 0) - 1;

    % Step 5: Position update
    positions = mod(positions + velocities, L);
end
```

This approach allows for highly customizable simulations. By adjusting parameters such as the number of lanes, density, braking probability, or maximum velocity, one can observe different emergent behaviors, including traffic waves and jams.

The algorithm was inspired and supported by the following references: [2, 3, 4].

This implementation provides a balance between simplicity and realism, allowing in-depth experimentation with real-world traffic dynamics. It also sets the stage for further enhancements, such as the introduction of adaptive cruise control systems or integration of real map-based road topologies for practical simulations.

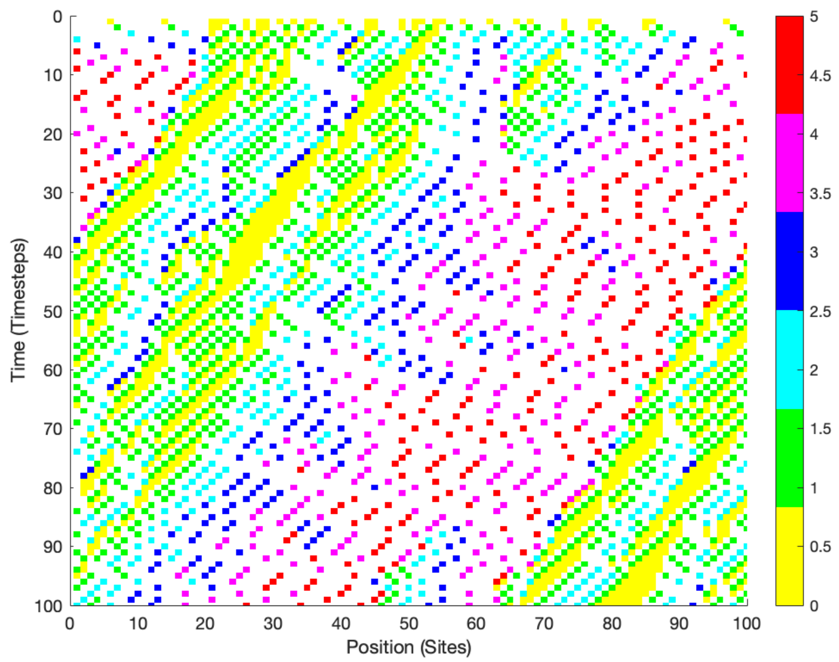

Figure 3: An example of a space-time plot of a one-lane periodic road with $\rho = 0.2$. The color bar on the right indicates vehicle speed in sites per timestep, from yellow (stopped) to red (maximum speed of 5). This color scale is used to visualize traffic jams, which appear as clusters of yellow or light green cells moving diagonally from top left to bottom right.

Traffic congestion remains a persistent feature under specific density conditions. The color scale used in the visualization represents vehicle speeds, with cars traveling at either 4 or 5 units per timestep remaining within the jam until preceding vehicles clear the way.

The animation captures a momentary state of traffic flow, proving highly effective in visualizing multiple lanes, as seen in Figure 4. Despite a high overall density, traffic does not fully utilize all five lanes due to overtaking constraints. The second and third lanes remain mostly empty, while congestion accumulates in the first lane. Minimal vehicle presence is observed in the fourth and fifth lanes due to

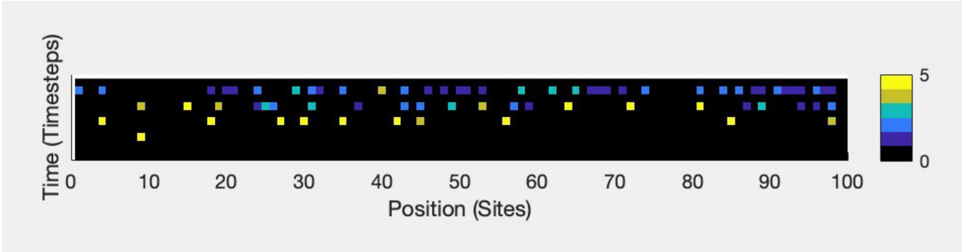

Figure 4: A snapshot of the animated plot of a five-lane road with $\rho = 0.7$. The visualization uses a different color scheme from Figure 3, where **dark blue indicates stationary vehicles** and **yellow marks cars moving at maximum speed**. This color inversion enhances contrast on dark backgrounds.

underutilization of the third lane. Unlike the space-time plot, the animated visualization employs a different color scheme, where dark blue represents stationary cars and yellow indicates vehicles moving at maximum speed.

The simulation demonstrates sensitivity to all factors except road length. Since the vehicle count is density-controlled, modifying road length alone does not alter density values. It is important to note that density in this context refers to vehicles per site rather than total road occupancy. On multilane roads, a density of one does not imply full utilization of available lanes. The program enables modifications to maximum velocity, braking probability, and the likelihood of lane return on a per-car basis, though this aspect was not explored in this study.

As discussed earlier, the road length was initially used for a transient simulation phase. To optimize computational efficiency and enhance visualization, road length was set to a reasonable value between 100 and 1000 units.

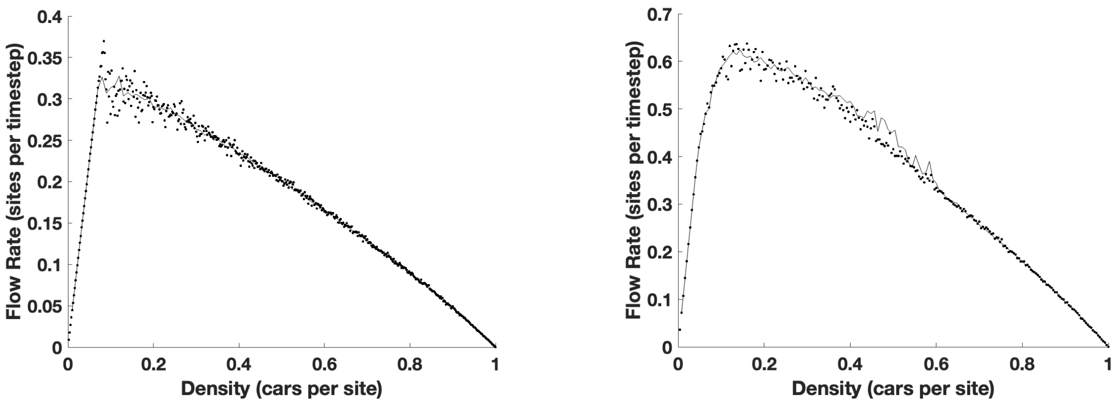

Figure 5: The effect of varying the density of cars along both a one-lane and two-lane road. The lines correspond to a time period of 10000 (1000 for the second figure), while dots represent a period of 100 timesteps.

The initial segment of the density graph, covering values from 0 to 0.08 cars per site, indicates minimal vehicle interaction. This represents a stable traffic phase where vehicles are sufficiently spaced to avoid congestion. The relationship between flow rate and density follows the equation:

$$Q = \rho \times v_{\max} \tag{1}$$

where Q represents the flow rate, ρ denotes car density, and $v_{\max}$ corresponds to the highest attainable velocity. The declining segment of the graph illustrates lane oversaturation effects. The shape of Figure 6 aligns with findings from the NS study, confirming consistency between both models.

A comparison between the model results and real-world traffic data further validates the findings. The real traffic flow rate peaks at a density of 0.2, which matches the simulated critical den sity. By

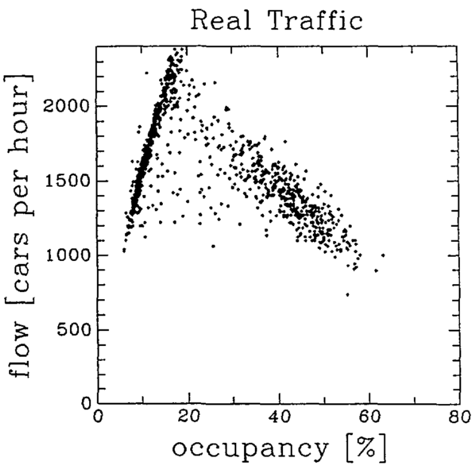

Figure 6: Flow rate of real traffic for varying occupancy ratios. Original source: [2,3].

adjusting the vertical scale of Figure 6 based on Equation (2), direct comparison with real traffic flow is achieved. The recalculated peak flow rates are:

$$0.37 \times 1.14 \times 60 \times 60 \approx 1520 \textit{ cars per hour.} \tag{2}$$

$$0.61 \times 1.14 \times 60 \times 60 \approx 2500 \textit{ cars per hour.} \tag{3}$$

The scaling applied in Equations (2) and (3) is based on two key assumptions. First, it is assumed that one timestep in the simulation corresponds to approximately one real-time second. Second, one site (or cell) is mapped to an average vehicle length of 7.5 meters, a standard convention in traffic flow studies. With these assumptions, the simulated flow in sites per timestep can be translated into a physical unit of vehicles per hour through multiplication by 3600 seconds per hour and an empirical scaling factor.

However, this conversion is approximate and should be interpreted cautiously. Cellular automaton models use discrete space and time steps with simplified rule sets. These do not capture finer aspects of real traffic behavior, such as variable acceleration, human reaction times, heterogeneous vehicle types, or external factors like road geometry and weather. Unlike real-world traffic, which exhibits continuous and often nonlinear dynamics, CA-based simulations rely on local rules and can exhibit abrupt transitions and exaggerated sensitivity to initial conditions—especially at higher densities.

Nevertheless, the visual and quantitative consistency between the simulation and empirical data reinforces the validity of CA models as a first-order approach for capturing macroscopic flow behavior, particularly under controlled assumptions.

The second dataset, corresponding to a two-lane configuration, closely matches the observed trend in Figure 6, suggesting that multiple-lane roads—such as highways—serve as a more reliable basis for traffic modeling compared to rural roads.

Investigation

One of the natural ways to explore the model is by modifying its parameters and analyzing how these changes affect the traffic flow along the road. A key aspect of this investigation was assessing the impact of 'smart cars' on the overall flow rate. In this model, smart cars operate without the random braking characteristic of human-driven cars, making them effectively perfect drivers. This implies that a smart car should never directly contribute to a traffic jam, so increasing their proportion in the system should result in a higher traffic flow.

The process of eliminating random braking from specific cars in this model is straightforward since the probability of braking for each car is stored within a 1×N array. Therefore, the investigation can be performed simply by setting a varying number of these cells to zero. This was done on road configurations just beyond saturation, the point at which it is not possible for all cars to move at the maximum velocity. These scenarios are particularly important because they are most sensitive to changes in vehicle behavior and present the highest risk of congestion.

Furthermore, several scenarios were simulated, ranging from 0% to 100% of smart cars, to identify thresholds where traffic flow significantly improves. These tests revealed that even a modest inclusion of smart cars (as low as 25%) already leads to a noticeable increase in flow efficiency, while 70% or more smart cars nearly double the throughput. This supports the hypothesis that traffic systems benefit greatly even from partial adoption of autonomous driving technologies.

It is important to note that the definition of a 'smart car' used in this simulation — as a vehicle that simply does not apply random braking — is a simplified abstraction. In reality, autonomous vehicles may exhibit a range of advanced behaviors such as smoother acceleration/deceleration profiles, predictive modeling of traffic flow, and vehicle-to-vehicle (V2V) communication, which enable coordinated movement and improved safety. Incorporating such features in future simulations could offer a more nuanced understanding of their impact on traffic dynamics. These extensions could also allow the study of cooperative driving strategies and emergent behaviors in heterogeneous traffic systems.

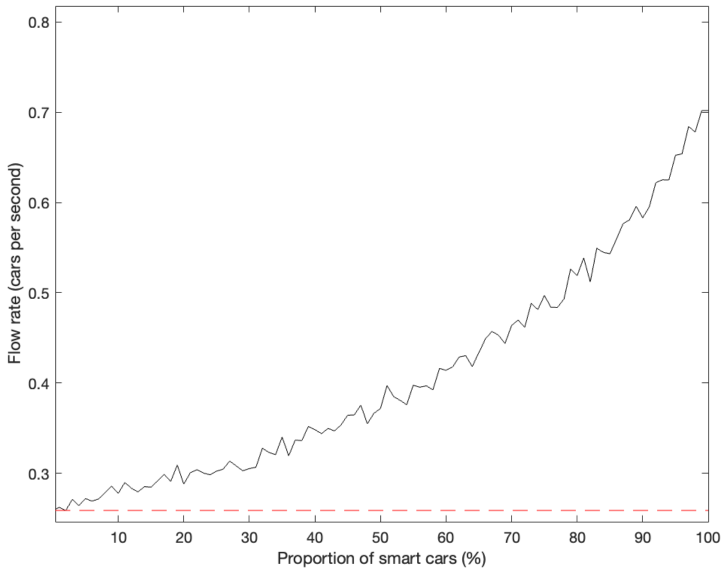

Figure 7: Graph showing the effect of replacing human-driven cars with smart cars on the flow rate. The road has one lane and a car density of 0.2 cars per site.

Figure 7 illustrates a significant increase in flow rate due to the introduction of smart cars. When around 70% of the vehicles on the road are 'smart', the traffic flow rate effectively doubles. The red dashed line represents the baseline flow rate in a system with no smart cars.

The difference between human-driven and smart cars is most evident because a substantial portion of traffic slowdowns results from the random braking behavior of human drivers. Replacing them with smart vehicles slightly reduces congestion formation beyond a critical density of $1/v_{max}$, since at this threshold, vehicles are forced to slow down at some point. However, traffic jams that would normally emerge due to random braking disappear when these vehicles are replaced with smart cars.

Next, a trial was run on a two-lane system with a density of 0.5, which corresponds to an ideal spacing of 4 sites per vehicle. Figure 8 shows that this system behaves very similarly to the single-lane configuration in Figure 7 . The flow rate of this system doubles when smart cars constitute approximately 90% of traffic. Systems with densities lower than this saturation threshold show even greater improvements as smart cars are introduced.

This result is particularly important for real-world applications, as they often improve flow rate

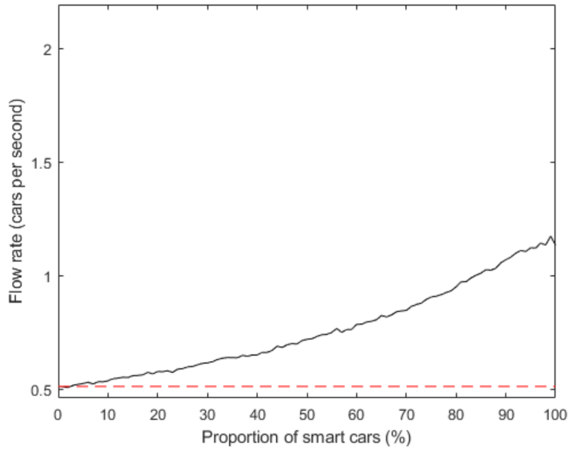

Figure 8: Graph showing the effect of replacing human-driven cars with smart cars on the flow rate. The road has two lanes and a car density of 0.5 cars per site.

dramatically with only a small number of smart vehicles being added. It also suggests that mixed-autonomy systems—where smart and human-driven cars coexist—can already offer significant benefits before reaching full automation.

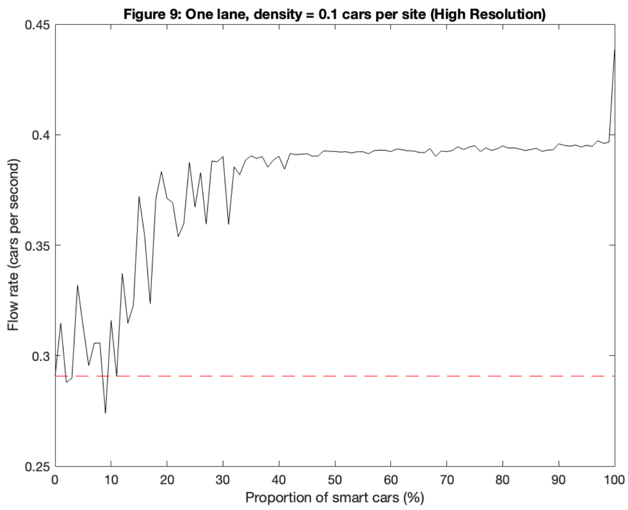

Figure 9: A one-lane system with a density of 0.1 cars per site.

The data presented in Figure 9 suggests that introducing just 25% smart cars leads to a flow rate increase of approximately 33%. Given the growing popularity of self-driving vehicles, this result is quite relevant—it implies that traffic efficiency will improve as autonomous cars become more widespread. Obviously, this change in the proportion of self-driven cars will be small at first, which is why this large flow rate increase with only a quarter of the cars being smart is notable.

There is currently limited data available on how autonomous vehicles influence real-world traffic. However, related studies using similar models have been conducted. A study using reinforcement learning [5] concluded that, although the autonomous vehicles were able to improve the flow rate substantially for several road configurations, they were not able to arrange themselves into the most optimal configuration. This opens further avenues for incorporating more advanced behavior learning algorithms or communication between vehicles to reach even higher efficiency in mixed traffic systems.

5. Summary

The developed model demonstrated a fairly accurate representation of traffic flow. Simulations were performed for different numbers of lanes and vehicle densities, with initial car placements along the road being randomized. The movement of vehicles was dictated by a straightforward set of rules, which led to the emergence of complex behaviors such as traffic congestion.

For a single-lane road, a phase transition was observed at a density of 0.08 ± 0.01 cars per site. Beyond this point, any further increase in density resulted in a decline in the overall flow rate. The two-lane system exhibited a similar behavior, with a phase transition occurring around 0.20 ± 0.01 cars per site. As expected, the peak flow rate for the two-lane road was nearly double that of the single-lane case. The simulated flow rate versus density graph closely matched the patterns found in real traffic data.

These results underscore the effectiveness of the modified cellular automaton model and its capacity to replicate real-world traffic dynamics while providing a robust framework for testing autonomous and hybrid traffic scenarios.

These two datasets, both simulated and real, were used to align arbitrary time steps and road segment lengths with real-world metrics such as meters and seconds per hour. This allowed the model's outputs to be converted into practical, real-world traffic values.

At lower densities, the system reached a stable configuration where cars maintained even spacing with minimal interaction. However, as density increased, congestion became a dominant feature, with most vehicles being stuck in recurring traffic jams. These jams persisted as fundamental characteristics of the road, spreading in a backward direction.

For very high densities, the system either failed to reach a stable state or required an impractically long time to do so. To address this, simulations were conducted on extended road lengths ranging from 250 to 1000 units. However, these high-density configurations were not particularly useful for modeling, as vehicles spent most of their time in congestion.

This modeling challenge arises due to the fundamental limitations of classical cellular automata (CA) in handling dense traffic conditions. The simple rule-based logic, operating on discrete cells and time steps, makes CA systems highly sensitive to minor perturbations such as sudden deceleration. In high-density regimes, these perturbations rapidly accumulate, leading to the formation of persistent congestion clusters that the model struggles to resolve. The localized nature of the interaction rules also prevents effective anticipation or cooperative behavior between vehicles, which are essential under saturated flow conditions.

To address this, future versions of the model could incorporate extended decision rules that account for multi-cell lookahead or probabilistic safety margins during braking and lane switching. Furthermore, hybrid integration with other computational intelligence techniques—such as fuzzy logic, agent-based modeling, or reinforcement learning—could enhance the model's adaptability and realism, especially under extreme density scenarios. These improvements would, however, require balancing the model's complexity with computational efficiency and interpretability.

On the other hand, low-density scenarios were of greater interest, as they allowed for a more detailed examination of individual vehicle behavior.

Potential improvements to the model could include incorporating additional traffic control elements, such as traffic lights and intersections, to study their impact on traffic dynamics. Additionally, the current model lacks support for non-periodic roads without significant modifications to the distance calculation functions.

Moreover, the simulation highlighted the contrast between low- and high-density regimes. In low-density conditions, vehicles tend to spread out, producing a steady-state configuration with little interaction. In contrast, at high densities, vehicles become locked in jams, leading to a breakdown of flow and persistent backward-propagating congestion waves. These jammed states rarely stabilized and often required road lengths of up to 1000 cells to observe significant trends.

While the current model effectively captures fundamental traffic behaviors, it can be further expanded. Future improvements may include support for non-periodic roads, simulation of dynamic elements such as traffic lights, roundabouts, or merging lanes, and integration with real geographic data using GIS. Additionally, incorporating driver variability, adaptive cruise control, or vehicle-to-vehicle communication could make the system more realistic.

For example, traffic lights could be integrated into the CA framework by introducing fixed or adaptive stop conditions at specific sites, simulating intersection control logic. This would enable analysis of phase timing, queue lengths, and the impact of signal coordination (e.g., green waves).

Vehicle-to-vehicle (V2V) communication, on the other hand, could be implemented by allowing cars to access information from multiple neighboring cells, effectively simulating a distributed sensing system. This would allow for anticipatory behavior (e.g., earlier braking) and enable testing of cooperative driving strategies, which are essential in highly automated or mixed-autonomy environments.

Such extensions would increase the model's applicability for urban planning, traffic optimization, and autonomous vehicle testing [6-11].

6. Key Results and Quantitative Evaluation

The proposed model achieved consistent and reproducible quantitative outcomes. For instance:

- In the single-lane scenario, a critical phase transition was detected at density $\rho_c \approx 0.08$, aligning with theoretical expectations from the NS model.
- The model demonstrated agreement with empirical data (see Figure 6), where the simulated peak flow matched the real-world peak near 2500 vehicles/hour, validating the scaling assumptions.
- The comparative experiments between one-lane and two-lane roads revealed a twofold increase in capacity under optimal conditions, highlighting the benefit of lane multiplicity in dense traffic systems.

Declaration on Generative AI

The authors have not employed any Generative AI tools

References

[1] A. Yuen, R. Kay, Applications of cellular automata, International Journal of Artificial Intelligence in Education (2009) 37–58.

[2] K. Nagel, M. Schreckenberg, A cellular automaton model for freeway traffic, Journal de Physique I 2 (1992) 2221–2229.

[3] S. Wolfram, Theory and Applications of Cellular Automata, World Scientific, Singapore, 1986.

[4] M. Rickert, K. Nagel, M. Schreckenberg, A. Latour, Two lane traffic simulations using cellular automata, Physica A: Statistical Mechanics and its Applications 231 (1996) 534–550.

[5] E. Vinitsky, K. Parvate, A. Kreidieh, C. Wu, A. Bayen, Benchmarks for reinforcement learning in mixed-autonomy traffic, in: Proceedings of the 1st Conference on Robot Learning, volume 78 of *PMLR*, 2017, pp. 517–528.

[6] H. F.L., B. L.A., G. M.A., Speed-flow relations for partially congested highways, Transportation Research Part A: General 20 (1986) 197–210.

[7] B. S. Kerner, S. L. Klenov, D. E. Wolf, Cellular automaton approach to three-phase traffic theory, Journal of Physics A: Mathematical and General 35 (2002) 9971–10013.

[8] Y. Zheng, J. Wang, K. Li, Smoothing traffic flow via control of autonomous vehicles, arXiv preprint (2018). arXiv:arXiv:1812.09544.

[9] Y. M. Wang, L. S. Zhou, Y. B. Lu, A divided two-lane cellular automaton model of traffic flow, Journal of Advanced Transportation 42 (2008) 179–206.

[10] B. S. Kerner, Three-phase traffic theory and highway capacity, Physica A: Statistical Mechanics and its Applications 392 (2013) 5261–5282.

[11] B. S. Kerner, Breakdown minimization principle for freeway traffic control, Transportation Research Part C: Emerging Technologies 77 (2017) 245–260.