IMS Project Documentation

Traffic Simulation Using Cellular Automata

Analysis of a Congested Intersection in Brno

Authors:

Michal Repčík (xrepcim00)

Adam Veselý (xvesela00)

Faculty of Information Technology
Brno University of Technology

December 6, 2025

Contents

1 Introduction

The traffic situation in Brno is heavily influenced by high vehicle intensity during weekday peak hours (15:00–17:00). This project builds a cellular automaton (CA) traffic model to evaluate how modifying a selected intersection would affect throughput and congestion.

The simulation is implemented in C++ using a two-dimensional cellular automaton based on the Nagel–Schreckenberg model [1], including acceleration, braking, random slowdowns, turning behavior, and traffic light interactions.

2 Simulated Intersection

The chosen location is the congested **Zvonařka–Dornych** intersection in central Brno, selected for its high traffic volumes, available traffic data, and potential for infrastructure improvements. The modification involves adding one additional straight lane to the eastbound approach (from 3 to 4 lanes).

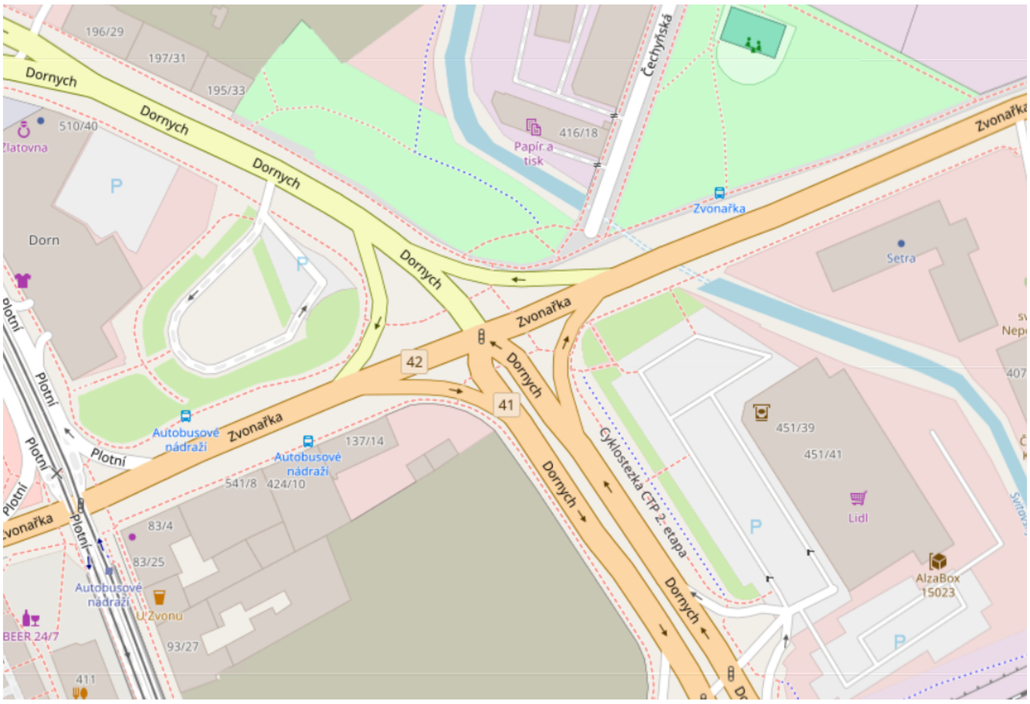


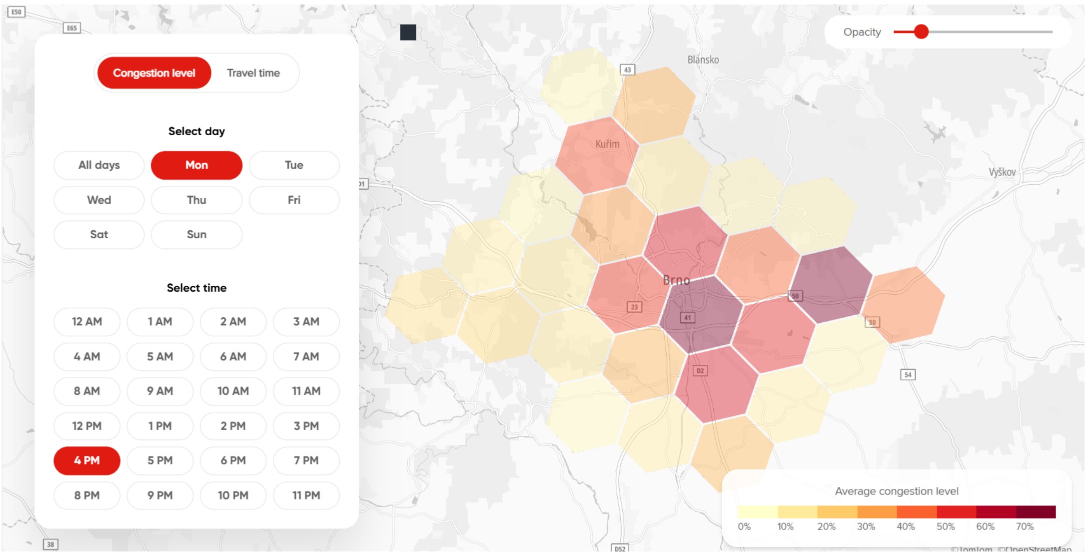
Figure 1: Zvonařka–Dornych intersection layout (OpenStreetMap [2]).

3 Data Sources

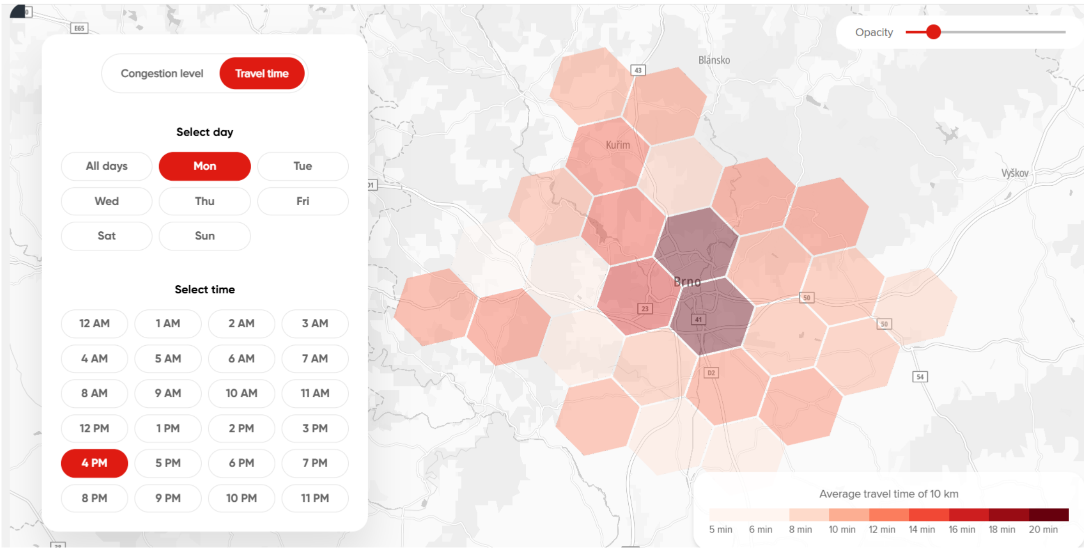
All traffic demand parameters were derived from publicly available sources to ensure realistic calibration.

3.1 TomTom Traffic Index

TomTom provides congestion and travel time data for Brno [3]. Figure 2 shows congestion patterns for Monday at 4 PM, corresponding to the peak hours simulated in this study.



(a) Congestion level heatmap

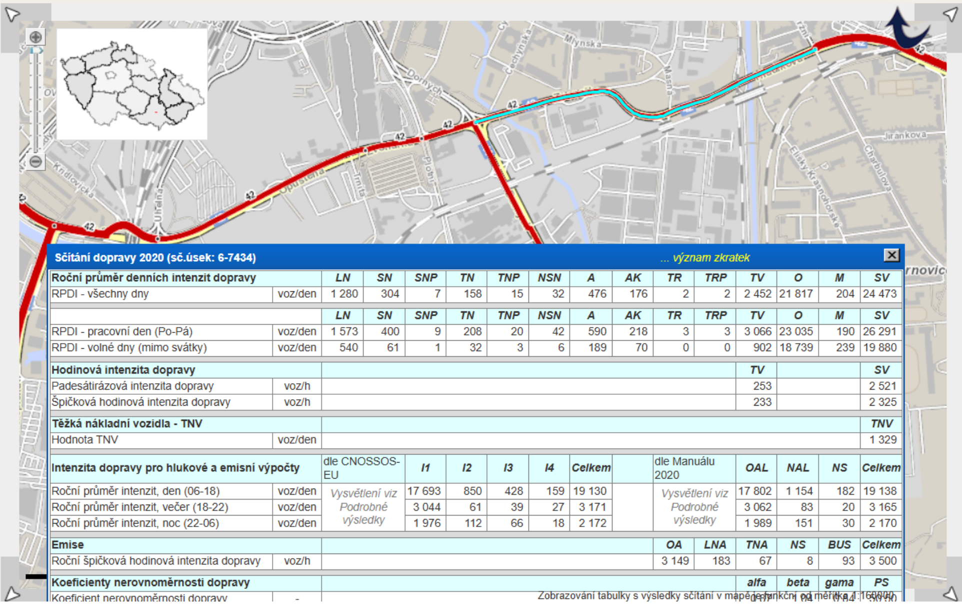


(b) Travel time heatmap

Figure 2: TomTom Traffic Index data showing congestion patterns on Monday at 4 PM. Darker/redder areas indicate higher congestion and longer travel times [3].

3.2 ŘSD ČR – Traffic Census

The Road and Motorway Directorate's traffic census provides detailed vehicle counts [4]. The data shows 26,291 vehicles/day on the eastbound approach, with peak hour intensities of 2,325 vehicles/hour.



Roční průměr denních intenzit dopravy		LN	SN	SNP	TN	TNP	NSN	A	AK	TR	TRP	TV	O	M	SV
RPDI - všechny dny	voz/den	1 280	304	7	158	15	32	476	176	2	2	2 452	21 817	204	24 473
		LN	SN	SNP	TN	TNP	NSN	A	AK	TR	TRP	TV	O	M	SV
RPDI - pracovní den (Po-Pá)	voz/den	1 573	400	9	208	20	42	590	218	3	3	3 066	23 035	190	26 291
RPDI - volné dny (mimo svátky)	voz/den	540	61	1	32	3	6	189	70	0	0	902	18 739	239	19 880
Hodinová intenzita dopravy												TV			SV
Padesátirázová intenzita dopravy	voz/h											253			2 521
Špičková hodinová intenzita dopravy	voz/h											233			2 325
Těžká nákladní vozidla - TNV															TNV
Hodnota TNV	voz/den														1 329
Intenzita dopravy pro hlukové a emisní výpočty		dle CNOSSOS-EU		I1	I2	I3	I4	Celkem		dle Manuálu 2020		OAL	NAL	NS	Celkem
Roční průměr intenzit, den (06-18)	voz/den	Vysvětlení viz Podrobné výsledky		17 693	850	428	159	19 130		Vysvětlení viz Podrobné výsledky		17 802	1 154	182	19 138
Roční průměr intenzit, večer (18-22)	voz/den			3 044	61	39	27	3 171				3 062	83	20	3 165
Roční průměr intenzit, noc (22-06)	voz/den			1 976	112	66	18	2 172				1 989	151	30	2 170
Emise										OA	LNA	TNA	NS	BUS	Celkem
Roční špičková hodinová intenzita dopravy	voz/h									3 149	183	67	8	93	3 500
Koeficienty nerovnoměrnosti dopravy												alfa	beta	gama	PS
Koeficient nerovnoměrnosti dopravy	-											[illegible]	[illegible]	[illegible]	[illegible]

Figure 3: ŘSD traffic census map for the Zvonařka–Dornych area [4].

3.3 NDIC – National Traffic Information Centre

The NDIC provides real-time traffic data and delay monitoring [5], confirming the congestion patterns observed in other sources.

4 Research Question and Hypotheses

Research Question: How does expanding the eastbound approach from 3 to 4 lanes (adding one straight lane) affect throughput, queue lengths, and congestion during weekday peak hours?

Hypotheses:

H1: Throughput increases by at least 20%

H2: Normalized queue length per lane decreases by at least 20%

H3: Average vehicle speed increases

5 Model Description

The Nagel–Schreckenberg model [1] applies four rules per timestep to each vehicle:

1. **Acceleration**: $v \to \min(v+1, v_{\max})$
2. **Slowing**: $v \to \min(v, d-1)$ where d is distance to next car
3. **Randomization**: $v \to \max(v-1, 0)$ with probability p (dawdling)
4. **Motion**: Move forward v cells

Our implementation extends this to two dimensions with multi-lane roads (100×100 grid), directional flow, dedicated turning lanes, traffic lights with multi-phase control, and comprehensive vehicle tracking.

5.1 Parameter Calibration

5.1.1 Spawn Rate Calculation

Peak-hour spawn rates were calculated from census data:

$$\text{Eastbound rate} = \frac{2{,}325\ \text{veh/h}}{3{,}600\ \text{s/h}} \approx 0.645\ \text{veh/s} \tag{1}$$

$$\text{Per-lane rate} = \frac{0.645}{3\ \text{lanes}} \approx 0.215\ \text{veh/s/lane} \tag{2}$$

$$\text{Capacity utilization} = \frac{0.215}{1.5} \approx 14.3\% \tag{3}$$

The 14% capacity utilization (well below theoretical maximum of 1.5 veh/s/lane at $v_{\text{max}} = 3$) allows realistic congestion formation at traffic lights rather than reaching fundamental flow capacity limits.

5.1.2 Key Parameters

Table 1: Simulation parameters

Parameter	Value
Grid size	100 × 100 cells
Cell length	5 m (car 4.5 m + gap 0.5 m)
Time step (Δt)	1 second
Max speed (v_{max})	3 cells/step (54 km/h)
Dawdling probability (p)	0.3
Initial density	0.5
Eastbound spawn rate	0.645 veh/s
Other directions spawn rate	0.2 veh/s each
Turning probability (mixed lanes)	0.4
Green durations	60–120 s (phase-dependent)
Simulation duration	3600 (1h)

6 Implementation

6.1 Software Architecture

C++ object-oriented design with key components:

- `Grid`: 2D cellular automaton, vehicle management, traffic lights
- `Cell`: Individual cells with car/traffic light/turn marker data

- `NSRules`: Nagel–Schreckenberg update logic
- `Logger`: Comprehensive data collection (5 CSV file types)
- `ArgParser`: Command-line argument parsing
- `Utils`: Turbo colormap visualization and PPM export functions

6.2 Grid Design Using Minecraft

Due to lack of suitable online grid editors for complex multi-lane intersections, the initial layout was designed using Minecraft. This allowed intuitive visualization, cell-by-cell placement, and rapid prototyping before implementation.

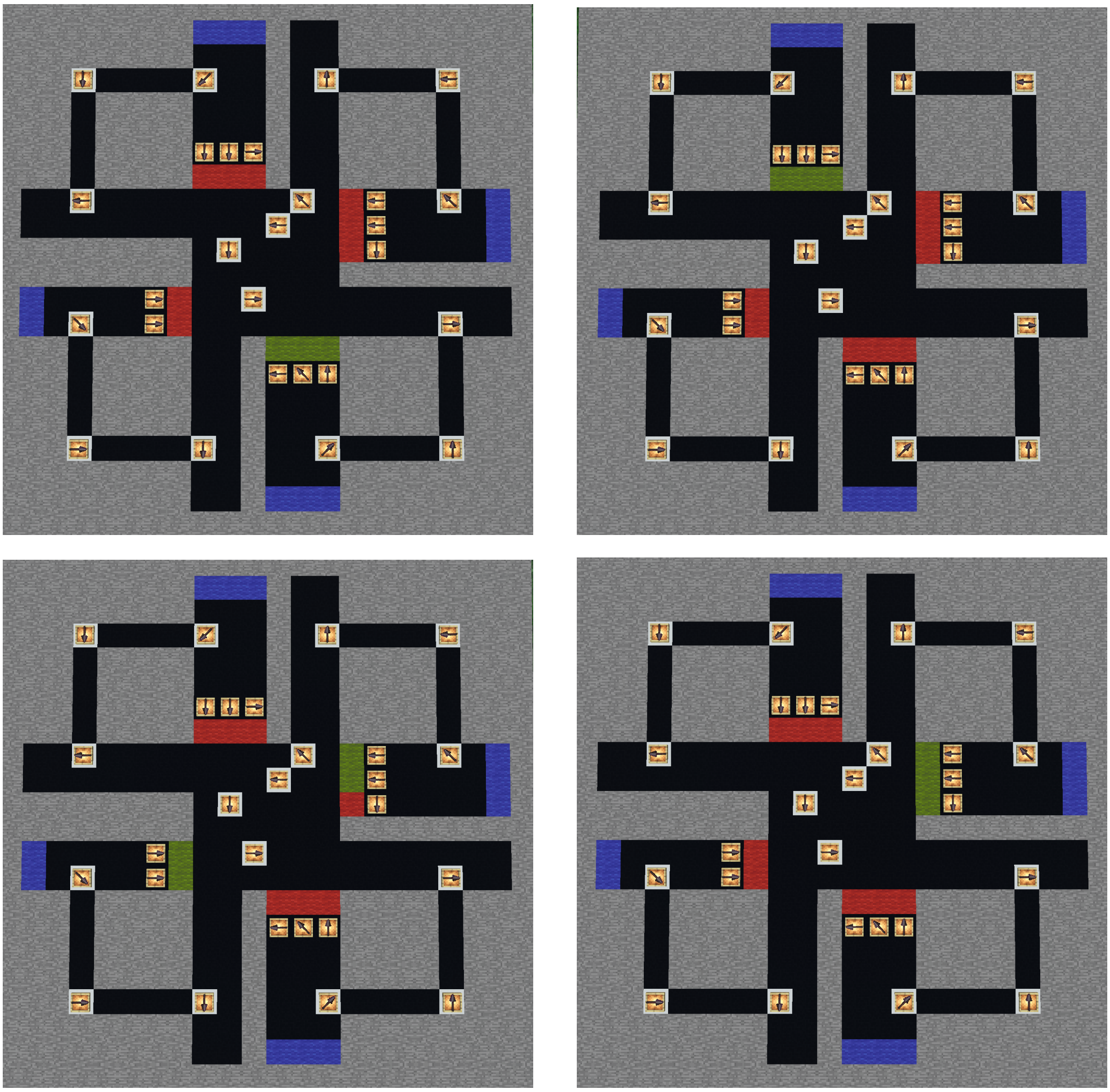


Figure 4: Minecraft-based intersection design with color-coded traffic lights, spawn points, turns and lanes

The final design includes:

- **North inbound**: 3 lanes (1 straight-only, 1 mixed, 1 turn-only)
- **South inbound**: 3 lanes (2 mixed, 1 turn-only)
- **East inbound**: 3 lanes baseline / 4 lanes modified (1/2 -only, 1 turn-only, 1 mixed)
- **West inbound**: 2 lanes (1 straight-only, 1 mixed)

7 How to Run

The project includes a `Makefile`, so compiling and running the simulation is straightforward.

- **Build:** `make`
- **Run simulation (3600 steps = 1h):** `make run`
- **Generate visualizations:** `make runvizmp4` or `make runvizgif`
- **Generate plots:** `make runplot`
- **Clean build:** `make clean`

All executable parameters (simulation length, visualization, plotting, output paths) are already handled inside the `Makefile`, so no manual parameter setup is required. For manual setup please refer to the attached **README.md** file or run the program with `-h/--help` flag.

8 Visualization Pipeline

If visualization is enabled (`-v`), the simulator outputs sequential PPM frames into the `viz/` directory. A video or GIF can then be generated automatically:

- `make runvizmp4` – runs the simulation and produces `output.mp4`
- `make runvizgif` – runs the simulation and produces `output.gif`

Plots of throughput, queue lengths, and waiting times are generated with:

- `make runplot` - user must be in a virtual environment or have required libraries installed, otherwise no graphs will be generated

This produces baseline and modified CSV data and automatically runs `plot_graphs.py` to generate the final graphs.

9 Experimental Setup

Two scenarios were compared with identical demand and signal timing:

1. **Baseline**: 3 eastbound lanes (1 mixed, 1 straight-only, 1 turn-only)
2. **Modified**: 4 eastbound lanes (1 mixed, 2 straight-only, 1 turn-only)

Both scenarios used the parameters from Table 1. Each simulation ran for 3600 steps (1 hour) with statistics collected throughout. Summary metrics were printed every 25 timesteps for monitoring, with final data exported to CSV files for analysis.

10 Results

The simulation was executed for the baseline setup and the modified rule set. Table 2 summarizes the key metrics and hypothesis evaluations.

Metric	**Baseline**	**Modified**	**Change**
Throughput [veh/min]	47.55	64.23	+35.1%
Queue length [veh/lane]	15.33	11.50	−25.0%
Avg. speed [cells/step]	0.665	0.703	+5.8%
Avg. waiting time [s]	110.5	100.5	−9.0%
Avg. time in system [s]	154.5	144.2	−6.7%

Table 2: Simulation metrics comparison.

10.1 Hypothesis Evaluation

The following hypotheses were evaluated based on the aggregated simulation results. Note that the exact values may vary slightly between runs due to inherent randomness in the model (e.g., spawn rates, turning probabilities, and the stochastic slow-down component of the Nagel–Schreckenberg model).

- **H1:** Throughput increases by at least 20%. **Confirmed** (≈35.1% increase)
- **H2:** Normalized queue length decreases by at least 20%. **Confirmed** (≈25.0% reduction)
- **H3:** Average vehicle speed increases. **Confirmed** (≈5.8% increase)

Note on Result Variability. The reported values represent typical outcomes but may vary between runs. This is expected due to the stochastic components of the model, including vehicle spawn randomness, turn selection probabilities, and the random slowdown step in the Nagel–Schreckenberg update rule. Multiple runs produce consistent trends, but individual metric values may differ.

10.2 Spatial Congestion Analysis

Figures 5 and 6 show the average vehicle speeds across the grid. The modified scenario exhibits substantially less congestion on the eastbound approach (lighter colors), with improved flow propagating through the entire intersection.

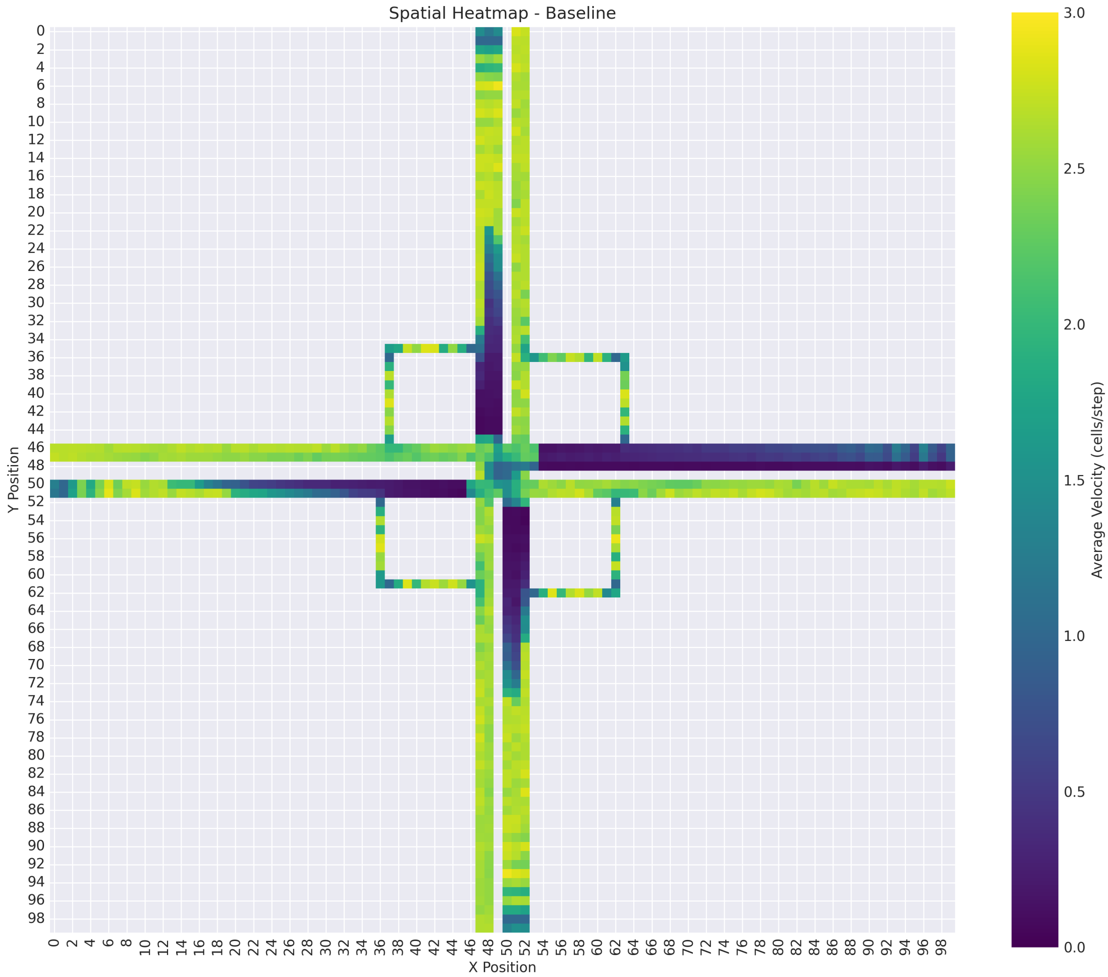


Figure 5: Baseline scenario (3 eastbound lanes). Darker colors indicate severe congestion. Generated from `spatial_heatmap.csv`.

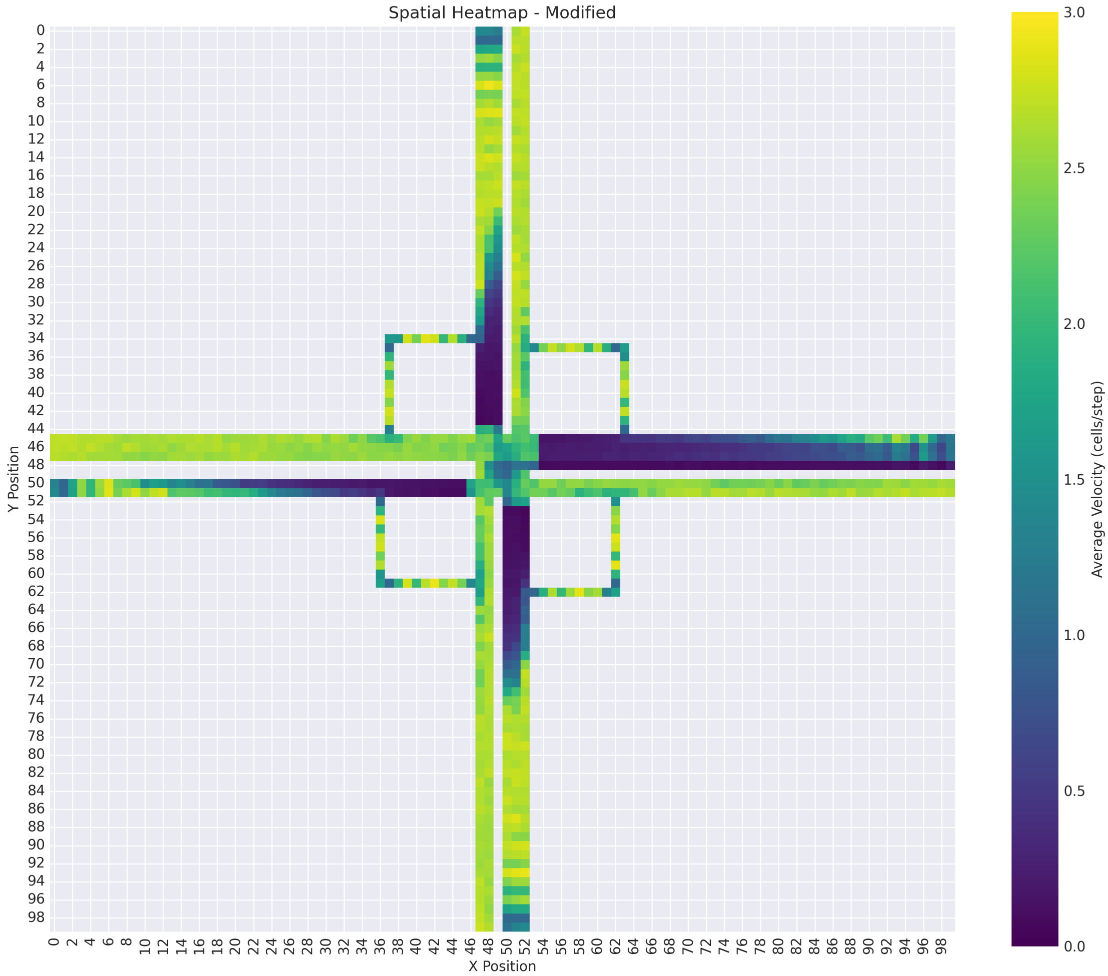
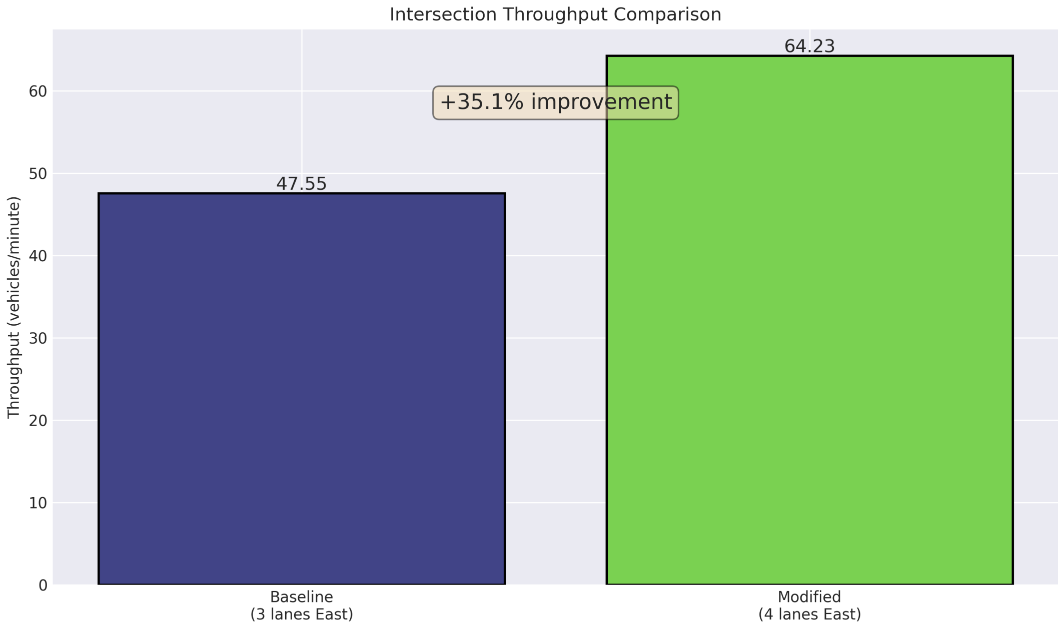
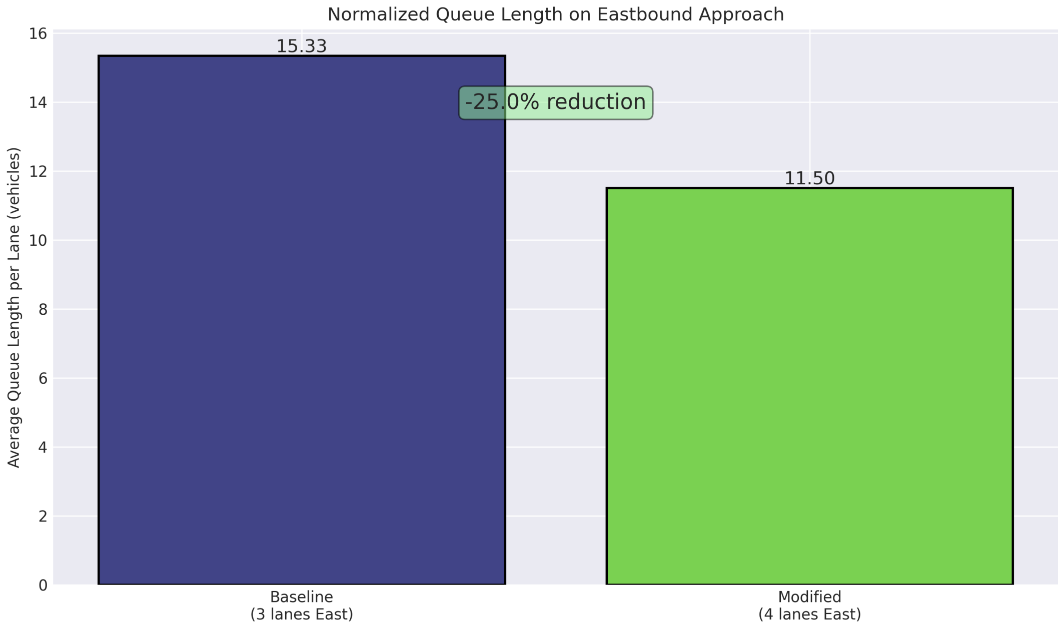


Figure 6: Modified scenario (4 eastbound lanes). Lighter colors indicate reduced congestion. Generated from `spatial_heatmap.csv`.

10.3 Performance Metrics



(a) Throughput improvement



(b) Queue length reduction

Figure 7: Key performance indicators from `summary_statistics.csv` showing substantial improvements in both throughput (+35.1%) and normalized queue length (-25.0%).

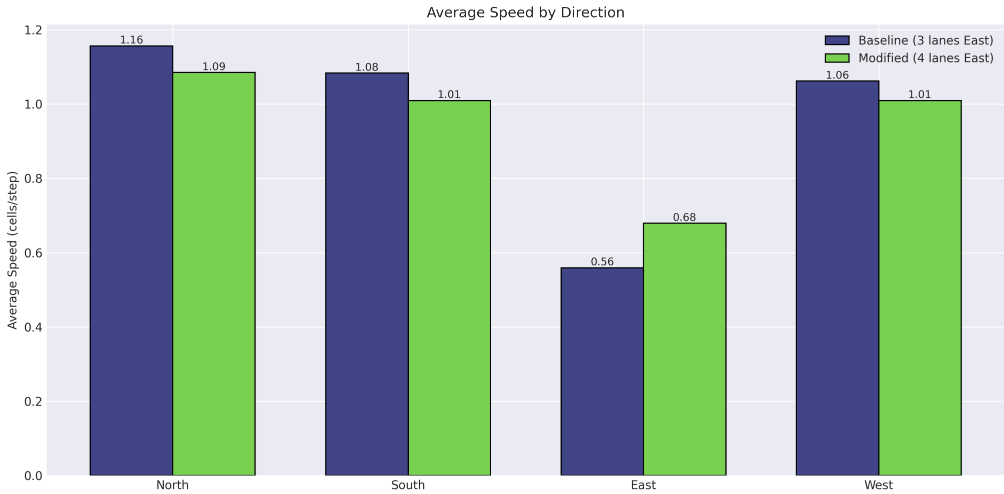


Figure 8: Average speed by direction from `timestep_metrics.csv`. The eastbound approach shows the most significant improvement, with positive effects on all directions due to reduced overall congestion.

10.4 Temporal Analysis

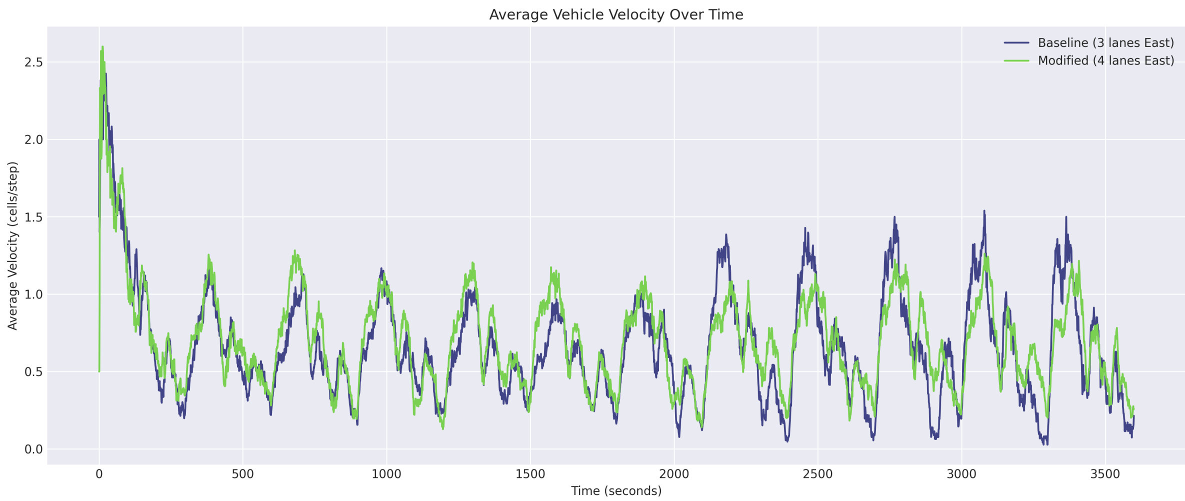


Figure 9: Average velocity over time showing consistently higher speeds in the modified scenario throughout the entire simulation period.

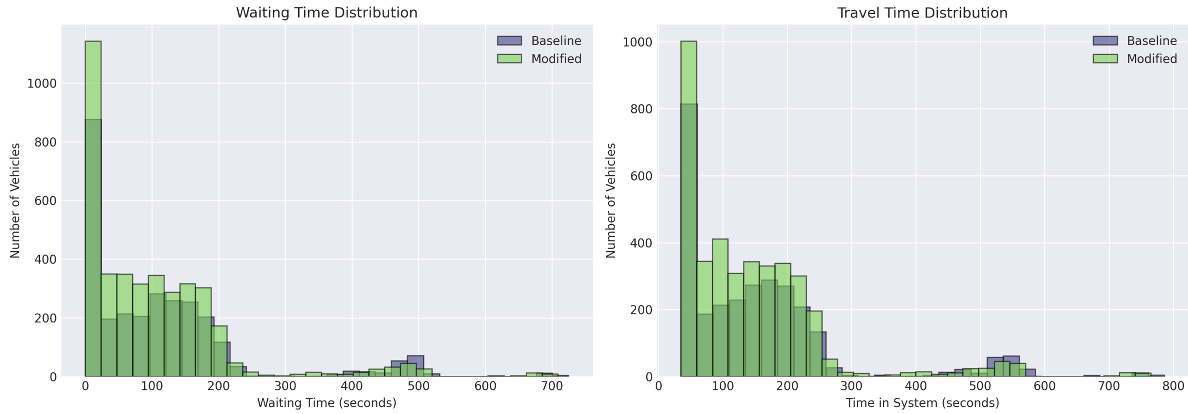


Figure 10: Distribution of waiting times (left) and total travel times (right) from `vehicle_trajectories.csv`. The modified scenario shifts both distributions toward lower values, benefiting all vehicles.

10.5 Visualization Examples

The complete visualization videos (MP4):

- https://nextcloud.fit.vutbr.cz/s/QsxPJzMJbpAyr9C.
- https://nextcloud.fit.vutbr.cz/s/NtYJTPM8fHpK9pf

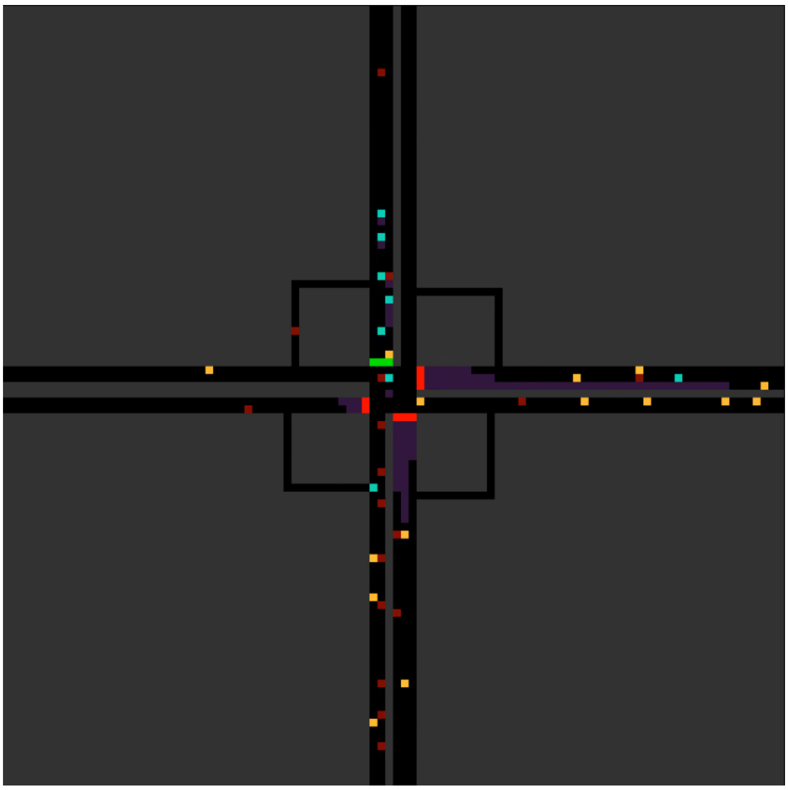


Figure 11: Example frame from the simulation visualization. The `turbo` colormap is used, where darker colors represent slower vehicles and brighter colors represent faster movement.

11 Conclusion

The cellular automaton simulation based on the Nagel–Schreckenberg model successfully demonstrated that adding one straight lane to the eastbound approach would significantly improve traffic flow during peak hours. All three hypotheses were confirmed with improvements exceeding the target thresholds.

11.1 Key Findings

- Adding a fourth eastbound lane increases throughput by ≈35.1% (exceeding H1's 20% target)
- Normalized queue length per lane decreases by ≈25.0% (exceeding H2's 20% target)
- Average speed increases by ≈5.8%, with waiting time reduced by ≈12.9%
- Benefits extend beyond the modified approach, improving flow throughout the intersection
- The improvement represents genuine congestion reduction, not just capacity distribution

11.2 Implementation Insights

- Cellular automata effectively model complex urban intersection dynamics
- Open data sources (ŘSD, TomTom, NDIC) enable realistic calibration
- Minecraft provided practical rapid prototyping for complex grid geometries
- The C++ implementation achieved good performance (with the exception of PPM file generation due to frequent IO operations)

11.3 Future Work

- Field validation with actual traffic measurements at Zvonařka–Dornych
- Network-level simulation with multiple interconnected intersections
- Adaptive traffic signal control (SCOOT, SCATS algorithms)
- Lane-changing behavior and overtaking maneuvers
- Mixed vehicle types (cars, buses, trucks, motorcycles)

References

[1] Kai Nagel and Michael Schreckenberg. A cellular automaton model for freeway traffic. *Journal de Physique I*, 2(12):2221–2229, December 1992. doi: 10.1051/jp1:1992277.

[2] OpenStreetMap contributors. Openstreetmap, 2025. URL https://www.openstreetmap.org.

[3] TomTom International BV. Tomtom traffic index 2024, 2024. URL https://www.tomtom.com/traffic-index/brno-traffic/.

[4] Ředitelství silnic a dálnic ČR. Celostátní sčítání dopravy 2020 (csd2020). National traffic census report, Road and Motorway Directorate of the Czech Republic, 2020. URL https://scitani.rsd.cz/.

[5] Ředitelství silnic a dálnic ČR. Národní dopravní informační centrum (ndic), 2025. URL https://www.dopravniinfo.cz.