SOITCS: Self-Organizing Intelligent Traffic Control Systems

Self-Organizing Systems 2025–2026

Student Name

January 18, 2026

Politehnica University of Bucharest

Introduction & Motivation

The Traffic Congestion Problem

- Urban traffic congestion costs billions annually in lost productivity
- Traditional fixed-timing signals cannot adapt to dynamic conditions
- Incidents and rush-hour patterns cause cascading delays

Why Self-Organization?

- Decentralized control enables local adaptation
- Emergent coordination without central authority
- Resilience to failures and changing conditions

Project Goals

Primary Objectives

1. Integrate 6 self-organizing algorithms into a unified simulation
2. Create an educational visualization tool
3. Demonstrate emergent behavior in traffic systems

Key Features

- Real-time traffic flow simulation
- Interactive algorithm comparison
- Incident injection and response
- Performance metrics dashboard

State of the Art

Foundational Approaches

Cellular Automata Nagel-Schreckenberg model (1992) captures realistic traffic flow with simple rules [1]

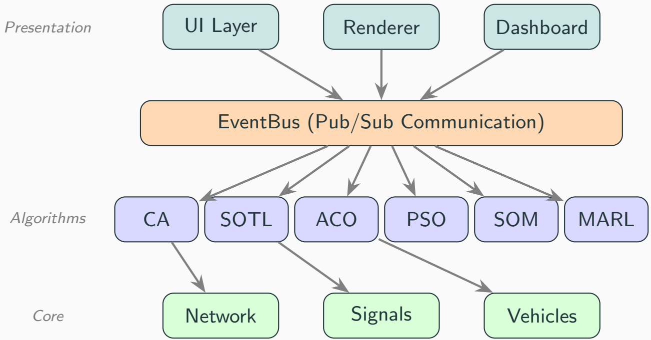
Self-Organizing Signals SOTL by Cools et al. demonstrates adaptive control via local queue sensing [2]

Swarm Intelligence ACO and PSO applied to routing and signal optimization [3, 4]

Deep RL Recent MARL approaches show promise for large-scale coordination [5]

Gap: Few systems integrate multiple paradigms for comparative study.

System Architecture



Event-driven architecture enables loose coupling and real-time coordination

Methodology: The Six Algorithms (1/2)

Cellular Automata

NaSch Model Rules:

1. Accelerate
2. Brake for obstacles
3. Random slowdown
4. Move forward

Realistic traffic flow from simple local rules

SOTL

Self-Organizing Signals:

- Monitor queue lengths
- Threshold-based switching
- Local adaptation only
- No global coordination

Emergent wave patterns

ACO

Ant Colony Routing:

- Pheromone trails
- Probabilistic path choice
- Evaporation mechanism
- Congestion avoidance

Distributed route optimization

Methodology: The Six Algorithms (2/2)

PSO

Signal Timing Optimization:

- Particles = timing configs
- Velocity updates
- Global/local best
- SOTL integration

Continuous parameter tuning

SOM

Pattern Recognition:

- Kohonen network
- Traffic state clustering
- Anomaly detection
- Visual state mapping

Unsupervised learning

MARL

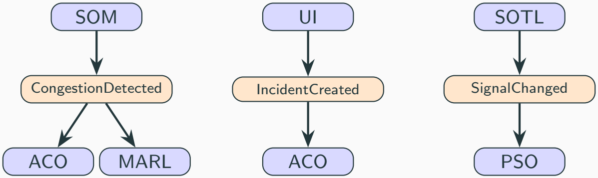
Q-Learning Agents:

- One agent/intersection
- State: queue + phase
- Reward: throughput
- Neighbor awareness

Adaptive control policies

Algorithm Integration

EventBus Communication



Key Integrations:

- SOTL + PSO: PSO optimizes SOTL threshold parameters (θ, t_{min}, t_{max})
- SOM $\rightarrow$ ACO: Pattern events trigger pheromone reduction (0.7$\times$ incident, 0.85$\times$ rush hour)
- SOM $\rightarrow$ SOTL: Pattern events adjust timing multipliers (e.g., rush hour: $t_{\text{max}} \times 1.3$)
- MARL + SOTL: MARL learns when to override SOTL decisions

Results: Visualization Features

Multi-Layer Rendering

- Road network with intersections
- Vehicle sprites with direction
- Traffic signal states
- ACO pheromone trails (heatmap)
- SOM cluster visualization

Interactive Controls

- Algorithm toggle switches
- Incident injection (click)
- Simulation speed control
- Scenario presets

Real-Time Dashboard

- Vehicle count and throughput
- Average speed metrics
- Per-algorithm statistics
- Congestion indicators
- Historical sparklines

[Screenshot of running simulation]

Results: Key Observations

Emergent Behaviors

- Traffic "green waves" form naturally with SOTL
- ACO pheromones create dynamic load balancing
- MARL agents develop coordination strategies

Algorithm Comparison

- SOTL: Fast adaptation, −29% delay
- ACO: Pheromone-guided routing, −30% delay
- SOM: Pattern detection enables integration
- Combined: −44.5% delay (synergistic)

Performance Metrics

Metric	Value
Frame Rate	30 FPS
Vehicle Capacity	100–500
Grid Size	Up to 10×10
Update Rate	60 Hz

Note: Educational focus prioritizes visualization clarity over maximum scale.

Results: Ablation Study (Additive Contributions)

Algorithm Contribution Analysis (340 runs, 20 seeds $\times$ 17 configurations)

Additive Contributions vs Baseline

(CA-only baseline: $\bar{d} = 36.0$ ticks)

Algorithm	**Δ Delay**	***p*-value**
SOTL	-29.4%	$< 0.001^{***}$
ACO	-29.9%	$< 0.001^{***}$
PSO	-1.5%	0.12
SOM	0.0%	–
Full Stack	-44.5%	$< 0.001^{***}$

$^{***}p < 0.001$, Welch's t-test vs baseline

Subtractive Importance

(delay impact when removed from full stack)

Removed	**Δ Delay**	**Sig.**
−SOTL	$+30.4\%$	***
−ACO	$+36.7\%$	***
−PSO	-7.6%	***
−SOM	$+11.3\%$	***

Key insight: SOM contributes via event-driven integration with SOTL/ACO, not direct control.

Hyperparameters & Experimental Setup

Algorithm Hyperparameters

Algorithm	Parameter	Value
CA	v_{max}	5 cells/tick
	p_{slow}	0.3
	lane change θ	2
SOTL	θ (queue)	5
	μ (platoon)	3
	$[t_{\text{min}}, t_{\text{max}}]$	[10, 60]
ACO	α (pheromone)	2.0
	β (heuristic)	1.5
	ρ (evaporation)	0.02
PSO	swarm size	30
	c_1, c_2	2.0
	w (inertia)	0.7
SOM	grid size	10×10

Experimental Protocol

- **Duration:** 5400 ticks (3 min sim time)
- **Runs:** 20 per configuration
- **Seeds:** 0–19 (sequential)
- **Network:** 4×4 grid (48 roads)
- **Road length:** 15 cells

Metrics Collected

- Average delay (ticks)
- Throughput (vehicles/tick)
- Average speed (cells/tick)
- Queue length at intersections

Scenarios Tested

- Normal traffic ($\rho = 0.08$)

Conclusions

Key Findings

- SOTL and ACO each reduce delay by ~30% independently
- Full stack integration achieves 44.5% delay reduction
- SOM's indirect contribution (+11.3% when removed) validates event-driven architecture
- Algorithm synergies: combined effect exceeds sum of individual contributions

Technical Contributions

- Event-driven pub/sub architecture enabling loose algorithm coupling
- Tuned ACO hyperparameters ($\alpha = 2.0$, $\beta = 1.5$, $\rho = 0.02$) for traffic domain
- SOM→ACO/SOTL integration via `PATTERN_RECOGNIZED` events

Future Work

- Hyperparameter optimization via grid search / Bayesian methods
- Extended scenarios: multi-incident, varying network topologies
- Deep RL integration (DQN/PPO) for MARL component

Questions?

Code: `https://github.com/username/soitcs`

References i

[1] Kai Nagel and Michael Schreckenberg. **"A cellular automaton model for freeway traffic".** In: *Journal de Physique I* 2.12 (1992), pp. 2221–2229. DOI: 10.1051/jp1:1992277.

[2] Seung-Bae Cools et al. **"Self-Organizing Traffic Lights: A Realistic Simulation".** In: *Advances in Applied Self-Organizing Systems*. Springer, 2013, pp. 45–55. DOI: 10.1007/978-1-4471-5113-5_3.

[3] Marco Dorigo and Luca Maria Gambardella. **"Ant colony system: a cooperative learning approach to the traveling salesman problem".** In: *IEEE Transactions on Evolutionary Computation* 1.1 (1997), pp. 53–66. DOI: 10.1109/4235.585892.

[4] James Kennedy and Russell Eberhart. **"Particle swarm optimization".** In: *Proceedings of ICNN'95 - International Conference on Neural Networks*. Vol. 4. IEEE, 1995, pp. 1942–1948. DOI: 10.1109/ICNN.1995.488968.

References ii

[5] Tianshu Chu et al. **"Multi-Agent Deep Reinforcement Learning for Large-Scale Traffic Signal Control".** In: *IEEE Transactions on Intelligent Transportation Systems* 21.3 (2020). arXiv:1903.04527, pp. 1086–1095. DOI: 10.1109/TITS.2019.2901791.