Computer Simulation of Traffic Flow Based on Cellular Automata and Multi-agent System

Magda Chmielewska, Mateusz Kotlarz, and Jarosław Wąs(✉)

Department of Applied Computer Science, Faculty of Electrical Engineering, Automatics, IT and Biomedical Engineering, AGH University of Science and Technology, Al. Mickiewicza 30, 30-059 Kraków, Poland
jarek@agh.edu.pl

Abstract. The paper describes a microscopic simulation of traffic flow phenomenon based on Cellular Automata and Multi-agent System. The simulation enables the study of the complexity of the traffic system and can provide current information about road capacity. A car is represented as a set of several neighboring cells, as an extension of Nagel-Shreckenberg model devoted for urban traffic simulation. The car is represented as an agent whose decisions are based on the actual situation on the road including neighboring cars decisions. The model also contains traffic lights and mechanisms such as right-of-way, route planning or lane changing which help to simulate more complex behavior of vehicles.

Keywords: Simulation · Traffic flow · Cellular Automata · Agent-based model

1 Introduction

Nowadays computer simulations are often chosen to study plenty of phenomenons. It is a safe way to expand our knowledge without conducting many expensive experiments. We can simulate phenomenons which are too difficult or simply cannot be recreated in the field. One of these phenomenons is traffic flow. Many scientists have become interested in methods, that would give the best traffic model and provide similar results in real life. We are going to mention two models that we have found especially interesting.

Nagel-Schreckenberg model [6] is a theoretical model of freeway traffic based on Cellular Automata where the car is represented as one cell and contains information only about velocity. In every iteration the car moves forward the number of cells that is equal to its velocity. The model is too simple to simulate all of the traffic occurring in the city, because it doesn't simulate any road regulations or the behavior of cars on crossroads. Another interesting model is proposed by Rolf Hoffmann [5]. It is based on Global Cellular Automata (GCA) with access algorithms to model traffic. This model uses dynamic links between

R. Wyrzykowski et al. (Eds.): PPAM 2015, Part II, LNCS 9574, pp. 517–527, 2016.
DOI: 10.1007/978-3-319-32152-3_48

potential global neighbors and gives us the possibility to change our neighbors state, something that is not allowed in classical CA.

In this paper, the authors present the model of traffic flow based on Cellular Automata and Multi-Agent System [1,3,7,10] containing simulation of cars and traffic lights. The result of this work is an application based on modificated Nagel-Schreckenberg model - extended by us to be used in urban environment.

2 Proposed Model

The main goal was to create traffic model with simple update rules, which takes into account all urban circumstances such as traffic lights, changing lanes etc. We propose discrete, non-deterministic, rule-based model, which extends well-known Nagel-Schreckenberg model [6].

The Nagel-Schreckenberg model was originally designed for freeway traffic. Hence it has some disadvantages in case of urban traffic simulation. The first one is unrealistic acceleration and deceleration of vehicles. In the NaSch model the system updates every second ($dt = 1\,\mathrm{s}$). This means that acceleration or deceleration rate of each vehicle equals $7.5\,\mathrm{m/s^2}$. This value is too high to represent urban traffic where drivers accelerate and brake more smooth in comparison to the highways (Table 1).

Table 1. Acceleration and deceleration rates (based on [2,9]).

Source	Typical values ($\frac{m}{s^2}$)
ITE (1982)	Maximal acceleration: 1.5–3.6
	Maximal deceleration: 1.5–2.4
	Normal deceleration: 0.9–1.5
Gipps (1981)	Normal acceleration: 0.9–1.5
	Maximal breaking: 3.0
Firstbus (private communication)	acceleration (buses): 1.2–1.6
Intelligent Driver Model (IDM)	Acceleration: 1.0
	Comfortable deceleration: 1.5

In order to adapt the model to city conditions, it was necessary to propose several modifications.

2.1 Road Representation

We propose to divide the road network into smaller cells (1 m). Due to such a spatial discretization each vehicle has to occupy more that one cell at a time-step (according to its length). This solution is inspired by the work [4]. Each car is divided into smaller parts: a *head* and a *tail* (other pieces). The most important part is the *head* which determines the position of the car on the road. This approach enables to distinguish several types of vehicles (for instance cars,

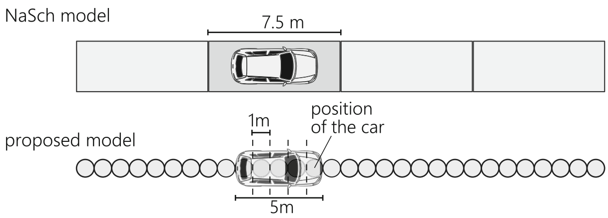

Fig. 1. A comparison between NaSch model and modified model.

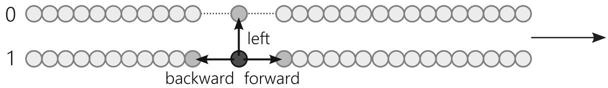

Fig. 2. Connections between cells at the road segment.

trucks, buses or HGVs). Each vehicle occupies as many cells as it is long (rounded up to the nearest number). For instance, if the car is 4.5 m long, it occupies 5 cells (Fig. 1).

The whole road network is composed of single roads. Each road consists of single sections which are divided into single cells. Road segments need to be connected in order to enable traffic flow (Fig. 2).

At single road segment each cell has a connection with its neighbors, i.e. the next cell, the previous one, and the cells on the left and right (if they exist). In this approach the road network is similar to the graph where cells correspond to the graph nodes (also called vertices) and connections between cells correspond to graph edges.

2.2 Modification of NaSch Rules

In the proposed model the system updates every 500 ms ($dt = 0.5$ s). The possible speeds of vehicles are shown in Table 2. In order to obtain the acceleration rates of 1–2 m/s^2 (for urban traffic) we introduced decimal fractions to represent speed and acceleration of each vehicle.

We propose following modification of update rules:

1. **Randomization** (with the probability $p = 0.15$):

$$behaviour_{t+1} = \begin{cases} max(behaviour_{t+1} - 2, -2) & \text{if the vehicle is entering the road,} \\ behaviour_{t+1} & \text{otherwise.} \end{cases} \tag{1}$$

2. **Accelerating or breaking:**

$$v_{t+1} = \begin{cases} max(v_t - dec, 0) & \text{if } behavior < 0, \\ v_t & \text{if } behavior = 0, \\ min(v_t + acc, v_{max}) & \text{if } behavior > 0, \end{cases} \tag{2}$$

Table 2. Speed discretization in the proposed model (n – number of cells, dt – time step).

Speed $[\frac{n}{dt}]$	Speed $[\frac{m}{s}]$	Speed $[\frac{km}{h}]$
10	20	72
9	18	64.8
8	16	57.6
7	14	50.4
6	12	43.2
5	10	36
4	8	28.8
3	6	21.6
2	4	14.4
1	2	7.2

where acc – acceleration which depends on a type of vehicle, dec – deceleration which depends on the road situation.

3. **Vehicle movement:**

$$x_{t+1} \rightarrow x_t + v_{floor}, \tag{3}$$

where v_{floor} is a speed rounded down.

We introduced variable *behavior* which represents the behavior of each vehicle. Values less than 0 represent braking, 0 means that vehicle should keep its speed, 1 means that it can accelerate (if the maximum velocity is not reached). Randomization probability was by default set to 0.15.

2.3 Modeling of Car Behaviour

Each vehicle treats other drivers on the road like obstacles. There are two kinds of obstacles:

– static – obstacles which don't move (velocity always equals 0),
– dynamic – obstacles which can move (velocity ≥ 0).

Visibility of the obstacle depends on its type. There are obstacles which are visible for every vehicle (e.g. cells occupied by other vehicles or traffic lights). The other obstacles can be ignored by the approaching vehicle under certain conditions (e.g. extreme points which are visible only for vehicles that have been assigned to the specific route).

Behavior of each vehicle depends on the obstacles which are within its sight:

$$s(n) = db_{normal}(v_n) + v_n, \tag{4}$$

where db_{normal} – normal braking distance (from v_n to 0), v_n – speed of vehicle n.

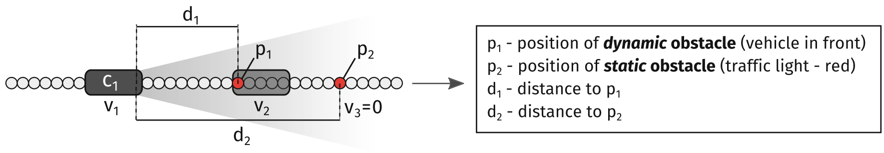

Fig. 3. The example of two obstacle types – dynamic (p_1) and static (p_2).

At this distance the car is looking for obstacles of both types. If some obstacle is found, the *real distance* is calculated. In case of dynamic obstacle, the real distance is the distance plus its emergency stopping distance:

$$d_{real} = d + db_{em}(v_{obstacle}), \tag{5}$$

where d – distance to obstacle, db_{em} – emergency braking distance (from $v_{obstacle}$ to 0), $v_{obstacle}$ – velocity of obstacle (equals 0 if obstacle is static) (Fig. 3).

Finally, the behavior of each vehicle is based on its speed and the minimum of calculated real distances to obstacles that were found (if both types of obstacles have been found):

$$d_{real} = min(d_{real_1}, d_{real_2}), \tag{6}$$

where: d_{real_1}, d_{real_2} – real distances to found obstacles.

2.4 Changing Lane and Route Assignment

There are two motivations to lane change maneuver:

- profitable – when it enables to keep higher speed and pass some slower cars ahead,
- necessary – in order to reach some point of the road network (and it's not possible from the current lane of the road) (Fig. 4).

Regardless of driver's motivation the safety criterion has to be satisfied before the maneuver is allowed:

$$\begin{cases} b(v_b) + d_1 - len > b(v), \\ b(v) + d_2 > b(v_f), \end{cases} \tag{7}$$

where $b(v_b)$, $b(v)$, $b(v_b)$ – emergency braking distance of vehicles c_b, c and c_f, d_1 – distance between c_b and c, d_2 – distance between c and c_f, len – length of vehicle c.

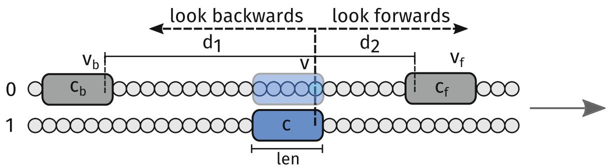

Fig. 4. The example of change lane maneuver.

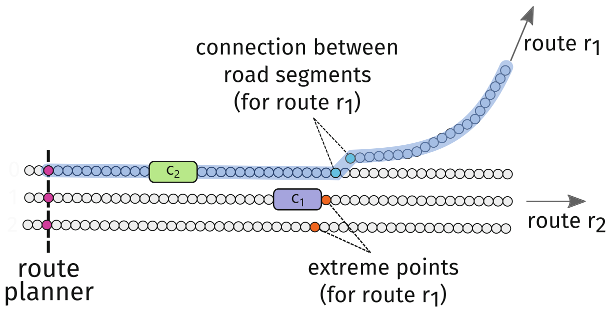

Fig. 5. The example of route assignment. Vehicles c_1 and c_2 are following route r_1. Vehicle c_1 has to change lane in order to follow assigned route.

Necessary lane changing is connected with route planning. Routes are assigned to vehicles approaching some crossroads. The point of such assignment is so-called route planner (see Fig. 5). For each route there are also some extreme points, which define the last possible road cells where the lane has to be changed. If lane change maneuver is not allowed (due to safety criterion which is not satisfied) the car has to stop before the extreme point and wait until the road is clean.

2.5 Right-of-way

In an urban area there are a lot of intersections without traffic lights where drivers must yield right-of-way. In proposed model such points are called priority points. There are two kinds of priority points:

– crossing points – places where vehicles enter the main road,
– entering points – places where vehicles only cut some other road lane.

This cells are blocked when entering the road is not allowed due to approaching vehicles. A priority point is connected with one or more checkpoints which are situated on the main road at the collision points (see Fig. 6).

3 Simulation Results

In order to check proposed model two tests have been made. The first one was carried out to obtain the relation between main traffic characteristics (velocity, density and flow) by increasing traffic density. The second test was comparing the arrival time of the bus at each bus stop with the real schedule.

3.1 The First Test

In the application a three-lane road was created in a shape of the loop. In one point of the road a car generator was placed. On the opposite site of the

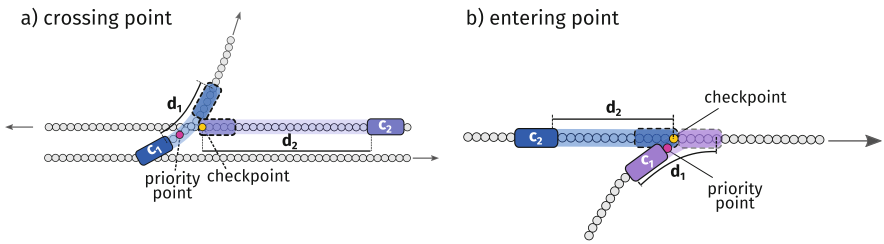

Fig. 6. The examples of both types of priority points (a – crossing point, b – entering point). Checkpoints are situated at the cross point of the roads.

loop the measurement point was placed. Measurement point has three counters: c_f (flow counter), v_c (velocity counter) and d_c (density counter) and it takes three parameters: dt_m (measurement interval) and d_m (measurement distance).

Three traffic flow characteristics (flow, velocity and density) were calculated in every measurement interval (formulas based on [8]):

$$J_n = \cdot \frac{60}{dt_m \cdot n_{lanes}} [veh/h/lane] \tag{8}$$

$$v_n = \frac{c_v}{c_f} [m/dt] \tag{9}$$

$$\rho_n = \frac{c_d}{dt_m \cdot 60 \cdot ips \cdot n_{lanes}} \cdot \frac{1000}{d_m} [veh/km/lane] \tag{10}$$

where dt_m – measurement interval, d_m – measurement distance, n_{lanes} – number of lanes, ips – iterations per second (Fig. 7).

The relation between traffic characteristics shows that the maximum flow (occurred by density about 45 vehicles per hour) is about 1800 vehicles per hour (at the single lane) and that the maximum density that can be reached is about 170 vehicles per kilometer (Fig. 8).

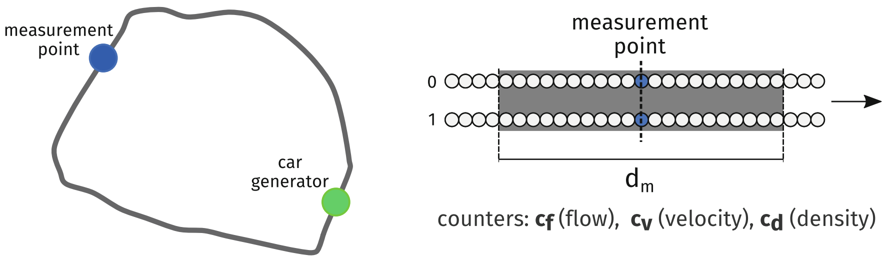

Fig. 7. The road designed for test purposes (left) and the example showing the placement of measurement point (right).

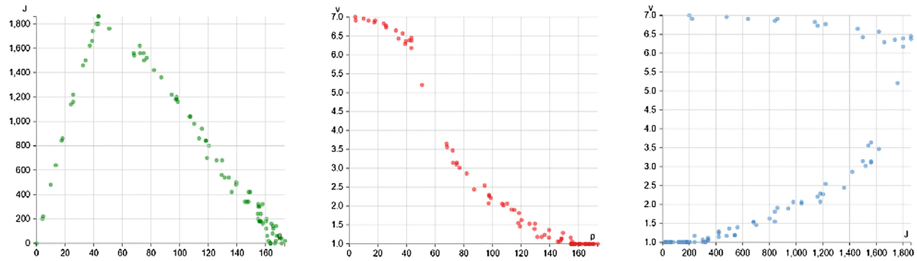

Fig. 8. Diagrams of relation between (from left): $J(\rho)$, $v(\rho)$, $v(J)$.

3.2 The Bus Test

The second part of model validation was comparing the time at each bus stop with the real schedule. All bus stops are shown in the Fig. 9. The whole route of the bus normally takes 20 min.

To compare the times, 10 single tests had been made. The results are shown in the Fig. 10. The minimum difference between schedule and simulation departure equals only 1 s (at the bus stop number five), and the maximum reached 70 s (at the first bus stop).

Fig. 9. All bus stops (line 168) in simulation area (in the order they are visited).

3.3 Traffic Jams in Simulation

The simulation showed that in some places traffic jams occur. Three examples of such locations are most interesting. The jams happen on yield-controlled intersections (without traffic lights) and they occurred in locations, which are also very problematic in reality.

It was useful and important observation, which shows that measured traffic data was correct and initial state of the simulation leads to the same traffic problems as in real life (Fig. 11).

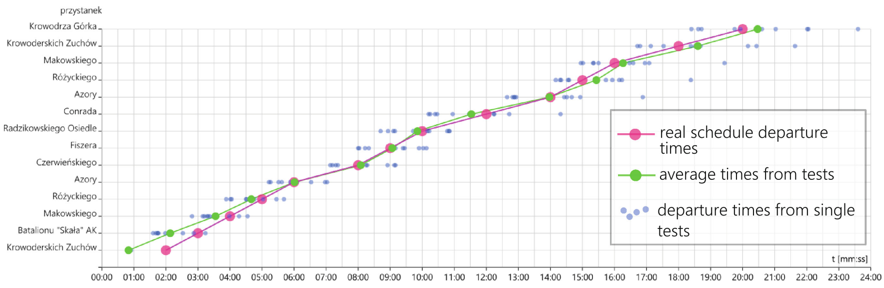

Fig. 10. Graph showing times of departure at each bus stop.

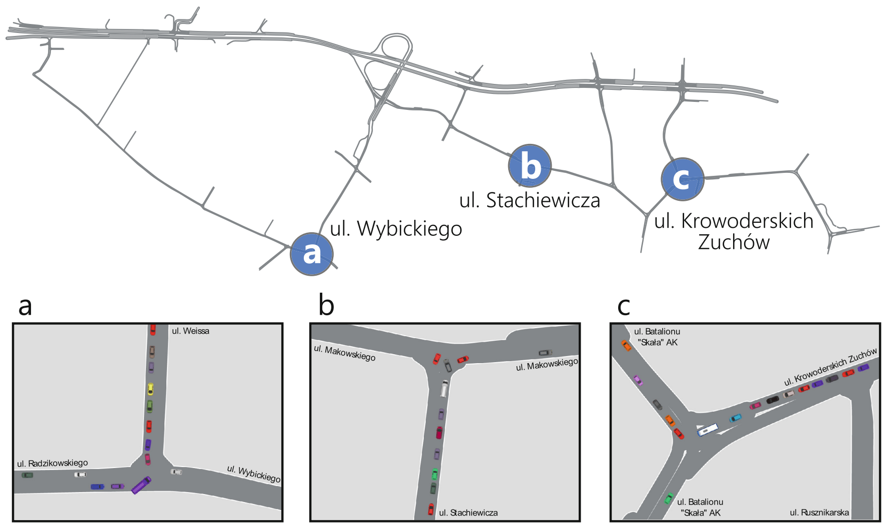

Fig. 11. Traffic jams occurring during simulation.

3.4 Simulation Performance

Although the application is divided into two separate threads (first thread is responsible for calculation and the second for visualization part) calculating of vehicle positions in next iteration is not parallel. After some modifications it could be implemented in parallel and for sure it would improve application performance.

The essential part of model testing was to measure the time of calculations. In order to obtain the maximum possible number of vehicles during the test we generated, at every generation point, as many vehicles, as possible (the number of vehicles reached over 1900). Obtained times (at each iteration of simulation) are plotted on a graph (Fig. 12). Time of calculation depends on simulation area – chosen area was the northwestern part of Cracow (Poland) with area containing roads with length of 33 km (considering each lane separately).

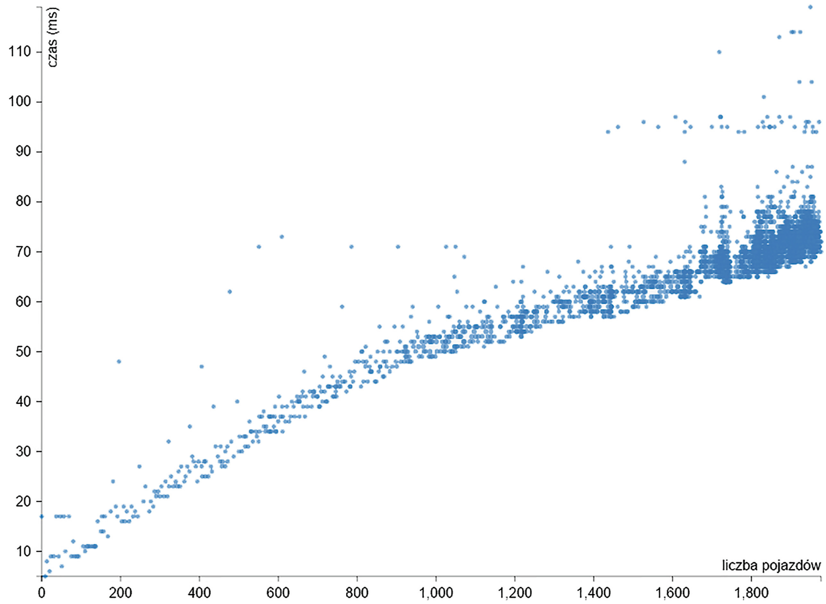

Fig. 12. The graph showing times of model calculations depending on number of vehicles.

4 Conclusions

The purpose of this paper was to develop a new method to study phenomena of traffic flow which combines Cellular Automata and Multi-Agent System. The model includes more elements which affect the traffic flow and represent the car as an autonomic unit whose decisions can affect other traffic participants decisions. This model is also enhanced with traffic lights, right-of-way and lane changing rules and route planning. The model can be further extended and the application can be modified to enable parallel calculations.

One of the most valuable advantages of the application is that it can be used as a powerful tool which can help to test different traffic scenarios on the roads (before they are even built). The biggest advantage of this solution is reduction of time, costs and resources.

References

1. Bonabeau, E.: Agent-based modeling: methods and techniques for simulating human systems. Proc. Natl. Acad. Sci. U. S. A. **99**(3), 7280–7287 (2002)
2. Bonsall, P., Liu, R., Young, W.: Modelling safety-related driving behaviour-impact of parameter values. Transp. Res. Part A Policy Pract. **39**(5), 425–444 (2005). http://www.sciencedirect.com/science/article/pii/S0965856405000261

3. Davidsson, P.: Multi agent based simulation: beyond social simulation. In: Moss, S., Davidsson, P. (eds.) MABS 2000. LNCS (LNAI), vol. 1979, pp. 97–107. Springer, Heidelberg (2001)
4. Hartman, D.: head leading algorithm for urban traffic modeling. In: 16th International European Simulation Symposium and Exhibition ESS, pp. 297–302 (2004)
5. Hoffmann, R.: GCA-w algorithms for traffic simuation. Acta Phys. Pol. B Proc. Suppl. **4**, 183–200 (2011)
6. Nagel, K., Schreckenberg, M.: A cellular automaton model for freeway traffic. J. Phys. I Fr. **2**(12), 2221–2229 (1992)
7. Płaczek, B.: A traffic model based on fuzzy cellular automata. J. Cell. Automata **8**, 261–282 (2013). ISSN 1557–5969
8. Schadschneider, A., Chowdhury, D., Nishinari, K.: Stochastic Transport in Complex Systems: From Molecules to Vehicles. Elsevier, Amsterdam (2011)
9. Treiber, M., Kesting, A.: Traffic Flow Dynamics. Springer, Heidelberg (2013)
10. Wąs, J., Bieliński, R., Gajewski, B., Orzechowski, P.: Problematyka modelowania ruchu miejskiego z wykorzystaniem automatów komórkowych. AUTOMATYKA **13**(3), 1207–1217 (2009)