Traffic Flow Simulation using Cellular automata under Non-equilibrium Environment

Hideki Kozuka, Yohsuke Matsui, Hitoshi Kanoh

Institute of Information Sciences and Electronics, University of Tsukuba, Tsukuba, Ibaraki, Japan 305-8573

Abstract

To evaluate dynamic route selection methods, we developed a traffic flow simulator that uses cellular automata in a non-equilibrium environment where traffic congestion occurs frequently. The simulator uses the S standard map of the Navigation System Researchers’ Association, which is the map used in actual car navigation devices, and produces environments where spontaneous traffic congestion occurs. The traffic flow model we propose is based on Nagel’s multiple-velocities model. Additionally, we propose an intersection model and a lane change model. We confirmed that our simulator works effectively for real road maps.

Keywords
Cellular automata, traffic flow, dynamic route selection method, simulation, car navigation, congestion.

1. Introduction

Car navigation devices are widely used as information terminals in Intelligent Transportation Systems (ITS) [7]. A provision for real-time information on traffic congestion began in 1996 in major urban areas in Japan. When traffic congestion occurs, the car navigation system finds an alternative route. Because existing car navigation systems have long calculating time, the detour sometimes can’t be displayed until after the car passes the intersection at which it should turn. Moreover, an unsuitable detour may be displayed.

We are investigating dynamic route selection methods for finding the easiest-to-drive and quasi-shortest route in real time and have developed a simulator for evaluating dynamic route selection methods under non-equilibrium environments where congestion occurs frequently.

This paper describes the simulator. The simulator uses the S standard map provided by the Navigation System Researchers’ Association, which is the map used in actual car navigation devices and produced environments where spontaneous traffic congestion occurs. This system can simulate any road in Japan. A fluid dynamical traffic flow model [5], a car-following model [6], and a cellular automata model have been proposed as traffic flow models. We use Nagel’s multi-speed model [1] with cellular automata [2]. This model regards a road as a belt that connects many cells, and represents the state of a road in terms of whether a car exists in each cell.

First, we show a supplementary examination of Nagel’s multi-speed model. Next, we describe traffic flow simulation using the S standard map. Finally, we describe the traffic flow simulation system and present experimental results.

2. Outline of research domain

2.1 Cellular automata

In cellular automata (CA), the state of a cell at a given time depends only on its own state one previous time step and the state of its nearby neighbors at the previous time step. In this paper, we use one-dimensional binary-state CA [2].

In one-dimensional CA, a major factor is how far one cell is from another. Let \( i \) be the position coordinate, \( t \) the time step, and \( a_i^t \) the state of the cell. A neighborhood consists of a cell and its \( r \) (“radius”) neighbors on either side. Thus the function \( F \) must have the form

\[
a_{i+1} = F(a_i, a_{i-r}, \ldots, a_{i-r+1}, \ldots, a_i^{t+r}, a_i^{t+r})
\]

Figure 1 shows an example of one-dimensional binary-state CA \( (r=1) \).

\[
\begin{array}{cccccccccccc}
\square & \square & \square & \square & \square & \square & \square & \square & \square & \square & \square & \square \\
\end{array}
\]

Figure 1: An example of one-dimensional binary-state CA \( (r=1) \)

If white is set to 1 and black is set to 0 in the figure, then

\[
\begin{array}{cccccccc}
111 & 110 & 101 & 100 & 011 & 010 & 001 & 000 \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\
\end{array}
\]

(1)

We look upon the lined up lower numbers 01011010 as binary numbers. These numbers are converted to
decimals, that is 90, and we regard the above (1) as the 90th rule. The 184th rule is used in the following traffic flow simulation.

2.2 Traffic flow simulation using CA

In Nagel's multi-speed model [1], a road is defined on a one-dimensional array of length $L$. The length of each cell is defined as the minimum distance between two cars. Each site of the array may be empty, or it may be occupied by one car having an integer velocity $v$. This integer number for the velocity is the number of sites each vehicle moves during one step. The choice of maximum velocity is somewhat arbitrary, but it can be justified by comparison between the model and real world measurements. One iteration consists of the following steps, which are performed simultaneously for all vehicles (gap is the number of unoccupied sites in front of a vehicle).

- **Acceleration of free vehicles:** Each vehicle of speed $v < v_{max}$ with $gap \geq v + 1$ accelerates to $v + 1$: $v \rightarrow v + 1$.

- **Slowing down due to other cars:** Each vehicle (speed $v$) with $gap \leq v - 1$ reduces its speed to $gap$: $v \rightarrow gap$.

- **Randomization:** Each vehicle (speed $v$) reduces its speed by one with probability $1/2$: $v \rightarrow max[v - 1, 0]$ (take into consideration individual fluctuations).

- **Movement:** Each vehicle advances $v$ sites.

This model is called the stochastic traffic cellular automaton (STCA) [4] because it uses probability. Moreover, if maximum velocity is set to 1, this model is 184th rule.

3. Proposed method

The S standard map has road characteristics such as speed limits, distances, road classes, the number of lanes, traveling direction, and the existence of signals. There are eight road classes: national roads, prefectural roads, principal prefectural roads, and so on. The traveling direction is expressed by 32 values, that is from 0 to 31. In this paper, a national road, prefectural road, or principal prefectural road is a main road. A road of any other class is a basic road.

We consider a route from a starting point to a destination. In real traffic, each car moves toward its destination. A car at an intersection moves according to the algorithm in Fig.2, where each car decides its traveling direction at random. There are two kinds of intersections. One has signals, and the other doesn’t.

(1) Transit rules at an intersection

The algorithm that determines the behavior of a car at an intersection is shown in Fig.2, where $gap_{facing}$ is the distance between a car approaching the intersection and a signal and $v_{facing}$ is the speed of the car.

```plaintext
if(with signals)
go or stop according to signal
else if(without signals)
  if(a preference road &
    (left-turn || straight drive))  go
  else if((a preference road & right-turn)
    || not a preference road)
    if(the car which had stopped)
      if($gap_{facing} \cdot v_{facing} > 0$)  go
    else  stop
  else  stop
Figure 2: The algorithm for determining behavior of a car at an intersection
```

We need algorithms that can determine the combination of signals because the S standard map only indicates the existence of signals. Our idea for such an algorithm is shown below.

(i) Three-road intersection

Step 1: calculate the difference of the direction of two roads respectively.

Step 2: select two roads that the absolute value of the difference is close to 16.

Step 3: let the two roads be one group and let the rest be another group.

(ii) Four-road intersection

Step 1: sort the roads according to the direction in small order

Step 2: let the 1st and the 3rd be one group.

Step 3: let the 2nd and the 4th be one group.

When the number of crossing roads is an odd number greater than 3, we decide the combination of signals by accordingly extending (i). Likewise, to decide the signal combinations for an even number of crossing roads greater than four, we extend (ii).

(2) Lane-change rules

- A car turning left at the next intersection must turn from the furthest left lane.
- A car turning right at the next intersection must turn from the furthest right lane.
- A car going straight at the next intersection can be in any lane.
- A car traveling behind a slow-moving vehicle can change lanes with a 20% probability. However, a car that turns at the next intersection can’t change lanes in front of the intersection.
- When a car wants to change lanes, it checks the distance and the speed of the car it wants to pass. If the speed of the car is too high, the car cannot
change lanes if it is traveling behind two cars in adjacent lanes traveling at the same speed.

The algorithm is shown in Fig. 3, where \( \text{gap}_{\text{back}} \) is the distance between two cars, \( v_{\text{back}} \) the speed of the back car, and \( v_{\text{forward}} \) the speed of the front car.

\[
\begin{align*}
\text{if (left-turn)} & \\
& \begin{cases} 
\text{if (gap}_{\text{back}} - v_{\text{forward}} > 0) & \text{change lane} \\
\text{else} & \text{not change lane}
\end{cases} \\
\text{else if (right-turn)} & \begin{cases} 
\text{if (gap}_{\text{back}} - v_{\text{forward}} < 0) & \text{change lane} \\
\text{else} & \text{not change lane}
\end{cases} \\
\text{else} & \begin{cases} 
\text{if (gap}_{\text{back}} - v_{\text{forward}} > 0 \& \text{v}_{\text{forward}} \geq 0) & \text{change lane} \\
\text{else} & \text{not change lane}
\end{cases}
\end{align*}
\]

Figure 3: Lane change algorithm

4. Experiment for a single lane

A simulation was carried out using the rules shown in section 2.2 to inspect the difference in traffic flows between road lengths \( L=100 \) (small system) and 10000 (large system).

Where one cell is 6.5 m, one time step is 1 sec, the maximum speed of cars is 5, probability \( p \) is 1/2 (refer to Section 2.2). The density \( (\rho) \) is the number of cells in which a car exists divided by the number of all cells. The traffic \( (q) \) is the number of cars that pass a certain cell within a fixed time.

![Figure 4: Density and traffic](image)

Figure 4 shows the relation between density and traffic. Here, we defined the number of cars that passed the 50th cell from \( t=0 \) to \( t=1000 \) as traffic. According to Fig. 4, this model reaches capacity (maximum throughput) \( q=415 \) at a density of \( \rho=0.1 \) for \( L=100 \) and it reaches capacity \( q=318 \) at a density of \( \rho=0.08 \) for \( L=10000 \). This means that short segments behave differently from long ones. Figure 4 also shows that traffic decreases at \( \rho=0.08 \). This is due to congestion.

5. Experiments using a real road map

5.1 Traffic volume

A simulation was carried out using the rules shown in section 3. The following data were obtained from a map of Tsukuba in Japan. We determine the traveling direction of cars so as to reflect the fact that more cars are driven on main roads than on basic roads. We first count the number of main roads. Then, the car travels on those roads with a 30% probability. Traveling direction is determined at random if a car doesn’t go those roads or a main road doesn’t exist. One cell is 5.5 m and one time step is 2 sec. The maximum speed is the speed limit of each road. The density on a basic road is 0.1 uniformly when the simulation starts. And we change the density of the main road from 0 to 1. The number of cars driven on a certain road is shown in Fig. 5.

In Fig. 5, traffic is maximum between 0.1 and 0.2. This is the same as in Fig. 4. Maximum traffic is 315 in Fig 4, but is only 72 in Fig. 5. This is because the data in Fig. 4 are for one road, but the data in Fig. 5 are for given categories, and because the density of the basic road was only 0.1 uniformly when simulation starts.

![Figure 5: Density of main road and traffic](image)

5.2 State of cells

We show the time series of the cell’s state in Fig. 6. When the simulation was started, the density of main road was 0.15, and that of basic road was 0.1. Moreover, the time step of the main road is from 420 to 429 and that of the basic road is from 1 to 10. It can be seen from in Fig. 6 that the congestion is solved and a basic road does not become congested.

5.3 Time-series behavior of congestion

The congested-roads information obtained every 5 min is shown in Fig. 7. When simulation started, the density on a main road was 0.15, and that on a basic road was 0.1. The heavy black lines show congested roads. Congested places changed in time series. Several basic roads became congested where these...
roads are concentrated. This is because cars turning right at an intersection could not be passed.

Figure 6: State of cells

Figure 7.1: The map 5 min after the start of simulation

Figure 7.2: The map 10 min after the start of simulation

Figure 7.3: The map 15 min after the start of simulation

6. Evaluation of real-time route selection

The composition of our system is shown in Fig. 8. Table 1 summarizes the data (D1-D5).

In this system, first, a starting point and a destination are decided. Next, a route recommendation

Figure 8: The structure of our system

Table 1: Data summary

<table>
<thead>
<tr>
<th></th>
<th>node number</th>
<th>neighbor node number</th>
<th>distance</th>
<th>road class</th>
<th>the number of lanes</th>
<th>direction</th>
<th>signal</th>
<th>coordinate</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D2</td>
<td>distance</td>
<td>road class</td>
<td>the number of lanes</td>
<td></td>
<td>coordinate</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D3</td>
<td>node numbers</td>
<td>coordinate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D4</td>
<td>the number of congested roads</td>
<td>congested road number</td>
<td>coordinate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D5</td>
<td>place where one is now</td>
<td>congestion position</td>
<td>present time</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
is given based on calculations by the Dijkstra algorithm and the car travels on that route to its destination. We update traffic information every 5 min. If congestion occurs, the Dijkstra algorithm or a Genetic algorithm [8] finds an alternative route. Figure 9 shows an example of route selection.

Figure 9: An example of route selection

7. Conclusions

In this paper, we first performed a supplementary examination of Nagel’s multi-speed model and proposed an intersection model and a lane-change model. We confirmed that the models work effectively in a real road network.

As future study, we will compare the simulation results with real traffic and deal with the case where each car moves toward its destination instead of at random like in the present experiments.

We are deeply indebted to the Navigation System Researchers’ Association for providing the digital road map database.

References