Scientific Journal of Impact Factor(SJIF): 3.134
e-ISSN(O): 2348-4470
p-ISSN(P): 2348-6406
International Journal of Advance Engineering and Research Development
Volume 2,Issue 2, February -2015

INTRODUCTION OF TRAFFIC FLOW MODELS BASED ON CELLULAR AUTOMATA

Mitesh Patel[1], A.K.Patel[2], P. J. Gundaliya[3]

[1]*P.G. Student, Civil Engineering Department, L.D. College of Engineering, Ahmedabad, mi_mitesh18@hotmail.com*
[2]*Assistant Professor, Civil Engineering Department, Government Polytechnic, Himmatnagar, transportationakp@gmail.com*
[3]*Associate Professor, Civil Engineering Department, L.D. College of Engineering, Ahmedabad, pjgundaliya@gmail.com*

Abstract *— Cellular automata (CA) has been used in the recent past by many researchers for homogeneous and heterogeneous traffic flow modelling. The position and speed of vehicles are assumed to be discrete in CA traffic flow model. The speed of each vehicle changes according to its interactions with other vehicles and is governed by some pre-assigned (stochastic) rules. The major advantages of CA is its ability to handle large microscopic simulation. In this paper, introduction of traffic flow modelling using CA is presented. The basic concept of CA is briefly explain. The paper study also highlights the advantages and limitation of CA modelling concept in traffic flow models.*

Keywords- *Traffic flow modelling; Cellular automata; discrete simulation; Traffic flow model, Traffic simulation.*

I. INTRODUCTION

The increasing use of computers has led to a new way of looking at the world. This view sees nature as a form of computation. A computer follows rules. At each moment, the rules determine exactly what the computer need to do next. The rules are implemented by discrete dynamics into algorithms to reflect the system behaviour. Cellular automata are discrete dynamical systems whose behavior is completely specified in terms of a local relation.

The word Automata is the plural of automaton. While the word 'automaton' brings up the image of a mechanical toy or a soulless organism, in computer science it has a very precise meaning. It refers to all machines whose output behavior is not a direct consequences of the current input, but of some past history of its inputs. They are characterised as having an internal state which is a repository of this past experience. The inner state of an automaton is private to the automaton, and is not avai1able to an external observer. One type of automaton that has received a lot of attention is cellular automata. For one thing, they make pretty pictures. For another, they are related to exciting new ideas such as artificial life and the edge of chaos. By building appropriate rules into a cellular automaton, it can simulate many kinds of complex behaviour, ranging from the motion of fluids governed by the Navier-Stoke's equations to outbreaks of starfish on a coral reef. Ce1ls are discrete dynamical systems whose behaviour is completely specified in terms of a local relation.

A cellular automaton can be thought of as a stylised universe. Space is represented by a uniform grid, with each cell containing a few bits of data; time advances in discrete steps and the laws of the "universe" are expressed in, say, a small look-up table, through which at each step each cell computes its new state from that of its close neighbours. Thus, the system's laws are local and uniform. Even in computer science various names for cellular automata were used, including tessellation automata, particle hopping, cellular spaces, iterative automata, homogeneous structures and universal spaces.

Many researchers have used Cellular Automata (CA) for traffic flow modelling efficiently for homogeneous traffic flow. Literatures shows that CA application is simple and computationally efficient for traffic flow modelling. In the present paper we discussed about the basic concept of the cellular automata and its application in traffic flow model. Nagel and Schreckenberg (1992) presented the cellular automata model for traffic flow well known as NaSch model. This model also describe in separate section as to understand how the CA work for traffic flow modelling.

II. HISTORY OF CELLULAR AUTOMATA

Despite their very simple construction, nothing like general cellular automata appear to have been considered before about the 1950s. Yet in the 1950s – inspired in various ways by the advent of electronics computers- several different kinds of systems equivalent to cellular automata were independently introduced. A variety of precursors can be identified. Operations on sequences of digits had been used since antiquity in doing arithmetic. Finite difference approximations to differential equations began to emerge in the early 1900s and were fairly well known by the 1930s. A Turning machines invented in 1936 were based on thinking about arbitrary operations on sequences of discrete elements.

The best-known way in which cellular automata were introduced (and which eventually led to their name) was through the work by Neumann (1959). The first, mostly in the 1960s, was increasingly whimsical discussion of building actual self-reproducing automata – often in the form of spacecraft. The second was an attempt to capture more of the essence of self-reproduction by mathematical studies of detailed properties of cellular automata. In 1967, Ulam has used CA for generating complicated pattern, and mentioned that this might be relevant to biology. But perhaps because almost no progress was made on this with traditional mathematical methods, the result was not widely known, and was never pursued. Ulam (1970) has used the similar concept for mathematical logic for "simulation games". Wolfram (1986) has made a CA rules based on von Neumann's cellular automata. This rule are used to various engineering and science field by certain modification in recent past. A graphics program of using cells concept specifically called a cellular automaton when it is

1. Parallel,
2. Local, and
3. Homogeneous.

Parallelism means that the individual cell updates are performed independently of each other. That is, we think of all the updates being done at once. Locality means that when a cell is updated, its new value is based solely on the old values of the cell and of its nearest neighbours. Homogeneity means that each cell is updated according to the same rules. Typically the values of the cell and of its nearest four or eight neighbours are combined according to some logico-algebraic formula, or are used to locate an entry in a preset lookup table.

III. WORKING PRINCIPLE OF CELLULAR AUTOMATA

In cellular automata space, time and state variables are discrete which makes them ideally suited for high-performance computer simulation. However, CA modelling differs in several respects from coutinum models. These are usually based on differential equation, which often cannot be treated analytically. One has to solve them numerically and therefore the equation have to be discretized. In general, only space and time variables become discrete whereas the state variable (e.g. the density or velocity) is still continuous. CA is discrete in space and time variables, this discreteness is already taken into account in the definition of the model and its dynamics. This allows for obtaining the desired behaviour in a much simpler way. The numerical solution of (discretised) differential equations is only accurate in the limit Δx, $\Delta t \rightarrow 0$. This is different in the CA where Δx and Δt are finite and accurate results can be obtained since the rules (dynamics) am designed such that the discreteness is an important part of the model.

In order to achieve complex behaviour in a simple fashion one often resorts to a stochastic description. A realistic situation seldom can be described completely by a deterministic approach. A cellular automaton is a discrete dynamical system. Each point in a regular spatial lattice, called a cell, can have anyone of a finite number of states. The states in the cells of a lattice are updated according to a local rule say *R*. That is, the state of the cell at a given time depends only on its own state one time step previously, and the states of its nearby neighbours at the previous time step. All cells in the lattice are updated synchronously. The state of the lattice advances in discrete time steps. In the above definition, the rule R is identical and homogeneous for all sites and applied simultaneously to each of them. The rule R is some well-defined function and a given initial configuration will always evolve the same way.

However it may be very convenient for some applications to have certain degree of randomness in the rule. It may be desirable for some instance, that a rule selects one outcome among the several possible states, with a probability *p*. Cellular automata whose updating rule is driven by external probabilities are called probabilistic cellular automata. On the other hand, those which strictly comply with the definition given above, are referred to as deterministic cellular automata.

CA is used in traffic flow modelling since last decade. The freeway being simulated is discretised into homogeneous cells of equal length, and time is discretised into time-steps of equal duration. These cells can be either in an occupied or empty state, depending on whether a vehicle is present at that location. The state of the cells is updated sequentially at each time step with a set of vehicle position updation rules.

A cellular automata rule is local, by definition. The updating of a given cell requires one to know only the state of the cells init's vicinity. The spatial region in which a cell needs to search is called the neighbourhood. For two-dimensional cellular automata, two neighbourhoods are often considered. The key idea of neighbourhood is that when updation occurs, the cells within the block evolve only according to the state of that block and don't depend on what is in the adjacent blocks. In practice when simulating a given cellular automata rule, it is not possible to deal with an infinite lattice. The system must be finite and have boundaries. A site belonging to the lattice boundary doesn't have the same neighbourhood as other internal sites. In order to define the behaviour of these sites, a different evolution rule can be considered, which sees the appropriate neighbourhood. The basic deterministic CA rules given by Wolfram (1986) is explained in following subsection.

3.1 Wolfram's CA rules

Wolfram (1986) worked with a one-dimensional variant of von Neumann's cellular automata; this was fully horizontal and occurred on a single line. Each cell touched only two other cells, its two immediate neighbours on either side, and each succeeding generation was represented by the line underneath the preceding one. A cell in generation two would determine its state by looking at the cell directly above it, i.e. in generation one, and that cell's two neighbours. Thus, there arc eight possible combinations of the states of those three cells ranging from "000" (all off, all white) to

"111" (all on, all black) as shown in Fig.1. Three cells and their updation state called an applet. He has specified such different combination and given them CA rule number from 0 to 254. The CA rule 0, CA rule 254 and CA rule 184 are illustrated in Fig. 1. The CA rule 0 means that whatever updation occurs in the neighbouring cells, the middle cell will be white (fixed rule). Similarly CA rule 254 updates middle cell to black for any colour combination in neighbouring cells. The working of CA rule 184 is illustrated as shown in Fig. 2. In this rule, status of cell updation is defined as described in CA rule 184 in Fig. 1. The cells updates their state according to neighbours state (flexible rule). Hence, in the second time-

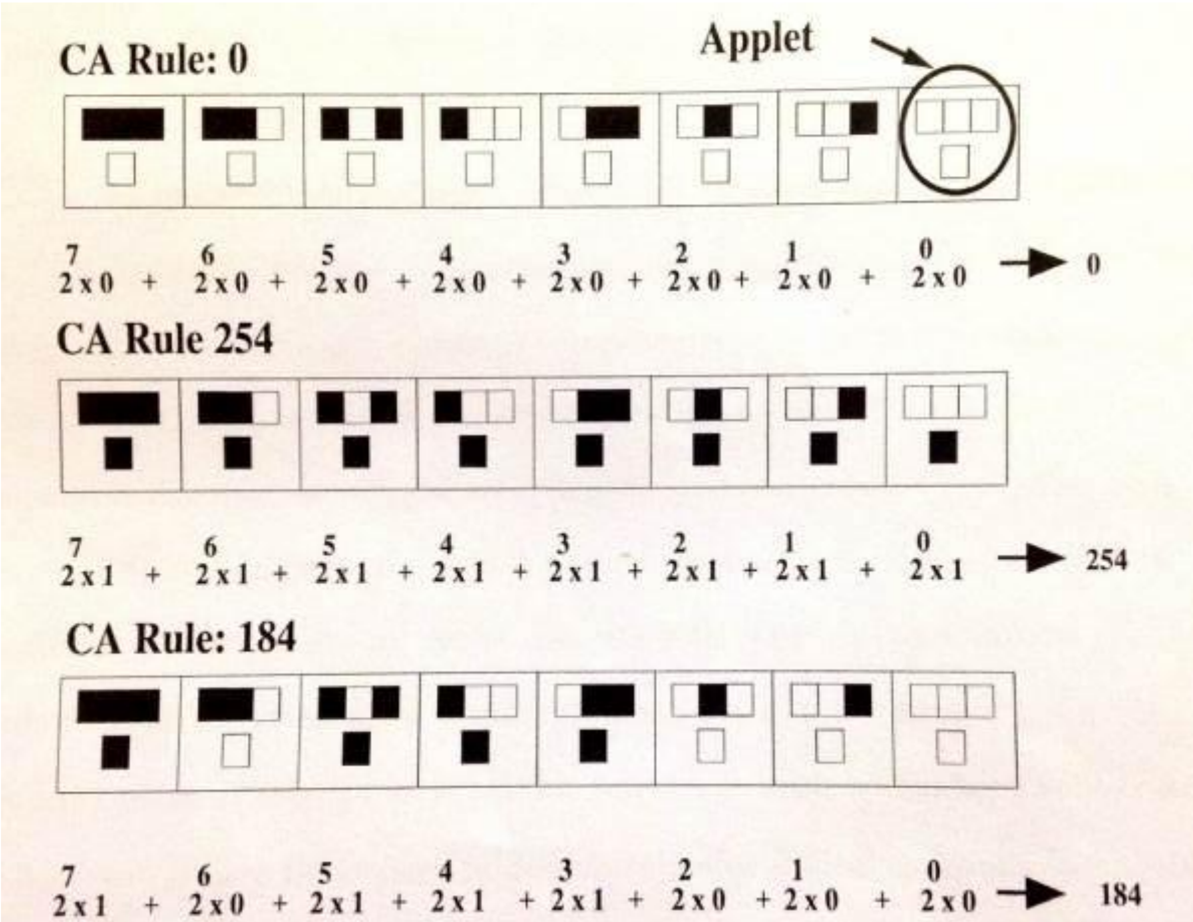

Figure 1. Illustration of CA rules described by Wolfram (1986)

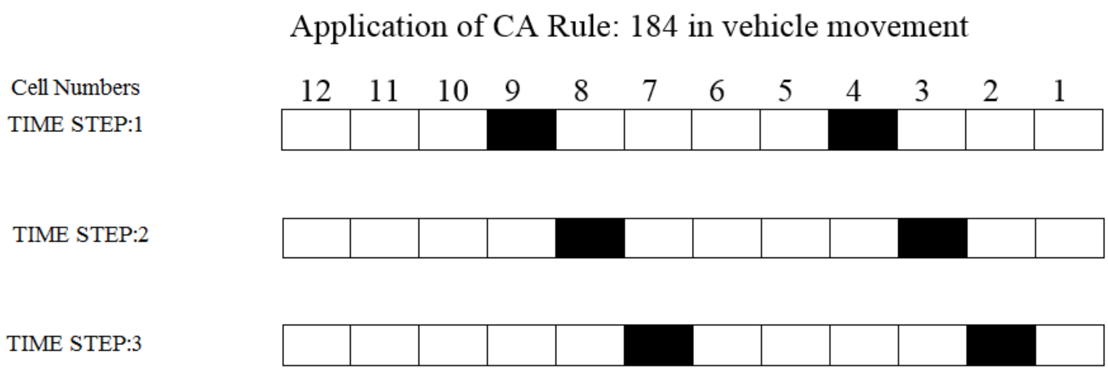

Figure 2. Updation of cell status in case of CA rule 184

step the cell number four, five, ten and eleven from the left, changes its state in next time step according to rule 184 as shown in Fig. 2. This is similar to the position updation of vehicles (or how the vehicle advances) at every time step. Cellular automata are the mathematical models for complex natural systems containing large numbers of simple identical components with local interactions. They consist of a lattice of sites, each with a finite set of possible values. The value of the sites evolve synchronously in discrete time steps according to identical rules. The value of a particular site is determined by the previous values of a neighbourhood of sites around it. Single dimension cellular automata are arrays of discrete cells with discrete values. Yet sufficiently large cellular automata often show seemingly continuous macroscopic behaviour. They can thus potentially serve as models for continuum systems, such as fluids. Their underlying discreteness, however, makes them particularly suitable for digital computer simulation and for certain forms of mathematical analysis. On a microscopic level, physical fluids also consist of discrete particles. But on a large scale, they, too, seem continuous, and can be described by the partial differential equations of hydrodynamics. The form of these equations is in fact quite insensitive to microscopic details. Changes in molecular interaction laws can affect parameters such as viscosity, but do not alter the basic form of the macroscopic equations. As a result, the overall behaviour of fluids can be found without accurately reproducing the details of microscopic molecular dynamics.

Cellular automata is developed in discrete dynamics of space and time, and is a discrete simulation method. Its buildup from the single string of one dimensional automata, and can be arranged in two or higher dimensional lattice for two or higher dimensional automata. All cells in CA are identical and having discrete state. The future state of each cell depends only of the current state of the cell and the states of the cells in the neighbourhood. The development of each cell state is defined by simple rule CA models are in principle amenable to single bit coding. This approach run extremely fast on traditional vector-computers.

3.2 Properties of CA

- CA developed in space and time.
- A CA is a discrete simulation method, hence space and time are defined in discrete steps.
- A CA is built up from cells, that are lined up in a string for one- dimensional automata.
- CA can be arranged in a two or higher dimensional lattice for two - or higher dimensional automata
- The number of states of each cell is finite.
- The states of each cell are discrete.
- All cells are identical.
- The future state of each cell depends only of the current state of the cell and the states of the cells in the neighbourhood.
- The development of each cell is defined by rules.
- CA models are in principle amenable to single bit coding. This approach run extremely fast on traditional vector-computers.

IV. APPLICATION IN TRAFFIC FLOW MODELLING

The CA recently used for traffic flow modelling in case of homogeneous and heterogeneous traffic due to its high computing power which makes possible on-line application of traffic flow models. CA has many distingue advantages in microscopic traffic flow modelling. CA models are Robust numerics, which seem at the first glance as too rough an approximation of reality, include the same range of dynamic phenomena as the most advanced fluid-dynamical models for traffic flow to date.

The consequence for traffic simulation is that, as long as one expects certain simple aspects of traffic jam formation to be realistic enough for the problem under consideration e.g. for large-scale questions, the simplest possible model will be sufficient for the task thus saving human and computational resources. The present results show that close up car-following behaviour is not the most important aspect of traffic to model. The important crucial aspect is to model deviations from the smooth behaviour and the ways in which they lead to jam formation. Another important aspect, which seems far from obvious, is the acceleration behaviour, especially when there are other cars ahead, since it is the acceleration behaviour that mostly determines the maximum flow out of a jam. Therefore, investigations such as CA models are important for microscopic modelling as long as one does not have the perfect model of driving or the computational resources to run it. This concept can be stated as minimal. Fast running and easy to implement CA can be very useful in interpreting measurements such as for the traditional 5-min-averaged fundamental diagrams (speed-flow-density). CA models are inherently microscopic, which allows one to add individual properties to each car such as the identity of travelers, route plan, and engine temperature for emission modelling. These properties are imperative for the kind of traffic models that are needed in current policy evaluation processes. CA models are stochastic in nature, thus producing different results when using different random seeds even when starting from identical initial conditions. At first, this is certainly considered a disadvantage from the point of view of policy makers or traffic engineers. However, the traffic system is inherently stochastic and the variance of the outcomes is an important variable itself. Furthermore, there is reason to believe that the average over several stochastic runs will not be identical to a deterministic run. In CA models have potential to apply at network level and parallel computation can be possible.

Due to discreteness of CA it have disadvantages like, speed of the vehicle are imitated to discrete time steps, acceleration and deceleration are more than the real, lane changing is done in single time step (one second) whereas it needs more time to change the lane.

4.1 Limitations of CA for Traffic Flow Modelling

Some of the limitations of CA based traffic flow models are listed below.

- Time setup is discrete hence the acceleration and deceleration is more than the real.
- If vehicle size and speed are widely differ, then its difficult to represent them as uniform cells.
- Accelerations and decelerations are much larger than in reality due to discreteness.
- Lane change is done in one second (or for given time-step) whereas actual time required more than that.
- The speed of vehicle are imitated to discrete time steps.

The Traffic flow modelling using CA is extensively done for an uninterrupted roads like freeway and arterial to model the traffic flow behaviour. In cellular automata the cells, which are either empty or occupied by exactly one vehicle. Movement takes place by distance between vehicle and vehicles characteristics. In CA models, a road is represented as a string. Initial proposition of a CA model for traffic is given by Gerlough (1956) and same is extended by Cremer and Papageorgio (1981), Cremer and Ludwig (1986) and co-workers. They implemented fairly sophisticated driving rules and also used single-bit coding with the goal to make the simulation fast enough to be useful for real-time traffic applications. The bit-coded implementation, though, made it too impractical for many traffic applications. In 1992, CA models for traffic were brought into the statistical physics community. Nagel and co-workers used model with maximum velocity for one and for two dimensional traffic. One-dimensional here refers roads includes multi-lane traffic. Two-dimensional traffic in the CA context usually means traffic on a two dimensional grid, as a model for traffic in urban areas (Intersection).

Traffic flow model developed by Nagel and Schreckenberg (1992) is defined as a one-dimensional array. This means the total number of vehicles N in the system is maintained constant. Each cell may be occupied by one vehicle or it may be empty. Each cell corresponds to a road segment with length L. In this approach, cars are represented as points moving on a discretised road with only a small set of possible velocities and accelerations.

Since, the introduction of NaSch model developed by Nagel and Schreckenberg (1992), many researchers has used CA for traffic flow models The CA is also applied by many researchers by modifying the basic concept for the heterogeneous traffic (Chowdhury et al., 2000, TRANSIM, 2000). The same is modified by applying the grid based approach by Gundaliya et al. (2004) and Lawarence and Chang (2004). In The following section the NaSch model is explain with the numerical example.

V. NAGEL-SCHRECKENBERG MODEL

Nagel and Schreckenberg (1992) presented the cellular automata model for single lane traffic flow for homogeneous traffic. This model is taken as the base model for present research work. The process of vehicle movement is described here. In CA traffic flow models, the position, speed, acceleration as well as time are treated as discrete variables. In a basic CA traffic flow model each vehicle has an integer speed with values between zero and the maximum speed of the vehicle. The speed of each vehicle can take one of the integer values out of say 0,1,2,3, ... ,v_{max}, in term of cells per time step. The road is represented by a cell which can be either empty or occupied by at most one vehicle at a given instant of time. At each discrete time step, the state of the system is updated following a well defined rule. In NaSch model vehicles updation are followed by four rules namely acceleration, deceleration, randomisation and updation. This model is developed for the homogeneous traffic flow. In these rules except probability p all operation are integer. A numerical example is taken for explaining vehicle movement on road in NaSch model. In this example noise probability (p) taken as 1/3 which reflect the driver behaviour. All vehicles are taken as cars with maximum speed (v_{max}) allowed as 2 cell/time step. In Fig. 3 the road stretch are represented with the uniform front gap of vehicle is taken numerically as number of empty cells ahead plus one. The updation of vehicle speed are indicated after applying each rule shown in Fig. 3. The uniform vehicle (car) is considered in this example. Due to noise probability an average one third of the cars qualified slowdown in the randomisation step. All cars update their speed parallaly at every time step as shown in Fig. 3 step by step. In Fig. 3 all four cars speed are indicated on right top corner of the cell at every step. The initial speed of car 1, 2, 3 and 4 are 0, 1, 2, and 1 respectively. The front gap of all cars 1, 2, 3, and 4 are -1, 1, 3, and 2 respectively. The front gap of all cars are indicated at the right bottom corner of the cell in Fig. 3. The first car front gap is taken as -1, means that there is infinite gap and vehicle can attain the speed of its maximum speed.

- Step 1: Acceleration

If $v_n < v_{max}$ then the speed of the nth vehicle is increased by one, $v_n = min\ (v_n + 1, v_{max})$ but remains unaltered if $v_n = v_{max}$. After applying this rule the speed of the cars are 1, 2, 2, and 2 respectively for car 1, 2, 3, and 4. The car 3 is going with its maximum speed hence its speed remain unaltered. This reflects the general tendency of the drivers to drive as fast as possible, if allowed to do so, without crossing the maximum speed limit.

- Step 2: Deceleration due to other vehicles (vehicle ahead)

If $gap_p^f \leq v_n$ in the speed of the nth vehicle is reduced to $gap_p^f - 1$, here gap is one even if a front vehicle is there in the next cell, $v_n = max\ (v_n, gap_p^f - 1)$. Here gap_p^f is front vehicle gap in the present lane. After applying this rule the speed of cars are 1, 0, 2, and 1 respectively for car 1, 2, 3, and 4. The car 2 and car 4 reduces its speed based on the available front gap in number of cell as 0 and 1 cells per time step respectively.

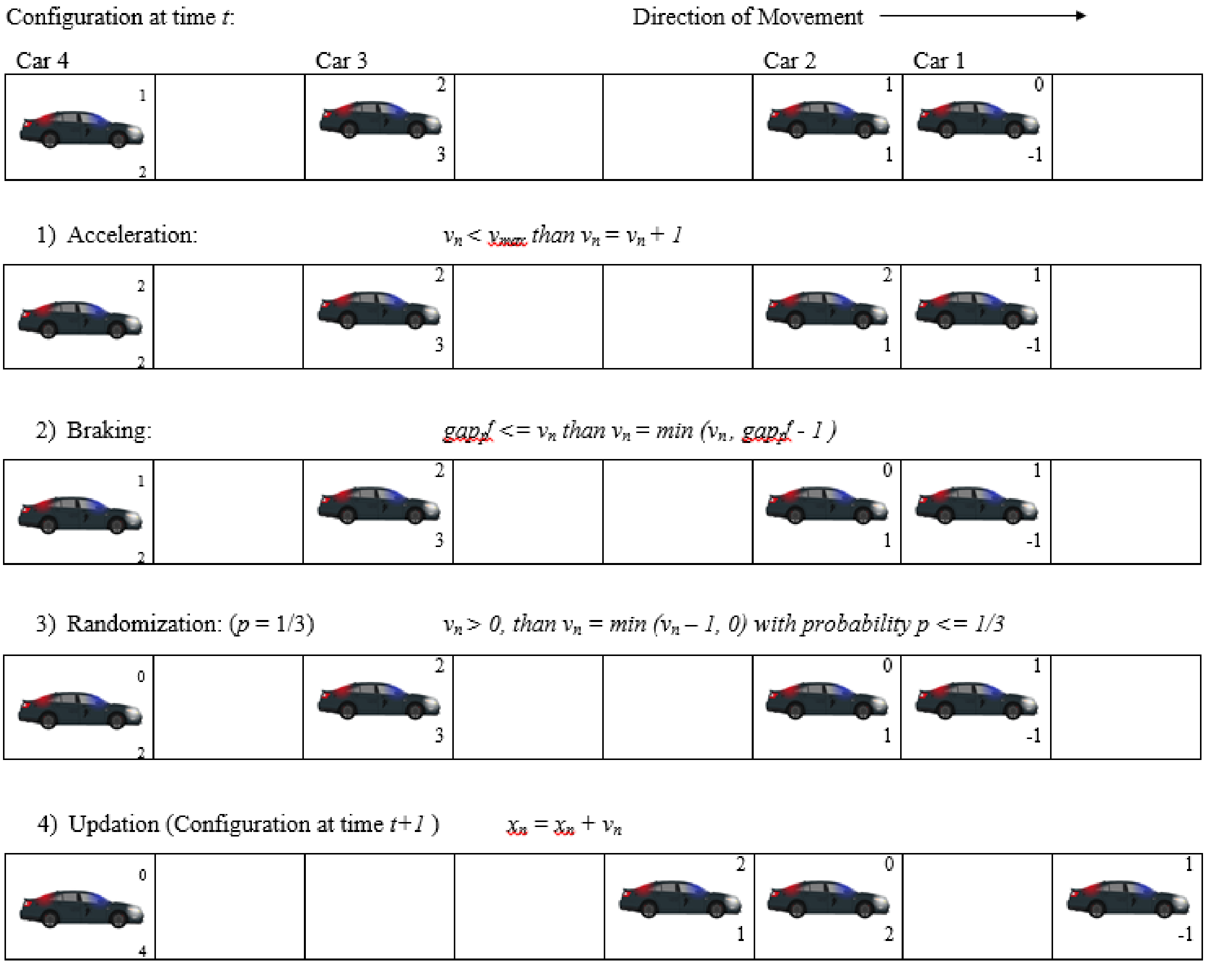

Figure 3 Numerical example for the application of CA rules for NaSch model

- Step 3: Randomisation

If $v_n > 0$, the speed of the nth vehicle is decreased randomly by one unit with the probability p but does not change $v_n = 0$, $v_n = max\ (v_n - 1,\ 0)$ but remains unaltered if $v_n = v_{max}$. After applying this rule the speed of the cars are 1, 0, 2, and 0 respectively for car 1, 2, 3, and 4. After applying this rule the car 4 has its speed reduced by 1 with the chance of probability p. This step takes into account the different behavioural patterns of the individual drivers, especially, non-deterministic acceleration and overreaction while slowing down; this is crucially important for the spontaneous formation of traffic jams.

- Step 4: Vehicle movement

Each vehicle moves forward according to its new speed determined in Steps 1-3, i.e.

$x_n = x_n + v_n$

The vehicle then forwarded with the number of cell (speed) as shown in step 4 in Fig. 3. After applying all rule the new speed of cars are 1, 0, 2, and 0 respectively for car 1, 2, 3, and 4. The front gap all cars are then updated as -1, 2, 1, and 4 respectively for car 1, 2, 3 and 4. This procedure again repeated for another time step. It is to be noted that even changing the precise order of the steps of the update rule stated above would change the properties of the model. This model may be regarded as a stochastic CA.

VI. SUMMARY AND CONCLUSION

In this paper basic principle of the cellular automata and its development are explained. The characteristics of CA and its application in traffic flow modelling is presented. The computational advantage of the CA and its capability to reproduce the complex phenomena is discussed in brief with highlighting some limitation of the CA based traffic flow models. Wolfram (1986) CA rules are explained taking the numerical example. The rule pertaining to the vehicular movement is explained to understand the working of CA in case of traffic flow model. The working principle of NaSch model for freeway is explained with a numerical example for understanding the implementation of the four basic rules (acceleration, deceleration, randomization and updation) in CA based model. This paper gives brief working principle of CA based traffic flow models and its application.

REFERENCES

[1] Chowdhury, D., Santen, L., and Schadschneider, A. (2000). “Statistical physics of vehicular traffic and some related systems.” Elsevier Science, Physics Report 329. http://www.elsevier.com/locate/physrep. Last accessed on 21/03/2004

[2] Cremer, M. and Ludwig, J. (1986). "A fast simulation model for traffic flow on the basis of Boolean operations." Mathematics and Computers in Simulation, 28, 297-303.

[3] Cremer, M. and Papageorgio, M. (1981). "Parameter identification for a traffic flow model." Automatica, 17(6), 837-843.

[4] Gerlough, D. L. (1956). "Simulation of freeway traffic by an electronic computer."Highway Research Board Proceedings, 35th Annual Meeting. p.p.-548-557.

[5] Gundaliya, P. J., Tom, M. V., and Dhingra, S. L. (2004). "Application of cellular automata in traffic flow modelling." Transportation systmem planning and operation, I.I.T. Madras, Madras (INDIA). 2, 659-668.

[6] Lawarence, L. W. and Chang, C.W.(2004)."In homogeneous particles hopping models for mixed traffic with motorcycles and car."International conference on application of information and communication technology in transportation systems in developing countries, University of Maratuway, SRILANKA.

[7] Nagel, K. and Schreckenberg,M.(1992)."Cellular automaton model for freeway traffic." Journal de physique I,2(12), p.p.-2221-2229.

[8] Neumann, J. (1959). "Theory of automata: Construction, reproduction, homogeneity." Theory of Self-Reproduction Automata, W. A. Burks, ed., University of Illinois Press, Urbaba, Iiinois, Pt.Two. 91-3801.

[9] TRANSIM (2000). "Transportation analysis and simulation system,. http://transims.tsasa.lanl.gov.Last accessed on 20/09/2002

[10] Ulam. S. M (1970). “ On some mathematical problems connected with patterns with patterns of growth of figure.” Essay on cellular automata, A. Burks, ed., University of Illinois Press, Champaign, IL.

[11] Wolfram,S. (1986). Theory and Application of Cellular Automata. World Scientific, Singapore.