Name: Tao Zhang **Student No. 7389480**

Shockwaves on Highway

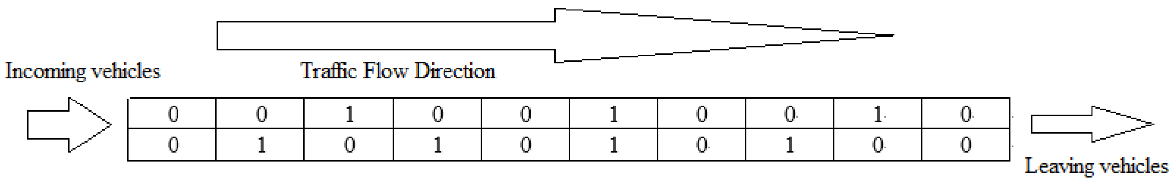
Part I: Original Conceptual Model

a) Motivation

I propose to use a Cellular-DEVS model to implement the shockwaves on Highway using cellular automata.

Recently I read an article entitled “The Cell Transmission Model: A Dynamic Representation of Highway Traffic Consistent with the Hydrodynamic Theory”. In this article, the author compared the characteristics of traffic flow with the hydrodynamic theory. They are consistent with each other, even when there are some disturbance in the flow.

The concept is simple: we consider a homogenous highway, at the initial state, let the traffic density be slow varying. The traffic are moving at the free flow speed. Then we add some disturbance into this flow. Because discontinuities are common—arising spontaneously when low density traffic catches up with denser, slower downstream traffic—their effect on the road should be examined.



0	0	1	0	0	1	0	0	1	0
0	1	0	1	0	1	0	1	0	0

In a two-lane one way road segment, how the traffic flow evolve with an increasing density? And how the heavy traffic will evolve with a decreasing density?

b) Cell-DEVS modelling

1) The time incoming vehicles are following the Poisson distribution and the rate of incoming vehicles is increasing with time.
2) Boundary conditions can be specified by means of input and output cells. The output cell, a sink for all exiting traffic, should have infinite size and a suitable, possibly time-varying, capacity.
3) The “source” cell with an infinite number of vehicles that discharges into an empty cell.
4) The model is non-wrapped. So the vehicles in the upper and lower lane can change their lane to the other, and any crash should be avoided.
5) Run this simulation again to show how the traffic flow evolve with a decreasing density.

Part II: Formal Specification

a. Cell-DEVS Atomic Model Specification

The following is the formal specification for the Cell-DEVS model:

CD = $< X, Y, I, S, \theta, N, d, \delta_{int}, \delta_{ext}, \tau, \lambda, D >$

X = { incomingVehicles1, incomingVehicles2 }

Y = Ø

S = { 0, 1 ,2}

N = neighborhood = { (-1,-1), (-1, 0), (-1, 1), (0, -1), (0, 0), (0, 1), (1, -1), (1, 0), (1, 1) }

d = 1ms

τ: N→S is defined by the traffic rules described in the previous section, i.e.:

S = 0 if cell(0,0) = 1 and cell(-1,0) = 0
S = 1 if cell(0,0) = 0 and cell(1,0) = 1
S = 0 if cell(0,0) = 2 and cell(-1,0) = 0
S = 2 if cell(0,0) = 2 and cell(1,0) = 1
S = 0 if cell(0,0) = 1 and cell(-1,0) = 1 and cell(0,1) = 0 and cell(-1,1) = 0
S = 2 if cell(0,0) = 0 and cell(1,0) = 0 and cell(1,-1) = 1 and cell(0,-1) = 1
S = 0 if cell(0,0) = 2 and cell(-1,0) = 2 and cell(0,-1) = 0 and cell(-1,-1) = 0
S = 1 if cell(0,0) = 0 and cell(1,0) = 0 and cell(1,1) = 2 and cell(0,1) = 2

b. Cell-DEVS Coupled Model Specification

The model specification from part (a) above gives rise to the following model definition:

```
[top]
components : shockwave  incomingVehicle1@MyGenerator
incomingVehicle2@MyGenerator
link : out@incomingVehicle1 inputVehicles1@shockwave
link : out@incomingVehicle2 inputVehicles2@shockwave

[shockwave]
type : cell
dim : (30,2)
delay : transport
defaultDelayTime  : 1
border : nowrapped
neighbors : shockwave(-1,-1) shockwave(-1,0) shockwave(-1,1)
```

```
neighbors : shockwave(0,-1)  shockwave(0,0) shockwave(0,1)
neighbors : shockwave(1,-1)  shockwave(1,0) shockwave(1,1)
in : inputVehicles1 inputVehicles2
link : inputVehicles1 in@shockwave(29,0)
link : inputVehicles2 in@shockwave(29,1)
zone : outVehicles { (0,0)..(0,1) }
InitialValue : 0
InitialCellsValue : vehicle.val
portInTransition : in@shockwave(29,0)  setVehicle1
portInTransition : in@shockwave(29,1)  setVehicle2
localtransition : traffic-rule

[traffic-rule]
rule : 0  1 { (0,0) = 1 and (-1,0) = 0 }
rule : 1  1 { (0,0) = 0 and (1,0) = 1 }
rule : 0  1  { (0,0) = 2 and (-1,0) = 0 }
rule : 2  1  { (0,0) = 0 and (1,0) = 2 }
rule : 0  1  { (0,0) = 1 and (-1,0) = 1 and (0,1) = 0 and (-1,1) = 0 }
rule : 2  1  { (0,0) = 0 and (1,0) = 0 and (1,-1) = 1 and (0,-1) = 1 }
rule : 0  1  { (0,0) = 2 and (-1,0) = 2 and (0,-1) = 0 and (-1,-1) = 0 }
rule : 1  1  { (0,0) = 0 and (1,0) = 0 and (1,1) = 2 and (0,1) = 2 }
rule : { (0,0) }  1  { t }

[setVehicle1]
rule : { portValue(in) } 0 { t }

[setVehicle2]
rule : { portValue(in) } 0 { t }

[incomingVehicle1]
distribution : exponential
mean : 1
initial : 1
increment : 0

[incomingVehicle2]
distribution : exponential
mean : 1
initial : 2
increment : 0

[outVehicles]
rule : 0 10 { t }
```

The model definition use .val file to define the initial traffic condition in the road segment by specifying the states of cells as 1's, 2's or 0's. For example, the following is a definition of an initial traffic condition which can be simulated using the Cell-DEVS model:

```
(2,0)=1
(3,1)=2
(5,1)=2
(7,0)=1
(10,0)=1
(11,1)=2
(13,0)=1
(15,1)=2
(18,1)=2
(20,0)=1
(20,1)=2
(21,0)=1
(21,1)=2
(22,0)=1
(24,0)=1
(26,0)=1
(26,1)=2
(27,1)=2
```

Here we use 1 to represent cars that in the left lane; and 2 to represent cars that in the right lane; 0 means no cars in the cell

There are two inputs in cell(29,0) and cell(29,1), representing the random arrival of vehicles. The intervaltime of vehicles is exponentially distributed. In order to examine the different rate of incoming vehicle, I let MyGenerator to change the mean value of the exponential distribution in the simulation.

The cell(0,0) and cell(0,1) are the two sinks.

The following are the explanation of traffic rules:

rule : 0   1 { (0,0) = 1 and (-1,0) = 0 }
rule : 1   1 { (0,0) = 0 and (1,0) = 1 }

if the vehicle is in the left lane and the cell ahead is empty, then the vehicle will move forward to the next cell.

rule : 0   1   { (0,0) = 2 and (-1,0) = 0 }
rule : 2   1   { (0,0) = 0 and (1,0) = 2 }

if the vehicle is in the right lane and the cell ahead is empty, then the vehicle will move forward to the next cell.

rule : 0   1   { (0,0) = 1 and (-1,0) = 1 and (0,1) = 0 and (-1,1) = 0 }
rule : 2   1   { (0,0) = 0 and (1,0) = 0 and (1,-1) = 1 and (0,-1) = 1 }

1	
1	

If the vehicle is in the left lane and the cell ahead is occupied, and if the two cells on the right lane are empty, then the vehicle will move the diagonal cell

	2

rule : 0 1 { (0,0) = 2 and (-1,0) = 2 and (0,-1) = 0 and (-1,-1) = 0 }

rule : 1 1 { (0,0) = 0 and (1,0) = 0 and (1,1) = 2 and (0,1) = 2 }

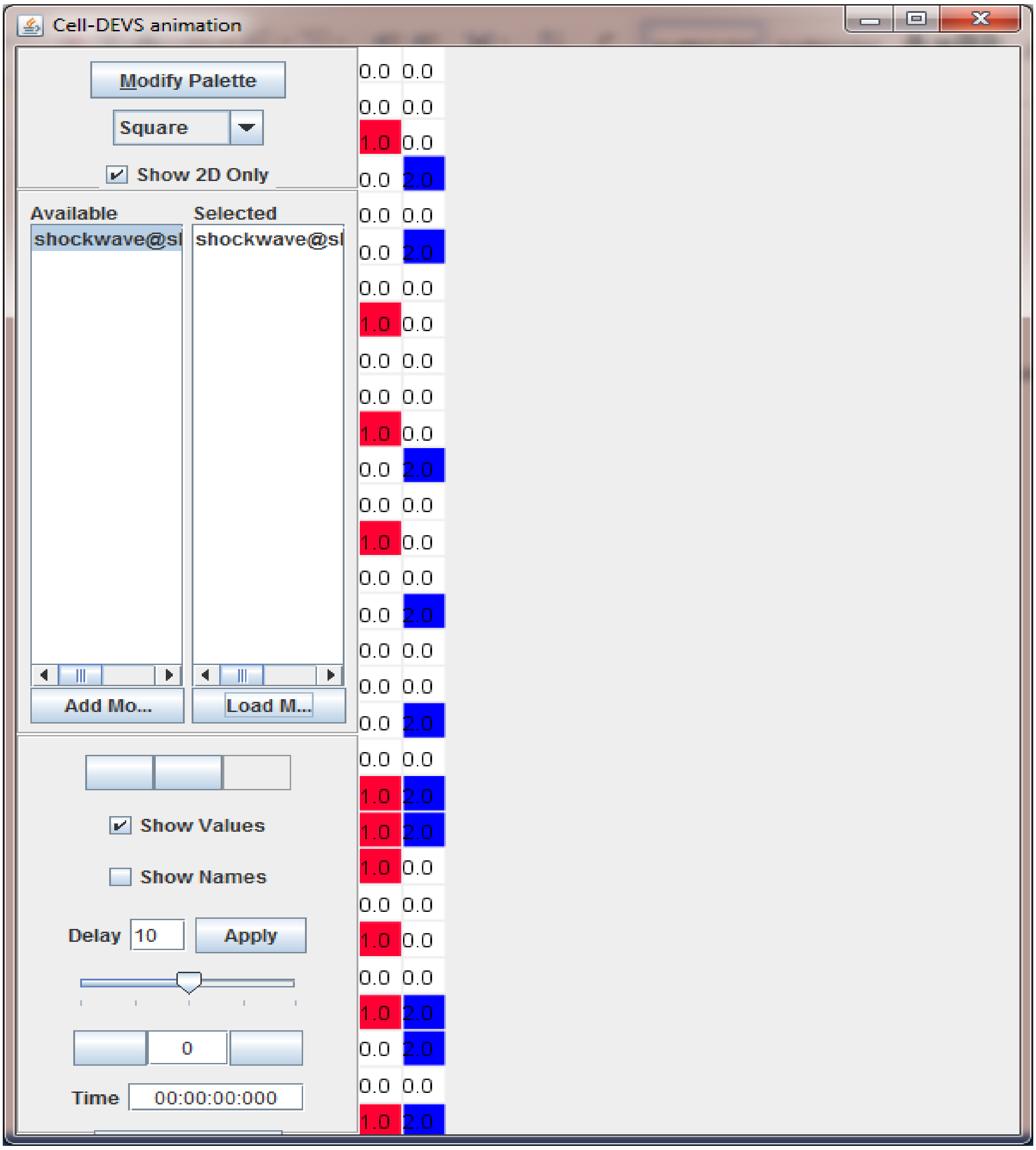
	2
	2

If the vehicle is in the right lane and the cell ahead is occupied, and if the two cells on the left lane are empty, then the vehicle will move the diagonal cell

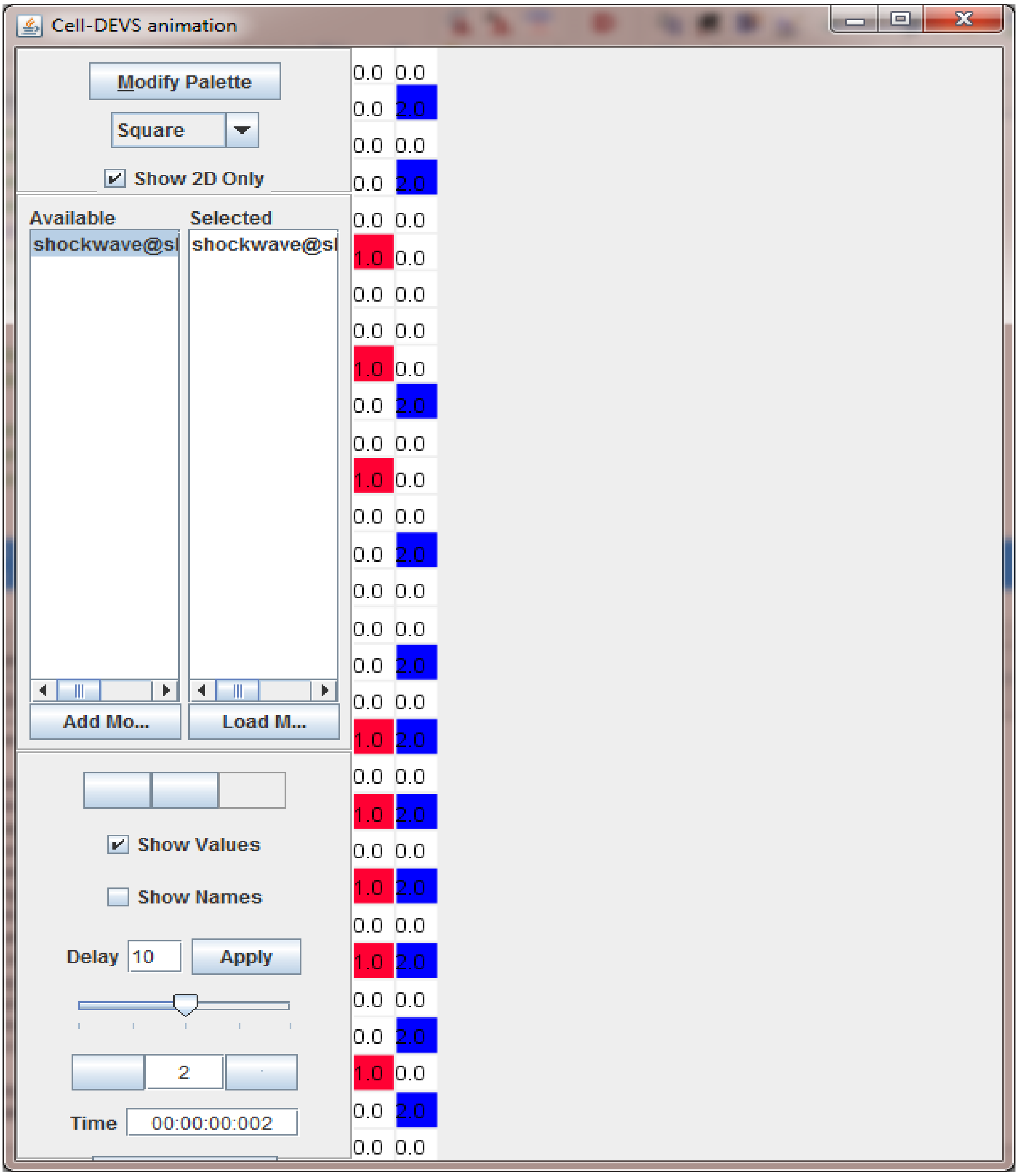
1	

c. Implementation and Testing

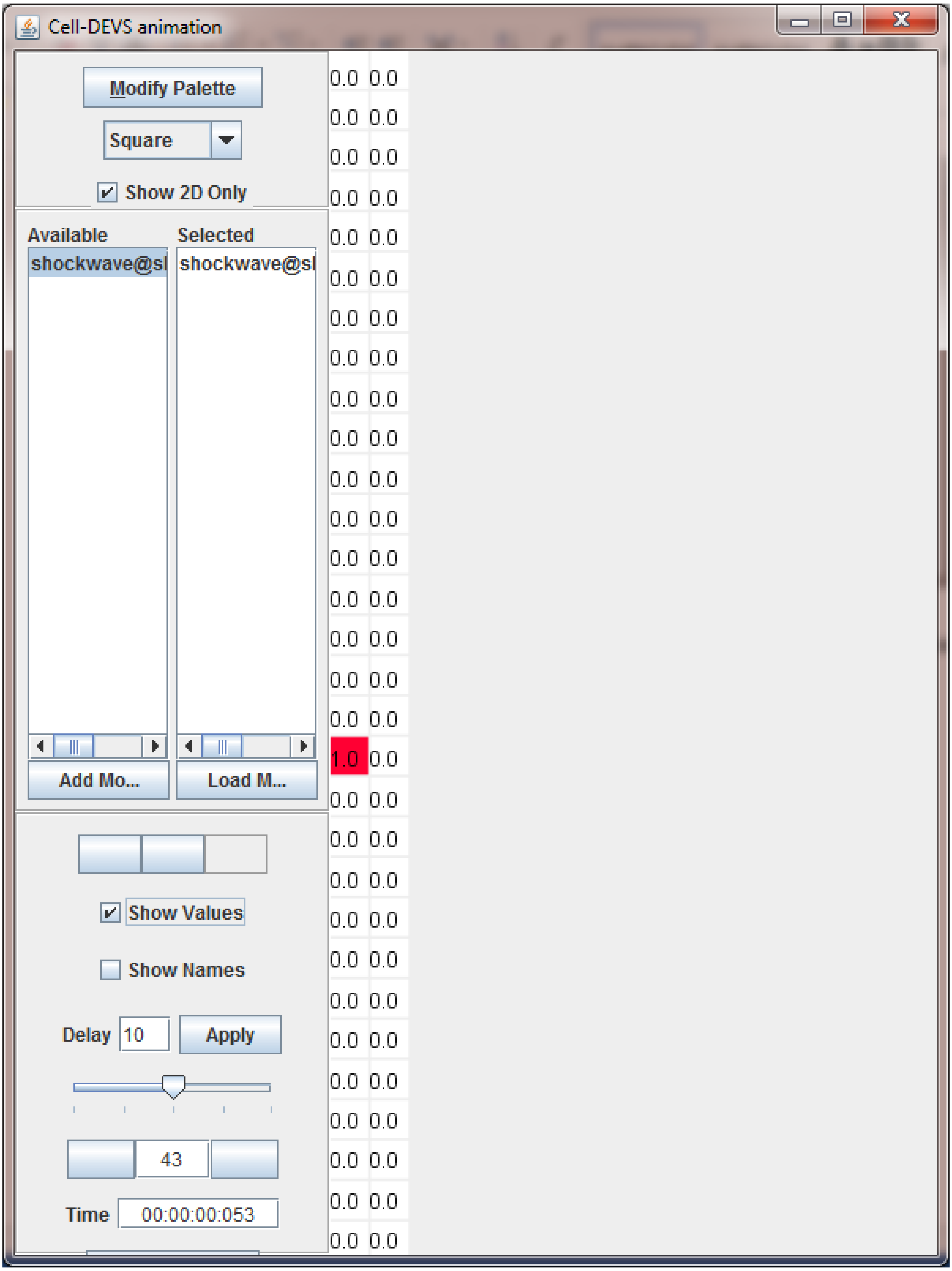
1) The initial traffic condition



2) After 2 steps:



3) After all the initial traffic leave this road segment, the incoming vehicles randomly arrive at this road segment from the bottom cells


Cell-DEVS animation
Modify Palette
Square
Show 2D Only
Available
Selected
shockwave@sl
shockwave@sl
Add Mo...
Load M...
Show Values
Show Names
Delay
10
Apply
43
Time
00:00:00:053
1.0
0.0

Part III: Conclusions

The Cellular-DEVS models included with this report and described herein correctly simulate the behaviour of the traffic flow. The road, as modeled as a cell space, can easily simulate the complicated traffic rules and the randomly arrival of the incoming vehicles.

At the beginning, the traffic density on this road segment is really high (because of the traffic light or accident). After everything goes right, the vehicles quickly dissipate and at the end become a free-flow traffic.

This assignment was also successful in demonstrating the use of the CD++ tool in simulating a cellular automata model, and using the associated tools (drawlog, Cell-DEVS animation) to visualize the outputs.