Scenario-based cellular automata urban growth modeling and policy application

Implemented using Extended CD++

Michael Dang'ana
Department of Systems and Computer Engineering
Carleton University
1125 Colonel By Drive, Ottawa ON CA, K1S 5B6
michaeldangana@cmail.carleton.ca

***Abstract*—This research explores the use of the DEVS formalism, and more specifically Cell-DEVS, to perform modeling and simulation for urban growth modeling. DEVS is a general modeling and simulation technique, while Cell-DEVS is an extension which supports cellular automata models. The paper covers the application of Cell-DEVS for population settlement modeling and policy application in the Changjiang Delta Region of China[1]. The primary focus of the paper is as an enhancement to the parent study[1], employing Extended CD++[2] to combine the separate models in the parent study into one optimized model. This allows for yet another formulation of urban networks in the study of metropolitan regions, where complex decisions on management and control of environment and land use factors regarding scale and arrangement of settlements are made. This study uses scenario-based Cellular Automata modeling to produce predictive scenario-based projections within the region of concern.**

Keywords—Cellular Automata; DEVS, Cell-DEVS, scenario, modeling, simulation

I. INTRODUCTION

Urban environments can be modelled as networks. Globally, there are diverging views of management and control of environment, land use, scale and arrangement of settlements. These can be assisted by simulation using now-widely available spatial information. This is a case study of the Changjiang Delta Region in China.

The study area covers 75,900 km^2 of land with 10,200 km^2 of water bodies. There are 16 regional-level cities, 28 county-level cities and 1700 towns.

Four growth scenarios were considered, depending on different growth factors, and are outlined below.

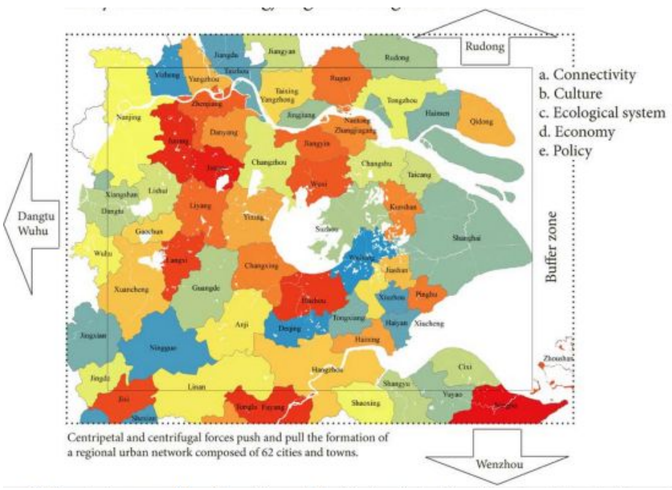

Figure 1: Study Area of the Changjiang Delta Region Urban Network[1]

Different scenarios for city growth are considered, like

- Development corridors
- Corridors with big city growth
- Ecological system concerns, plus development corridors
- Disaster prevention with development corridors

The urban growth prediction model (modified with these scenarios) is given below, and is used as the basis for the Cell DEVS models created for this assignment, covering the above mentioned scenarios.

The goal of this paper is to combine the four scenarios outlined above into one model, rather than the four models as defined in the reference paper[1]. The model produced takes advantage of the input and output ports and state variable features available in Extended CD++[2].

II. Background

A. Urban Growth Prediction Model

The urban growth prediction model uses eight factors to calculate the growth at any particular location in the region, at each point in time. The formulae used are shown in the figure below, outlining how the growth for each scenario is calculated.

Scenario one considers the settlement, spread growth from previous settlements, edge growth from previous settlements, road and corridor settlement factors. Scenario two considers the effect of cities, while scenario three looks at ecological systems concerns, and scenario four considers disaster preventions growth.

$$(a)\ G^{t+1}_{total} = \sum G^{t+1}_{spontaneous(i,j)} + G^{t+1}_{spread(i,j)} + G^{t+1}_{edge(i,j)} + G^{t+1}_{road(i,j)} + G^{t+1}_{corridor(i,j)}$$

$$(b)\ G^{t+1}_{total} = \sum G^{t+1}_{spontaneous(i,j)} + G^{t+1}_{spread(i,j)} + G^{t+1}_{edge(i,j)} + G^{t+1}_{road(i,j)} + G^{t+1}_{corridor(i,j)} + G^{t+1}_{city(i,j)}$$

$$(c)\ G^{t+1}_{total} = \sum G^{t+1}_{spontaneous(i,j)} + G^{t+1}_{spread(i,j)} + G^{t+1}_{edge(i,j)} + G^{t+1}_{road(i,j)} + G^{t+1}_{corridor(i,j)} + G^{t+1}_{ecological(i,j)}$$

$$(d)\ G^{t+1}_{total} = \sum G^{t+1}_{spontaneous(i,j)} + G^{t+1}_{spread(i,j)} + G^{t+1}_{edge(i,j)} + G^{t+1}_{road(i,j)} + G^{t+1}_{corridor(i,j)} + G^{t+1}_{disaster(i,j)}$$

Where,

G^{t+1}_{total} =Total urban growth prediction, at year t+1
$G^{t+1}_{spontaneous(i,j)}$ = Spontaneous growth, the occurrence of random urbanization of land, at year t+1
$G^{t+1}_{spread(i,j)}$ = Spread growth, the urban spreading of newly urbanized land cell, at year t+1 (pay attention that this excluded the spread growth of existing urbanized land cell)
$G^{t+1}_{edge(i,j)}$ = Edge growth, the further expansion of newly spread urbanized cell, at year t+1
$G^{t+1}_{road(i,j)}$ = Road influenced growth, at year t+1
$G^{t+1}_{corridor(i,j)}$ = Development corridor growth, at year t+1
$G^{t+1}_{city(i,j)}$ = Big cities growth, at year t+1
$G^{t+1}_{ecological(i,j)}$ = Ecological system concerns, at year t+1
$G^{t+1}_{disaster(i,j)}$ = Disaster preventions growth, at year t+1

Figure 2: Urban Growth Prediction Model[1]

A. Rationale

This project was conceived of as an enhancement of the approach in the first project that used separate models for each scenario.

Some of the issues encountered in the previous CDBuilder for Eclipse project are outlined below:

Note that due to memory constraints (Java Heap & Out Of Memory exceptions on CD++ Eclipse), the size of the input data was rarefied for non-primary settlement data. The full data set would be a 3 by 3 by 8 matrix, which would not get processed in a reasonable amount of time, and usually did not even terminate due to the above-mentioned errors.

The structure of the matrix consisted of an initial layer (layer zero) of settlements across the Chianjiang region in 2D, plus higher layers containing corridor settlements, road settlements, spread and edge hotspots, ecological and disaster relief areas, etc. These subsequent layers were then condensed into two layers, providing a 30 by 30 by 3 matrix, which was processed in a reasonable amount of time by CD++ (although a much longer execution time when compared to the Lopez simulator).

Layers two and three consisted of interleaved rows of settlement influencing data as described earlier such as road, corridor settlements, spread hotspots, etc. They were interleaved in groups of 3, allowing three different datasets per layer. Having three interleaved datasets was important because it allowed all the datasets to be accessible from a cell's neighbours, in higher layers.

III. Models Defined

B. Combined Scenarios Model Specification

CDR = <X, Y, I, S, Θ, N, d, $\boldsymbol{\delta}_{int}$, $\boldsymbol{\delta}_{ext}$, τ, λ, D>
Where
CDR = Changjiang Delta Region
X = {X_{region}} where
X_{region} = Region year zero settlement data, e.g road, corridor settlements
Y = { G_{total} } where G_{total} = Total growth of the region
I = <η, μ, P^x, P^y> model's modular interface, where
η= 8 (neighbourhood size)
μ= 5 (number of other ports)
P^x = {$scenario_1$, $scenario_2$, $scenario_3$, $scenario_4$, out}
P^y = {$scenario_1$, $scenario_2$, $scenario_3$, $scenario_4$, out}
S = {{$G^t_{total} \in$ R}}
Θ = {$G^t_{spontaneous}$, G^t_{spread}, G^t_{edge}, G^t_{road}, $G^t_{corridor}$, G^t_{city}, $G^t_{ecology}$, $G^t_{disaster}$}
Where
G^t_{total} = Total urban growth prediction at year t
$G^t_{spontaneous}$ = Spontaneous growth, the `` of newly urbanized land cell, at year t
G^t_{edge} = Edge growth, the further expansion of newly spread urbanized cell, at year t
G^t_{road} = Road influenced growth, at year t
$G^t_{corridor}$ = Development corridor growth, at year t
G^t_{city} = Big cities growth, at year t
$G^t_{ecological}$ = Ecological system concerns, at year t
$G^t_{disaster}$ = Disaster preventions growth, at year t
N = { [-1,-1], [0,-1], [1,-1], [-1,0], [1,0], [-1,1], [0,1], [1,1] }
$\boldsymbol{\delta}_{int}$ = { }
$\boldsymbol{\delta}_{ext}$ = {
If (X_{road} == 0) passivate
Else if (P_{xy} == [0, 0])
G^t_{road} += X_{road}
}
τ = {
G^{t+1}_{total} =

$$\sum G^{t+1}_{spontaneous(t,f)} + G^{t+1}_{spread(t,f)} + G^{t+1}_{edge(t,f)} + G^{t+1}_{road(t,f)} + G^{t+1}_{corridor(t,f)}$$

}
λ = { G^t_{total} }

C. Combined Scenarios Model Definition in Extended CD++

```
[top]
components : Region

[Region]
type : cell
dim : (30,30)
delay : transport
defaultDelayTime  : 100
border : wrapped

neighbors : Region(-1,-1) Region(-1,0) Region(-1,1)
neighbors : Region(0,-1)  Region(0,0)  Region(0,1)
neighbors : Region(1,-1)  Region(1,0)  Region(1,1)
initialvalue : 0
localtransition : RegionBehavior
StateVariables: settlement spread edge road corridor city
disaster ecology
StateValues: 0 0.0 0 0 0 0 0 0
InitialVariablesValue: Region.stvalues

NeighborPorts: scenario1 scenario2 scenario3 scenario4



[RegionBehavior]

% Calculate spread & edge
rule : { $spread } { $spread := (-1,0); } 100 { (-1,0)>0 and
(0,0)=0 and (1,0)=0 }
rule : { $spread } { $spread := (1,0); $edge := (1,0); } 100 {
(0,0)=0 and (1,0)>0 }
rule : { $spread } { $spread := (0,-1); } 100 { (0,-1)>0 and
(0,0)=0 and (0,1)=0 }
rule : { $spread } { $spread := (0,1); $edge := (0,1); } 100 {
(0,0)=0 and (0,1)>0 }

% Calculate edge
rule : { $edge } { $edge := 0; } 100 { (-1,0)>0 and (0,0)>0 and
(1,0)=0 }
rule : { $edge } { $edge := 0; } 100 { (0,-1)>0 and (0,0)>0 and
(0,1)=0 }

% Calculate Gtotal
rule : {
```

```
~scenario1 := randInt(1) + $settlement + $spread + $edge +
$road + $corridor + (0,0) +
(1,0) + (0,-1) + (-1,-1) + (1,1) + (-1,0) + (0,1) + (-1,1) + (1,-1);
~scenario2 := randInt(1) + $settlement + $spread + $edge +
$road + $corridor + $city + (0,0) +
(1,0) + (0,-1) + (-1,-1) + (1,1) + (-1,0) + (0,1) + (-1,1) + (1,-1);
~scenario3 := randInt(1) + $settlement + $spread + $edge +
$road + $corridor + $ecology + (0,0) +
(1,0) + (0,-1) + (-1,-1) + (1,1) + (-1,0) + (0,1) + (-1,1) + (1,-1);
~scenario4 := randInt(1) + $settlement + $spread + $edge +
$road + $corridor + $disaster + (0,0) +
(1,0) + (0,-1) + (-1,-1) + (1,1) + (-1,0) + (0,1) + (-1,1) + (1,-1);
~out := randInt(1) + $settlement + $spread + $edge + $road +
$corridor;
}
100
{ $settlement > 0 or $spread > 0 or $edge > 0 or $road > 0 or
$corridor > 0 or $city > 0 or $ecology > 0 or $disaster > 0 }

% Clear cell if no state variables are set, to avoid
indeterminate values in the ports
rule: { ~out := 0; ~scenario1 := 0; ~scenario2 := 0; ~scenario3
:= 0; ~scenario4 := 0; } 100 { t }
```

C. Settlement population levels and the colour palette

Settlement Population Range	Colour (RGB)
0 - 0	255, 255, 255 - white
0.1 - 49	0, 51, 204 - blue
50 - 499	0, 255, 255 - turquoise
500 - 4999	255, 204, 0 - orange
5000 - 49999	255, 0, 255 - pink
50000 - 999999	255,0,0 - red
1000000 - 9.99e7	51,204,0 - green

From the color palette one can get an intuitive feel for how populous an area is based on the colour displayed. Patterns of cell colours provide growth potential information and help to uncover growth trends and relationships

D. Growth factors and state variables

In this model, all the settlement growth factors, and corresponding growth scenarios, were considered in one combined model. They include corridor settlement growth, using corridor and road settlement data for year zero, as well as (randomized) town settlement data. Spread and edge growth arounds these settlements were also simulated.

In addition, big city settlement growth was tested, using big city settlement data. Spread and edge growth arounds these settlements was simulated as well.

The impact of ecological concerns on growth was tested, using ecological location data. Spread and edge growth arounds these settlements was simulated.

Lastly, the impact of disaster prevention on growth was tested, using disaster prevention settlement data, such as moving settlements away from flood zones, landslide zones and other high risk areas.

IV. Simulation Results

The combined model was developed using the Extended CD++ Lopez version. The Lopez simulator RISE client and API was used for testing and verification[3]. The model was developed using a text editor, along with the input variables file. The palette file was developed using CD++ Modeller in CD++ Builder plugin for Eclipse 3.6.0.

No compilation is needed to run the models. Simply follow the Lopes simulator RISE client user manual[3]. This requires an internet connection as it accesses an Amazon web service. The Webviewer[4] tool was used for loading and visualization of the model and related log files created by the Lopez simulator.

B. *How to run a simulation - an example*

Using the RISE client[3], open a command terminal, change directory to the unzipped project folder:

unzip RegionEnhancedSimulationProject.zip

cd RegionEnhancedSimulationProject/

You may then use the RISE client to put the XML model descriptor into Lopez:

java -jar <path_to_RISE_REST_client_jar> PutXMLFile test test test/lopez/RegionModelTest Region.xml

You can then upload the model zip file:

java -jar <path_to_RISE_REST_client_jar> PostZipFile test test test/lopez/RegionModelTest?zdir=Region Region.zip

It is then time to run the simulation:

java -jar <path_to_RISE_REST_client_jar> PutFramework test test test/lopez/RegionModelTest/simulation

Should you encounter a problem, or would just like to restart the simulation, you may delete the uploaded framework and restart the steps above, using the documented[3] command below:

java -jar <path_to_RISE_REST_client_jar> DeleteFramework test test test/lopez/RegionModelTest

The simulation should complete within seconds, and the results (results.zip) archive is now available on the Lopez framework for download at:

http://vs1.sce.carleton.ca:8080/cdpp/sim/workspaces/test/lopez/RegionModelTest

C. *How to visualize and validate simulation results - an example*

Download and unzip the results.zip file from the previous section. Note the location of the Region.log01 file.

You may then launch the ARSLab CellDEVS Simulation Viewer[4]. The input model file (Region.ma), values file (Region.stvalues), and palette file (Region.pal) are found in the project archive:

<unzipped_project_dir>/Region

The required log file is to be uploaded from the results.zip archive downloaded from the previous step (Region.log01).

When the files are uploaded on the Webviewer[4], you may start the log parsing action, and when it is complete, navigate to different frames of the visualization, recording video if needed.

D. *Interpreting the results - an example*

The Webviewer[4] should show four grids, one for each output port, which translates to one grid for each scenario. They should look as shown below:

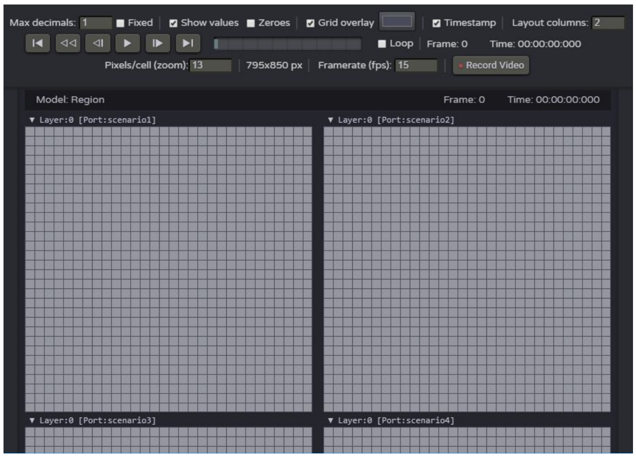

Figure 3: Grids for each port/scenario on Webviewer[4]

Advancing through the simulation reveals the initial positions of the settlement types in each scenario, outlined below.

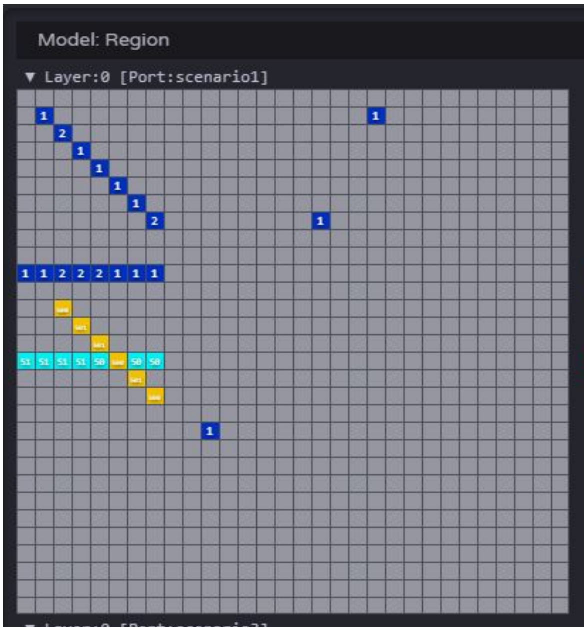

Figure 4: Scenario 1 initial positions: blue are basic settlements, turquoise are the road, and orange are corridor settlements

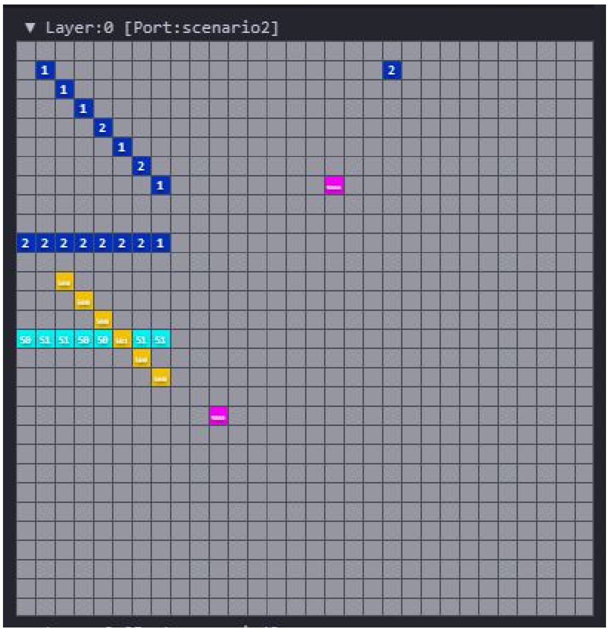

Figure 5: Scenario 2 initial positions: blue are basic settlements, turquoise are the road, and orange are corridor, and pink the city settlements

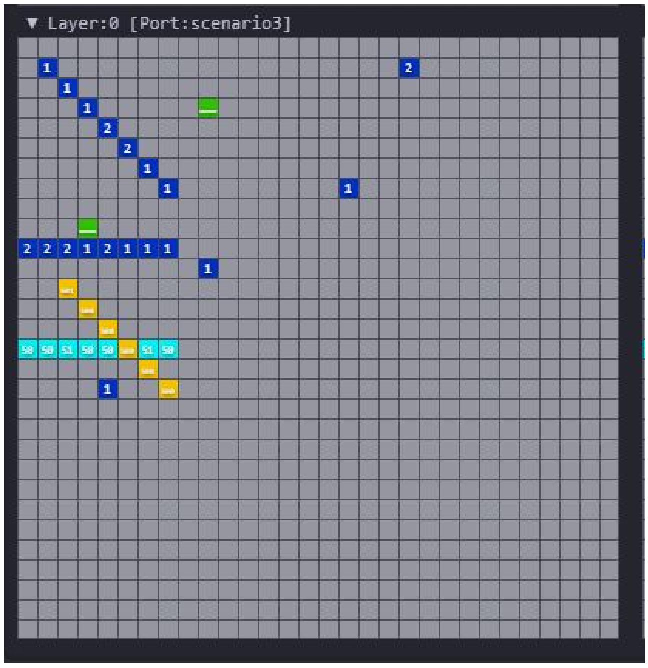

Figure 6: Scenario 3 initial positions: blue are basic settlements, turquoise are the road, and orange are corridor settlements, and green are the ecological zones

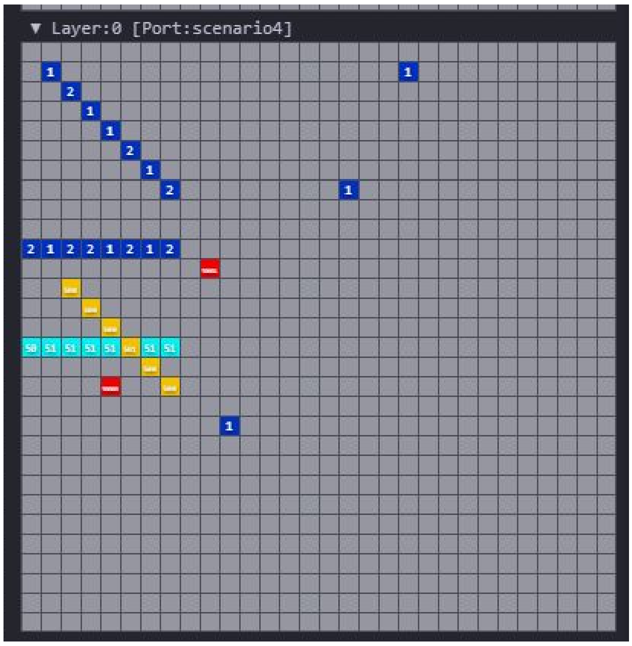

Figure 7: Scenario 4 initial positions: blue are basic settlements, turquoise are the road, and orange are corridor settlements, and red are the disaster prevention settlements

Frame 3, a few frames down from the initialization frame, is shown below for all scenarios, capturing the initial population spread and edge growth.

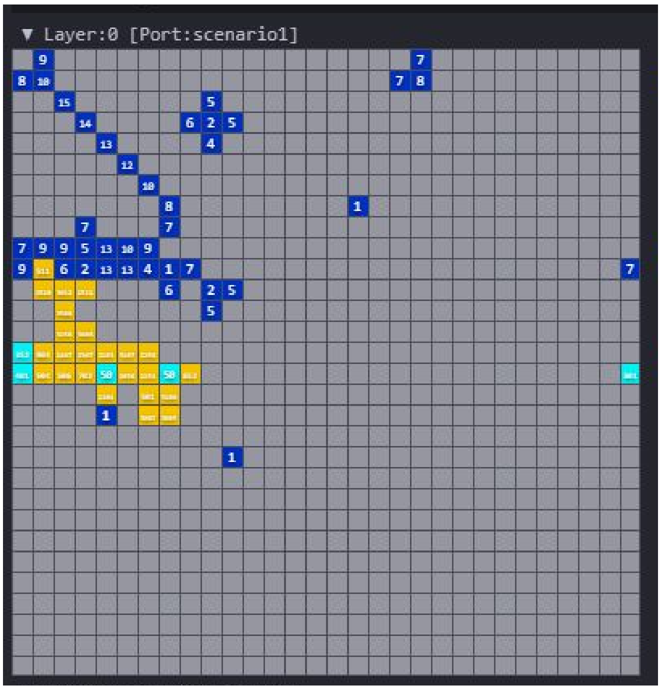

Figure 8: Scenario 1 three frames on: blue are basic settlements, turquoise are the road, and orange are corridor settlements. There is initial spread and wrap-around for the road settlements.

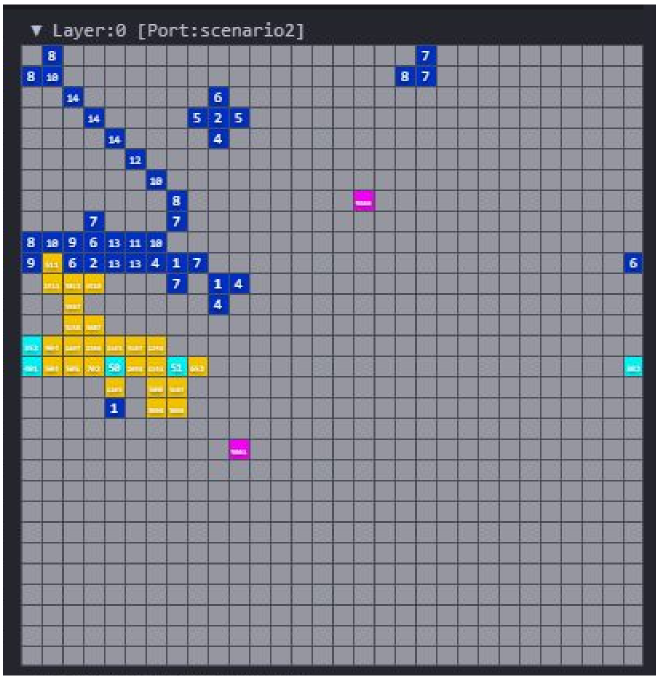

Figure 9: Scenario 2 three frames on: blue are basic settlements, turquoise are the road, pink are big city and orange are corridor settlements.

There is initial spread and wrap-around for the road settlements. Minimal growth around cities

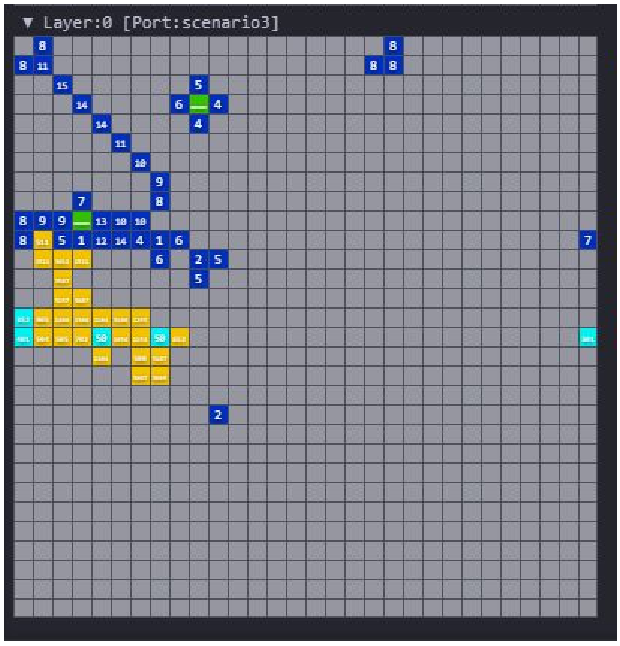

Figure 10: Scenario 3 three frames on: blue are basic settlements, turquoise are the road, green are ecological zones and orange are corridor settlements.

There is slight growth around the ecological zones

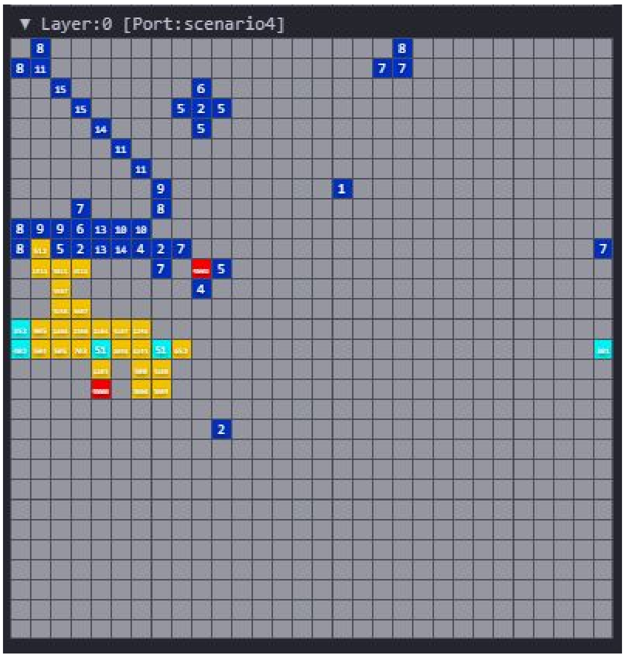

Figure 11: Scenario 4 three frames on: blue are basic settlements, turquoise are the road, red are disaster relief and orange are corridor settlements.

There might be some growth around the disaster relief settlement sites, but its not very clear.

After a few more frames, population levels that indicate big cities are popping up spontaneously in most scenarios, as pink settlements

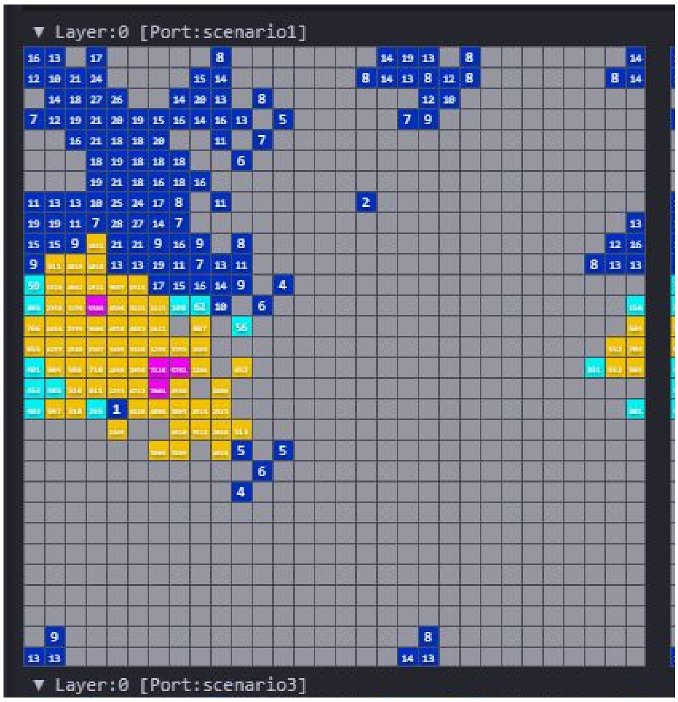

Figure 12: Scenario 1 five frames on: blue are basic settlements, turquoise are the road, pink city and orange are corridor settlements.

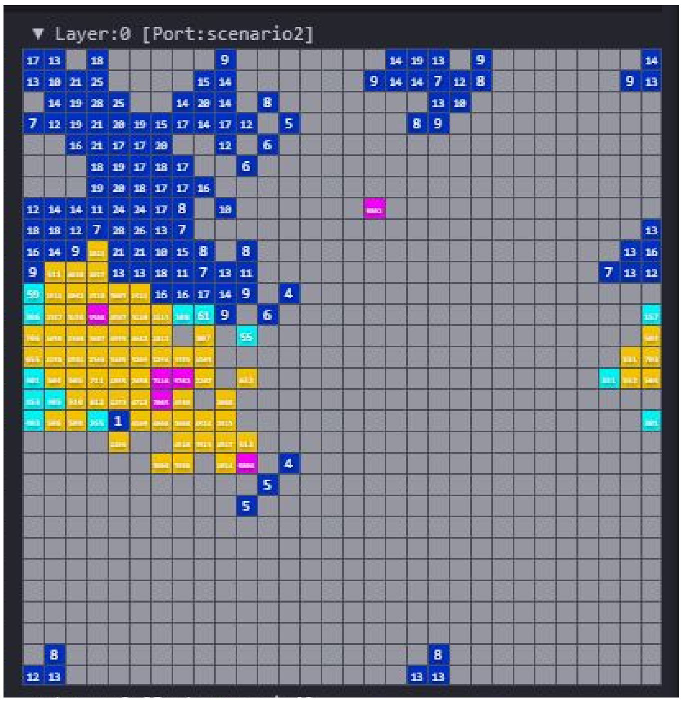

Figure 13: Scenario 2 five frames on: blue are basic settlements, turquoise are the road, pink city and orange are corridor settlements.

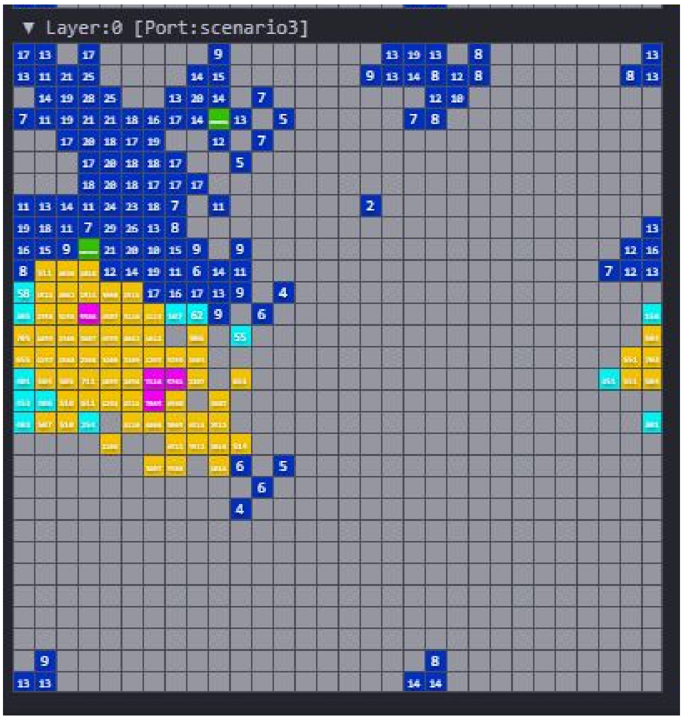

Figure 14: Scenario 3 five frames on: blue are basic settlements, turquoise are the road, pink city, green ecological zones and orange are corridor settlements.

The newly formed city settlements are fully surrounded, but themselves aren't expanding.

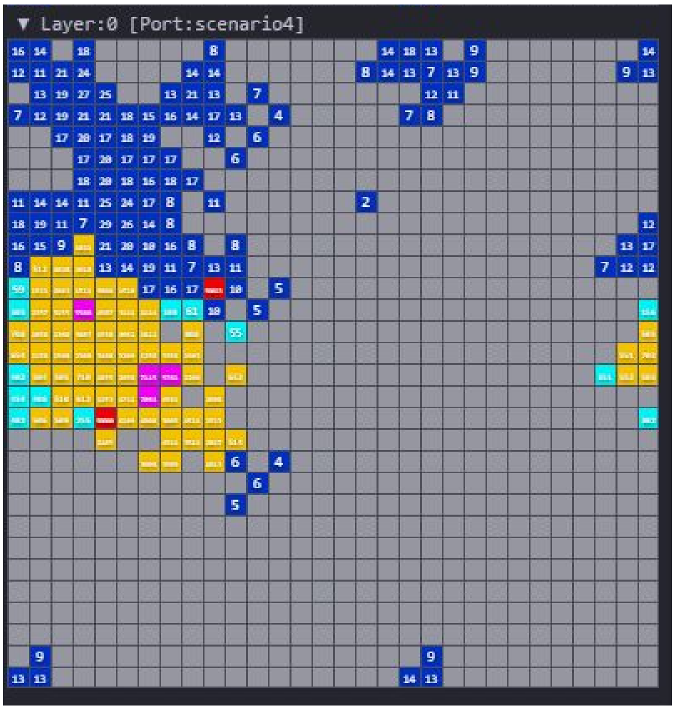

Figure 14: Scenario 4 five frames on: blue are basic settlements, turquoise are the road, pink city, red disaster relief and orange are corridor settlements.

Seven frames in, it appears that the road settlements are surrounding the corridor settlements in most scenarios.

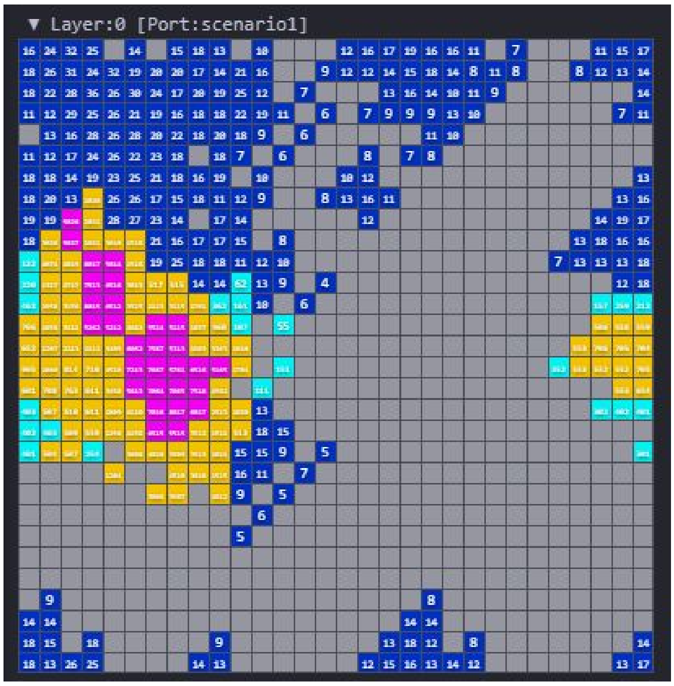

Figure 15: Scenario 1 seven frames on: blue are basic settlements, turquoise are the road, pink city, red disaster relief and orange are corridor settlements.

Road settlements appear to be surrounding corridors, and cities are expanding at the center of these corridors.

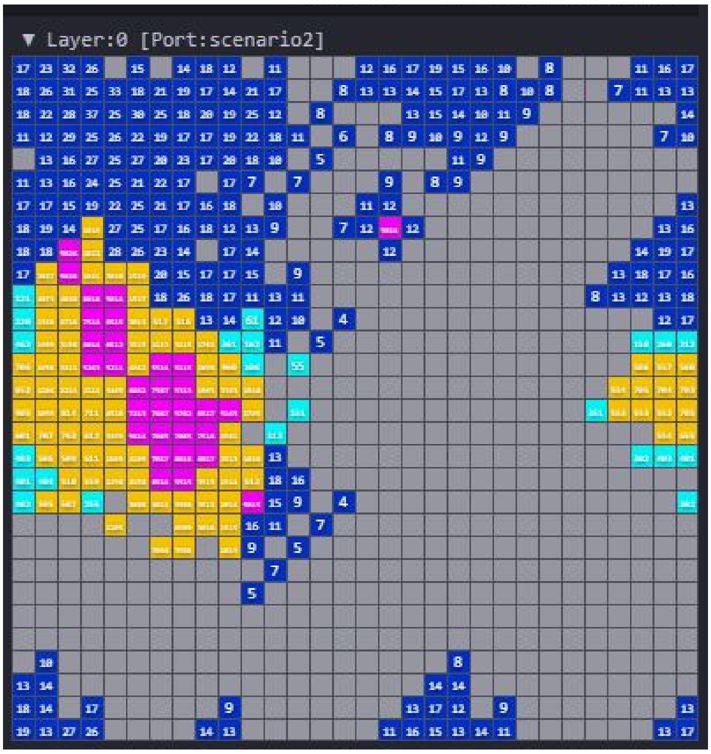

Figure 15: Scenario 2 seven frames on: blue are basic settlements, turquoise are the road, pink city, red disaster relief and orange are corridor settlements.

Similar story to scenario 1 with the road to corridor to city relationship.

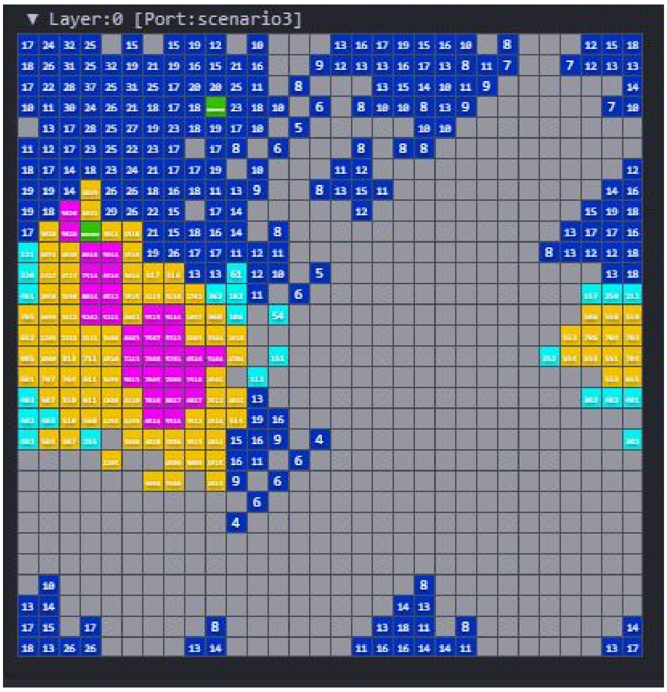

Figure 16: Scenario 3 seven frames on: blue are basic settlements, turquoise are the road, pink city, red disaster relief and orange are corridor settlements.

Similar behaviour to scenario 2 at this point, plus ecological zones which appear to be attracting low levels of settlement growth.

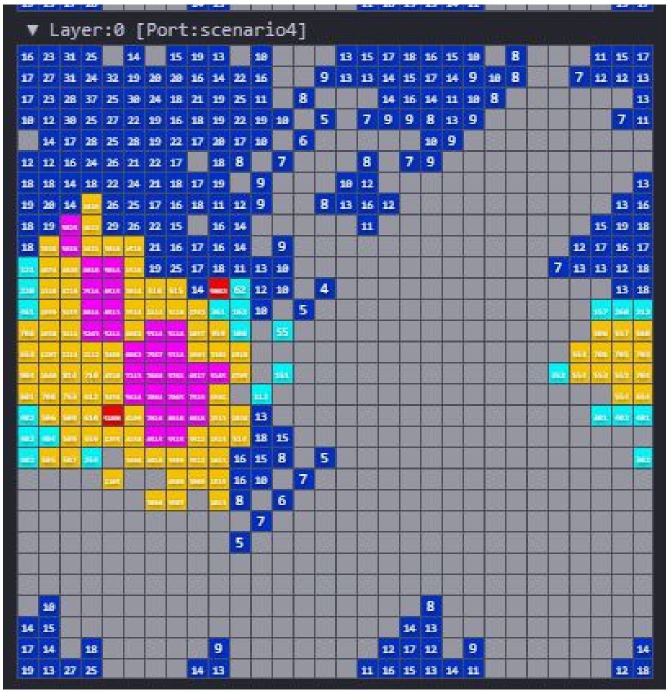

Figure 17: Scenario 4 seven frames on: blue are basic settlements, turquoise are the road, pink city, red disaster relief and orange are corridor settlements.

Not as much growth around disaster zones, although they are fairly surrounded with low levels of settlement.

In the end, 11 frames on, there appears to be universally present a cluster of cities in the corridor to the north west area of the region.

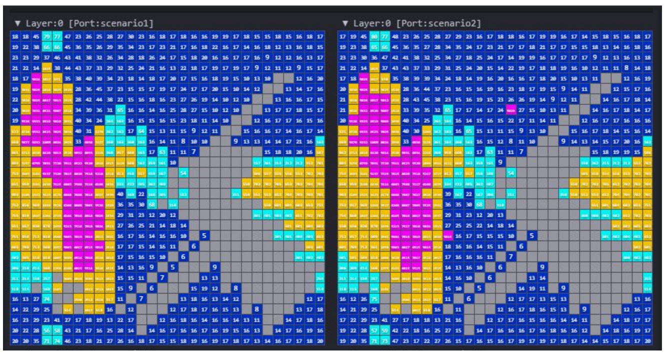

Figure 18: Scenarios 1 and 2 eleven frames on: blue are basic settlements, turquoise are the road, pink city, red disaster relief and orange are corridor settlements.

There is a cluster of big cities over the western part of the region

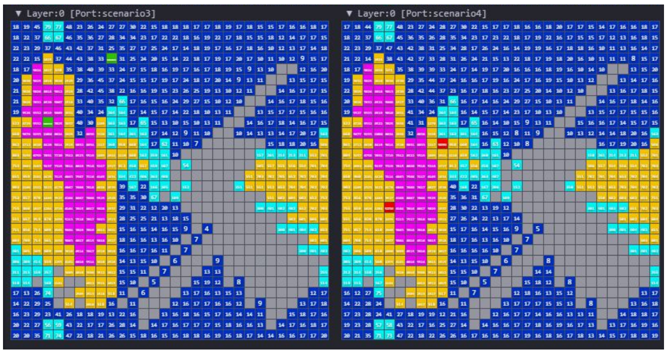

Figure 19: Scenarios 3 and 4 eleven frames on: blue are basic settlements, turquoise are the road, pink city, red disaster relief and orange are corridor settlements.

There is a clustering of big cities to the west, and a large uninhabited zone to the east of all the scenarios.

E. Contextualizing the results within real world geography

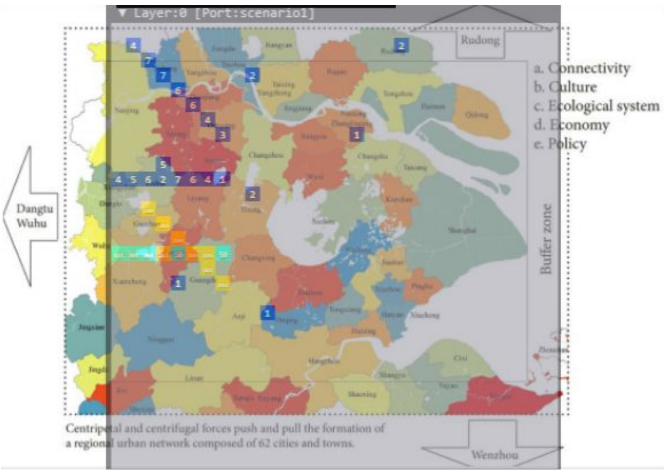

Figure 20: Scenarios 1 initial frame overlay on Chainjiang region map: blue are basic settlements, turquoise are the road, pink city, red disaster relief and orange are corridor settlements.

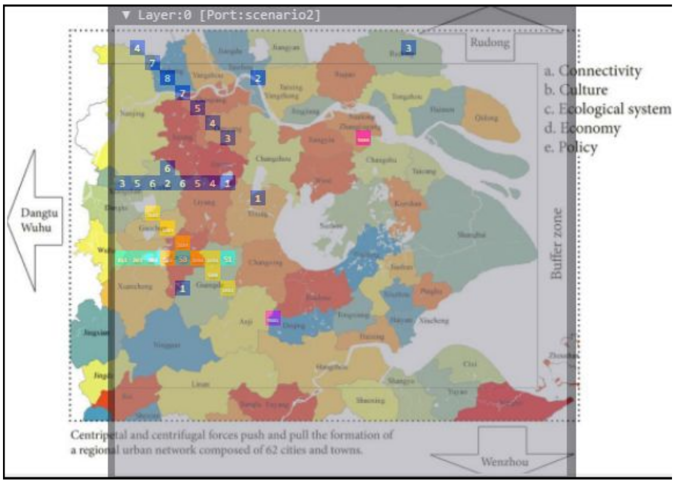

Figure 21: Scenarios 2 initial frame overlay on Chainjiang region map: blue are basic settlements, turquoise are the road, pink city, red disaster relief and orange are corridor settlements.

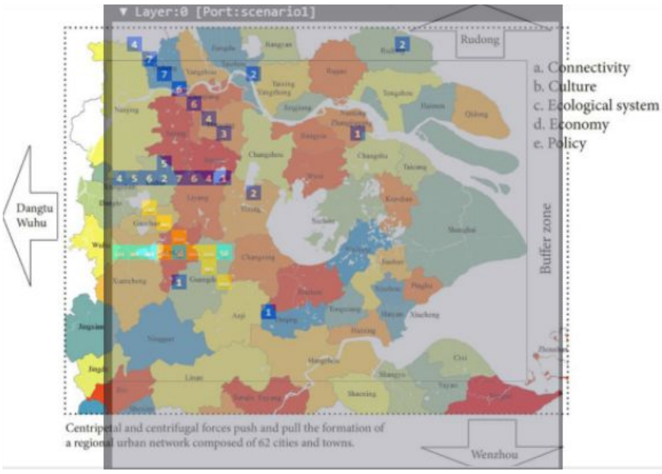

Figure 22: Scenario 1 frame 3 overlay on Chainjiang region map: blue are basic settlements, turquoise are the road, pink city, red disaster relief and orange are corridor settlements.

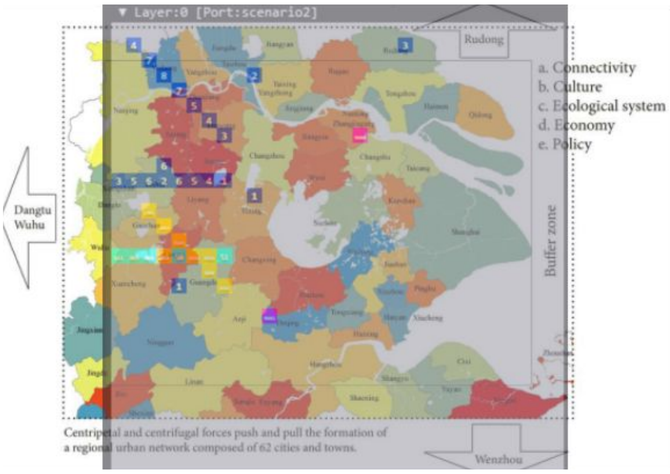

Figure 23: Scenario 2 frame 3 overlay on Chainjiang region map: blue are basic settlements, turquoise are the road, pink city, red disaster relief and orange are corridor settlements.

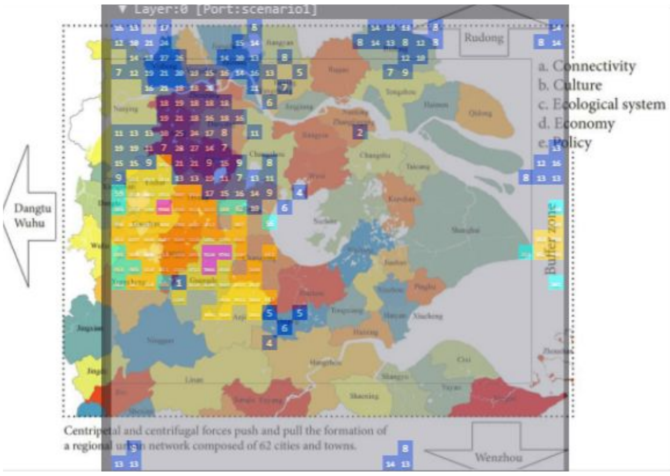

Figure 24: Scenario 1 frame 5 overlay on Chainjiang region map: blue are basic settlements, turquoise are the road, pink city, red disaster relief and orange are corridor settlements.

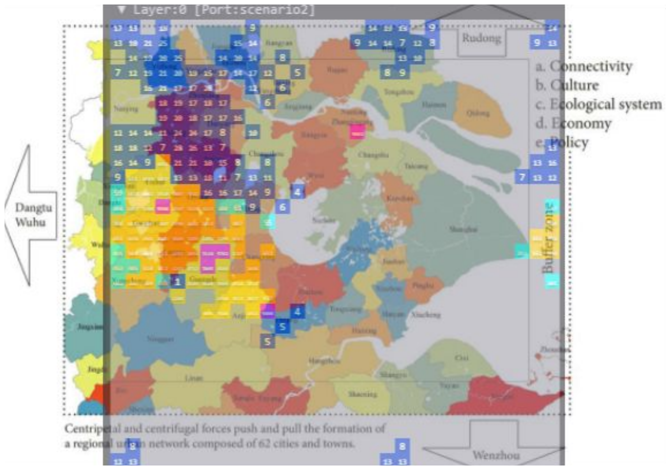

Figure 25: Scenario 2 frame 5 overlay on Chainjiang region map: blue are basic settlements, turquoise are the road, pink city, red disaster relief and orange are corridor settlements.

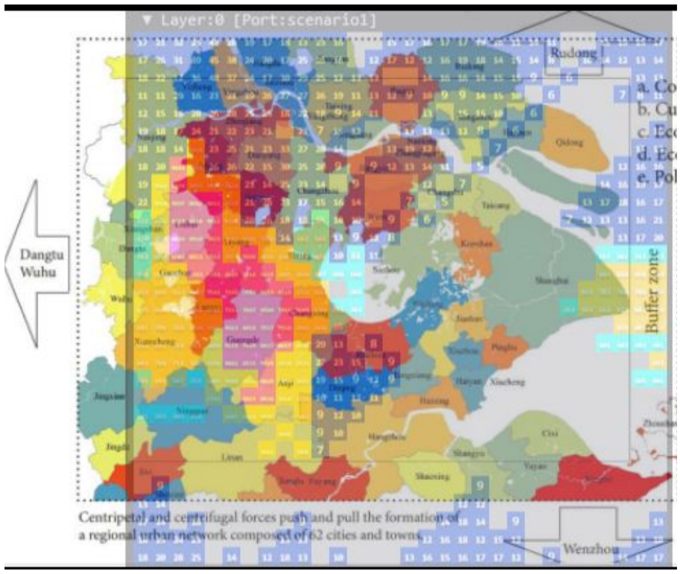

Figure 25.1: Scenario 1 frame 8 overlay on Chainjiang region map: blue are basic settlements, turquoise are the road, pink city, red disaster relief and orange are corridor settlements.

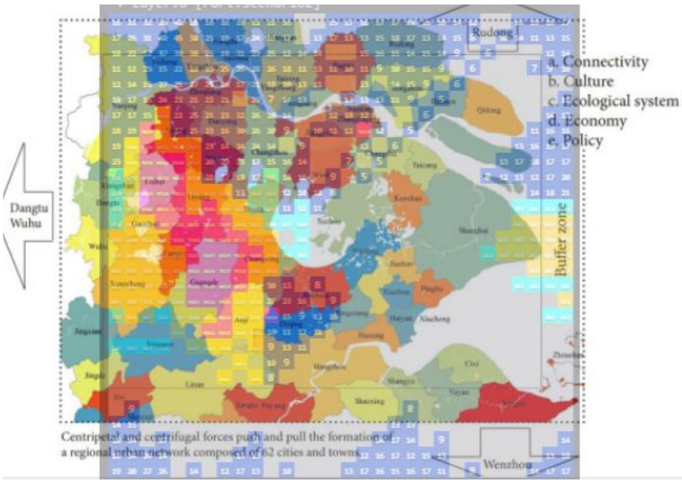

Figure 25.1: Scenario 2 frame 8 overlay on Chainjiang region map: blue are basic settlements, turquoise are the road, pink city, red disaster relief and orange are corridor settlements.

A. Performance Analysis

The execution time of Extended CD++[2] was compared to that of CDBuilder for Eclipse 3.6.0. This sort of comparison is not completely accurate due to the fact that the Lopez simulator used was operating remotely on a different platform (Linux) and host. Also, since it is a dedicated system, it is quite likely that the dedicated RISE REST server[3] used for running the simulations was configured in an expert manner and would be likely to have superior performance. The comparison system used in assignment 2 was ran on the local machine on Eclipse.

Overall there were orders of magnitude improvement in execution time. We shall now refer to the assignment 2 local system as system 1, and the REST based Lopez simulator as system 2.

Typically, system 1 would complete execution in 2 to 3 minutes. System 2 would complete execution in seconds, very often under 3 seconds

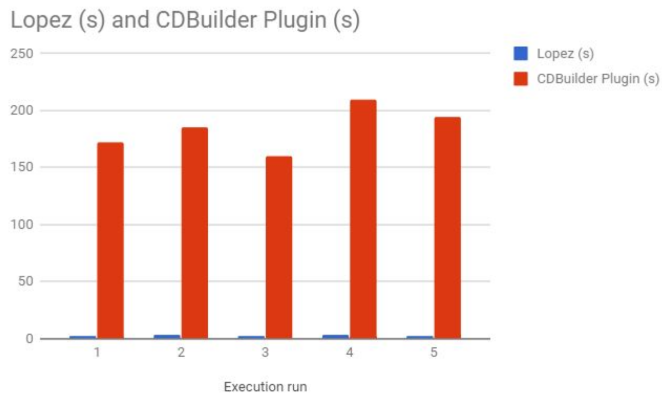

Figure 26: Plot of Lopez simulator execution time versus CDBuilder Plugin execution time over multiple runs.

B. Performance analysis: Sanity and expected result

Analysis of the two systems and their algorithms would expect the Extended CD++ Lopez simulator (system 1) to take longer, and to utilize more resources than the CD++ Plugin for Eclipse (system 2), since it has a much more expansive feature set, and handles more complex models than system 2.

System 1 also appears to have a more complex algorithm, necessary to support some of its new features, like multiple input and output ports, multiple states per cell.

Conclusion

Using the Extended CD++ Lopez simulator was easier to use than expected, and certainly runs more quickly than a locally ran CDBuilder Plugin simulator, which is more tied to the state of the machine.

As mentioned earlier, this is probably an effect of using the online Lopez simulator rather than the compiled desktop version, allowing for the benefits of the cloud and other optimizations to be brought to bear.

Extended CD++ was much easier to use when developing the model in this project, because it involved taking four different models and merging them into one model, which is not too complicated, certainly appearing less complicated then the sum total of the original four models.

From an algorithmic standpoint, using state variables as opposed to extra layers of cells reduces the size of the grid under consideration in CellDEVS. This certainly reduces the overall memory footprint, and probably also the processing time (when compared to having extra layers).

References

The following references were used in this report.They are mostly resources made available by the professor during the course of the semester. If there was no journal or other publication, the origin URL or source document is referenced instead.

[1] Guan, ChengHe; Rowe, Peter Grimmond, “Should big cities grow? Scenario-based cellular automata urban growth modeling and policy application”, https://dash.harvard.edu/handle/1/33747421

[2] Alejandro Lopez, Gabriel Wainer, “Extending CD++ Specification Language for CellDEVS Model Definition”, Carleton University (not for distribution).

[3] Sixuan Wang, Gabriel A. Wainer, “RISE User Manual”, Carleton University (not for distribution)

[4] Omar Hesham, CellDEVS Simulation Viewer, http://omarhesham.com/arslab/webviewer/, Carleton University ARS lab.