Asymmetric Cell-DEVS-Based Modelling for Multistate voter models

by

Alexandre Marques
Student ID: 101189743

Submitted to: Professor Gabriel A. Wainer

Term Project
SYSC 5104 - Methodologies for Discrete Event Modelling and Simulation

Department of Systems and Computer Engineering
Faculty of Engineering
Carleton University

April 26th 2025

Contents

1 Introduction

The asymmetric Cell-DEVS voters model is a variation on the Cell-DEVS voters model done from Assignment 2 [3]. The motivation behind this conversion was to observe the impact of asymmetric relationships and varied neighborhood sizes for cells. This is because in real life, each individual has a different network of connections, it is much more accurate to model relationships as graphs instead of cell grids. There can also be asymmetric influences, for instance, a parent may have a stronger influence on their child than the child on the parent.

The goal of this version of the voters model is to evaluate how different networks of connections between individuals affect the evolution of their political views. This should be more helpful in making predictions on the evolution of political views in societies as a whole.

To keep the scope of testing manageable, the simulations are variations on one scenario. The network is formed of two complete graphs with three cells with equal connection weights. These graphs are connected one cell which serves as a mutual connection. The controlled variable is the asymmetry of the connection weight on this mutual connection. The two graphs start with opposite positions (values, or votes), and leave no cell undecided initially. This allows us to observe if the positions change, or stabilize eventually.

2 Background

2.1 Voter Model

The Multistate Voter Model is a simple probabilistic cellular automata model. It is a useful model to study trends relating to how sharing beliefs can influence others in societal settings [1]. In this cellular automata model, each cell represents a voter, with a defined set of chosen **preferences**. Each cycle, cells which have a neighbor with a different preference have a chance to change their preference to this neighbor's preference, with a chosen **probability weight**.

2.2 Asymmetric Cell-DEVS

Models made with Asymmetric Cell-DEVS (*ACD*) are a variation of typical cellular automata [2]. Cells in ACD have a current state, inputs from their neighbors, outputs to their neighbors, and a function to describe how the cell changes based on inputs, time and state. What makes ACD different from most cellular automata is that each cell can have its own set of input neighbors, and a separate set of output neighbors. It also allows for these connections to have different weights.

The advantages of ACD is the fine grained control it provides over the composition between each cell. While most cellular automatas can be seen as a grid, ACD is better visualized as a graph. Graphs are useful structures to analyze complex connection behaviors, as is often the case in human relationships.

This extra control is also a drawback since scaling the model requires more considerations to compose it. To circumvent this, it is possible to use cells to represent varying group sizes by taking advantage of the asymmetry. For example, it is just as feasible to model a simulation between five people as it is to model five regions in ACD, as long as their average behaviors are known.

3 Models

3.1 Voter Cells

3.1.1 Description

Each cell in the model represents a voter. Voters have a *preference*, initially set based on the test scenario. This preference can change if the voter has an *input neighbor* with a different preference than them in the current cycle.

Input neighbors are neighbors which can influence the cell, and they can have a *strong*, *normal* or *weak* influence on the cell. A voter also has a set of *output neighbors*, which are the cells which it influences itself.

In any individual cycle, a voter can choose to keep their position, or change it to any of their input neighbors, with a higher weight on stronger influences. A voter is treated as their own neighbor with a normal influence on themselves.

The more neighbors a cell has, the more likely it is to change. Considering it this way also allows the definition of more or less "stubborn" cells which are less likely to be influenced. Since we do not vary these factors, it is not relevant for our simulations but it could be another factor to control in future simulations.

3.1.2 Voter Implementation

The formal specification for a voter Cell is defined as follows:
$< X, Y, S, N, d, \tau, \delta_{int}, \delta_{ext}, \lambda, D >$

- $S = \{B, R\}$
- $X = \{x | x \in S\}$
- $Y = \{y | y \in S\}$
- $N = S^2 \vee S^3 \implies$ All cells have either two neighbors, or three*
- d = 1 (time unit)
- $\delta_{int}, \delta_{ext}, \lambda, D$ defined using Cell-DEVS specifications.

*All cells connect to their two neighbors with their shared initial preferences. Depending on the simulation scenario, cells can also connect to an additional cell. The different scenarios are described in detail in section 3.2.1.

- $\tau : N \to S$ is defined by this pseudocode:

    1. Repeat the following instructions for each cycle.
    2. Def bag = {}
    3. For all neighbors, add to bag neighbor.status as many times as their weight indicates.
        - Important note: A cell is in its own neighborhood
        - Weight in this case is an int value indicating the cell vicinity.
    4. If the bag is uniform, do nothing (becomes quiescent until a new input).
    5. Else, grab a random status from the bag, set it to self (with float noise).**
    6. Wait for next cycle.

**In implementation, the statuses are implemented as floats to be able to force a check even if states remain logically the same. The simulator attempts to optimize outcomes because it assumes the model is deterministic, if the state remain the same with no changes in inputs, it believes it should be quiescent. In reality, we can detect if it is possible for a change to occur if the bag is not uniform. In this case, we force an update on the next cycle by changing the float value while remaining in the same status range.

3.2 Model Graph

3.2.1 Graph Description

The graph used varies slightly for all tests, but in a controlled fashion. Each cell is a node in one of two complete graphs of size three. The initial state of the positions is always the same, both graphs are fully saturated with a different position.

There is a single mutual connection between the graphs to establish communication. The weight of this connection varies to evaluate the impact it has through different tests.

For one test, an even normal weight is given. This could be a connection between two colleagues, or friends from two different groups.

Another test uses a skewed weight. A strong influence is given on one side, while a weak influence is given on the other. This could be a connection between a parent (strong) and their child (weak) for example.

The last test uses a unilateral weight, this connection could represent a broadcast, movie, political propaganda, or any sort of influence which cannot be changed itself by the ones it influences. These are common in reality, but likely less influential than closer bonds. For this reason, a weak influence is evaluated for this unilateral bond.

3.2.2 Formal Graph Coupling

The formal specification for the coupled voter graph is defined as follows:
$< X, Y, C, EIC, IC, EOC >$

- X, Y, EIC, EOC: N/A - No external input, output or connections.
- C: The set of voter cells, as defined in section 3.1.2. For the simulations, there are always exactly six cells, forming two isolated neighborhoods (or *sub-graphs*).
- IC: N/A - Cells are coupled through neighborhoods

3.2.3 Graph Coupling Implementation

Most of the implementation for the coupling is actually done through JSON configurations.

```
{
    "cells": {
      "default": {
        "delay": "transport",
        "model": "voter"
      },


      "cb1": {
        "state": {"preference": 0},
        "neighborhood": {"cb1": 3, "cb2": 3, "cb3": 3, "cr1": 1}
      },
      "cb2": {
        "state": {"preference": 0},
        "neighborhood": {"cb1": 3, "cb2": 3, "cb3": 3}
      },
```

Figure 1: A portion of the configuration for the unilateral graph.

In figure 1 there is an element to configure a default cell, and the first two blue cells. "cb1" is of particular importance as this is the blue cell which communicates with the red cells.

For the unilateral test, the normal weight is 3, and the weak influence from the red side is 1. This means the red cell has a third of the influence from an individual blue node. In the initial state, thus the total probability of "cb1" switching becomes 1/9.

For other tests, a normal weight of 3 is used everywhere unless a weak weight is mentioned (1), or strong weight (6).

4 Simulations

4.1 Graph Legend

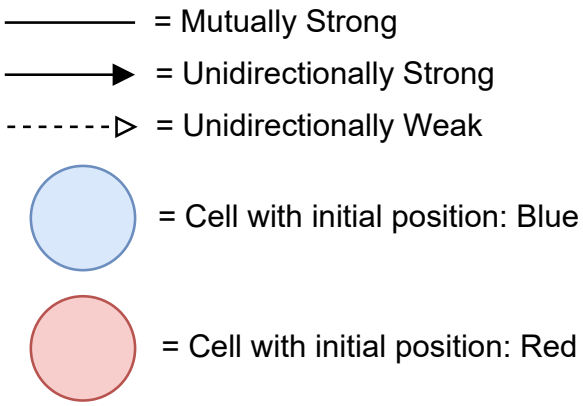

Figure 2: The legend of graph symbols used

Different connection types are used, and two colors are used to show voter initial positions. Figure 2 shows the symbols which will be used to display the initial states of each simulation to follow.

4.2 Variance in Results

Due to the stochastic elements of the simulation, one outcome does not always show the complete picture. Ideally, data collection should have been automated to perform stochastic analysis and get more rigorous conclusions. To keep the scope of this project manageable, this was not done. Instead, each simulation was ran a dozen times to take multiple observational samples. The observations described in the conclusions of each simulation factor multiple tests, not just the ones displayed in the result.

4.3 Equal Weight

4.3.1 Initial Configuration

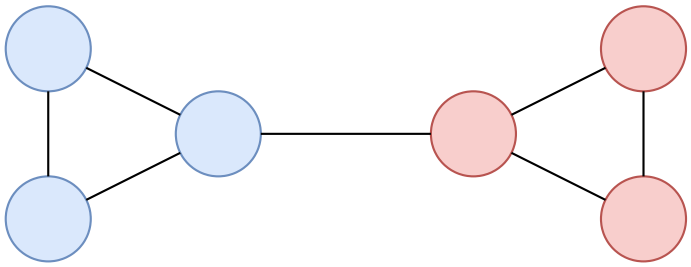

Figure 3: Graph of the initial configuration for the Equal simulation.

There are two sub-graphs in figure 3: A blue sub-graph (left) and a red sub-graph (right). There are two cells connecting them with a mutually strong connection. This could simulate be a relationship between two friends, for example.

4.3.2 Hypothesis

Due to the initial configuration, the cells connecting the subgraph are more likely to stay the same due to their neighbors in the complete sub-graph. In the event that they do flip, they may eventually convert their local complete graph, but it is equally likely they revert to the initial state. There is a very rare chance that both of the communicating cells flip at the same time, and an even rare chance that both sub-graphs fully flip positions from that point.

With this in mind, the hypothesis is that the graph will fully convert to one position, but it will resist this change due to the connected graphs. Stronger connections would slow the full conversion even more.

4.3.3 Results

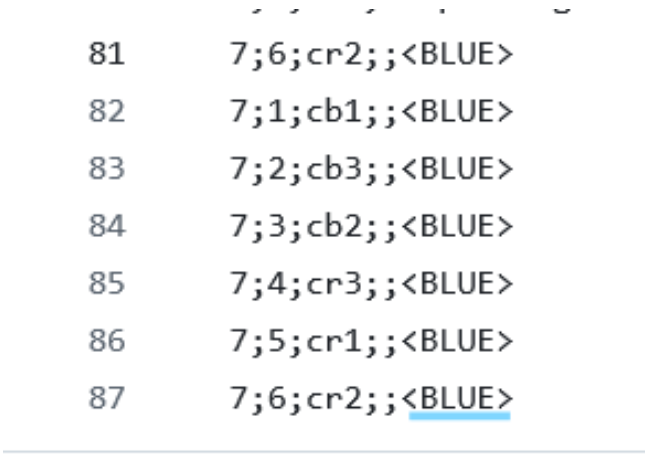

Figure 4: An end result of the even weight test. Stabilizes Blue at 6 cycles.

4.3.4 Conclusion

There was no clear bias towards one specific color, as expected in the hypothesis. The cycles required to fully convert were lower than expected. Often staying around five cycles on average, and rarely more than twelve cycles.

Note: The first number in the logs show the time elapsed in the simulation. The last value shows the voter preference (blue, or red).

4.4 Skewed Weight

4.4.1 Initial Configuration

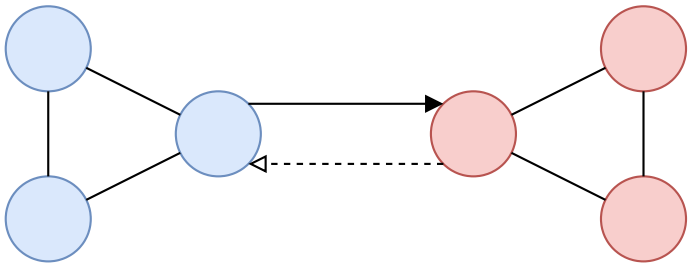

Figure 5: Graph of the initial configuration for the Skewed simulation.

To be concise, the "red node" is defined as the node in the red sub-graph connected to the blue sub-graph, and the "blue node" is the node in the blue sub-graph connected to the red sub-graph. This will be common notation for the next two tests.

The red node in figure 5 has a weak influence on the blue node. The blue node has a strong influence on the red node. This could represent an asymmetrical relationship between a parent and a child, for example.

4.4.2 Hypothesis

In contrast with the even weight, there is a stronger influence from the blue node onto the red node. This means the red node will flip to blue more often than the blue node will flip to red. The additional neighbors again provide stability making a full conversion more difficult. A full red conversion is made more difficult because it is harder to stabilize the flipped blue node due to the weak supporting influence compared to the strong support it would influence on the flipped red node.

With this in mind, the hypothesis is that the graph will fully convert to blue most of the time. The disproportionate weights make it so there are more opportunities to flip a node on the red side and cause a full conversion. The neighbors also make the weight more important, since the stronger it is, the more time the dissenting node has to convert the others. The odds of a full blue conversion are hypothesized to be even likelier than the odds of converting the red node in the initial configuration.

4.4.3 Results

```
323     38;6;cr2;;<BLUE>
324     38;1;cb1;;<BLUE>
325     38;2;cb3;;<BLUE>
326     38;3;cb2;;<BLUE>
327     38;4;cr3;;<BLUE>
328     38;5;cr1;;<BLUE>
329     38;6;cr2;;<BLUE>
```

Figure 6: An end result of the skewed weight test. Stabilizes Blue at 38 cycles.

4.4.4 Conclusion

There was a clear bias towards a stable blue ending, but stabilizing red still occurred rarely. This concurs with the hypothesis for this test.

One interesting behavior that was observed was the difference in the time required to stabilize. Simulation times were almost always longer than the even configuration. Due to the increased likelihood to fully convert, it was believed it would actually take less time to do so (and thus fewer cycles). To get a better understanding of why this occurs, a more rigorous probabilistic analysis may be needed over multiple samples.

4.5 Unilateral Weight

4.5.1 Initial Configuration

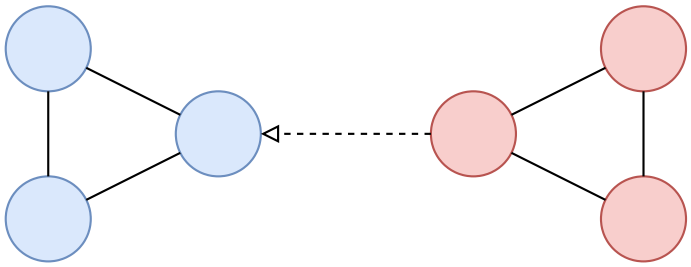

Figure 7: Graph of the initial configuration for the Unilateral simulation.

The red node in figure 7 has a weak unilateral influence on the blue node. This could be an influence by a commercial or broadcast by the red node seen by the blue node.

4.5.2 Hypothesis

In this graph, it is impossible to flip the red sub-graph to blue. However, it is unlikely to flip the blue node. Once it is flipped, it should still be more likely to return to blue than to fully convert the graph.

The hypothesis for this test is a full conversion to red will happen, although it may take a long time. There will always be a possibility to flip a node to red, but the same cannot be said for blue. Given infinite time, the graph should stabilize to red.

4.5.3 Results

```
194     37;3;cb2;;<RED>
195     37;1;cb1;;<RED>
196     37;2;cb3;;<RED>
197     37;3;cb2;;<RED>
198     37;4;cr3;;<RED>
199     37;5;cr1;;<RED>
200     37;6;cr2;;<RED>
```

Figure 8: An end result of the unilateral weight test. Stabilizes Red at 37 cycles.

4.5.4 Conclusion

It always stabilized to red, given enough time. This concurs with the hypothesis for this test.

5 Conclusion

Using asymmetric Cell-DEVS, it was possible to observe changes in preferences between different entities and relationship types. Specifically, this project analyzed the changes in behavior when the type of connection between two groups with distinct preferences change.

Although the focus in this paper is on one specific scenario, it should show the potential of Asymmetric Cell-DEVS for voter analysis. In the future, the influence of different initial position configurations could be analyzed, as well as different initial networks. With complex asymmetric graphs, small changes in initial positions may lead to chaotic and unpredictable changes.

For a practical use case, nodes could be replaced to represent demographics or groups and influences of importance. Data could be collected to find their initial positions, and influencing weights. Once the model is simulated, the evolution of positions could be observed over time to predict voting outcomes.

References

[1] S.G Alves, N.M Oliveira Neto, and M.L Martins. "Electoral surveys' influence on the voting processes: a cellular automata model". In: *Physica A: Statistical Mechanics and its Applications* 316.1–4 (Dec. 2002), pp. 601–614. ISSN: 0378-4371. DOI: 10.1016/s0378-4371(02)01208-6. URL: http://dx.doi.org/10.1016/s0378-4371(02)01208-6.

[2] Román Cárdenas and Gabriel Wainer. "Asymmetric Cell-DEVS models with the Cadmium simulator". In: *Simulation Modelling Practice and Theory* 121 (2022), p. 102649. ISSN: 1569-190X. DOI: https://doi.org/10.1016/j.simpat.2022.102649. URL: https://www.sciencedirect.com/science/article/pii/S1569190X22001198.

[3] Alexandre Marques. *Cell-DEVS-Based Modelling for Multistate voter models*. 2025. URL: https://github.com/orselane/Voters-cell-DEVS/blob/main/docs/FinalReport.pdf.