

**SYSC 5104/HCIN 5405 (METHODOLOGIES FOR DISCRETE EVENT MODELLING AND SIMULATION) - ASSIGNMENT 2 (SYMPATRIC SPECIATION)**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Name: Daniel Melean

Student ID: 101028978

Date: 11/24/2017

**1. Conceptual model**

**1.1 The model that was used for reference**

Speciation is a process where individuals from a population become a distinct species. There has been a variety of theories for how this process specifically happens, one of them being Sympatric Speciation [1]. This theory is somewhat surprising because proposes the differentiation to happen within groups of individuals that live in the same place, compared to other theories that explain speciation as consequence of spatial isolation of parts of the population from the rest. Adaptive Sympatric Speciation proposes that the differentiation is caused by the variable adaptation of groups of individuals to a particular resource, and that individuals with similar adaptation tend to mate. El Yacoubi and Gourbière (2006) built a cellular automaton as a model for Adaptive Sympatric Speciation. The model is a 2D square lattice with size of 50x50 cells, where each cell can be empty or occupied by individuals with different ecological and mating alleles (A or a and B or b, leaving four different types of individuals that are represented with the numbers 0, 1, 2 and 3). Cells also have a characteristic habitat, which can be of two different types (H1 and H2) and it is chosen at the beginning of the simulation. The neighbourhood is of the Von Neumann type. The transition function has three steps:

* Survival, where individuals survive or not depending on their ecological alleles, cell habitat and a fixed survival probability (s). For example, individuals with an “a” allele are more likely to die in H1 and more likely to survive in H2, both with probability s. The opposite can be said for individuals with an “A” allele.
* Mating, where the mating alleles and a fixed mating probability (p) determine the presence of each individual in two different mating pools. For example, individuals with a “b” allele are more likely to be in the first mating pool (S1) and inidividuals with a “B” allele are more likely to be in the second mating pool (S2). The occurrence of mating and the alleles of the offspring depend on probabilities that are computed each time the function is executed, and they are based on the total number of individuals for each allele combination and the size of the mating pools
* Dispersal, where the offsprings are randomly placed in the neighbourhood of one of the parents, as long as there is an unoccupied cell.

**1.2 Changes in the model that was built for this assignment**

A model for Sympatric Speciation is built and implemented in this assignment. The implementation on CD++ required the speciation not to be adaptive for the difficulty of getting information for all the cell states at a time, and the differences from the model developed by El Yacoubi and Gourbière (2006) are:

* For the mating process, the generation of the offspring is only based on the parent’s alleles and the mating probability (p). In the adaptive model, the probabilities of generating an offspring depended on the total number of individuals of each combination of alleles and mating pools. For the model that was built for this assignment, the probabilities were computed considering all mating scenarios:
* P(No offspring) = 1 – p
* P(Offspring is the same as the parent) = (1/4)\*(1 + 1\*(1/2) + (1/2)\*1 + (1/2)\*(1/2))\*p = (9/16)\*p
* P(Offspring has the same ecological allele but a different mating allele) = (1/4)\*(1\*0 + 1\*(1/2) + (1/2)\*0 + (1/2)\*(1/2))\*p = (3/16)\*p
* P(Offspring has the same mating allele but a different ecological allele) = (1/4)\*(0\*1+ 0\*(1/2) + (1/2)\*1 + (1/2)\*(1/2))\*p = (3/16)\*p
* P(Offspring has both opposite alleles) = (1/4)\*(0\*0+ 0\*(1/2) + (1/2)\*0 + (1/2)\*(1/2))\*p = (1/16)\*p

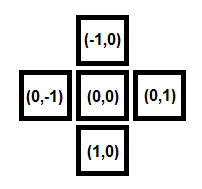
As can be seen, the mating alleles don’t have a practical use any longer, but they were kept in order to maintain the number of types of individuals as 4.

* For the dispersal process, the model that was built for this assignment randomly generates a preferred cell to place the offspring. If the cell is already occupied, the offspring dies. This contrasts with the adaptive model, where the offspring only dies if there are no surrounding cells in the neighbourhood.

The survival process remained intact from the referenced model.

**2. Formal specifications**

Each cell is an atomic Cell-DEVS model, with the neighborhood defined as a Von Neumann type (figure 1).

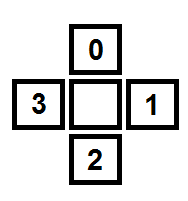


**Figure 1. Neighborhood for the built model.**

The state of a cell is S = xyzw, where x ∈ {1,2} and denotes the cell habitat, y ∈ {0,1,2,3,4} and denotes the type of individual in the cell (or emptiness if y = 4), z ∈ {0, 1, 2, 3, 4} and denotes the type of the generated offspring (or no offspring generated if z = 4) and w ∈ {0, 1, 2, 3} and denotes the chosen neighbour cell for the offspring (0 if (-1,0), 1 if (0,1), 2 if (1,0) and 3 if (0,-1)). When y = 4, z and w are automatically set to zero, as they don’t contain information for empty cells. When z = 4, w is set to zero because it doesn’t contain information for cells that didn’t generate an offspring. See figure 2 for description of the different cell states and figure 3 for an illustration of the meaning of w.



**Figure 2. Possible states for the cells.**



**Figure 3. Offspring dispersal based on the value of w, shown inside the cells.**

The transition rules can be divided in two groups: The first group implements the survival and mating processes for cells that are not empty. First, a random number is generated and according to the survival probability (s), the cell’s state can be changed to empty (1400 or 2400) or remain the same. If the state does not get changed to empty (the individual survived), and offspring is generated according to the probabilities computed in the last section, including the mating probability (p). The chosen cell for the offspring is randomly generated between the neighbors. The rule for ab individuals in H1 with s = 0.8 and p = 0.85 is shown below. Note that the habitat type (first digit) is unchanged in any case. Analogous rules were implemented for other types of individuals in both habitats.

rule : { if( random < 0.8, 1400, if( random >= 0.85, 1040, if( randInt(15) <= 8, 1000 + randInt(3), if( randInt(6) <= 2, 1010 + randInt(3), if( randInt(3) <= 2, 1020 + randInt(3), 1030 + randInt(3) ) ) ) ) ) } 100 { (0,0) = 1040 or (0,0) = 1000 or (0,0) = 1001 or (0,0) = 1002 or (0,0) = 1003 or (0,0) = 1010 or (0,0) = 1011 or (0,0) = 1012 or (0,0) = 1013 or (0,0) = 1020 or (0,0) = 1021 or (0,0) = 1022 or (0,0) = 1023 or (0,0) = 1030 or (0,0) = 1031 or (0,0) = 1032 or (0,0) = 1033 }

The second group of rules implements the dispersal process. If a cell is empty and has a neighbor that generated an offspring for it, it changes the state according to the offspring type (third digit of the state of the neighbor cell). The rule for empty cells that get an offspring from the cell above ((-1,0), with the fourth digit of neighbor’s state being 2) is shown below. The other 3 cases (offspring from left, right and down) are implemented with similar rules.

rule : { if( (-1,0) = 1002 or (-1,0) = 1102 or (-1,0) = 1202 or (-1,0) = 1302 or (-1,0) = 2002 or (-1,0) = 2102 or (-1,0) = 2202 or (-1,0) = 2302, (0,0) - 400, if( (-1,0) = 1012 or (-1,0) = 1112 or (-1,0) = 1212 or (-1,0) = 1312 or (-1,0) = 2012 or (-1,0) = 2112 or (-1,0) = 2212 or (-1,0) = 2312, (0,0) - 300, if( (-1,0) = 1022 or (-1,0) = 1122 or (-1,0) = 1222 or (-1,0) = 1322 or (-1,0) = 2022 or (-1,0) = 2122 or (-1,0) = 2222 or (-1,0) = 2322, (0,0) - 200, (0,0) - 100 ) ) ) } 100 { ( (0,0) = 1400 or (0,0) = 2400 ) and ( (-1,0) = 1002 or (-1,0) = 1102 or (-1,0) = 1202 or (-1,0) = 1302 or (-1,0) = 2002 or (-1,0) = 2102 or (-1,0) = 2202 or (-1,0) = 2302 or (-1,0) = 1012 or (-1,0) = 1112 or (-1,0) = 1212 or (-1,0) = 1312 or (-1,0) = 2012 or (-1,0) = 2112 or (-1,0) = 2212 or (-1,0) = 2312 or (-1,0) = 1022 or (-1,0) = 1122 or (-1,0) = 1222 or (-1,0) = 1332 or (-1,0) = 2022 or (-1,0) = 2122 or (-1,0) = 2222 or (-1,0) = 2332 or (-1,0) = 1032 or (-1,0) = 1132 or (-1,0) = 1232 or (-1,0) = 1332 or (-1,0) = 2032 or (-1,0) = 2132 or (-1,0) = 2232 or (-1,0) = 2332 ) }

The type of delay for the cells is transport, and its magnitude is always 100 ms.

The coupled Cell-DEVS model specifications < Xlist, Ylist, I, X, Y, n, {t1,...,tn}, N, C, B, Z, select > are:

Xlist = ∅;

Ylist = ∅**;**

I: The couple model definition is like in Cell-DEVS;

X =∅;

Y =∅;

n = 2;

{t1,...,tn} = {20, 20};

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

C = {Cij / i ∈ [1,20] ^ j ∈ [1,20]}, where Cij = < Iij, Xij, Yij, Sij, Nij, dij, dintij, dextij, tij, lij, Dij> is an atomic component;

B = {Cij / (i = 1 ^ i = 20) ∧ (j = 1 v i = 20)} nowrapped;

Z : The couple model definition is like in Cell-DEVS;

select = {(-1,0), (0,1), (1,0) , (0,-1)}};

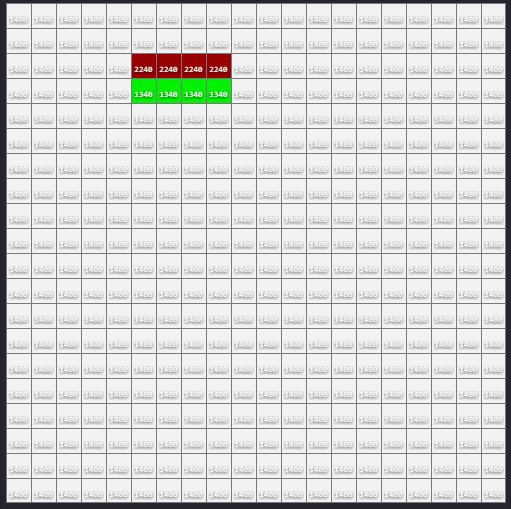
**3. Simulation results**

Five simulations were run for testing the model, each of them with a duration of 5 seconds.

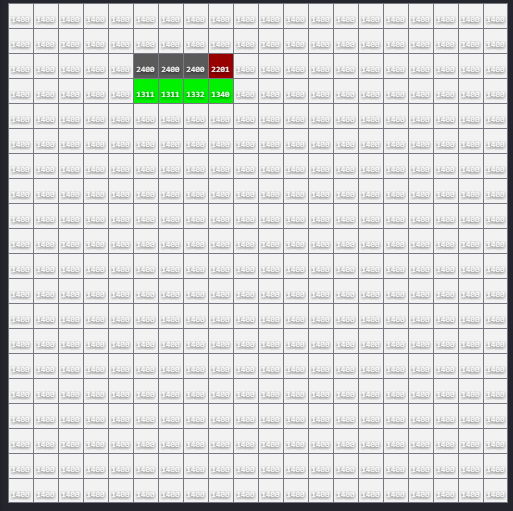
**2.1. Test for the execution of the survival, mating and dispersal processes**

Files: (1) Testing Processes.bat, (1) Testing Processes.me, TestingProcesses.ma, TestingProcessesLOG.log, Initial\_Values\_for\_testing.val, SS.pal, (1) TestingProcesses.webm

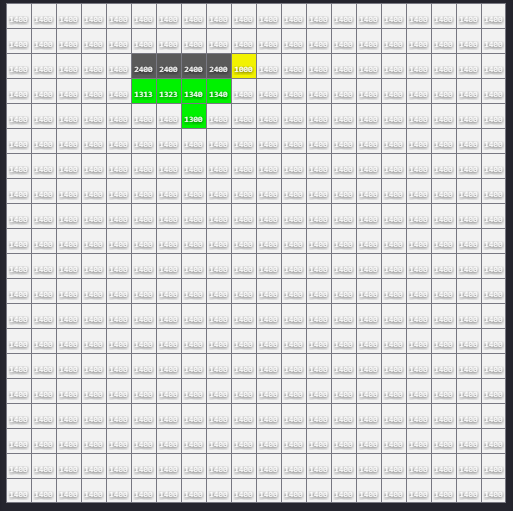
A simple simulation was run for validating the model. The initial conditions were as seen in figure 4, and s and p were set to 0.8 and 0.85 respectively. There were four Ab individuals (red colour) next to each other and four AB individuals (green colour) similarly placed in the row below. The Ab individuals are placed in H1 and the AB individuals are placed on H2, hence the light and dark colours. It is expected that most of Ab individuals fail to survive after 100 ms as they are not in their preferred habitat, and the opposite is expected for the AB individuals. This is confirmed in the next step, seen in figure 5. Note that, despite becoming empty cells, the colours remain dark as the habitat continues to be H2. This is required because the habitats cannot change throughout the simulation. The initial third digits as 4 indicated none of the individuals was going to produce an offspring, but after delay we can see that some of the third and fourth digits have changed, indicating the individuals will reproduce and place their offsprings after the delay. It is expected that the dark red cell produces a light yellow cell to its right (the third digit is 0 and the fourth one is 1). The only green cell to be able to produce an offspring is the third one from right to left, since the other ones chose to place it in occupied cells. The offspring is supposed to be green and be placed in the cell below (the third digit is 3 and the fourth digit is 2). This is exactly what happens (figure 6). Note that the habitat is unchanged again. The simulation keeps running for five seconds with normal behavior. A video of this simulation is available in the file (1) TestingProcesses.webm.



**Figure 4. Initial conditions for the execution test.**



**Figure 5. Step 2 for the execution test.**

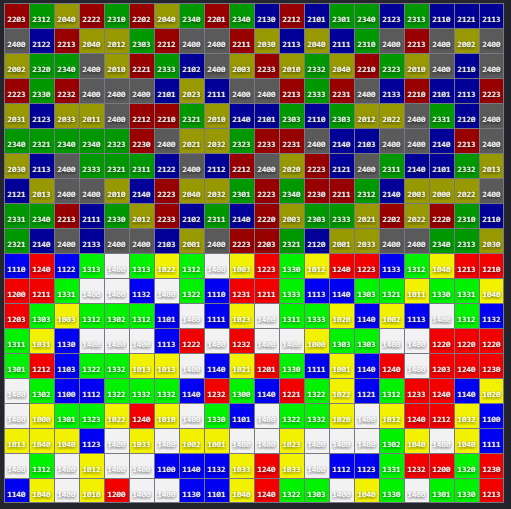


**Figure 6. Step 3 for the execution test.**

**2.2. Simulation with structured landscape, s = 0.8 and p = 0.85**

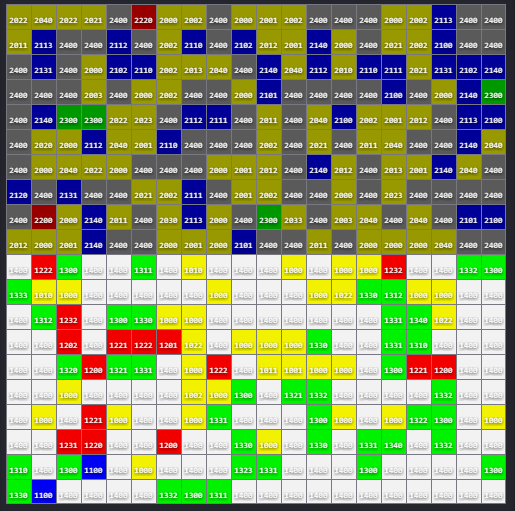
Files: (2) Structured Landscape with s = 0.8 and p = 0.85.bat, (2) Structured Landscape with s = 0.8 and p = 0.85.me, Structured1.ma, Structured1LOG.log, Structured\_Landscape.val, SS.pal, (2) Structured1.webm

In this simulation, the initial states of the cells were randomly generated, but the habitats were set as H2 for the upper half (dark colours) of the cell space and H1 for the bottom half (light colours). This can be seen in figure 7. A video of this simulation can be seen in the file (2) Structured1.webm.



**Figure 7. Structured landscape with initial states.**

The simulation was run, producing normal behavior. The states after the simulation ends are seen in figure 8. Note that there is a concentration of yellow and blue cells in the upper half and red and green cells in the bottom half. This is expected because those pairs of individuals have the same ecological allele and consequently have the same preferred habitat. The probability of survival is not very large, so there is good number of empty cells in both halves of the cell space. There is also a significant number of yellow cells (ab individuals), which is probably related to the seed of the random generator used by the randInt() function of CD++. The relatively high mating probability (0.85) favours this situation, since the most likely offsprings for ab individuals are those of the same type.

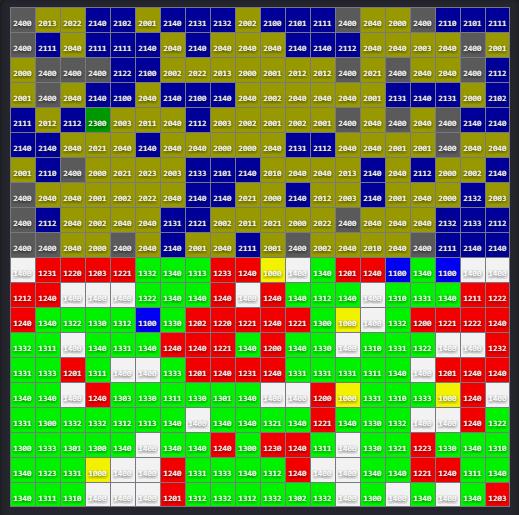


**Figure 8. Final states for the second simulation.**

**2.3. Simulation with structured landscape, s = 0.95 and p = 0.6**

Files: (3) Structured Landscape with s = 0.95 and p = 0.6.bat, (3) Structured Landscape with s = 0.95 and p = 0.6.me, Structured2.ma, Structured2LOG.log, Structured\_Landscape.val, SS.pal

For this simulation, the survival probability is increased and the mating probability is decreased. The initial conditions, however, remain the same. As expected, the pairs of individuals that have the same ecological allele tend to concentrate in one of the halves of the cells space (figure 9), even more clearly than in the previous simulation for the increase in survival probability. The decrease in mating probability causes the generation of offsprings to be slowed down, but this is not very significant since the survival rate is high and the majority of the cells tend to be occupied anyways.

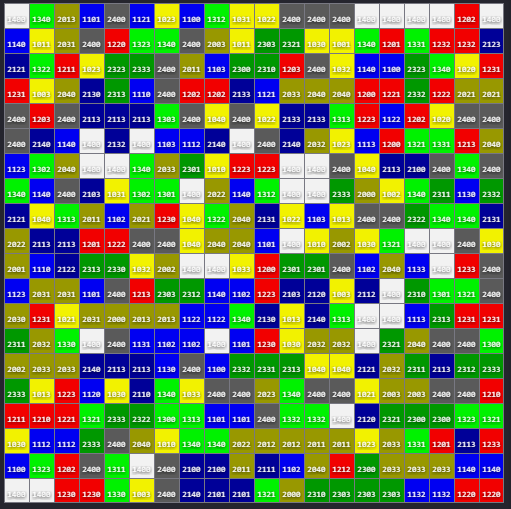


**Figure 9. Final states for the third simulation.**

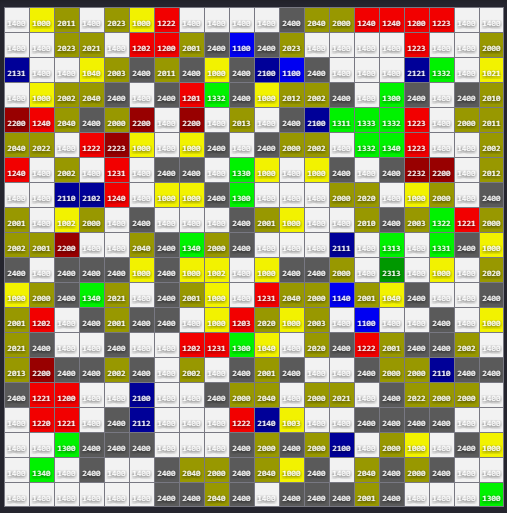
**2.4. Simulation with non-structured landscape, s = 0.8 and p = 0.85**

Files: (4) Non-structured Landscape with s = 0.8 and p = 0.85.bat, (4) Non-structured Landscape with s = 0.8 and p = 0.85.me, Non-structured1.ma, Non-structured1LOG.log, Non-structured\_Landscape.val, SS.pal

The initial values are changed for this simulation. Not only are the genotypes of the individuals still randomly generated, but also the habitats of the cells (figure 10). This makes the individuals of the different genetic types to be more equally distributed throughout the cell space. As the survival probability is the same as in the second simulation, the number of empty cells is considerable in the end of the simulation despite the relatively high mating probability (figure 11). Note that, even though the landscape is non-structured, the individuals with the same genotypes tend to gather around each other because the mating process favors the generation of individuals of the same genotypes as their parents.



**Figure 10. Non-structured landscape with initial sates.**

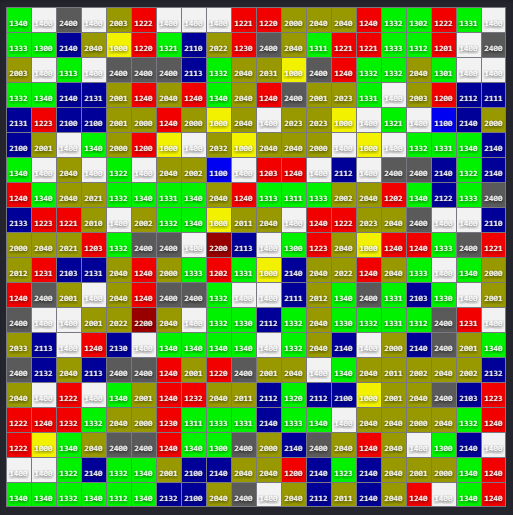


**Figure 11. Final states for the fourth simulation.**

**2.4. Simulation with non-structured landscape, s = 0.95 and p = 0.6**

Files: (5) Non-structured Landscape with s = 0.95 and p = 0.6.bat, (5) Non-structured Landscape with s = 0.95 and p = 0.6.me, Non-structured2.ma, Non-structured2LOG.log, Non-structured\_Landscape.val, SS.pal, (5) Non-structured2.webm

As expected, in this simulation the number of empty cells decreases because of the high survival probability (figure 12). Here, the small “colonies” of individuals of the same genotypes are easier to spot because individuals tend to get more changes to reproduce and generate offsprings that are most likely to be similar to themselves, even though the mating probability is a little lower than before. A video of this simulation is provided in the file (5) Non-structured2.webm.



**Figure 12. Final states for the fifth simulation.**

**References**

[1] El Yacoubi, S. & Gourbière, S. (2006). *A Cellular Automata Model for Adaptive Sympatric Speciation*. University of Perpignan. Retrieved from http://dunlop.u-cergy.fr/mosv/yacoubi\_gourbiere\_06.pdf