Anant Marur
19237018
January 23, 2018
CMPLXSYS 530

Langton's Ant Implementation

Git Repository: https://github.com/amarur/langton-ants

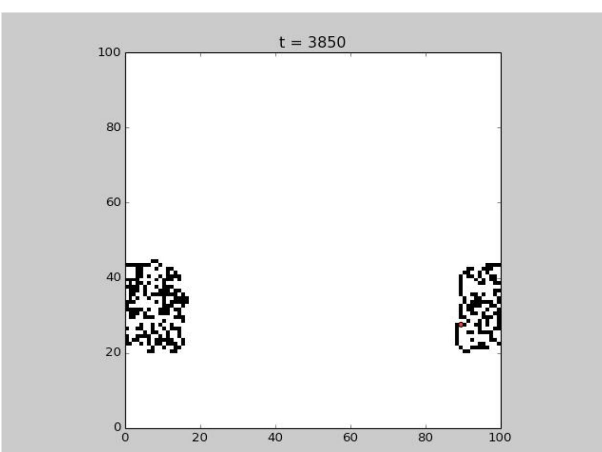
The usage details of the implementation can all be found in README.md in the git repository.

I implemented two modifications to Langton's Ants model. The first of my changes was simple, rather than clipping along the borders of the environment, I made the environment wrap around. This was implemented in the moveForward() member function of the Ant class, rather than building a custom environment, which I found to be the easiest way to implement wraparound.

The second of my changes was creating an 'inverse' breed of Langton's Ant, which turns the opposite direction on the respective colors. I allowed the proportion of 'true' Langton's Ants to be toggled with a variable at the top. I wanted to make all of the toggle variables (described in my README) accessible via PyCX parameters, but I couldn't figure out how to make it work, so the toggling can only be done by editing langAnt.py.

Single Ant Simulations:

Running the model as a traditional Langton's Ant simulation, it holds true to predictions. We see the Ant slowly grow its path in a seemingly random manner (Fig. 1, 2) until it hits 10,000, when it begins to cycle and translate itself diagonally (Fig 3). The inverse Ant behaves similarly, as is evident in the figures. Interestingly, the development of the inverse Ant's cycles is actually the inverse of the true Ant's cycles, a result I loosely suspected but was unsure about.



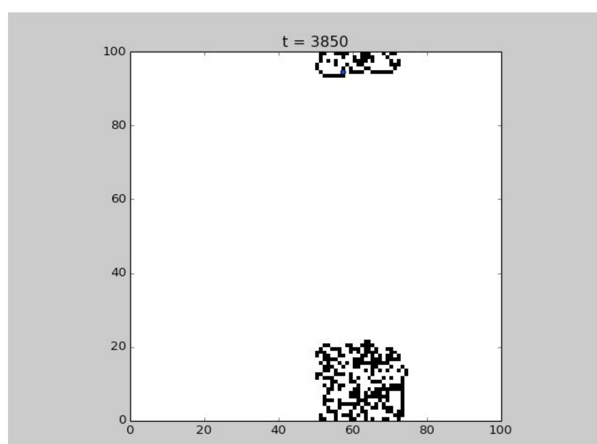


Figure 1: True ant (left) and inverse ant (right) after ~3,800 cycles

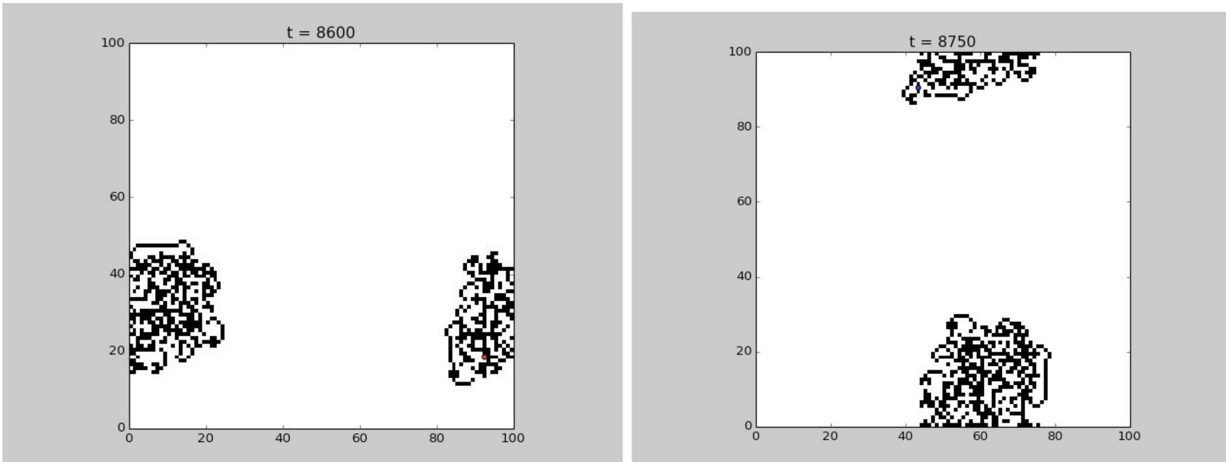


Figure 2: True ant (left) and inverse ant (right) after ~8,700 cycles

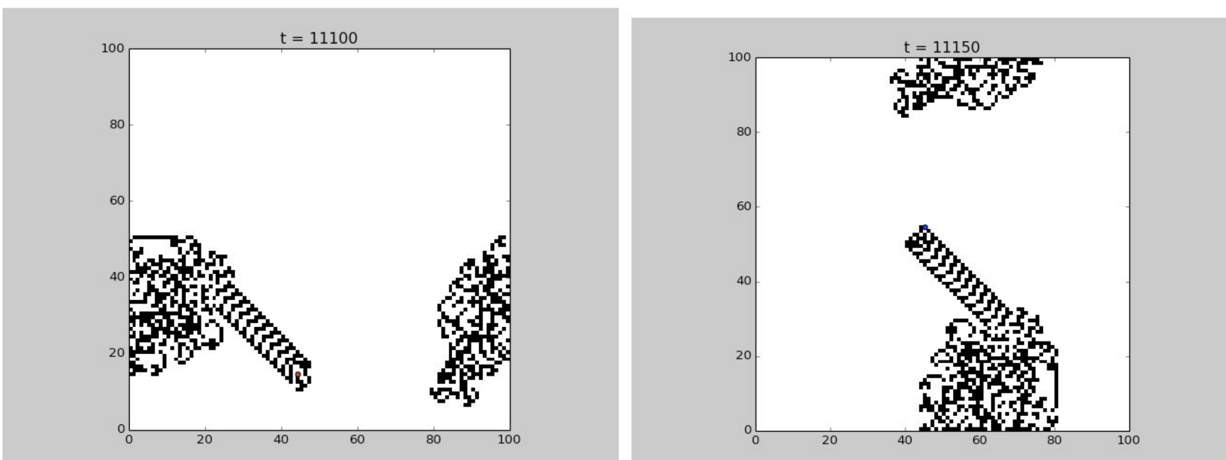


Figure 3: True ant (left) and inverse ant (right) after 11,100 cycles

<u>The Face Off: Dual Ant Simulations:</u>

Next, I ran the simulation with one of each type of Ant, to see how they would interact. To force interaction, I cut the environment down to 50x50 as opposed to the 100x100 used in the previous section.

This run yielded some interesting results (Fig. 4-7). It actually took longer for the Ants paths to begin intersecting each other than I anticipated, especially since I shrunk down the size of the environment so much. The inverse ant ended up traversing into the true ant's path and getting muddled in there, never reaching cyclical equilibrium and thus never reaching the point of translating itself. The true ant, on the other hand, got lucky and managed to avoid sections of its path that had been corrupted by the inverse ant, achieving cyclical equilibrium. This confirmed my hypothesis that these Turmites are incapable of recovering back to equilibrium once they encounter paths that contradict the pattern of their own paths.

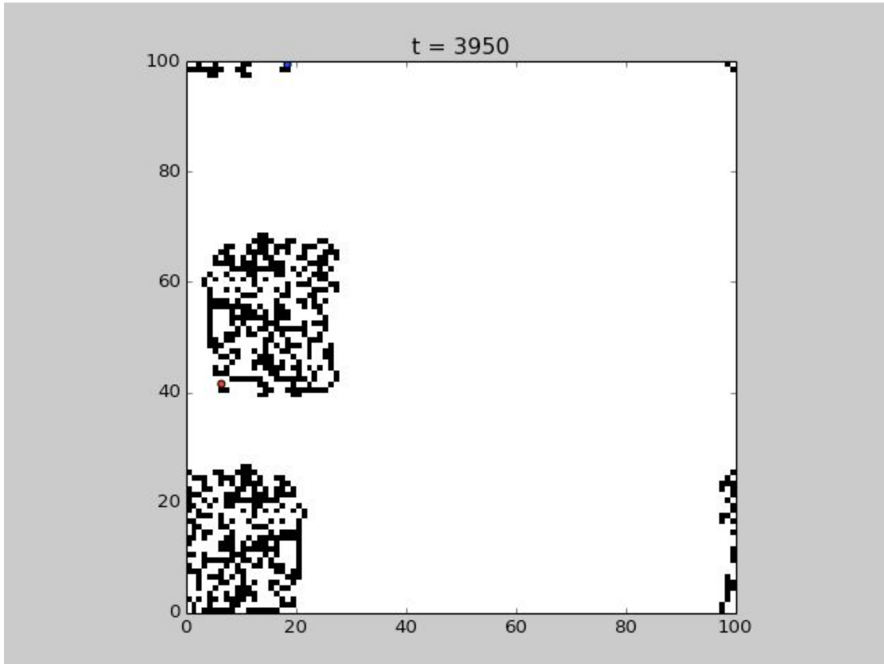


Figure 4: After ~4k steps, the ants had still not intersected with each other. True's location is ~(5,40), and Inverse's location is ~(10,20)

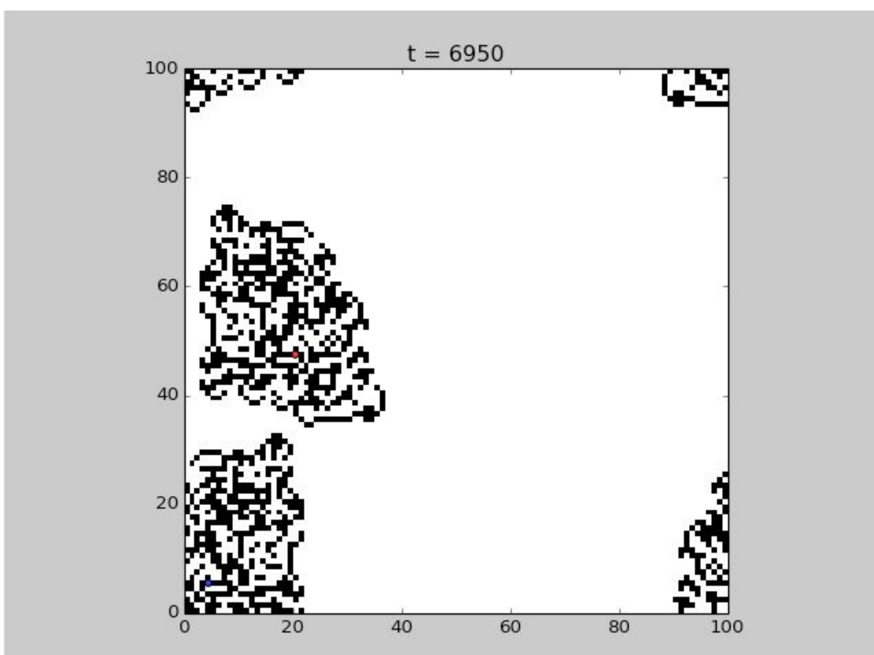


Figure 5: After ~7k steps, the ants still have not intersected paths at all, although they are coming extremely close. True's location is ~(20,45), and Inverse's location is ~(20,30)

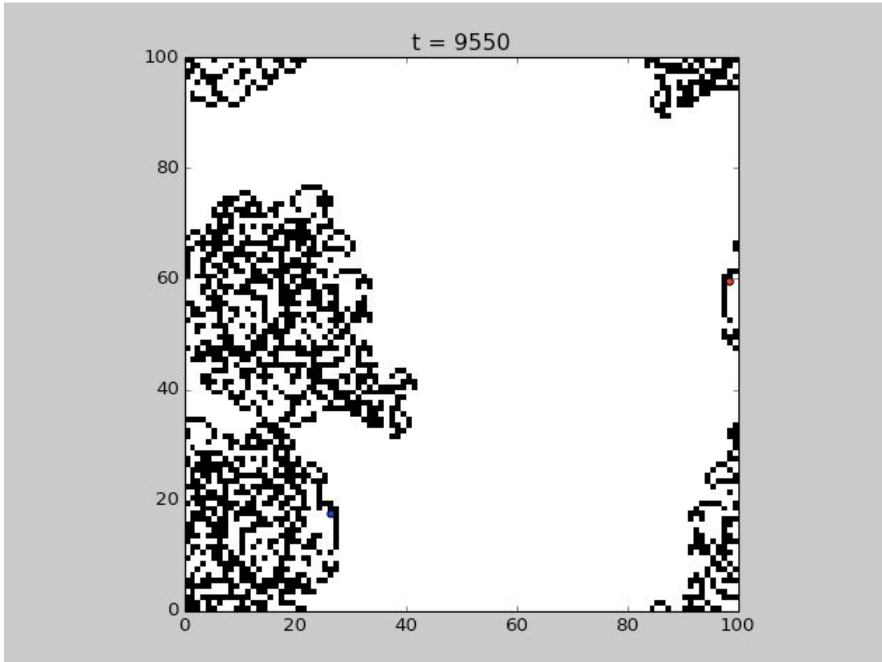


Figure 6: After ~10k steps, the ants have finally intersected paths. Specifically, Inverse has traversed along the very edges of True's path, although True has still only traversed around its own path. True's location is ~(100,60) (due to wraparound), and Inverse's location is ~(25,20)

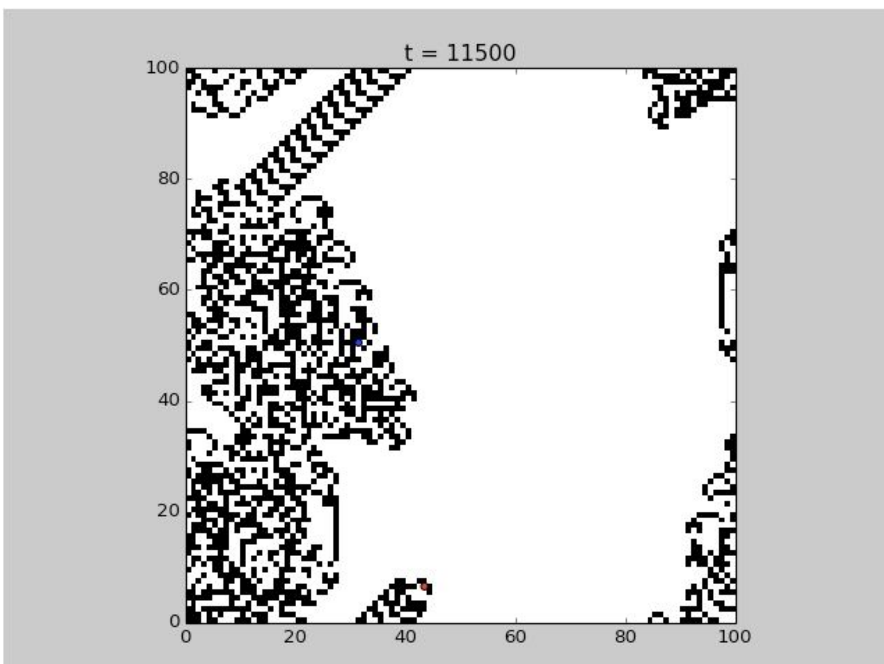


Figure 7: After ~12k steps, the results of the run are clear. True has been cycling and translating for a few hundred steps, as is clear from the figure. Inverse, on the other hand, has been stuck in the heart of True's old path, muddling around confused. True's location is ~(45,10), and Inverse's location is ~(30,50)