CX4230 3E Summary

Kartikeya Kumar and Karan Shah for Dr.Vuduc, Spring 2016

We had implemented population growth/death in the last milestone. In 3E, we implemented the actual migration of people. In a lot of migration models based on cellular automata, only neighboring cells are taken into account. We think this is not true for our world because it has become easier to move across countries since the time those papers were published. As stated in the conceptual model, we planned on comparing each cell to every other cell in the grid to decide on the desirability of each cell. Our first implementation was a brute force $O(n^2)$ algorithm that took a very long time to run.

To get around it, we wrote a variant of Barnes-Hut algorithm (http://arborjs.org/docs/barnes-hut is a good introduction) which is used to solve N-body problems. Our algorithm recursively divides the area into 4 parts and calculates the average values(analogous to centers of mass) of properties such as wealth and population (analogous to mass) till the smallest area is one unit. At each step of the recursion, it only recurses to the division with the highest averages (like going down in a quad-tree). This is more efficient as its complexity is $O(n \log n)$. Now, unlike a simulation of massive bodies, there are no areas in our system without a "mass". So, if we made a quad-tree, it would have the same number of leaves as there are cells in the grid. This function is called find_local_dest(). The analogy for force in our case is the desirability of one point with respect to some other point. There is some loss in accuracy because we are eliminating areas with low averages at every step. It might be possible that the most desirable cell is eliminated because it's surrounding cells are less desirable. In our testing, we have seen that it returns the best cell most of the times and returns cells that are pretty close to the highest value sometimes.

We also started running some tests by varying parameters to see if any trends could be observed for our migration framework. Right now, we ran our model without migration turned on to see how population varies purely on the basis of birth rate/death rate. Then, we ran our model with both migration as well as natural birth rate/death rate on to see how the results varied. Both the results have been provided in the PrelimResults.pdf file provided in the repo.

Finally, we also implemented a function called desirability() which calculates the 'desirability' of a destination location with respect to a source location, using the formula we had in our conceptual model. We also had to reimplement our desirable_location_country() function which returned a list of the most desirable locations in every country on the basis of wealth as we discovered some bugs in it.

Functions implemented in this milestone:

- desirability() : Return desirability score of one point wrt another point
- find_local_dest(): Recursive method to find the most desirable cell within the same country
- makeLocalDestMatrix(): Makes a matrix with the location and desirability of the most desirable destination cell.
- migration(): Initiates all migration functions
- engine(): Meant to act as an entry point to our program
- globalDesirability(): Returns the most desirable location in each country
- findIntMatrix(): Returns a matrix of 'best' countries for each cell to move to
- return_international_matrix(): returns a matrix consisting of the most desirable foreign country with respect to a cell as well as the desirability of moving to that country

FINAL TODOS BEFORE THE PRESENTATION:

1. CLEAN CODE! It is an illegible mess right now and we will write some documentation to make it easy for the reader to understand.
2. Implement different proportions of migration based on region. Right now, migration happens from any cell to any cell in the world in the same proportion. We will change this so that the proportion of people moving in their neighboring cells is higher than people moving across the country which will in turn be higher than people moving internationally.
3. Keep a better count of how people move. A case study could be how people from 1 particular region move and mapping their tracks.
4. Set parameters that represent real world countries (eg higher population rate,

lower average wealth in developing countries).

5. Try out different functions to calculate desirability.