Agent-based modelling of complex systems

Assignment 4

Janusz Szwabiński

Consider a version of the Schelling's segregation model implemented in the following way:

1. N agents ($\{n_1, n_2, \ldots, n_N\}$) are generated ($N^b$ blue, $N^r = N - N^b$ red).
2. Each individual is initially placed (uniformly) at random on a 1000×1000 square lattice.
3. Start with individual n_1 (who has type $t \in (b, r)$) the proceed with the algorithm.
   (a) Take individual n_i who has type t. It at least j_t of its m_t closest neighbors are of the same type, move on to next individual.
   (b) If fewer neighbors than j_t are of the type, move the individual to a randomly selected new empty location. Then move on to next individual.
   (c) Continue this process until no individuals remain who wish to move.

Implement the above model. Your program should:

- Take the sizes of each population as well as j_t and m_t for each type as command line options.
- Plot both the initial and final distributions of agents, as well as some intermediate configurations of the lattice.
- Return data for the final allocations of agents.
- Return a segregation index of the configuration. You may use the *similar neighbor index* as the measure of segregation: for each individual calculate the proportion of the nearest neighbors that are of his type and take the average over this number across individuals.
- Return the number of cycles the algorithm takes.

To do:

1. Run a baseline model with two populations of size 250 each. For each group assume $m_t = 8$ and $j_t = 0.5$ (work with periodic boundary conditions to treat the agents on the edges on the same footing).
2. Plot the initial and final distributions (for single runs). Run your program several times to see how stable the results are.
3. Plot the number of iterations as a function of population size (changing from 250 to 4000).
4. Plot the execution time as a function of group size.
5. Plot the segregation index as a function of j_t (changing from 0.1 to 0.9).

6. Plot the segregation index as a function of m_t (changing from 8 to 128).
7. Comming back to the baseline simulation, see what happens if $j_r = 6/8$ and $j_b = 3/8$?