Agent Based Modelling

0.1 Model

The algorithm used in this model is the following.

Algorithm 1 Agent Based Model Algorithm

1. Let i_t be the i-th individual agent at time t, with i = 1,...,1000 and t = 1,...,100. Randomly locate each agent at coordinates (x_i, y_i) with $x_i \sim U(0,1)$ and $y_i \sim U(0,1)$

2. Randomly locate 1000 objects k_{1-1000} at coordinates (x_i, y_i) with $x_i \sim U(10,-10)$ and $y_i \sim U(10,-10)$

3. At time (t + 1) each agent i moves a random distance in both x and y directions such that $x_{i,new} = x_{it} + d_{ix}$ and $y_{i,new} = y_{it} + d_{iy}$, where $d_{ix} \sim U(-1,1)$ and $d_{iy} \sim U(-1,1)$.

4. Each agent i identifies the f closest neighbours in each direction and halves the average distance to these neighbours such that the new coordinates at time (t + 1) are $x_{i,t+1} = 0.5(x_{i,new} + x_{ft})$ and $y_{i,t+1} = 0.5(y_{i,new} + y_{ft})$. x_{ft}) and y_{ft}) are the average distances in the x and y direction respectively of agent i with its f closest neighbours. Euclidean distance is used here to find the nearest neighbours.

5. The Euclidean distance $g(i,k)$ is calculated between each agent i and object k. Let $i_t + d = (x_{i,new}, y_{i,new}$ be the coordinates of the agents after the initial movement and before regrouping. If
$$g(i_t + d, k) < a$$
then
$$i_{w,t+1} = i_{w,t} + 1$$
where $i_{w,t}$ is the wealth count of agent i at time t.

In this model we have used f = 5 and a = 0.01. The figures below show the positions of the agents and objects at times t = 0 and t = 100 and also the accumulated wealth of the agents over time.

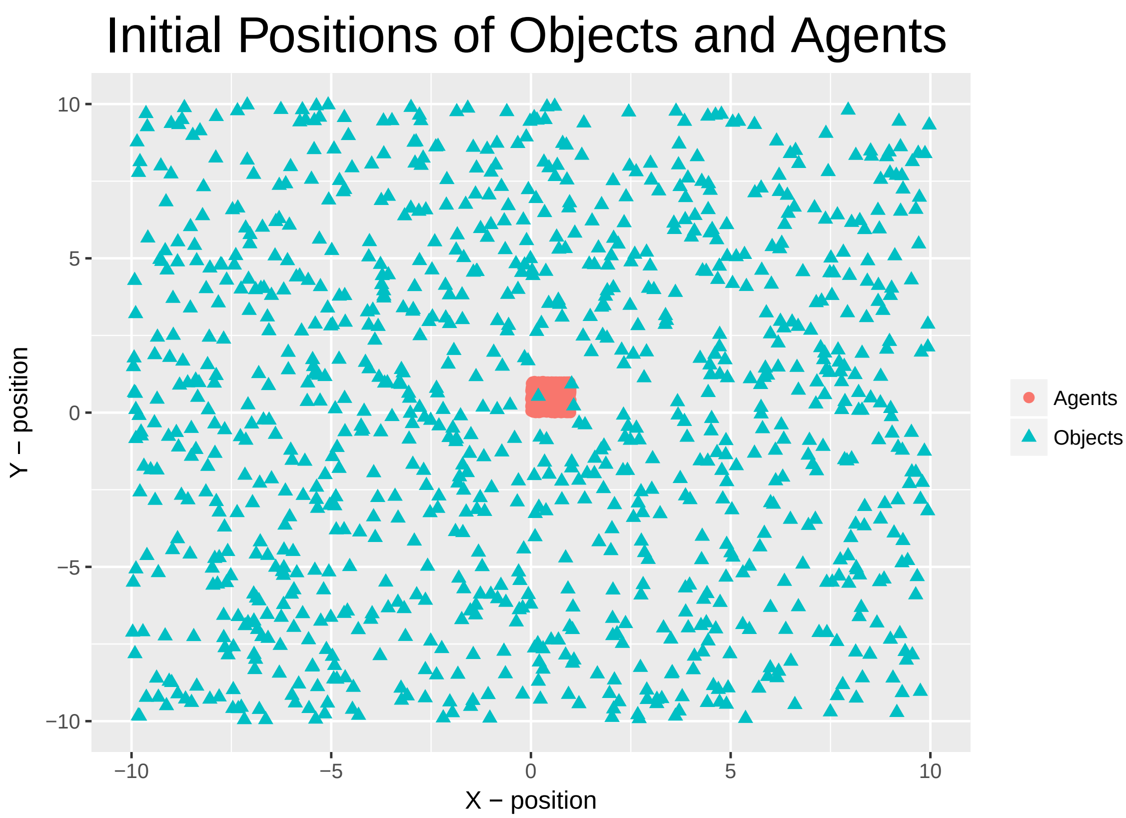

Figure 1:

0.2 Proposition

In the initial description of the model, there seemed to be another inconsistency. The description suggests that the average distance of the agents with the closest 5 agents should be halved. However in the mathematical notation, as seen in algorithm 1, step 4, the coordinated $x_{i,new}$ also gets multiplied by 0.5. This means that the new coordinate after the random movement between -1 and 1 also gets halved. Wouldn't it be better to only half the x_{ft} such that $x_{i,t+1} = x_{i,new} + 0.5(x_{ft})$ and $y_{i,t+1} = y_{i,new} + 0.5(y_{ft})$. This seems to be the correct way of halving the distance between each agent and the 5 nearest neighbours.

Using this model and putting a constraint on the maximum distance the agents can explore (in this case, a constraint on -10 to 10 on both axes seems appropriate), the agents seem to spread out more and explore the space better, unlike what we can see in figures 3 and 4. In figures 6 and 7 we can see the final positions of the agents and the accumulated wealth over time using the proposed algorithm. The agents are more spread out around the axis, hence exploring a larger area. This has lead to a higher accumulated wealth compared to the previous algorithm.

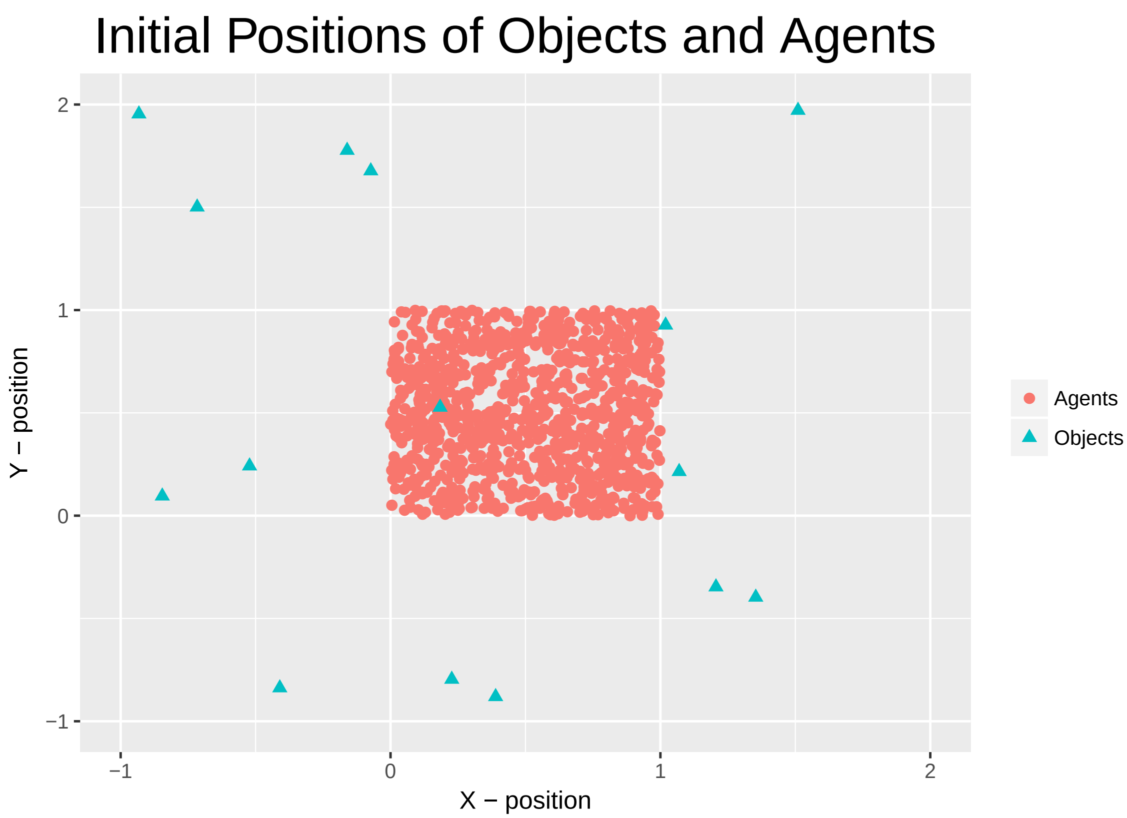

Figure 2: Zoomed in image of the positions of agents and objects at time t = 0

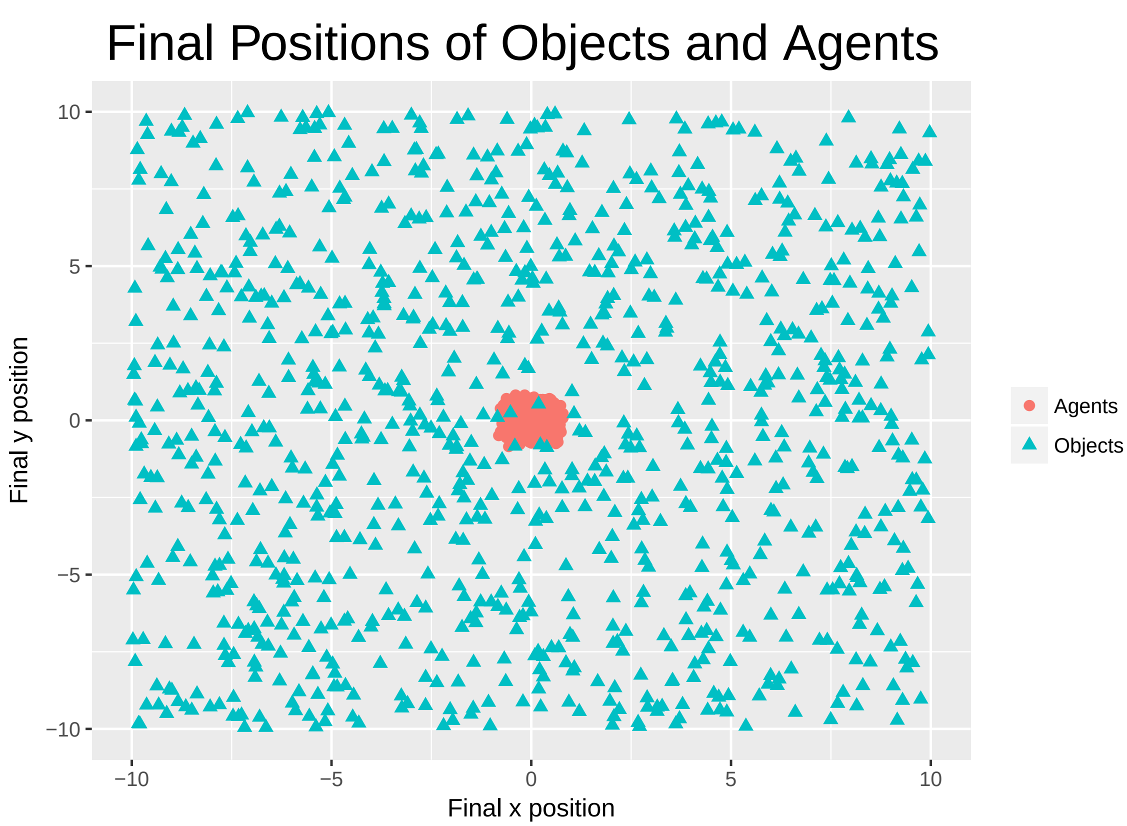

Figure 3:

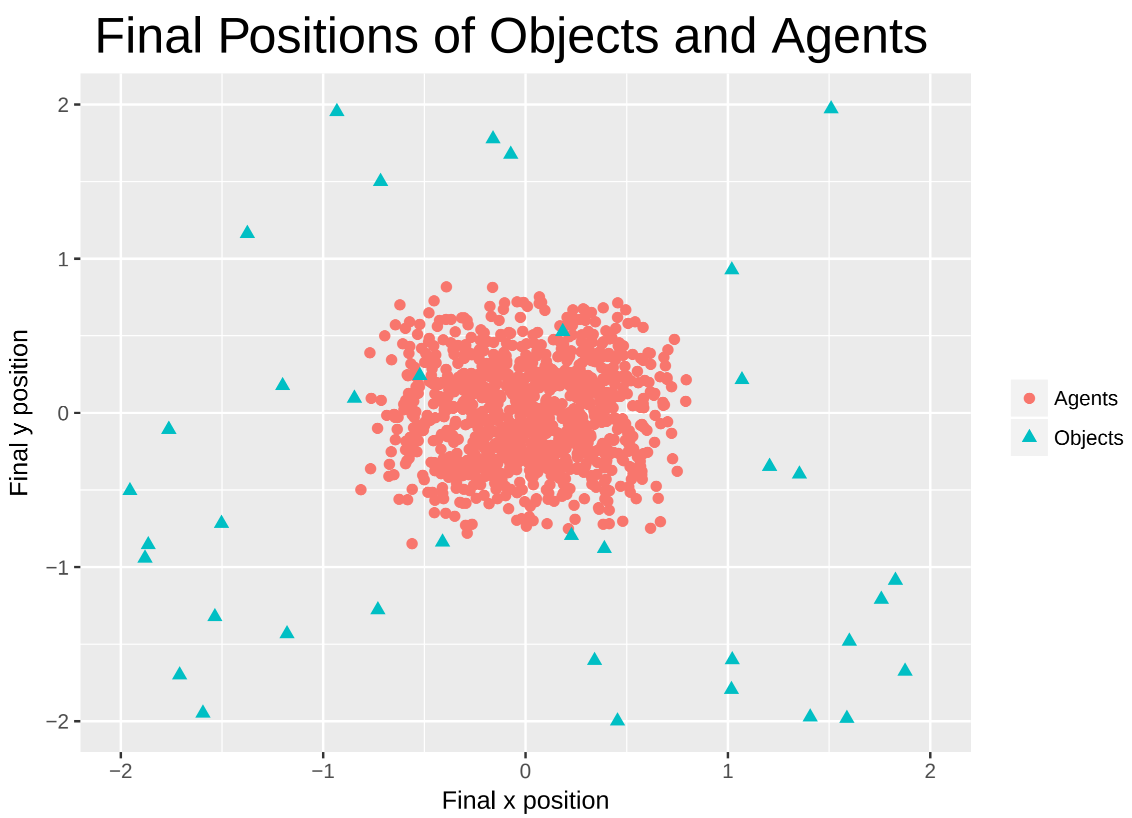

Figure 4: Zoomed in image of the positions of agents and objects at time t = 100

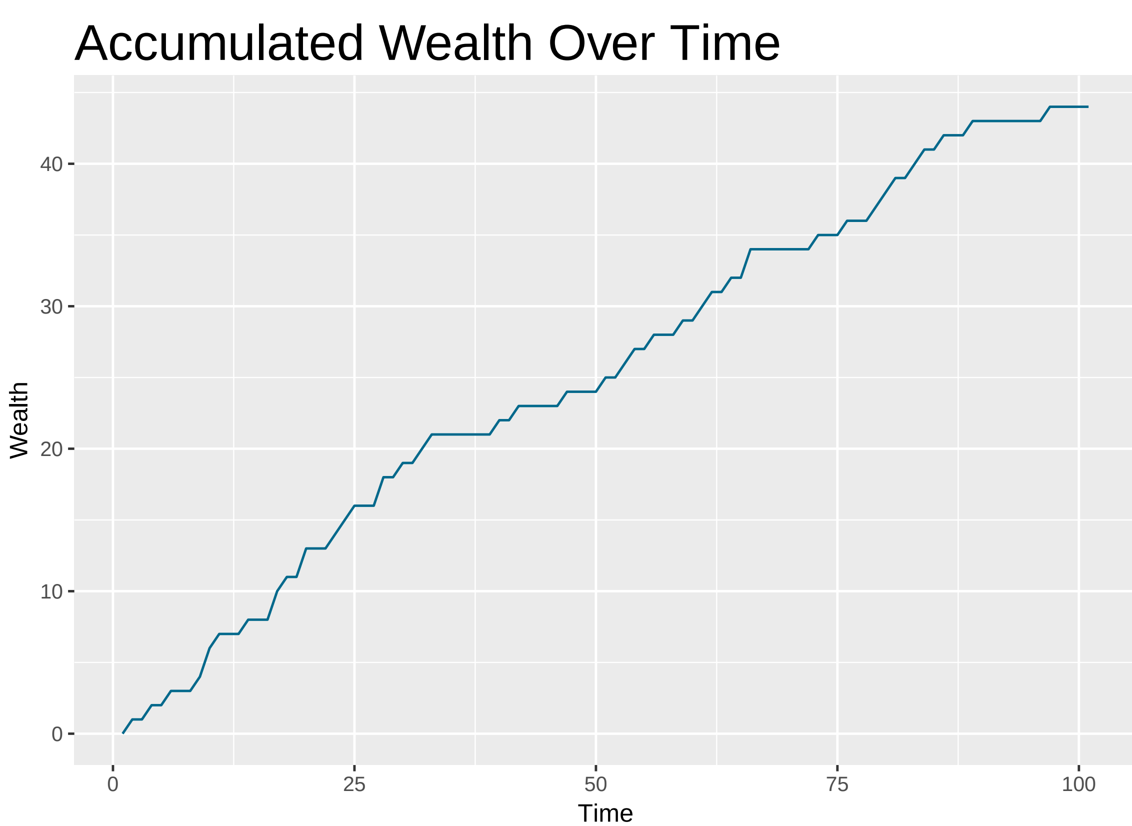

Figure 5:

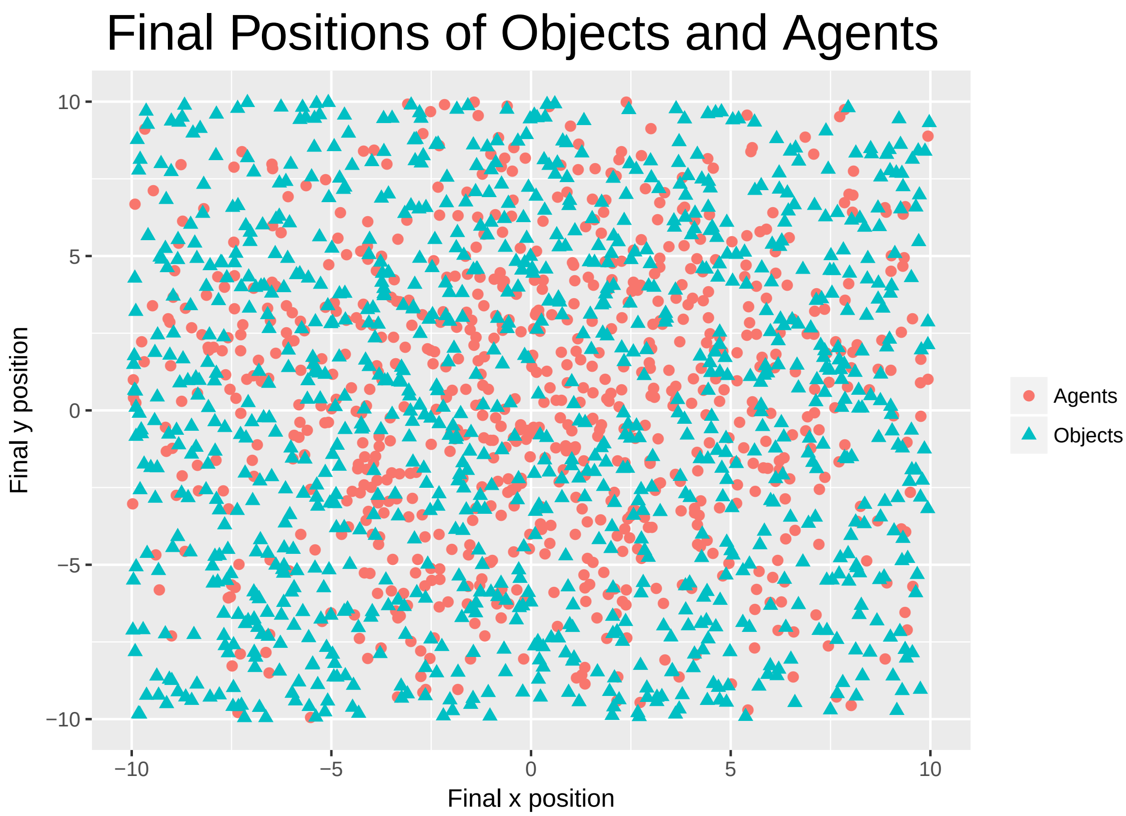

Figure 6:

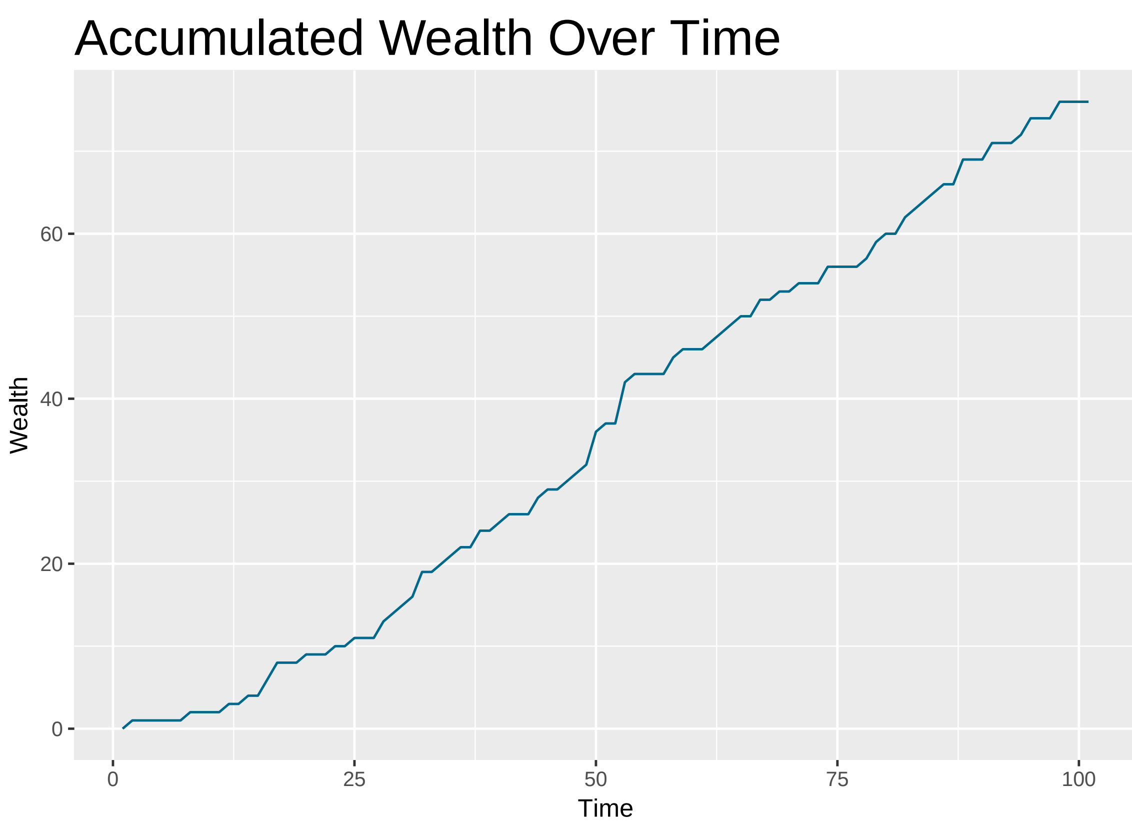

Figure 7: