Conrad Coutinho

Racial Preferences and Macro-Segregation

I. Introduction

The word ‘segregation’ calls to mind images of black-only water fountains, police brutality and other infamous symbols of the pre-civil rights era. Segregation certainly *feels* like it only exists in our history books, but it is far from history. Why are so many major cities still divided along racial lines? Everyone knows about the so-called “white flight” of whites to the suburbs of major urban centers. This racial divide is even evident in seemingly liberal cities such as Austin, Tx. Many explanations have been given for this phenomenon; these explanations have typically run the gamut between two extremes: “people move because they are racist” or “people move because of factors that happen to correlate with race but are not race itself”. I hypothesize that much of this de facto segregation is a result of slight racial preferences among the population. De facto segregation might be caused by a rather slight and, perhaps, even unconscious discomfort with people of a different race among certain populations. As a disclaimer: I am not attempting to justify de facto segregation or any type of racial preferences; rather, I am attempting to offer a reasonable explanation of the facts. I must also emphasis that I do not seek at this time to *prove* this hypothesis. This would require a large amount of empirical information and, like many things in social science, it would be difficult to prove beyond doubt. Once again, I only strive to offer a plausible explanation.

One of the most famous conceptions of the idea that slight preferences can have a large effect on macro-behavior was offered by economist Thomas Schelling. Schelling demonstrated this hypothesis using a simple model: a chessboard with pennies and dimes and a rule about the coins’ “preferences”; that the coins slightly preferred to be next to coins of a similar type. In this demonstration, Schelling moved the coins according to the preference rule and found a general tendency to move toward complete segregation among different types of coins even if the initial preferences were very slight. The upshot of all this is that racial segregation could arise when no single individual or family had extreme racist tendencies, but rather many had slight racial preferences.

The premise of my model is very similar. I took the agent based sugarscape model and created two different “races” of agents. I then programmed them with certain racial preferences of varying degrees in addition to their preference for sugar. I did this by adding a simple utility function that contained sugar and racial preference. This allowed each agent to consider and compare not only the amount of sugar but the racial composition of various areas on the board before moving. The general form of this racial rule was: “the agent slightly prefers to live amongst empty squares and agents of a similar type.”

Note: This model actually required a significant amount of code modification so, unlike my other papers, I will not go through every single change. I will attempt to hit what I believe to be the most important changes. I will then present a general summary of the entire model and how it all works together. Then I will review the general results of my model and the results of two experiments that I ran. I will then consider the successes and the faults of my model and what I could have done to make it better.

.

II. Agent Creation

The original sugarscape model had only one type of agent distributed over the 20% of the sugarscape; this was stored in the matrix "a_str" where a 1 meant an agent and 0 meant no agent. I added a second type by modifying the "initagent" function to add one type of agent 10% of the time and another type 10% of the time. They are still randomly distributed throughout the sugarscape and the *only* difference between agents is the "a_str.active" variable which is set to 0 for an empty space, 1 for type 1 agent and 2 for type 2 agent. They do not differ in any way but their "race". I accomplished this by first filling the "a_str" matrix with 0's or non-agents. I then added the lines:

```
a = rand;
if(a<=.1)
        *create type one agent*
elseif( a<= .2)
        *create type two agent*
end
```

This generates a random number "a" and if that number is below .1 a type one agent is created and if it is between .1 and .2 a type 2 agent is created. Each agent creation is nearly identical to the original model except I added the lines:

```
a_str(i,j).active = X;
```

Where X =1 for the first "if statement" and X =2 for the second "if statement". This is the only real difference between the two types of agents; this is their "race"

And I added the lines:

```
N = rand;
        if(N <.5) a_str(i,j).racepref = ceil(rand * raceprefv);
        else a_str(i,j).racepref = 0;
        end
```

Racepref is the max or strongest racial preference allowed in the model; it is equal to 20. These lines randomly assign 50% of each population uniformly distributed "racial preferences" from 0 to 20. The other 50% have a 0 for racial preferences. The higher the "a_str(i,j).racepref" the stronger the preference. The idea is that most of society has no racial preference, some of society has weak racial preferences and some have strong racial preferences. These racial preferences come into play later in the model in the utility function of the agents.

III. Checking the Race and Calculating Utility

I needed a way to calculate utility at any spot on the board. This would make checking each square very efficient; each time I needed to calculate an agent's utility at either the current square or at a square the agent is considering moving to, I could just call the same function but send it different coordinates. I essentially call this function under two occasions: 1) when calculating an agents utility at the spot he is in and 2) when calculating an agents utility at a spot he is considering moving to. The function would consist of the following:

```
d = 0;

  if(v<size)
     if(a_str(i,j).active == a_str(u,v+1).active||a_str(u,v+1).active == 0)
        d= d+1;
     end
  end

  if(v>1)
     if(a_str(i,j).active == a_str(u,v-1).active||a_str(u,v-1).active == 0 )
        d= d+1;
     end
  end

  if(u<size)
     if(a_str(i,j).active == a_str(u+1,v).active||a_str(u+1,v).active == 0 )
        d= d+1;
     end
  end

  if(u>1)
     if(a_str(i,j).active == a_str(u-1,v).active||a_str(u-1,v).active == 0 )
        d = d+1;
     end
  end
```

This basically lets the agent look at each square next to the spot being checked. The variables "u" and "v" represent the coordinates of the spot being considered and "i" and "j" are the coordinates of the agent itself When the agent is considering his utility at his current spot i = u and j = v.

If the spaces surrounding the spot being considered have an agent of the same type or is empty the counter "d" goes up. The integer "d" is therefore a measure of the racial homogeneity (or empty squares) of any given square on the board. Notice that by this measure, when "d" is included in the agent's utility function, each agent is exactly indifferent between living next to someone of the same race or an empty space. The following is the utility calculation:

```
Utility =  a_str(i,j).racepref * d +  s(u,v) ;
```

The utility is essentially just the sugar in the spot (“s(u,v)”)plus the homogeneity of “neighborhood” (“d”) around the spot weighted by the agent’s racial preference (“a_str.(i,j).racepref”). That is, if an agent has stronger racial preferences it is more beneficial to be in a homogenous neighborhood than if an agent has weaker racial preferences.

IV. Comparing Utilities

The function “neighbor” of the original model would compare the amount of sugar at the original spot to the spot under consideration. In my model, it simply compares the utilities in the same way with following lines:

```
if (a_str(u,v).active == 0)
   Utility = checkrace(i,j,a_str,size, u, v, s);
   if(tempUtility< Utility)
         tempUtility = Utility;
         tempi = u;
         tempj = v;
   end
end
```

The first if statement is ensuring that the spot under consideration is empty. The second line simply checks the utility available at the spot under consideration by calling the “checkrace” function. Recall that the “checkrace” function returns the utility at any spot on the board if the proper coordinates are sent to it. The variable tempUtility contains the the utility of the best spot checked thus far. So, if the utility of this spot is greater than other spots considered so far, the information of the spot is noted in the variables tempUtility, u and v. After checking all the spots, the “moveagent” function moves the agent to the spot with the highest overall utility. This is much clearer if you think about it behaving much like the original program but with utility instead of sugar.

V. Agent Display and Other Changes

These major changes to this program involved many more changes than have already been noted. However, many of these changes were small and for the most part uninteresting. Often, I would have to change variable names throughout the entire program or change function calls to include more variables. One of the most notable changes, however, was how the agents were displayed.

```
for i = 1:size;
   for j = 1:size;
      if (a_str(i,j).active == 1)
               a(i,j) = a_str(i,j).active;
         av(i,j) = a_str(i,j).vision;
         am(i,j) = a_str(i,j).metabolism;
      end
      if (a_str(i,j).active == 2)
               a(i,j) = a_str(i,j).active;
         av(i,j) = a_str(i,j).vision;
         am(i,j) = a_str(i,j).metabolism;
      end
   end
end

figure(2);
subplot(ceil(sqrt(nruns)),ceil(sqrt(nruns)),runs), imagesc(a);
```

Much like the original program, I would simply store the active value in the "a" matrix and display the "a" matrix. However, I had to store different values in different slots of the "a" matrix. In addition, I had to change the image view from "spy" to "imagesc" so that the proper contrast of colors could be taken into account

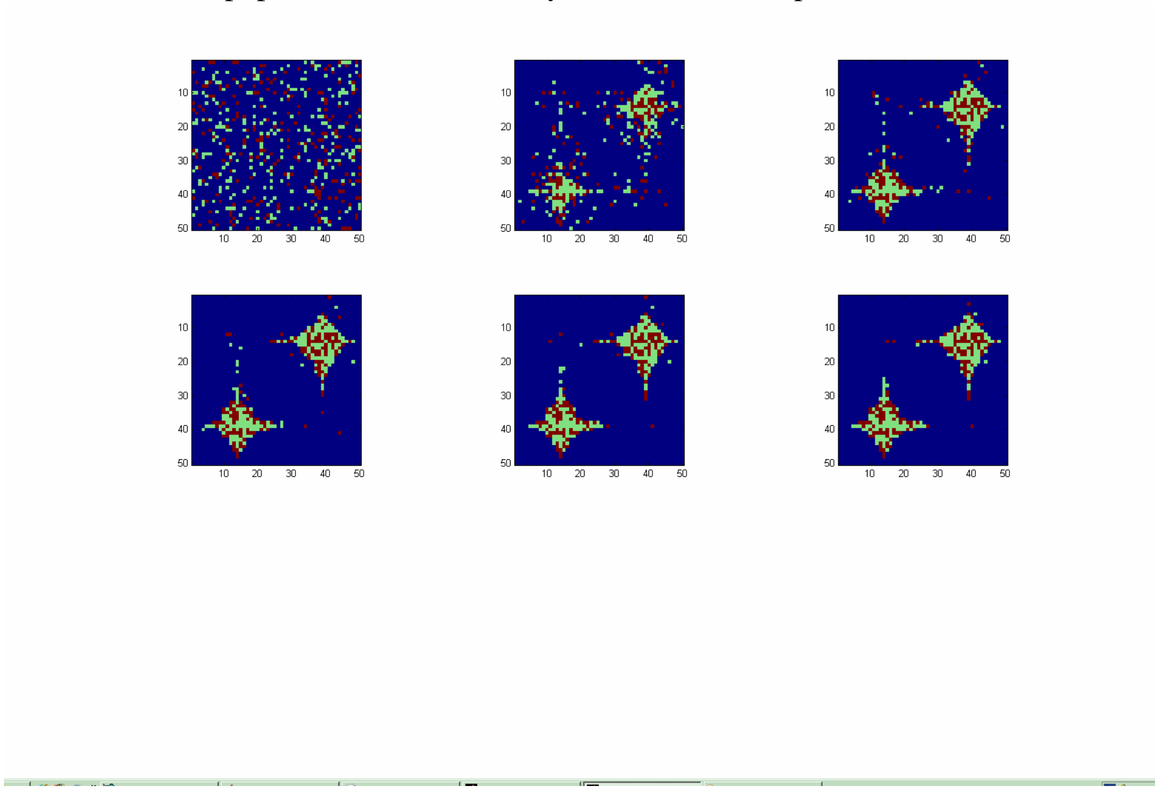
VI. Putting It All Together

In summary: Two different kinds of agents are created with certain varying racial preferences. In the programs main loop, each calls "checkrace" in order to check the utility of the current spot. It then stores this value as well as the coordinates of where it currently is in temporary variables. It then calls function "see" which checks each spot in the agent's vision. For each spot, function "see" calls function "checkrace" to calculate the utility of the spot being considered. Function "see" then calls function "neighbor" which checks if the utility in the temporary variables is lower than the utility at the spot being considered; if this is the case, then the new spot is stored in the temporary variables. The main difference between my model and the original is that the agents are not just comparing sugar, but overall utility. Often this means the agent must trade off between sugar and living next to other agents they may not like. After checking each spot in the agent's vision, the function "moveagent" is called and the agent is moved to the spot with the highest utility. It then consumes the sugar at its spot and if its overall wealth is lower than its metabolism it dies; it should be noted, that the agent lives and dies dependent on its sugar still and not its overall utility. The entire matrix is then displayed.

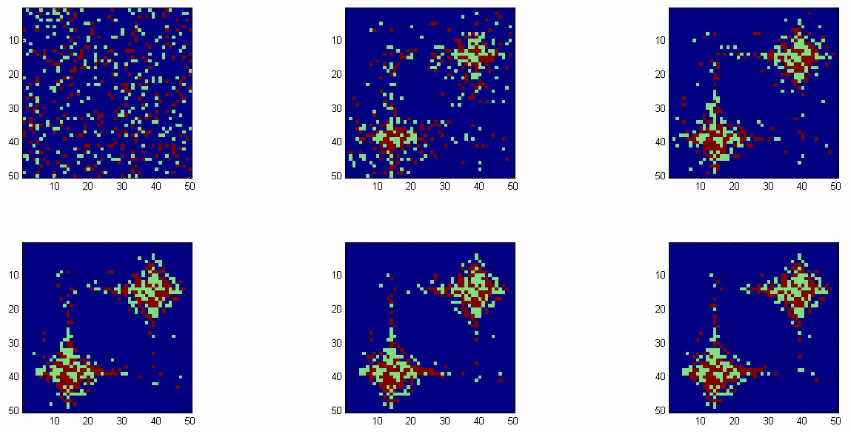
VII. The Results

The results were often consistent with observable phenomena. The agents would move towards the sugar hills and segregate themselves off into several "neighborhood" areas. As I increased the proportion of the population with uniformly distributed racial preferencesng, these neighborhoods became more and more concentrated.

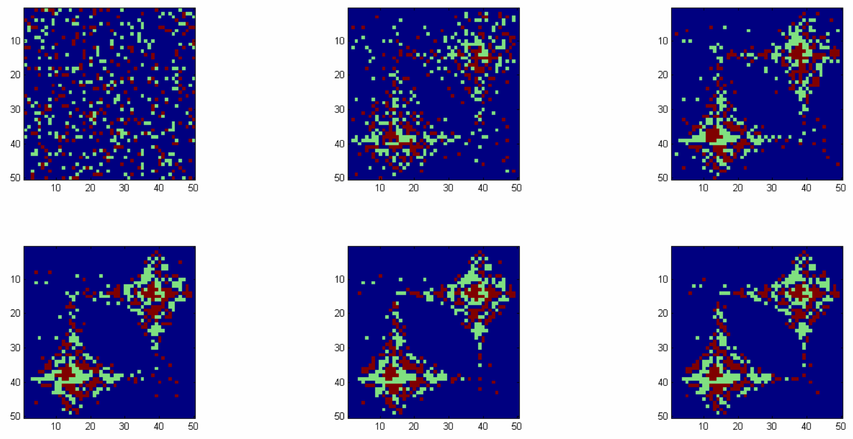
With 10% of the population with uniformly distributed racial preferences:



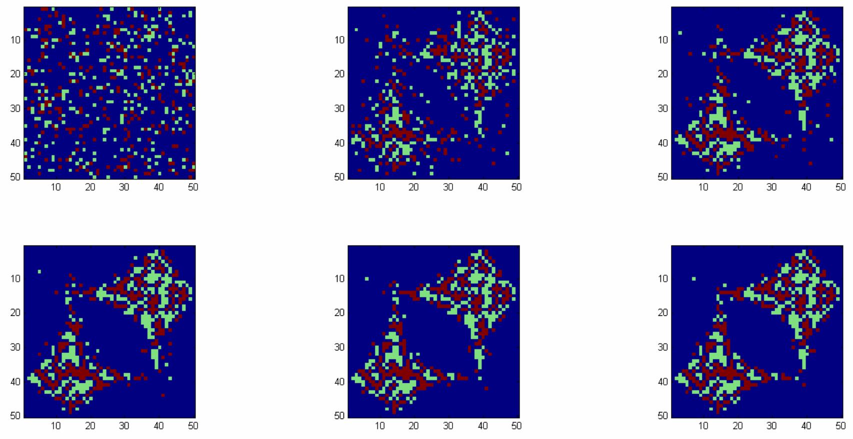
With 30% of the population with uniformly distributed racial preferences:



With 50% of the population with uniformly distributed preferences:

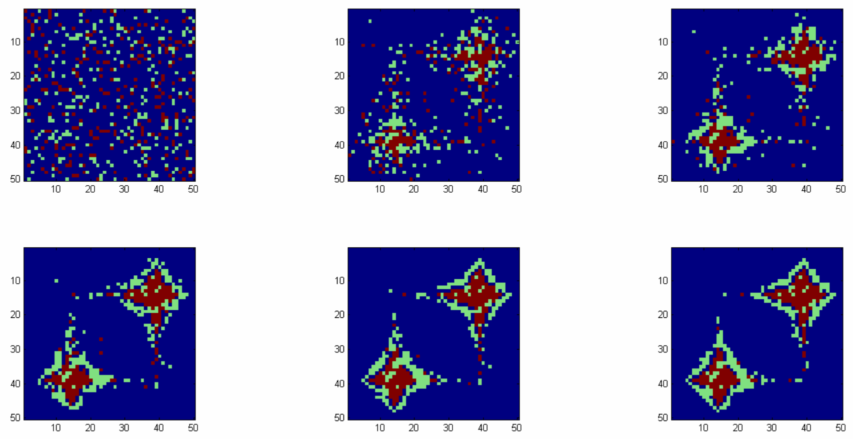


And with 80% of the population with uniformly distributed racial preferences.



As you can see, the areas of racial homogeneity of the “neighborhoods” became more and more concentrated as the proportion of uniformly distributed racial preferences grew. As the racial preferences became stronger, the agents started to leave empty lots between themselves and another type of agent. This seems very much like

My program was *extremely* successful at modeling one aspect of defacto segregation. When I set 30% of one type of agent population with racial preferences and the other population with no racial preferences (that is, one type of agents are racist and the others are not) I get the following results:

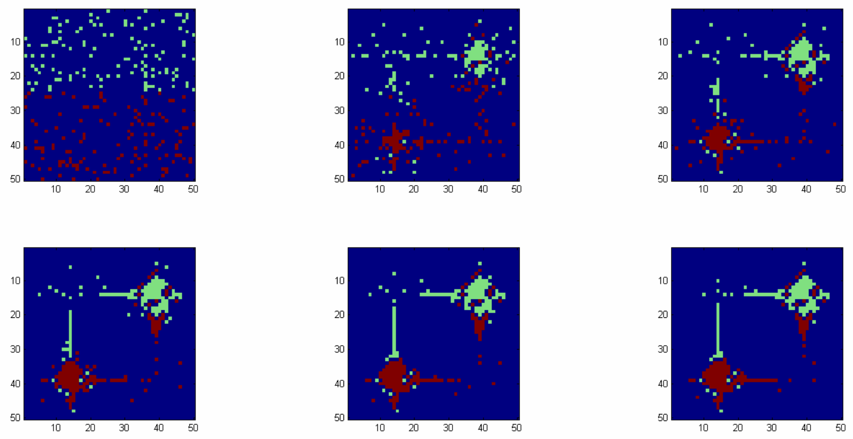


As you can see, the agents with no racial preference move straight for the sugar and the agents' with stronger racial preferences "flee" the sugar hill and sit on the outskirts. I found this result to be particularly interesting. It bears a strong resemblance to the "white flight" to the suburbs. What is equally interesting is this flight occurred with a relatively low amount of racial preference (30% of one population had uniformly distributed racial preferences and the other population had no racial preferences whatsoever).

VIII. Experiment 1

Obviously, following the civil rights movement, blacks and white were not just thrown together randomly in the same geographic space. In fact, the move towards integration was slow. For my first experiment, I wanted to see what would happen if I initialized the agents to be placed on the sugarscape pre-segregated.

As you can see from the figure below, the results were mostly uninteresting to my hypothesis: in general, the amount of segregation was constant *regardless* racial preferences. I am only going to show one trial because, frankly, every trial looked very similar. No matter how much I tweaked the racial preferences, the results were more or less the following:



The results were two separate racial cities with some agents heading into the other sugarhill and forming their own neighborhoods. Some may argue that this seems realistic. This is fine, but this would imply that current defacto segregation has little to do with current racial preferences and almost all to do with the "starting" locations; this idea may be something along the lines of: "the reason there is still segregation is people do not drift far from their starting positions."

IX. Experiment 2

My second experiment what I like to call the "neighborhood tax." I added the following lines to the function "checkrace"

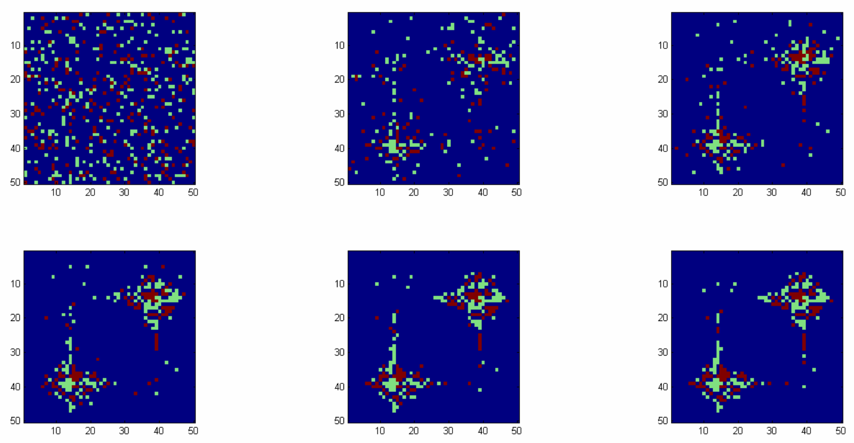
```
tax = 0;
   tax = d* R;
```

Where R is the "tax rate" and I modified the utility function in the following way:

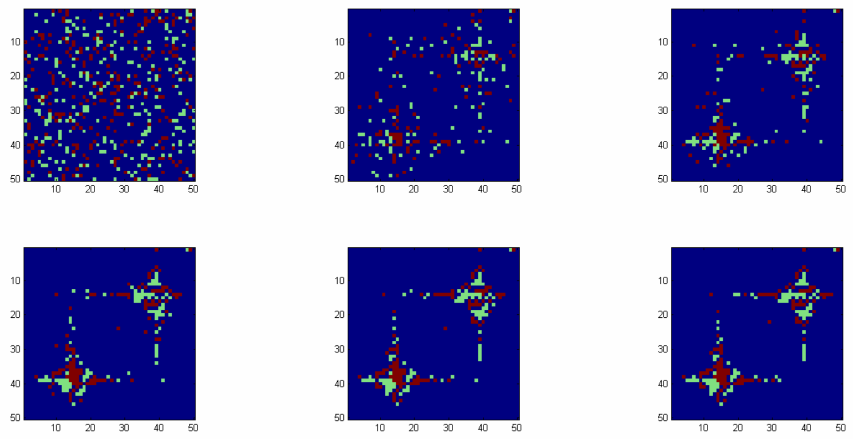
```
     tx2 = a_str(i,j).racepref * d +  s(u,v) - tax;
```

Essentially, this means that the more racially homogenous your neighborhood, the higher your tax rate. This could an example of two separate events: 1) property taxes correlate with racial homogeneity or 2) a law that "taxes" in order to force de-segregation to maintain equality in quality and access to public goods. This should discourage people from being very segregated, however, my results were a little confusing.

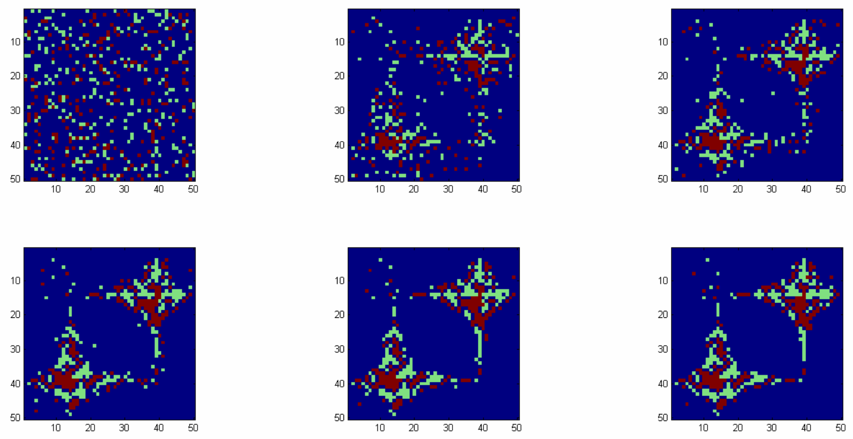
For a 15% tax with 50% of the population with uniformly distributed racial preferences:



For a 25% tax with 50% of the population with uniformly distributed racial preferences:



And finally, a 20 % tax with 50% of the population with uniformly distributed racial preferences:



As you can see, the inclusion of a tax did not vary the amount of segregation much, but rather changed the shape of the “cities” and “neighborhoods”. In fact, the only difference I can see is that the agents spread themselves thinner but are still in the same neighborhoods; I believe they do this because they are able to avoid a high concentration of people with a similar race and therefore a higher tax.

X. Conclusion

In general, I believe my model provided an interesting look at how weak preferences can create large macro-phenomena. My model’s original premise did emulate “white flight” and the branching off into racially homogenous neighborhoods very well. The first experiment revealed that it’s possible that “starting positions” matter much more than was once expected. The second experiment’s results seemed somewhat unrealistic; a tax on segregation resulted in almost the same amount of segregation.

Despite its successes, my model had what some might construe to be a major flaw. It would seem that, as the population became more and more racist, the agents of different races wouldn’t just be in separate neighborhoods in the same cities but in different cities altogether. However, in my model, once they agent was near a “sugarhill” it stayed near the sugar hill and carved out its own neighborhood within it no matter how racist the population became. This is because the agents themselves were only allowed to look at squares immediately surrounding the spot under consideration when calculating utility. If I had perhaps allowed them to “look” further, the result would be much more drastic segregation. There is a certain contention as to which outcome is the more realistic and I believe it is largely an empirical matter.

Work Cited

Schelling, Thomas. *Micromotives and Macrobehavior*, W. W. Norton and Co., 1978, pp. 147-155