Multi-Agent Modelling: Evolution and Skull Thickness in Hominids

Supervised by Brian Mayoh, University of Aarhus,Denmark

Abstract:

Within human evolution, the period of Homo Erectus is particularly interesting since in this period, our ancestors have carried thicker skulls than the species both before and after them. There are competing theories as to the reasons of this enlargement and its reversal. One of these is the theory that Homo Erectus males fought for females by clubbing each other on the head. The other one says that due to the fact that Homo Erectus' did not cook their food at all, they had to have strong jaw muscles attached to ridges on either side of the skull which prohibited brain and skull growth but required the skull to be thick.

The re-thinning of the skull on the other hand might be due to the fact that a thick skull provided poor cooling for the brain or that as hominids started using tools to cut their food and using fire to cook it, they did not require the strong jaw muscles anymore and this trait was actually selected against since the brain had a tendency to grow and the ridges and a thick skull were preventing this.

In this paper we simulated both the fighting and the diet as ways in which the hominid skull grew thicker. We also added other properties such as cooperation, selfishness and vision to our agents and analyzed their changes over generations.

Keywords: Evolution, Skull Thickness, Hominids, Multi-Agent Modeling, Genetic Algorithms

Introduction

Artificial Life

The question of what artificial life is concerned with can be found in the definition of life, since artificial life is interested in modeling real life situations. Can it be possible to approach real life as something that can be modeled? An interesting comment is made on this issue in Grimshaw's notes on his website: "Before the study of physics became important, everything was alive: the stars, the skies, the rivers and mountains, etc. There was no non-life, so the concept was of no importance. It is only when the deterministic mechanics of moving bodies become dominant that the question arises. If all matter follows simple physical laws, and we need no vitalistic explanations of the world's behavior, then what is indeed the difference between life and non-life, between biology and physics?" **(Grimshaw, 2001)** This is a crucial step to leave the prejudice in our minds that life cannot be modeled in computer environment.

Physical symbol system hypothesis, saying that any facet of human intelligence can be understood and described precisely enough for a machine to simulate, was also a crucial inspiration to break the prejudices about the difficulties of modeling real life contexts in a simulation**. (Callan, 2003)** Being courageous enough to dare to model life in a simulation resulted in many interesting projects. This includes anthills, wasp nests, larger forests, towns and cities. To date, very complex and interesting systems have been created by a multitude of very simple entities. For example many ants programmed by very small programs would potentially create an entire system with signs of emergent intelligence **(Champanlard, 2003)**, which can be the reason to believe that the simulation is successful in the sense that intelligence could be modeled within the simulation. The motivation behind the attitude to model real life is closely related to the need for the systems that are based on complex individuals, capable of learning.

Our motivation to work on a project about artificial life came from the fact that there has not been any serious study on this subject. At some point this was risky because we had no idea where simulation results would take us. However the idea of analyzing this interesting evolution process from Homo Erectus to Homo Sapiens, which will be just presented briefly in the next section, was the main motivation source for us.

Homo Erectus and Homo Sapiens

In analyzing the evolution of homo erectus and homo sapiens our starting point was the skull thickness which has been a point of discussion among many scholars. We essayed to see the success of different species in which skull thickness was and was not a factor in survival. However we think concentrating only on the skull thickness would not be very useful to analyze this evolution process. That is why we preferred to create an environment where we can simulate as many factors as possible to be able to demonstrate any aspect that might have had an influence on the evolution of Homo erectus. In this section we will present the scientific approaches to the evolution of Homo erectus, which will be useful especially to compare the results of these approaches to the results of the simulation.

Since the whole process concerning the studies on Homo Erectus skulls is beyond our scope, we will start with a crucial consequence of this process. Despite the early belief that H. erectus had descended from a line of massive, indeed gigantic ancestors, and that modern H. sapiens was the end result of a down-scaling trend, the discovery of new hominid fossils pointed out that the ancestors of H. Erectus did not have massive bones, and neither did H. erectus and in fact except for its strange skull, the skeleton of Homo Erectus resembled our own. **(Boaz & Ciochon, 2004)** At this point the question appears: what could be the reason that would make Homo erectus' skulls different than its predecessors and successors?

One possibility for or contributor to the need for these protective thick skulls seems to be the possible sources of traumatic injury that hominids faced. The thickness of the bone most probably depended on the need to be protected from any impact on the brain. We are not going to go into the details about the works concerning the bone structure of the skull and etc., since it is out of interest at this point. However we will take the results of these studies as the starting point. The idea of defense against trauma to the lower face was described as the pressure behind the evolution of thicker skulls by these studies. The next step of these studies was to find out what kind of interpersonal violence was it that led to the evolution. Sexual selection was the explanation that Darwin offered a long time ago to this kind of behaviour. In other words males' ability to win access to females through competition could be the motivation behind this interpersonal violence. This is why the fight action between the male agents in our simulation is one of the main components of Homo erectus behaviour.

There is a contending theory about the reason for the thinning of the hominid skull between erectus and sapiens. This theory claims that as hominids started cooking their food, they did not depend as much on their jaw muscles as they used to, thus freeing their skulls from the ridges onto which these muscles held and allowing enlargement of the cranial cavity which was required by brain evolution. We tested this theory in our simulation by including a food source that required the skull of the forager to be thicker than a constant times the value of the food to be eaten.

Homo sapiens evolved a larger and thin walled skull, which obviously could not create the protective feeling that Homo erectus skulls did. Since human violence still continues, the stress will be on what kind of strategy that Homo sapiens evolved to. As we introduce Homo sapiens in our simulation, the behavioural change of Homo sapiens will also give a chance to be able to consider its evolution. However skull thickness will not obviously be the only criteria that we are going to use to reach a conclusion about the evolution process. We preferred to create an environment where we can observe Homo erectus and Homo sapiens behaviour naturally up to a point. This attitude will be described in more detail in the next section, where we mainly try to describe our motivation and our goals about the project.

Goals

Our main goal about this project was to be able to create a similar environment, where the evolution process concerning Homo erectus and Homo sapiens took place. To prevent any kind of misunderstanding it should be noted down that the ultimate point we want to reach in this project has nothing to do with confirming or disproving the theoretical studies that have been done. We believe that this kind of attitude would result in prejudiced misleading results. That is why creating the simulation without focusing on any possible consequence was our attitude.

The more concepts we introduced in the simulation, the more results would be complicated to analyze. However it was necessary to represent as many factors as possible to be able to simulate any aspect that could have an effect on the process. That is because, there is a wide range of agent types: male & female homo erectus, male & female homo sapiens, predator and animals; and a wide range of attributes assigned to these agents: vision, skull thickness, courage, amount of sugar in body, metabolism rate, current age, sawAgent, pheroragent, cooperation, which will be explained in detail later. Additionally the concept of cooperation and selfishness was also included in these attributes. The simulation is in fact just a tool to demonstrate the interaction between these agents, follow the change in the values of these attributes and figure out what kind of parameters cause what kind of changes during this evolution process.

The other two attributes that we expect to observe to change during the execution of the simulation are *vision* and *cooperation. Vision* is started inversely proportional to skull thickness in the first generation and it determines how far agents see. It undergoes uniform crossover in the new generations. We wanted to make thick skull disadvantageous for intelligence as it is in theory and we represented intelligence by vision.

Coordination is exclusive to Homo Sapiens and we included it just as another switch which we could turn on or off or change its threshold in order to give or take back slight advantages to the Homo Sapiens in the simulations.

Having mentioned the main goals and motivation of this project, we find it useful to give the reader a basic understanding of our simulation before going into details. The main concepts of multi agent systems mentioned in **(Vlassis, 2003)** will be followed in this section.

Design

Agents:

In our case the design will be based on heterogeneous agents, meaning that agents' attributes are not going to be necessarily similar. Besides the fact that there are six different agent types in the beginning, the evolution process will also lead to different attributes among all agents. As the generations change as a consequence of evolution process the agents will face natural and sexual selection, which will result in the change of attributes. Agents' perceptions and behaviors concerning the environment will be autonomous. In other words criteria concerning an agent's decisions or behaviors will be defined in the design process.

We can talk about the concept of optimal decision making for the agents in our simulation in some sense. All the agents would try to choose the best action for the next step rationally. The concepts of best woman spot; best pheromone spot and best sugar spot are for instance some of the criteria that will help the agents to choose the best action for them. However it is important to keep in mind that the best action for the next step will depend on the knowledge and the perception of each agent, since the agents are heterogeneous. Also the generation, which the agents belong to, will be a decisive criterion to understand the best action for the agents. Especially in the situation of cooperation or selfishness the best action for the agent as an individual does not have to be the best situation for the sake of the community. For instance Homo erectus will not be able to cooperate like Homo sapiens, although cooperating may provide the best consequence for them at some point.

Agent Attributes

There are basically six agent types in our model. The classification is as follows:
Homo Erectus
Homo Erectus Males……..represented by red rectangles
Homo Erectus Females….. represented by blue rectangles
Homo Sapiens
Homo Sapiens Males ……..represented by green rectangles
Homo Sapiens Females……represented by black rectangles
Animal …………………….....represented by yellow rectangles
Predator …………………..….represented by orange circle

Each agent type has its own attributes yet they also share some common attributes and functionalities as well.
The main difference between Homo Erectus agents and Homo Sapiens agents reveals itself in the male agents. While Homo Erectus fights for mating, there is no fight in Homo Sapiens agents and this favors thin skull and more vision which corresponds to more intelligent agents in the next generations as expected.
Another difference between Homo Erectus and Homo Sapiens agents in both males and females is the existence of the cooperation behavior in Homo Sapiens which is for getting large chunks of food corresponding to animals and predators which Homo Erectus cannot gather.
Outline of the attributes according to agents:
Some of these attributes are genetic some of them are decided randomly and given later and some of them are physical properties of the agents
1) Homo Erectus males attributes:

Coordinates: This physical property indicates where the agent stands at a particular instance, the coordinate of an agent changes at each **step()** function and it is determined according to **movetobestspot()** method.

Amount of sugar in body: Sugar is the main need for survival in Homo Erectus males. At each spot they move, they take the sugar at that spot and the accumulation of sugar they collected minus the amount of sugar they consumed for metabolism and for activities like fighting, walking and climbing gives the amount of sugar in body. If this number becomes 0, the agent dies.

Metabolism rate: this is the number of sugar units an agent consumes at each tick or turn regardless of the activity that it does. It is determined randomly at agent birth in the interval between 1 and maxsugar.

Vision: Explained below under the section entitled “Genetic Algorithms”.
Skullthickness: Explained below under the section entitled “Genetic Algorithms”.

Current age: Each thick or turn corresponds to age. Age effects some factors like mating where an agent cannot mate under 10 years of age, this also prevents mating between parents, kids, sisters and brothers since if we let them mate at any age, multiplication would occur over and over at the same spot since the child is put at the same spot with the parents. Similarly, after a certain age, the agents can not mate either. This age limit is determined by the 60 % of the maximum death age. This maximum death age is set in the beginning randomly specific to each agent and an agent dies at that age if he hadn't died because of other factors such as starving or fighting. It is a random number between the minimum and maximum death ages that can be changed at the beginning of the simulation at the user interface.

SawAgent: sawAgent is a boolean variable, it is true if the male agent has more than 4 sugars (the amount it can spend as metabolism in a turn) and it either sees pheremone or a female agent around it.

PherorAgent: This variable is set to 0 as long as the male agent does not see a female. For Homo Erectus, if he sees some pheremone, this variable is set to 0 and sawAgent is set to true. At the point it sees the first female, this variable becomes 1 and the agent quits this for loop, starting on another search, this time searching for the closest female. If the first search ends with the agent not seeing any female but pheremone, than he looks for the maximum pheremone at the closest spot. If the first search returns neither a female nor pheremone, sawAgent is set to false and the agent starts looking for the maximum sugar at the closest location.

Courage: Courage variable is either 1 or 0, if it is 1, the agent fights with a rival for every challenge that he encounters, whereas if it is 0, it looks at the skullthickness of the rival and makes a decision upon whether the rival's skullthickness is greater than that of his or not. In this sense being timid is actually being cautious.

2) Homo Erectus females attributes:
Homo Erectus females have similar attributes yet they do not have the attributes such as sawAgent, PherorAgent and courage which effect the strategy in mating. Moreover, they have the skullthickness as it is inherited but this does not affect any behaviour in females unlike males' fighting. Also they do not use their vision for mating, they use it for finding food. One different property that females have is that they leave pheromones after the age of 10 and before age maxAge*0.6 for the males to follow. Females should be older than 10 years and younger than maxAge*0.6 to mate as well.

3) Homo Sapiens males attributes:
no courage variable
coordination gene
selfishness gene
Coordination: Explained below under the section entitled “Genetic Algorithms”.

Selfishness: Explained below under the section entitled “Genetic Algorithms”..

4) Homo Sapiens females attributes: As opposed to Homo Erectus females, Homo Sapiens females do not emit pheromone, and they cooperate. They can be selfish or not, like the males.

5) Animal Attributes: Animals have their sugar value (the food in them) and a random vision given in the beginning of the simulation.

6) Predator Attributes: Predators also have their sugar (dynamic) and vision(static) values. In addition they have a metabolism value which is subtracted from their sugar at every turn.

	Erectus male	***Erectus female***	***Sapiens male***	***Sapiens female***	***animal***	***predator***
coordinates	+	+	+	+	+	+
metabolism	+	+	+	+	-	+
sugar	+	+	+	+	+	+
vision	+	+	+	+	+	+
skull	+	+	+	+	-	-
age	+	+	+	+	-	-
sawAgent	+	-	+	-	-	-
PherorA.	+	-	-	-	-	-
cooperation	-	-	+	+	-	-
courage	+	-	-	-	-	-

Table 1. Attributes found in each agent type.

GENETIC ALGORITHMS AND INHERITANCE

In the simulation, there are mainly 4 traits that are passed to the other generations by inheritance. These are:
vision in HomoErectus and HomoSapiens
skullthickness in HomoErectus and HomoSapiens
cooperation in HomoSapiens
selfishness in HomoSapiens

All these traits are represented as separate genes given by integer arrays of 10 cells. Each cell is either 0 or 1. However, some limits apply for some of the genes.
The map is of size 50x50. So, since we wrap around to the beginning if we exceed the maximum coordinates, a vision of 25 should be enough for everyone. Thus, in creating the first generation, at least the first 3 cells of the visiongene can be filled with zeros. Putting a one at the 4th cell ensures that the vision of the agents (only that of the Homo Genus, since the animals' vision values are determined randomly) never falls below 32. The rest of the cells can be filled randomly for the first generation. If the sights of the agents proves to be too small or too big (causing slowdown or preventing natural selection), it is possible to set which cell we want to put this “1” at in the user interface.
Currently, we give a random 0 or 1 to the digit that is to the left of this mandatory one, too. So, by

default this value is 6, i.e the agents' vision can not fall below 8 and it can increase up to 31.

The skullthickness gene for the first generation of agents is generated by complementing each agents' vision gene. As an illustration:
let an agent's vision gene be:

0	0	0	0	0	0	1	0	1	1

Then, taking the complement, this agent from the first generation has a skullthickness gene as:

1	1	1	1	1	1	0	1	0	0

This random determination of vision gene and then taking the complement to create the skullthickness is just valid in the first generation where we have to create the agents from scratch.
The numerical value of these traits vision and gene is done by converting the binary value of these sequences into the decimal form. For example, the numeric vision value for the agent that we stated above is:
1+2*1+8*1=11
the skullthickness value is:
4*1+16*1+32*1+64*1+128*1+256*1+512*1=1012
For the other generations after the first generation of agents, traits are inherited from the parents via uniform crossover. There is an integer array variable called chromosome[10], which stores the current value of the gene created as a result of crossover. During crossover, each cell in the gene is either coming from the mother or father in a random fashion. As an illustration for the inheritance of vision gene:
Let g1 be the vision gene of mother and g2 be the vision gene of father:
g1:

0	0	0	0	1	0	1	1	0	0

g2:

0	0	0	1	0	0	0	0	1	0

A possible crossover between two genes would result in the new agent's vision gene as such:

0	0	0	1	1	0	0	1	1	0

One expectation from this inheritance and crossing over is that agents with more favored characteristics would survive and mate, thus the later generations will inherit such characteristics. For example, skull thickness in homoerectus is essential in mating and plays role in sexual selection.

Individual genes

Vision: The first generation of agents which are put in to the space have random vision genes of 10 digits filled with either 0 or 1 and the integer equivalent of this binary number indicates the multitude of the vision, the larger this number is the better agents see the sources and also the females, thus they are more advantageous in finding sugar and female. The later generations inherit vision from their parents via crossover of genes. We have set upper and lower limits at vision. The lower limit is $2^{visiongeneoneat-1}$ and the $2^{visiongeneoneat+1}$. We needed to have an upper limit since the simulation runs slower the higher the vision of each agent.

Skullthickness: This gene, as described above, is the complement of the visiongene for the first generation and for the offsprings, it is determined by the crossover of the mother's and father's skullgenes. Its value is essential in determining which Homo erectus male wins a fight (and which one dies) when they are fighting for a female..

Cooperation: In the first generation of Homo sapiens, by default this gene is filled with 0's up to the 4th slot. It is either 1 or 0 from the 4th slot on so the maximum value a Homo sapiens' cooperation can have is 63. The slot number can be changed at the user interface. This value is used when hunting a herbivore (animal) or killing a predator. If the cooperation values total of all HomoSapiens in the predetermined neighborhood of the animal is greater than the animal's sugar, the animal is “reborn” in a different part of the map so these hunters are clear of danger al least for a while.
In the same way, they hunt herbivores (animals) and if they are successful the sugar of that animal is distributed among participating hunters, if not, the animal runs away.

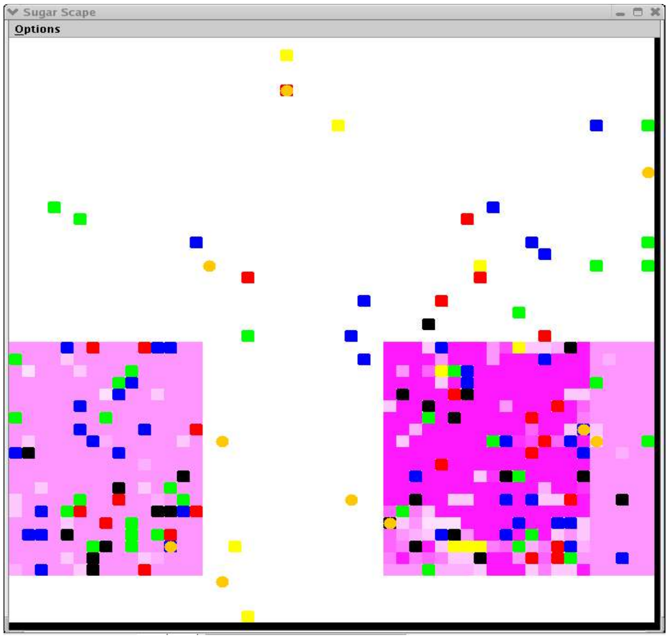
Selfishness: This is also a property of Homo sapiens. Only the last two slots of this gene are used. The selfishness gene is determined randomly for the first generation and later, it is passed on to offspring by uniform crossover. In the GUI, if selfishness gene is selected to be recessive, the gene should have 1's in both slots to be selfish. Otherwise, if it is selected as a dominant gene, only one slot to be 1 is also enough to be selfish. How it works is when an animal (herbivore) is near, normally all surrounding Homo sapiens' coordination values are added up and the success of the hunt is determined. If one of the agents turns out to be selfish though, he/she will try to hunt the animal itself. If his/her sugar exceeds that of the animal, he/she gets 90% of the animal's sugar. If not, the animal runs away and no member of the hunting party gets anything.

We are aware that one gene of two nucleotides cannot determine a mammal's selfishness. This applies to all our other genes as well. Therefore, although we call these constucts genes, they should really be regarded as phenotypes that are the end product of the interaction of numerous genes. Thus, collectively, our selfishness variable simulates the probability of selfishness in the population.

The Environment

The environment in the simulation can be described as dynamic. Although there will be some factors such as the altitude and the map (in the sense that coordinate structure will be the same during the simulation) which will be static, the amount of food for instance will be dynamic. Furthermore from the perspective of a single agent, the change in positions of the rest of the agents will lead to a dynamic feeling.

The world where agents will interact will be discrete meaning that there will be only finite number of places that agents can move to. Another important point to mention about the world is that it will be partially observable by the agents depending on the vision attribute of the agents. However the average vision is also supposed to change as the generations change in the simulation.



a) Attributes

There are mainly four attributes of the space in which the agents act. They are:
- pheromone
- sugar
- altitude
- size

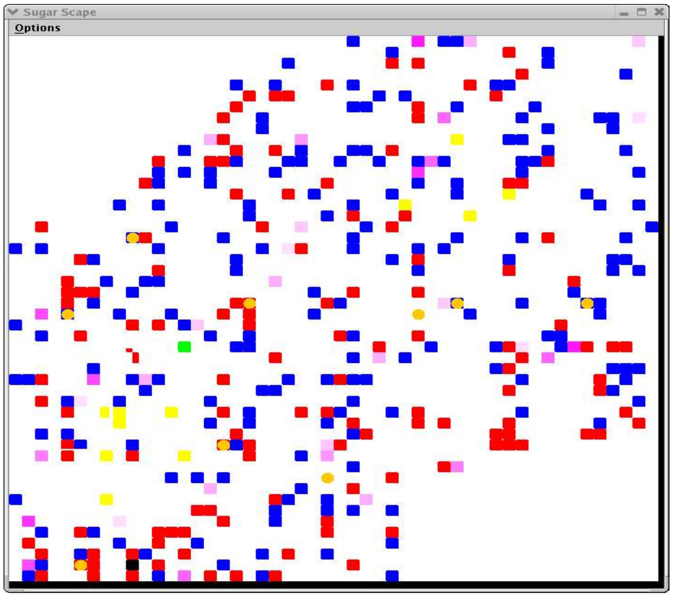


Figure 1: Pheromones are seen trailing behind female agents in this map.

i) Pheromone:

Pheremone level at a particular coordinate is depending on the HomoErectus female's path. On the path of the female agent, she puts 5 units of pheromone whereever she passes. In addition to this, pheromone also evaporates by time and this is done by subtracting 1 unit of pheremone at each turn from the spots that have pheromone on it. Pheromone putting is controlled by the method:
space.putPheremoneAt(x,y, 5) , here x and y are the coordinates.

ii) Sugar:

Figure 2: Two sugar squares of different values (the lighter one has wrapped around)

Sugar level is basically determined randomly. There are two main rectangle sugar areas each of size 20*20=400 unit squares. The areas outside these two rectangles do not have any sugar on them. However, these sugar attractions's positions and the sugar level they have change randomly.

iii) Altitude:
Altitude is another important factor in determining agent's movements and displacements due to its cost as we have explained in the Agents and Actions section.
Altitude is set in different coordinates in a random fashion. Within the map, for each coordinate, a random number between 0 and 100 is given as the altitude value. With the getcurrentAltitude(x,y) method, we can get the altitude at (x,y) coordinate.

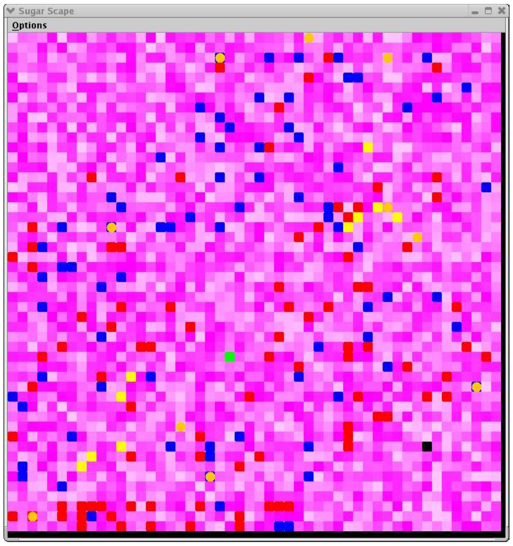


Fig 3: Higher altitudes are dark purple while shallow areas are lighter

iv) Size:
Size of the map is 50 units in the x-coordinate and 50 units in the y-coordinate, leading to a total area of 2500 unit square.

Natural Selection & Sexual Selection

Evolution refers to change over time, or transformation over time. Evolution assumes that all natural forms arose from their ancestors and adapted over time to their environments, thus leading to variation. In evolution, there are many rules the environment places upon the survival of a species. There are also numerous ways in which evolution occurs, the most noted are Natural Selection, Sexual Selection and Adaptation.

Natural Selection

In our simulation, natural selection will be mostly concerned with the agents' abilities of vision and the amount of food that the agents have. The vision is the main criterion to evaluate an agent's knowledge about the environment. That is why the better vision an agent has, the more aware of the environment it is. Consequently the agent will be able to make better decisions for future steps and be able to adapt to environment easier. Secondly, the amount of food that an agent has is also important. For instance when agents fight predators, if they lose they get injured and lose some food. Because of these kinds of situations it is more likely to expect a tendency towards cooperation among the agents.

Sexual Selection

In our simulation female agents do not decide on mates, so sexual and natural selection depend on male actions. The results are average skull thicknesses of both species, i.e those of Homo Sapiens' who do not fight and those of Homo Erectus' who do.

Repast

For our simulation we chose the Repast library. The University of Chicago's Social Science Research Computing's RePast is a software framework for creating agent based simulations using the Java language. **(**An Agent Based Modeling Toolkit for Java**)**

We built our model on the SugarScape model that comes with Repast 2.0. Since the environment was already ready, the programming that needed to be done was based on adding new agents to the simulation and assigning attributes to these agents.

Methods and Behaviours

Moving to the best spot: This is directed by the MovetoBestSpot() method. Homo Erectus and Homo Sapiens agents basically need sugar for survival, therefore finding sugar to sustain their living is their first aim in their movement. As second priority, if they already have enough sugar which corresponds to sugar level more than 4 units, they aim for mating and the criteria is that if they have a sugar amount of more than 4 units, they go and search for mates. However there are some differences between Homo Erectus males and females and Homo Sapiens males and females. According to the classification, the basic structure of moving to best spot is:

In Homo Erectus males, the basic mechanism for moving is as follows:

```
if male's sugar >4  // so that the male agent can survive
        look around for women or pheromone
        if sees women or pheromone-> sawAgent variable becomes true
                        if sees pheromone->pherorAgent becomes 0, thus the agent
             goes to the best pheromone spot or stays at the same spot
                        else if sees female->pherorAgent becomes 1, thus the agent
             goes to the best woman spot or stays at the same spot
        else // no women nor pheromone around
                        ->go to the best sugar spot
```

In this loop, the male agent looks for pheromone until he cannot see women and this is controlled by the pherorAgent variable. After it sees woman, it gives up tracing the pheromone and traces the woman instead. This favours the number of population since just searching for pheromone is not realistic or practical in finding a mate.

Vision and potential spots for movement

The group of potential spots that will be analyzed by the agents depends on the agents' vision capacity. The agents basically look in four directions from the center.

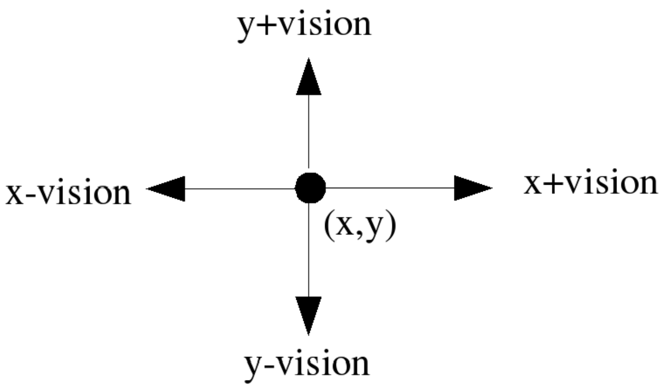


Figure 4. Potential spots of movement

The agent looks at all possible destinations within the above stated limits. This is maintained by two nested for loops, one for the x direction, another for the y direction.

<u>Explanation of the best woman spot, best pheromone spot and best sugar spot:</u>

The space has the altitude attribute and climbing some altitude and walking some distance requires the agent to consume sugar(energy). Moreover, climbing some altitude consumes twice as more sugar(energy) units than walking does. Therefore, the total energy consumed to go to a specified destination is the sum of these two criteria. The agents in our simulation make a calculation before moving to a spot and choose the optimal destination according to this energy consuming criteria.

The formulation of the total energy consumption is:

(x-xLook)+2*(space.getAltitudeAt(xLook, yLook)-space.getAltitudeAt((x,y),0))

where **x** and **y** is the current location of the agent
xLook and **yLook** is the potential spot for the agent to go
getAltitudeAt(x,y) method returns the altitude of an x,y spot

Thus,
x-xLook: The energy consumption for walking a distance of x-xLook long
2*(space.getAltitudeAt(xLook, yLook)-space.getAltitudeAt((x,y),0): The energy consumption of the agent to climb the altitude difference between his current altitude and the potential spot's altitude.

In choosing the best woman spot, the nearest woman is not determined solely by the 2 dimensional measure of the distance. For instance, a woman that is 1 unit far on the x-y coordinate from the male agent yet 100 meters higher than the altitude of the male agent in the z coordinate is less favorable than a woman 20 units away in the x-y coordinate and 50 meters higher than the male in the z direction. In the first case the energy consumption of the male to reach that woman would be 100*2(coming from the altitude)+1(coming from walking)=201 units of sugar whereas in the second case it would be 50*2+20=90 units of sugar which is of course better. After deciding this best place where the female stands, the agent subtracts this energy requirement from the current sugar level of himself and decides to go to that spot or not accordingly. If he finds out that his sugar level will go below 4 units he does not go to the female.

For choosing the best pheromone spot, the agent looks at the pheromone density in all the squares it can see and chooses the square with most pheromones if going there does not endanger his life (i.e runs him out of sugar). If there is more than one square with the same density of pheromones, he chooses the square in going to which he will consume the least amount of energy. Pheromone is traced in the case that there exists no woman within the vision capacity of the agent.

For choosing the best sugar spot, the energy to be consumed in order to go to the destination is subtracted from the potential spot's sugar level and the difference is added to the current sugar level of the agent. Among the options he can survive, the one with the biggest gain is selected.
The net sugar gain is formulated as follows:

```
                    space.getSugarAt(xLook, yLook)-(x – xLook)+
        2*(space.getAltitudeAt(xLook, yLook)-space.getAltitudeAt(x,y),0))
```

where,
getSugarAt(x,y) gives the amount of sugar at the x,y spot in the space.

Moreover, if they would not have a net gain from this displacement, they remain in their existing position.

For Homo erectus females, the only concern is finding the best sugar spot and the above mechanism applies to them too. In the case that they would not gain from this displacement, they do not move either. Moreover, they look at the potential spots within their vision capacity as stated in vision and potential spots for movement section. Thus, women with more vision have advantage in finding food.

For Homo Sapiens, the same rules for Homo Erectus apply, however one difference is that Homo Sapiens males do not trace pheromone, they search for women by only visually. Moreover, Homo Sapiens eventually have more vision and less skull thickness which increases their fitness for finding food. As the area that they search for food is larger, they are also better explorers. Homo Erectus tend to stay at a particular position for a long time and combined with the fact that they can just see four directions, the sugar spots that are actually close but out of their vision cannot be explored by them, whereas for Homo Sapiens, although they also scan 4 directions, they change positions more frequently and therefore they can check sugar spots from more angles relatively.

Animals do not consume energy by walking or climbing so they do not consider those factors. They only look for the nearest among the spots with the greatest amount of sugar so that they do not take into account the altitude. Their move to best spot function is very simple. They look as far as their vision allows in the x and y directions and migrate to the spot that contains the most amount of sugar. If there are several spots with the same sugar, they choose the closest one.

Predators just eat animals, Homo sapiens and Homo erectus. They look in the x and y directions as far as their vision allows and when they spot anything, they keep its distance and update their best distance variable when a new shortest distance is found. In the end, they move to the best spot and kill the agent that was there. To kill an agent at that point and transfer its sugar to the predator, the Kill() function is called.

Of course, all agents that do not have a reason for being in the same spot with another one check if a spot is occupied before considering it. The only agents that move to a spot occupied by another are

Homo Erectus (male): wants to move to a spot where a female Homo Erectus is in order to mate.

Homo Sapiens (male): wants to move to a spot where a female Homo Erectus is in order to mate.

Predator: wants to move to a spot occupied by anything but a predator in order to eat it.

Mating

The prerequisite for mating in both kinds is that one male and one female should be found at the same location. In addition to this, it should be noted that two agents of the same sex cannot be found at the same location. There are some differences in the mating mechanism of the two types Homo Erectuses and Homo Sapiens arising from fighting and tracing pheromone. As stated before, Homo Erectus males may fight with a rival for a particular female depending on the courage value of the agent, whereas Homo Sapiens do not fight, the only criteria is for the male and female to be found on the exact same location at an istance.

Mating in Homo Erectuses: Mating in Homo Erectuses is controlled by the method SeekandDestroy(). The whole map is spanned by two for loops and each HomoErectus male is evaluated in this method. Among these males, the ones who satisfy the conditions for age limits are elected and the others are filtered. According to this age limit, any agent should be older than 10 years old and younger than 60 % of the maximum death age.
So the first condition to be eligible for mating is that:

10 < age of the agent < 0.6*maximum death age of the agent

In the second step, for the male agent that is being evaluated which we called Rocky, namely the male agent in the center corresponding to the one at the (i,j) coordinate, the rival agents for Rocky are determined. This determination process is as follows: the male agents in the four sides of Rocky are stated,

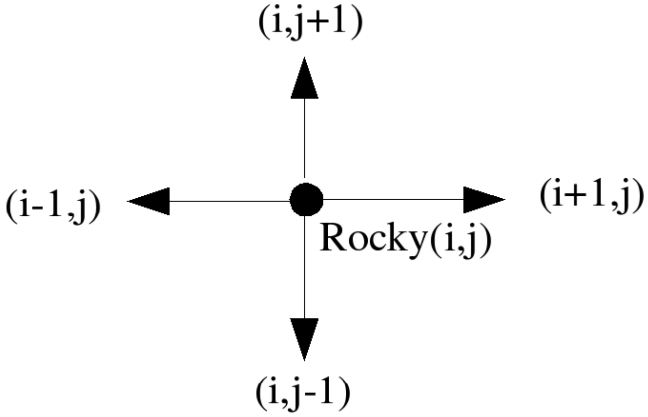


Figure 5. Potential rivals of Rocky

The male agents that are found on those 4 spots are evaluated to see if they are really rivals or not.

If their sawAgent variable is 1 (meaning they are on the lookout for a female) and they also comply with the age limit for mating, then they are found to be rivals to Rocky and we call the rivals as Ivan. One by one, Rocky is pitted against Ivans on the four sides. The main criteria for determining who will win the fight is the skull thickness plus the level of sugar in the agent's body, the agent with a greater sum of these two factors wins the fight and is thus granted the right to mate with the female. This can be outlined simply as follows:

```
if Ivan.bone/2+Ivan.sugar/2 < Rocky.bone/2 +Rocky.sugar/2 --> Rocky wins
else --> Ivan wins
```

However the courage values of agents play an important role at this point that can change the whole situation. To remind how courage variable affects, if courage variable is 1(randomly), the agent fights without evaluating the rival's strength, so such an agent fights in every situation. However, if the courage variable is set to 0, the agent acts smarter, evaluates the rival's skull thickness and decides to fight or not based upon the rival's skull thickness.

One thing to be pointed out here is that for the actual fight to be won both sugar level and skull thickness play role, however for the agent to decide to fight or not, only the skull thickness plays role since the agent cannot guess the other's sugar level in the body.

To summarize:

```
if Rocky.courage == 0
        Rocky looks to Ivan's skull
        if Rocky.skull < Ivan.skull -->
                Rocky runs away from the fight so Ivan'll mate whatever the
                sugarlevel+skullthickness is
        else
                Rocky fights with Ivan so that  sugarlevel+skullthickness
                determines who'll mate and who'll die
```

Eventually, after fighting one of the agent dies, whereas if he was not courageous, he could have saved his life. Moreover, 4 sugar units are subtracted from an agent who fights as a cost of fighting. Also, when an agent is born, it is put on the same location as with his/her mother.

In our simulation, Rocky's and Ivan's courage values are looked at at the same time.

There are four different conditions:

(1,1) Rocky.courage=1, Ivan.courage=1
(1,0) Rocky.courage=1, Ivan.courage=0
(0,0) Rocky.courage=0, Ivan.courage=0
(0,1) Rocky.courage=0, Ivan.courage=1

there are four different results:

A) Ivan dies, Rocky mates
B) Ivan mates, Rocky dies
C) Ivan runs away, Rocky mates
D) Rocky runs away, Ivan mates

There are two different skullthickness possibilities:

Rocky.skull > Ivan.skull
Ivan.skull > Rocky.skull

There are two different final conditions for fight

R) Ivan.bone/2+Ivan.sugar/2 < Rocky.bone/2 +Rocky.sugar/2
I) Ivan.bone/2+Ivan.sugar/2 > Rocky.bone/2 +Rocky.sugar/2

To illustrate the relationship between all possible situations:

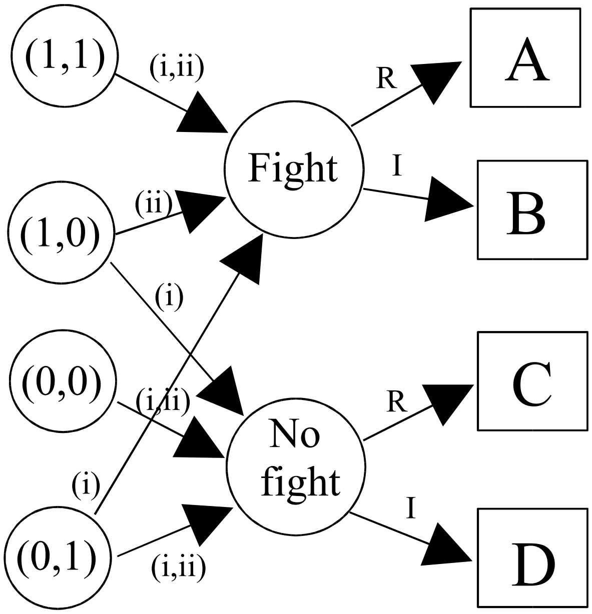


Figure 6. Possible fight states

Mating in Homo Sapiens: Mating in Homo Sapiens is fairly simple since no fight takes place. It is controlled by the SeekandCreate() method and the same rule for age limit applies as with Homo Erectus. After all, when the above conditions are met, in the case that a male and a female agent is found at the same spot, a new agent is created with traits inherited from the mother and father.

Killing and eating by a predator

The predator calls this function right after moving to a new position. The function is called Prey() and it's found in SugarModel. If the predator is small (has less than 50 sugar) it can only hunt animals. Else it can look for all other agents. If an agent is present in the coordinate of the predator, it is removed from its relevant agent list and 90% of its sugar is transferred to the predator.

Hunting

Hunting is the killing by and the transfer of sugar of animals to Homo Sapiens'. The hunt function is called in SugarModel. At every turn, the map is scanned for animals. If an animal is found, we look at the Homo Sapiens males and females in a specified square whose size is determined by the user in the GUI surrounding the animal. If each of these Homo Sapiens people are non-selfish, then we add up their coordination points and if this total exceed the sugar of the animal, the animal is killed and its sugar is distributed equally to the participants of the hunt.
If, however, one of the participants turns out to be selfish, he/she tries to have the animal to himself and if his sugar exceeds that of the animal he gets all the animal's sugar. If not, the animal runs away and no one gets anything.
The *HuntAnimalsAtAge* variable at the user interface controls how old an animal should be before Homo sapiens people can try to hunt it. This is done to make sure that not all animals are small and easily huntable.

Killing Predators
Predators are dangerous animals. If they are close by, this means somebody's going to die soon. Therefore, Homo sapiens people have decided to take preventive action. Thus we search for Homo sapiens' surrounding a predator at every turn just as we do for the other kind of animal. We add up the coordination points of these people and if this number exceeds the sugar value of the predator, the predator is killed. If not, the participators of this kill attempt are subtracted the amount "sugar of the predator/number of participators/2" each from their sugar values. This amounts to getting wounded in the attempt. If any of the agents' sugar falls below 0 in this process, he/she dies a hero.

The *CanKillPredatorsAtAge* variable in the user interface determines how old a predator should be before Homo sapiens can attempt to kill it. This is done so that the predators can build up some sugar and are not constantly killed and reborn in a crowded map.

GENERAL MECHANISM OF THE SIMULATION

There are 8 important classes that we've either created from scratch or modified in the RePast library, these are:
SugarAgent--> representing HomoErectus males
SugarAgent2--> representing HomoErectus females
HomoSapiens--> representing HomoSapiens males
HomoSapiens2 --> representing HomoSapiens females
Animal → representing herbivores.

Predator
SugarSpace--> implements the environment in which the agents act
SugarModel--> implements the core of the simulation where execution takes place.

We've explained in the previous sections the attributes of the agents and the space in which they act. In addition to these, SugarModel operates and organizes the systematics of the program.

The BuildModel() function is called first. This creates the first generation of Homo Erectus and the whole animal and predator populations, also one Homo Sapiens male and one female are created in order to open the agentlist of the HomoSapiens, since if one of the agent type's agentlist becomes null, the simulation halts.

In the SugarRunner class within the SugarModel, execution of the whole simulation takes place. The order in which the simulation works in SugarRunner is as follows:
The Homo Erectus and Homo Sapiens offsprings conceived in the previous turn are born (added to the simulation).
New animals can be born but we don't do this and keep our animal number constant.
New predators can be born but we don't do this and keep our animal number constant.
HomoErectus males are shuffled in order to change the order of agents randomly in the agent list to make a fair distribution

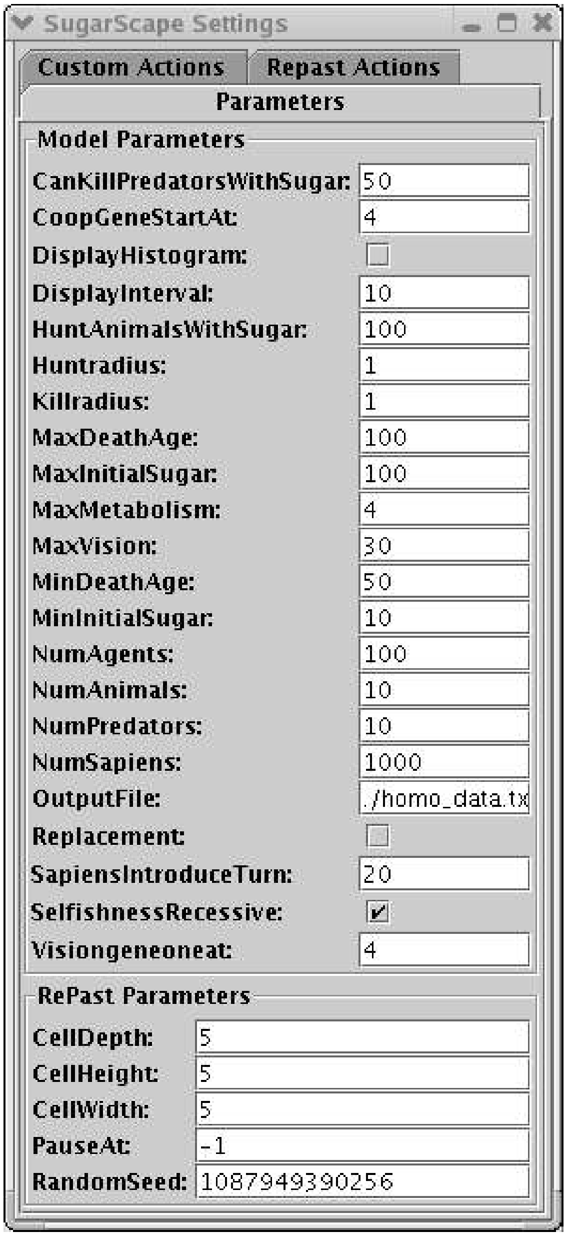
HomoErectus females are shuffled
Animals are shuffled
Predators are shuffled
Homo sapiens males and females are shuffled separately.
HomoErectus males are stepped*
HomoErectus females are stepped*
HomoSapiens females are stepped if HomoSapiens females have been introduced*
HomoSapiens males are stepped if HomoSapiens females have been introduced*
*The order of the species changes randomly in order not to give advantage to one over the other. i.e, Homo Erectus males and females are stepped first with 50% chance. We step the animals and predators after the humans because their numbers are constant and they do not pose a threat to the humans' food supply.
Animals are stepped
Predators are stepped
Predators eat the agent they find at the coordinate they go to
Sugar amount in the space is updated which means one unit of sugar growth in the rectangles if the rectangles are not saturated (did not reach the maximum level of sugar) and distribution of sugar at this point is that that has formed after all agents except predators have taken the sugars at the spots that they were standing on in the step() functions above.
HomoSapiens attempt to hunt
HomoSapiens attempt to kill predator
The two rectangles of sugar areas are updated randomly as explained in the space and its attributes section
The possible mating situation of the HomoErectuses is checked for each male agent in each coordinate, new offsprings are created accordingly, taking into account the fighting between rivals
The possible mating of the HomoSapiens is performed for each male agent found in each coordinate, new offsprings are created
If the time has come to introduce the Homo Sapiens population, (this time can be changed at the user interface) this first generation of Homo Sapiens males and females are added to the simulation.
The agent display, the agent attributes plot and the agent wealth histogram are updated.
For all agent types, the agents who have died in the previous actions and been put into their respective reaperqueues are reaped,i.e removed from their lists and grids. However, we chose to do this at the moment an agent is killed without waiting for the reapagents function because otherwise dead agents would still be in their respective grids and would be considered again.

Here, agentList is an important data structure as a variable of ArrayList type defined in the RePast library. It is basically a list of a specified agent type so that every type has its own list. The size of the agentlist returns the number of that agents in that list or in the simulation. If a new agent is created or removed, it should be added or deleted from the agentlist too.
Step() is a method of the classes for each agent type. Basic actions like moving to best spot, taking sugar at the current spot and increasing the currentage are performed in the step() function. Overall, the sequence of actions that are explained above happen within one turn which means one tick.

USER INTERFACE & HOW TO USE THE SIMULATION

The user interface is self-explanatory, yet we need to explain the fields that take the parameters which change the results of the simulation.

Warning: These parameters can only be changed before a simulation is started, not during it. CanKillPredatorsWithSugar determines the amount of sugar that is needed to be found in the body of the predator so that the agents decide to kill the predator or not, e.g, if the sugar amount in the predator is too low, it won't be worth killing the predator.



Model Parameters

CanKillPredatorsWithSugar:	50
CoopGeneStartAt:	4
DisplayHistogram:	□
DisplayInterval:	10
HuntAnimalsWithSugar:	100
Huntradius:	1
Killradius:	1
MaxDeathAge:	100
MaxInitialSugar:	100
MaxMetabolism:	4
MaxVision:	30
MinDeathAge:	50
MinInitialSugar:	10
NumAgents:	100
NumAnimals:	10
NumPredators:	10
NumSapiens:	1000
OutputFile:	./homo_data.tx
Replacement:	□
SapiensIntroduceTurn:	20
SelfishnessRecessive:	☑
Visiongeneoneat:	4

RePast Parameters

CellDepth:	5
CellHeight:	5
CellWidth:	5
PauseAt:	-1
RandomSeed:	1087949390256

CoopGeneStartAt: this sets the maximum value of cooperation. With 4, it is 63, with 5 it is 31 etc.

Figure 7. User interface

Display Histogram: Histogram is the window displaying the average wealths of the agents

DisplayInterval: At specified intervals, the display will be changed. For example if it is selected to be 10, after every 10 turns the display will be updated.

HuntAnimalsWithSugar: Minimum amount of sugar that should be found within the animal for the agents to hunt them.

Huntradius: In the nested for loops that we use in the hunt() function in SugarRunner, we search for Homo Sapiens around the animal starting at (i-huntradius, j-huntradius) and ending at (i+huntradius, j+huntradius) where(i,j) is the animal's coordinate.

Killradius: In the nested for loops that we use in the killpredator() function in SugarRunner, we search for Homo Sapiens around the predator starting at (i-killradius, j-killradius) and ending at (i+killradius, j+killradius) where(i,j) is the predator's coordinate.

MaxDeathAge and MinDeathAge: The death age of each agent except the animals and the predators is determined in between these two values.

MaxVision: This is set for the animals and the predators. Their vision is between 10 and maxVision.

MinInitialSugar and MaxInitialSugar: Each agent starts with an amount of initial sugar in between these values in their body.

NumAgents: Number of HomoErectus from each sex in the beginning of the simulation.

NumAnimals: Number of animals throughout the simulation.

NumPredators: Number of predators throughout the simulation.

NumSapiens: Number of HomoSapiens from each sex when they are first introduced to the simulation.

SapiensIntroduceTurn: The turn when HomoSapiens are introduced to the simulation.

SelfishnessRecessive: Determines if the selfishness gene is recessive or dominant.

Visiongeneoneat: The location of the obligatory 1 in the vision gene. This sets a minimum value of vision for the agents. The slots of the vision gene to the left of this “1” are 0, the slots to the right are random. 4 corresponds to 32, 5 to 16 and so on.

Disp: You must go to the SugarModel.java file and change the initial disp parameter to altitude or pheromone in order to view these attributes of the space. Since the map is created in the beginning of the program, we could not make this so it could be done through the user interface.

RESULTS OF THE SIMULATION

The results of the simulation which can be related with the graphics representing the population dynamics, average skullthickness, average vision and other properties of the agents are dependent on the parameters that we supply both in the GUI and within the program. These parameters have been explained in the previous section. Starting with simulation # 11, Homo Erectus' are required to have a somewhat thick skull (which correlates with strong jaw muscles) in order to chew their food.

Constant factors:
huntradius: 1
killradius:1
maxsugar:100
minsugar:30
# of predators: 10
# of animals: 10
can kill animals that have at least sugar: 100
can kill predators that have at least sugar: 50
Selfishnessgene is : Recessive*
Agent vision values minimum: 16*
Cooperation gene max: 63*
*these parameters change in the later simulations and you'll be told so when they do.

Simulation run # 1:
Variables:
Initial population of Homo Erectus : 100
Initial population of Homo Sapiens: 10
Homo Sapiens Introduced At Turn: 20

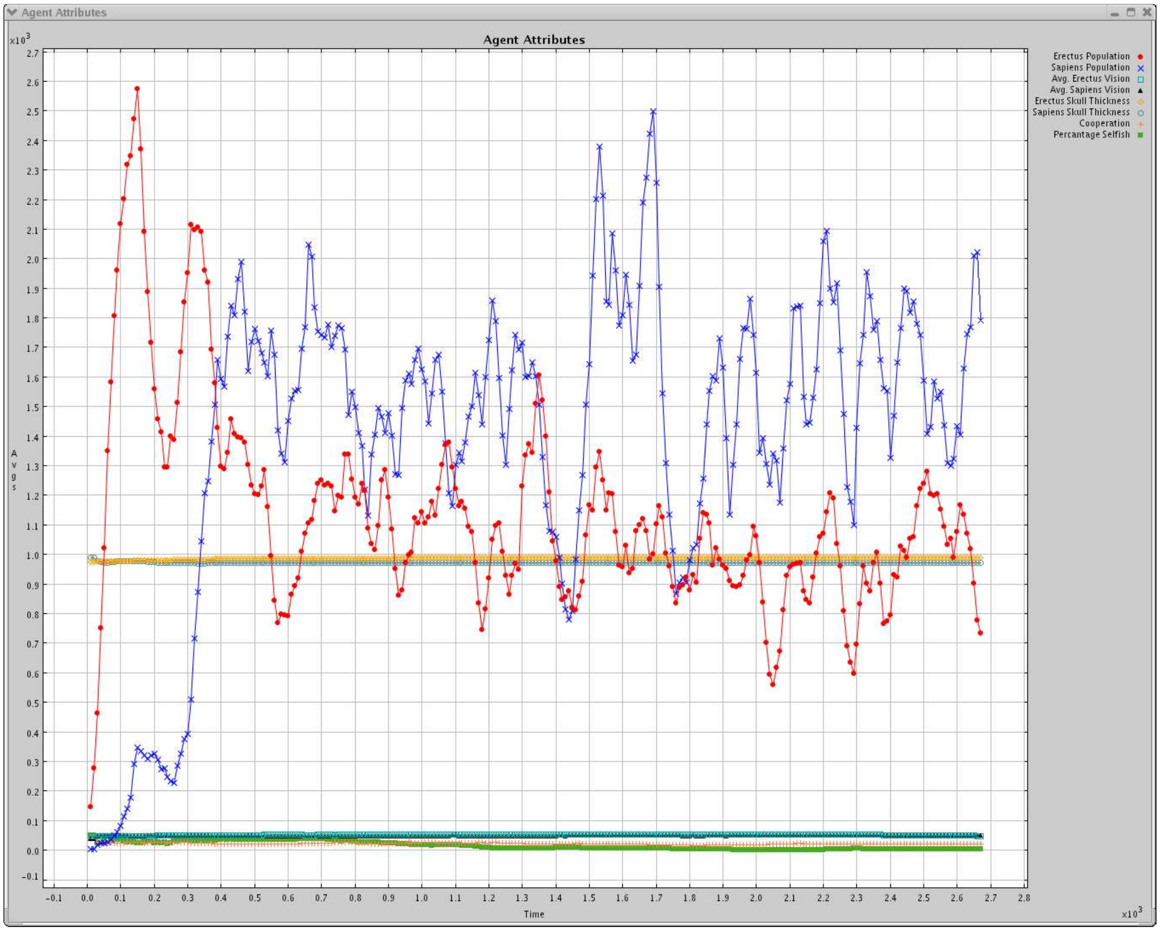


Figure 8

According to this graph, after the Sapiens are introduced, there is a rapid decline in the number of Erectus'. But after a while, both populations seem to oscillate around certain points, with the Sapiens population taking over and exceeding the Erectus' although it starts off much smaller.

Sapiens' vision slightly increased

Cooperation fluctuated around 22.

Selfishness increased first and then dramatically decreased to almost 0 in the end.

Erectus skull thickness has increased from 975 to 990.

Erectus vision hasn't changed.

We then tried to see if increasing the initial populations of both species by a magnitude would change the population dynamics and species' evolution.

Simulation run # 2:

Variables:

Initial population of Homo Erectus : 1000

Initial population of Homo Sapiens: 100

Homo Sapiens Introduced At Turn: 10

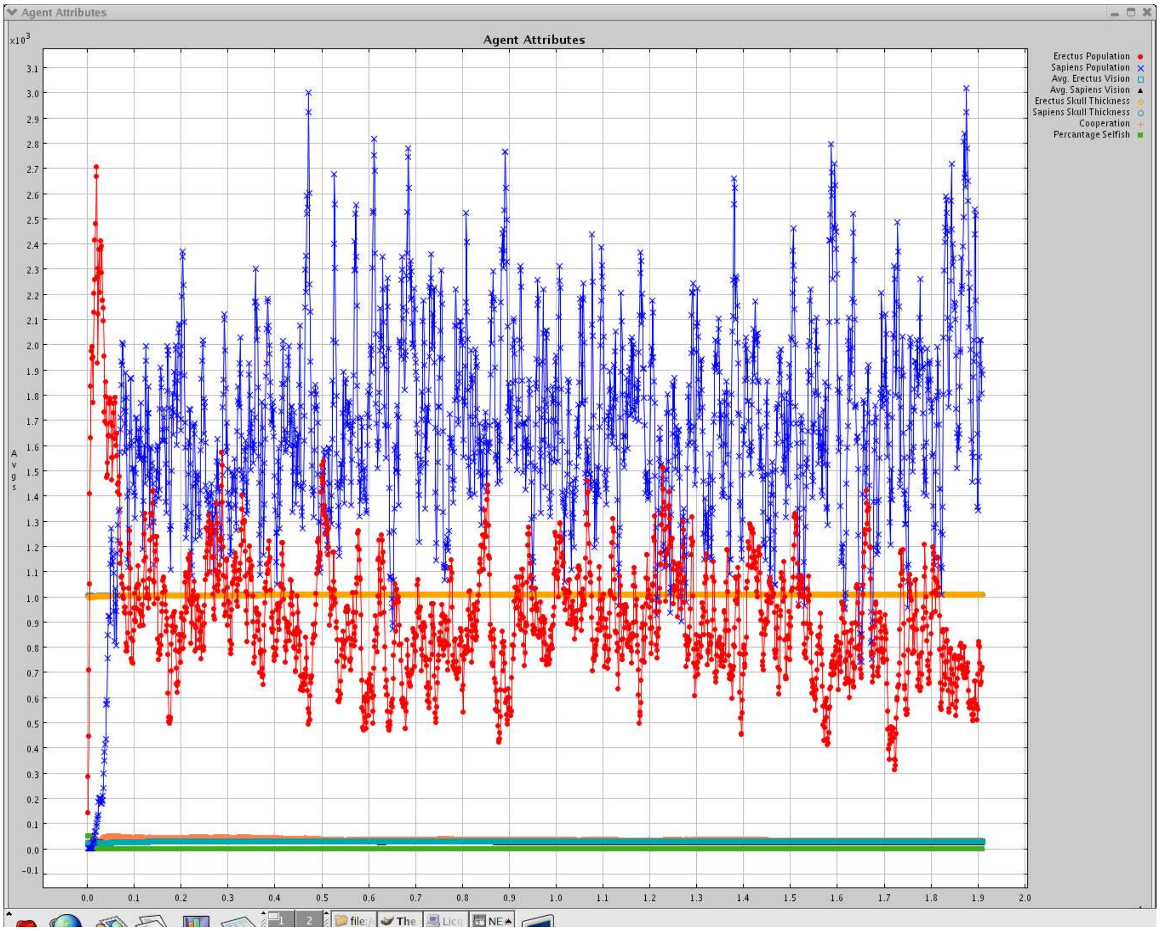


Figure 9
As seen in the data file, the Sapiens' vision has increased from 21 to 24 between the 100th and last turns.
The Erectus' vision increased from 23 to 30.
Cooperation value has not changed from 34.
Selfishness has decreased from 18 to 0.
Erectus skull thickness has increased from 999 to 1007.

Next, we introduce the sapiens at a much later turn, the 1000th as opposed to 10th and 20th in the previous two runs. This will let the erectus' populate the environment before the appearance of sapiens and will thus make it harder for the sapiens to find food and prosper.

Simulation Run #3
Variables:
Initial population of Homo Erectus : 200
Initial population of Homo Sapiens: 10
Homo Sapiens Introduced At Turn: 1000

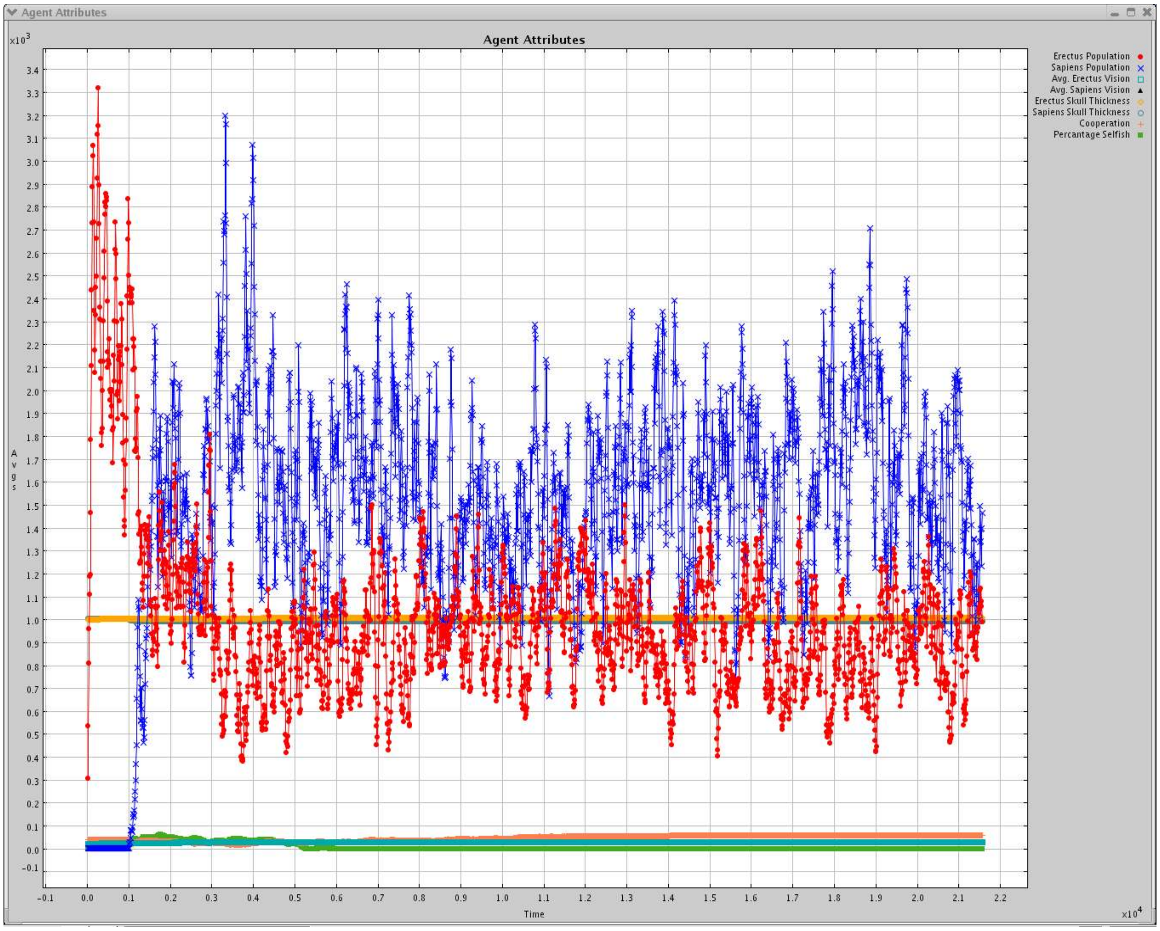


Figure 10

Erectus vision: Start:23 End: 30.
Sapiens Vision: Start: 24 End: 31.
Cooperation: Start: 32 End : 58.
Selfishness: Start: 27 End: 0.
Erectus skull thickness: Start: 999, End: 1007.

The next run is a repeat of run # 2 with the sapiens introduced at turn 20 instead of 10.
Simulation Run #4
Variables:
Initial population of Homo Erectus : 1000
Initial population of Homo Sapiens: 100
Homo Sapiens Introduced At Turn: 20

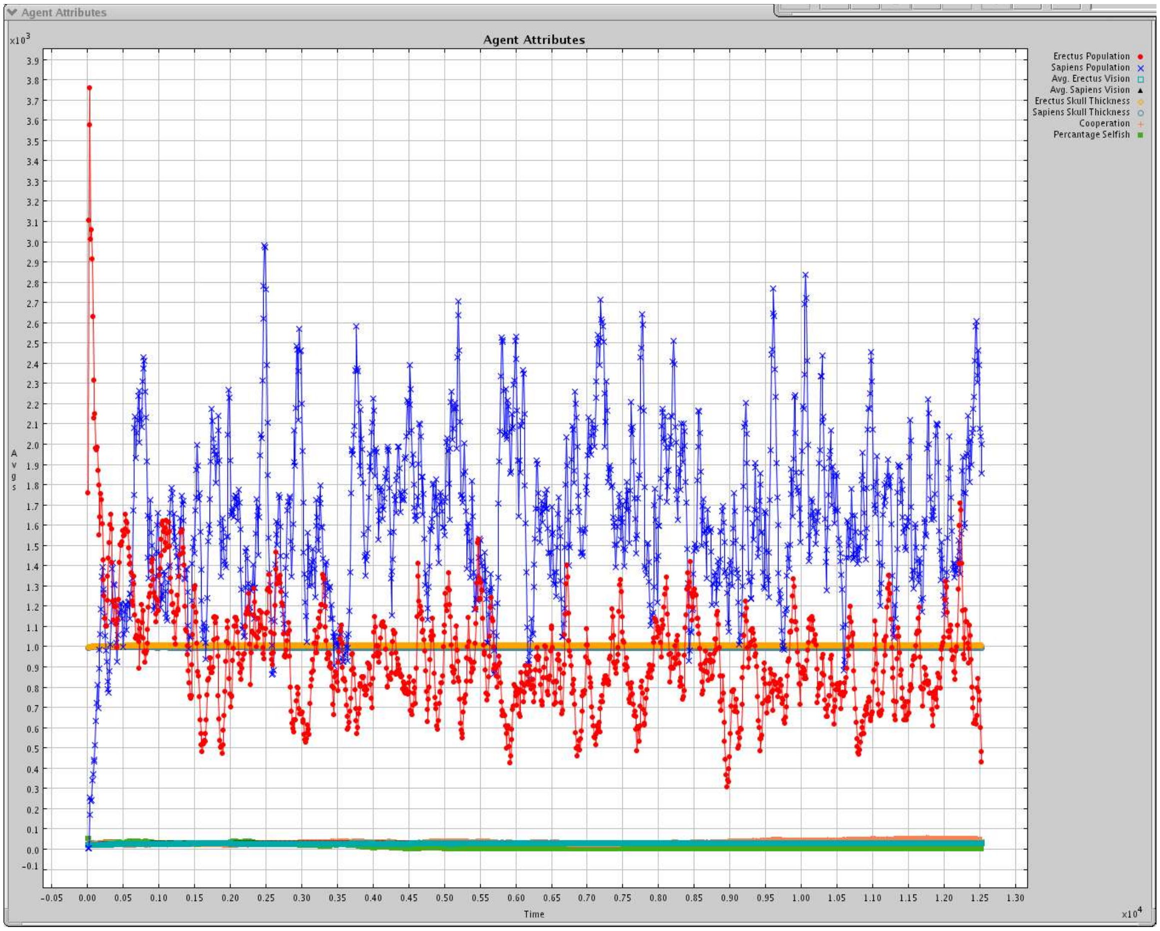
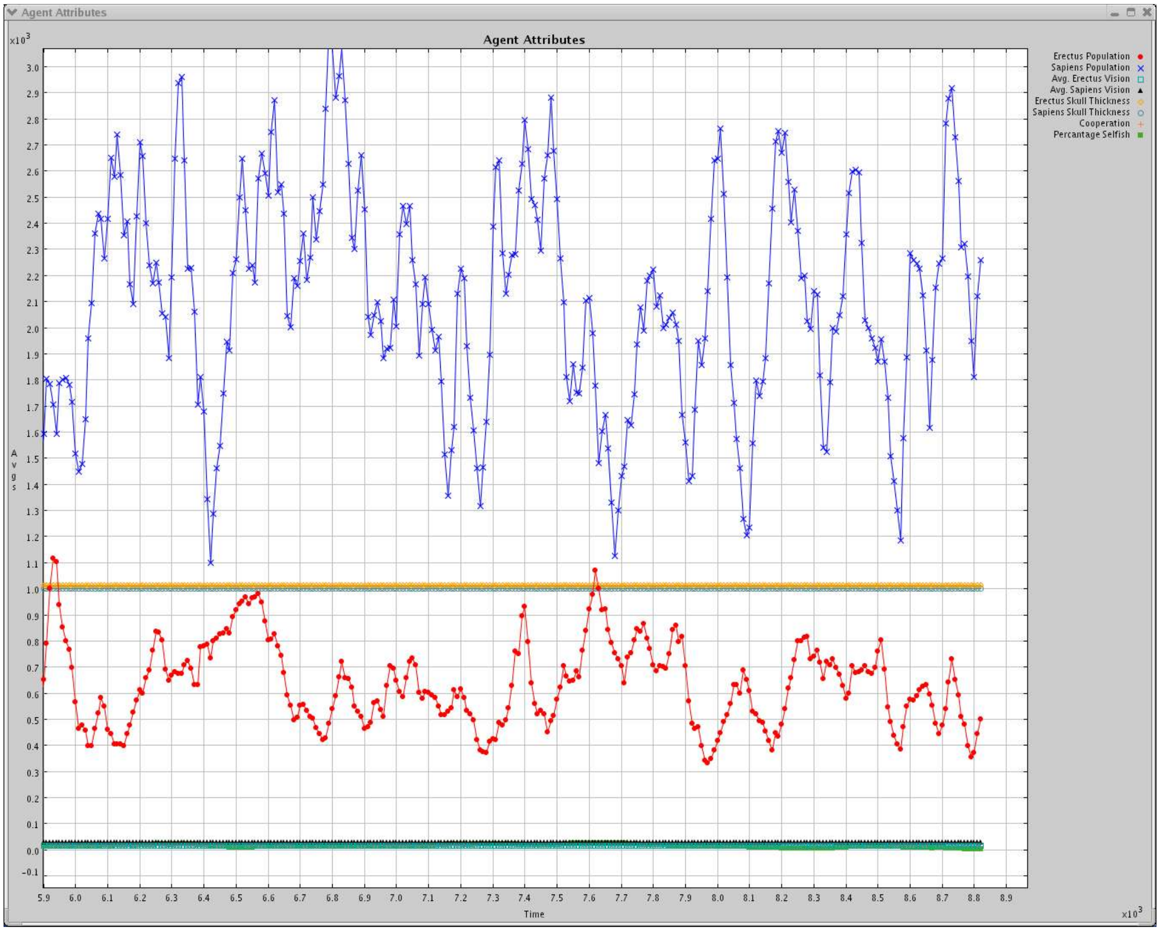


Figure 11
Erectus vision: 23 -> 31
Sapiens Vision: 23 -> 27
Cooperation: 27 -> 47
Selfishness: 29 -> 0
Erectus Skull Thickness: 999 -> 1007

Next we gave the erectus a certain disadvantage, namely their minimum possible vision was lowered to 8 from 16.

Simulation run # 5:
Variables:
Initial population of Homo Erectus : 100
Initial population of Homo Sapiens: 10
Homo Sapiens Introduced At Turn: 100
Erectus minimum vision: 8
Sapiens minimum vision: 16

When the visions of erectuses are started lower when compared to the sapiens, the population



difference between the sapiens and the erectuses get larger.

Figure 12
Erectus vision: 11 -> 15
Sapiens vision: 28 -> 28
Cooperation: 28 ->23
Selfishness: 5->4
Erectus skull thickness: 1011 -> 1015

With the same handicapped erectus, we introduced the sapiens at a later turn this time, so as to give the erectus more time to grow. We also made the selfishnessgene dominant so a homo sapiens will now be selfish if it has only one 1 in its gene.

Simulation run # 6:
Variables:
Initial population of Homo Erectus : 100
Initial population of Homo Sapiens: 10

Homo Sapiens Introduced At Turn: 300
Erectus minimum vision: 8
Sapiens minimum vision: 16

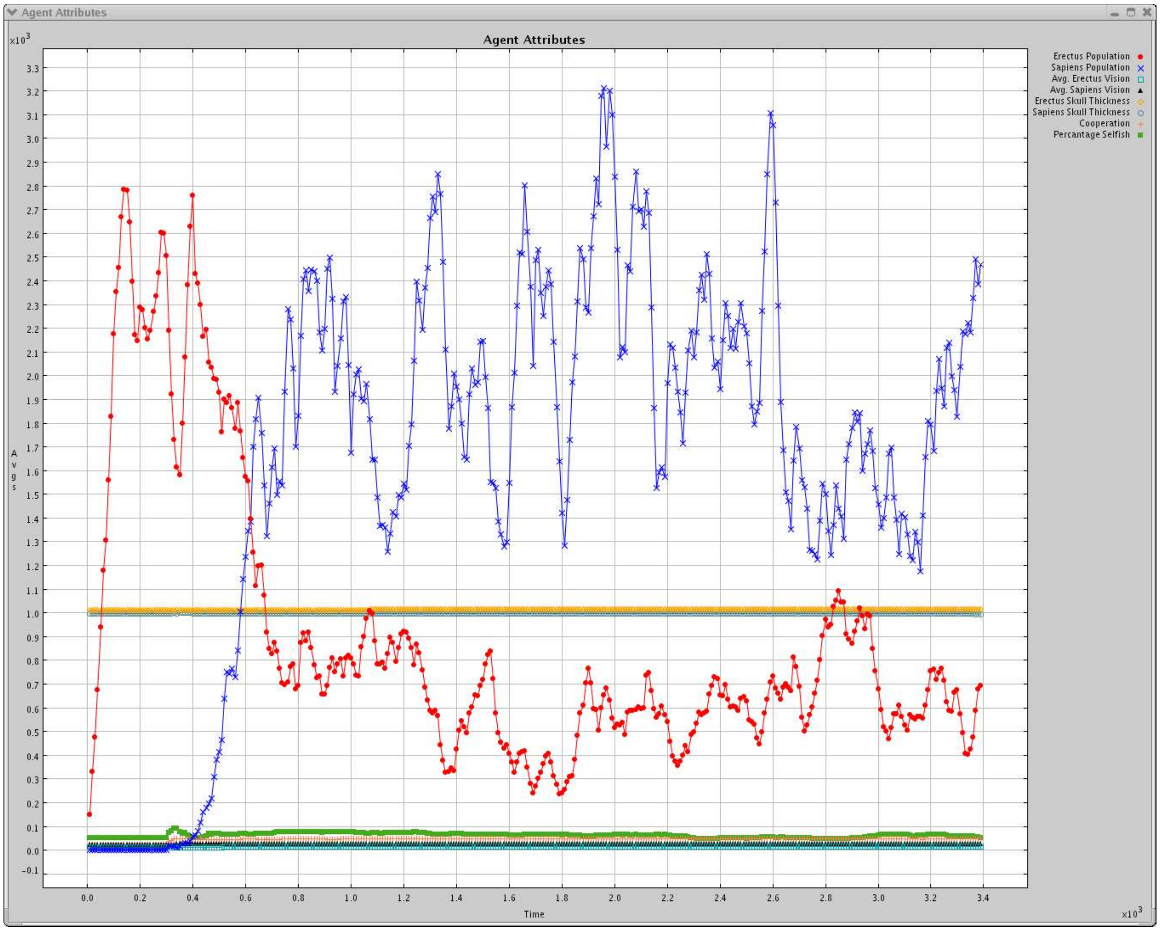


Figure 13
Erectus vision: 11 -> 15
Sapiens Vision: 23 -> 25
Cooperation: 35 -> 46

Here is the same plot zoomed in to the cooperation, selfishness and vision values of the last turns:

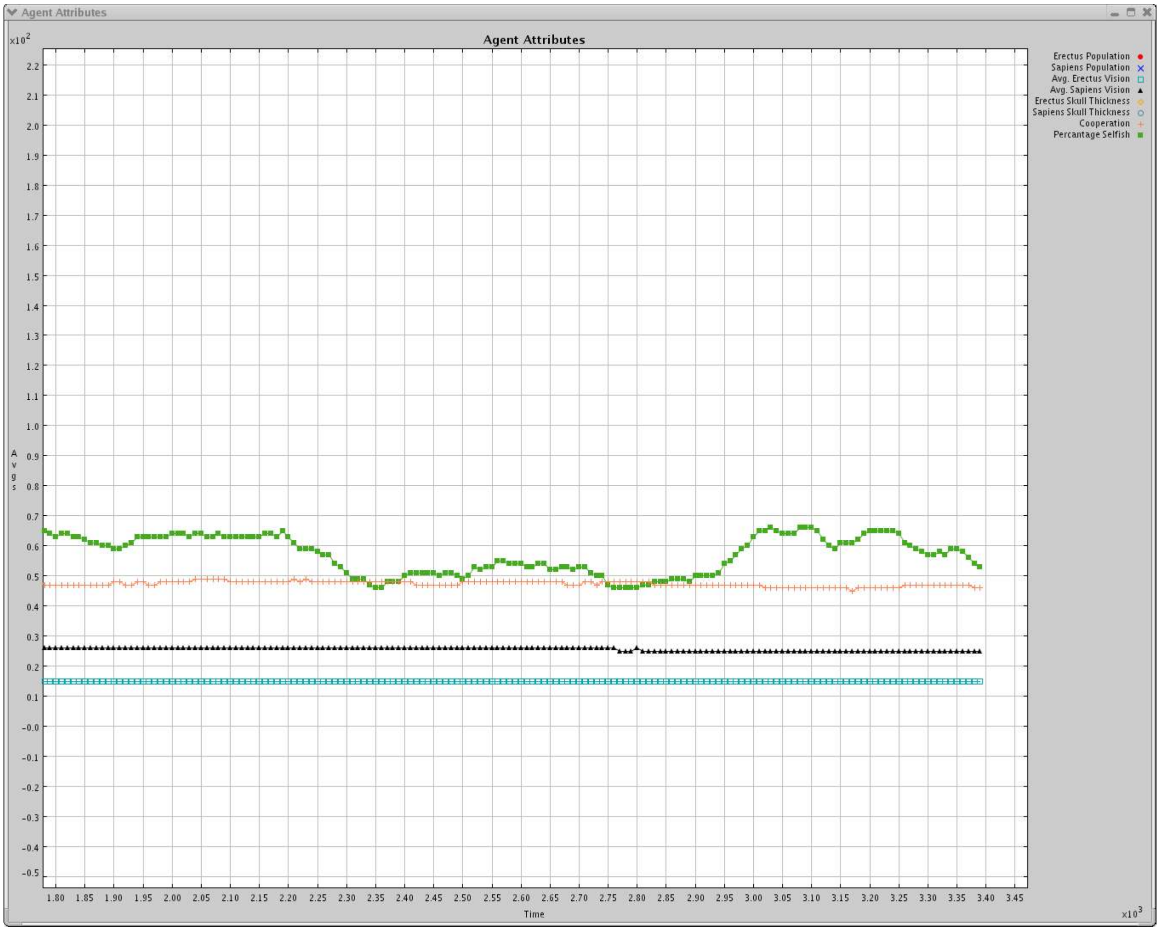


Figure 14

The selfishness in the population does not seem to have changed a great deal.
Erectus skull thickness: 1011 -> 1015
Sapiens skull thickness: 999 -> 995

In this run we lowered the maximum sugar an agent can be born with from 100 to 30.
Simulation run # 7:
Variables:
Initial population of Homo Erectus : 100
Initial population of Homo Sapiens: 10
Homo Sapiens Introduced At Turn: 400
Erectus minimum vision: 16
Sapiens minimum vision: 16

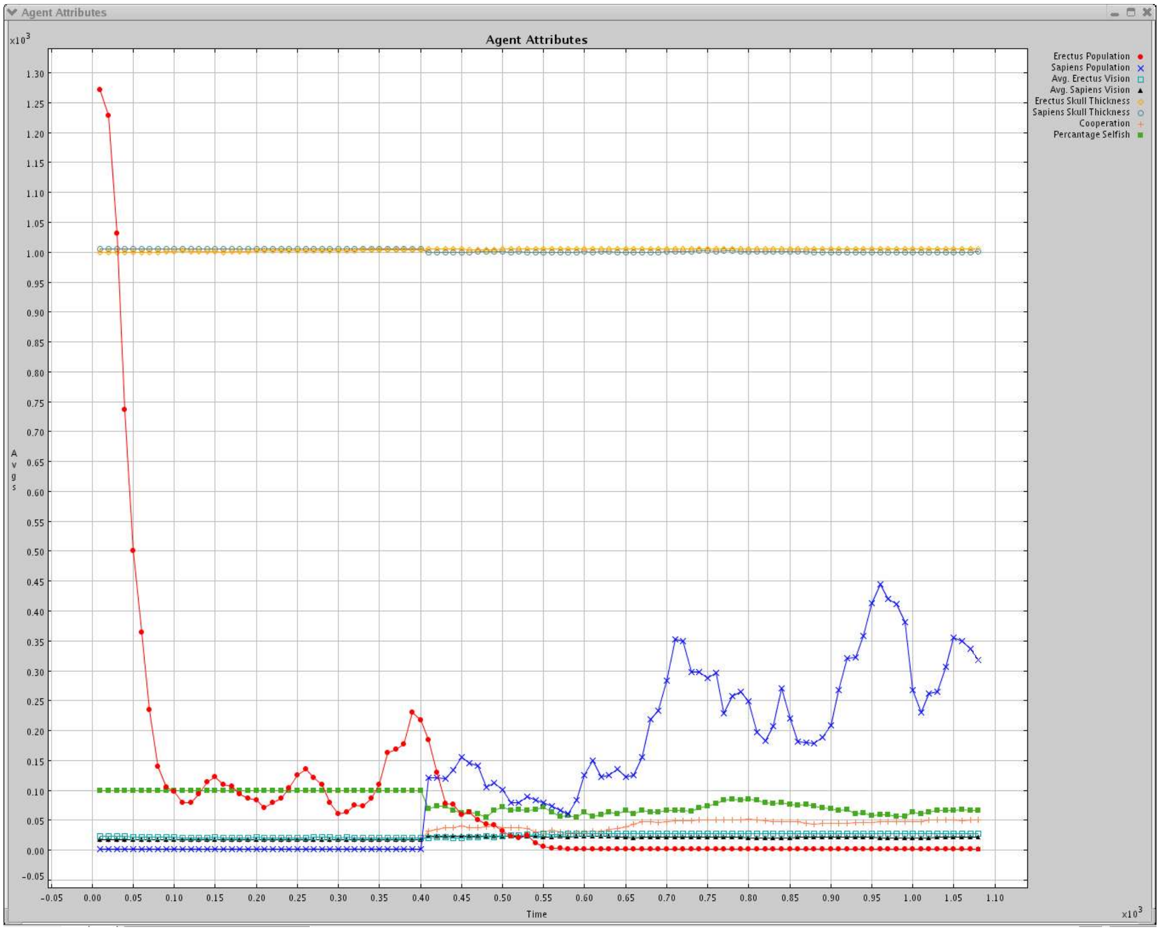
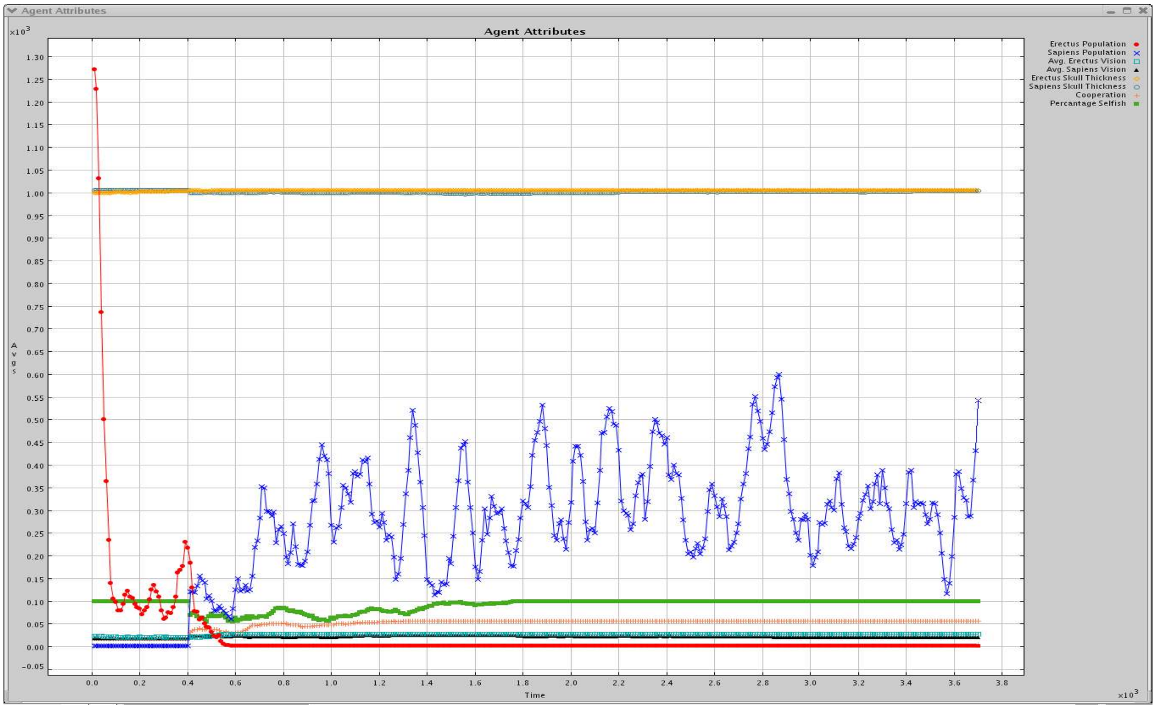


Figure 15
this part of the simulation clearly shows that the Erectus died out after the introduction of Sapiens to the scene.

Figure 16



This is the final situation with the sapiens' fluctuating. Selfishness has increased.

Results:
Erectus vision: 23 -> 28.
Sapiens vision: 23 ->25 -> 21
Cooperation: 32 -> 57
Selfishness: 69 ->100
Erectus skull thickness: 999 -> 1006

Simulation run # 8:
In this simulation we have lowered the maximum cooperation value that the first generation starts with two-fold in order to see more increase in the cooperation value. We thought that maybe in the previous cases the cooperation of the agents was enough so it did not tend to increase much.
Variables:
Initial population of Homo Erectus : 1000
Initial population of Homo Sapiens: 100
Homo Sapiens Introduced At Turn: 1000
Erectus minimum vision: 16
Sapiens minimum vision: 16
Cooperation maximum: 31
Selfishness: dominant

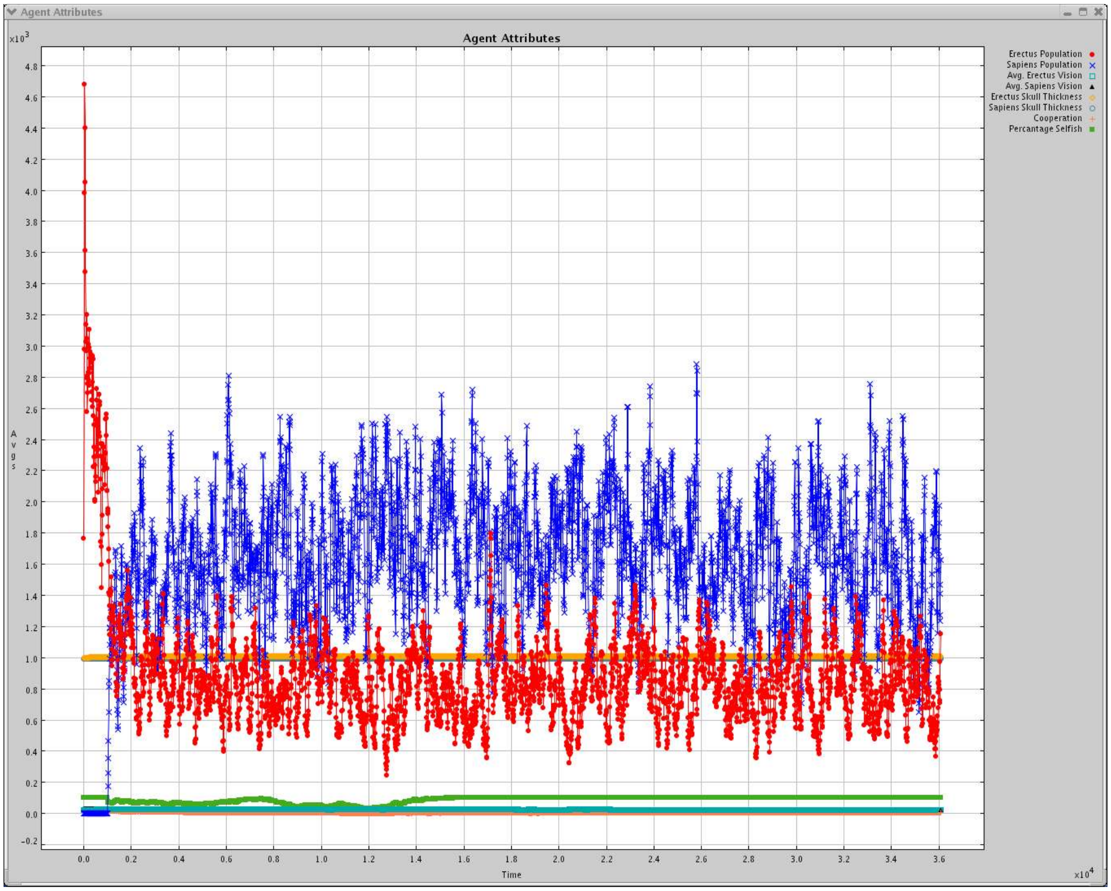


Figure 17
Erectus vision: 23 -> 24
Sapiens vision: 23 -> 24
Cooperation: 16 -> 8
Percentage Selfish 70 -> 100
Erectus skull thickness: 999 -> 1007
Sapiens skull thickness: 999 ->992
here is the last part of the graph in close up:

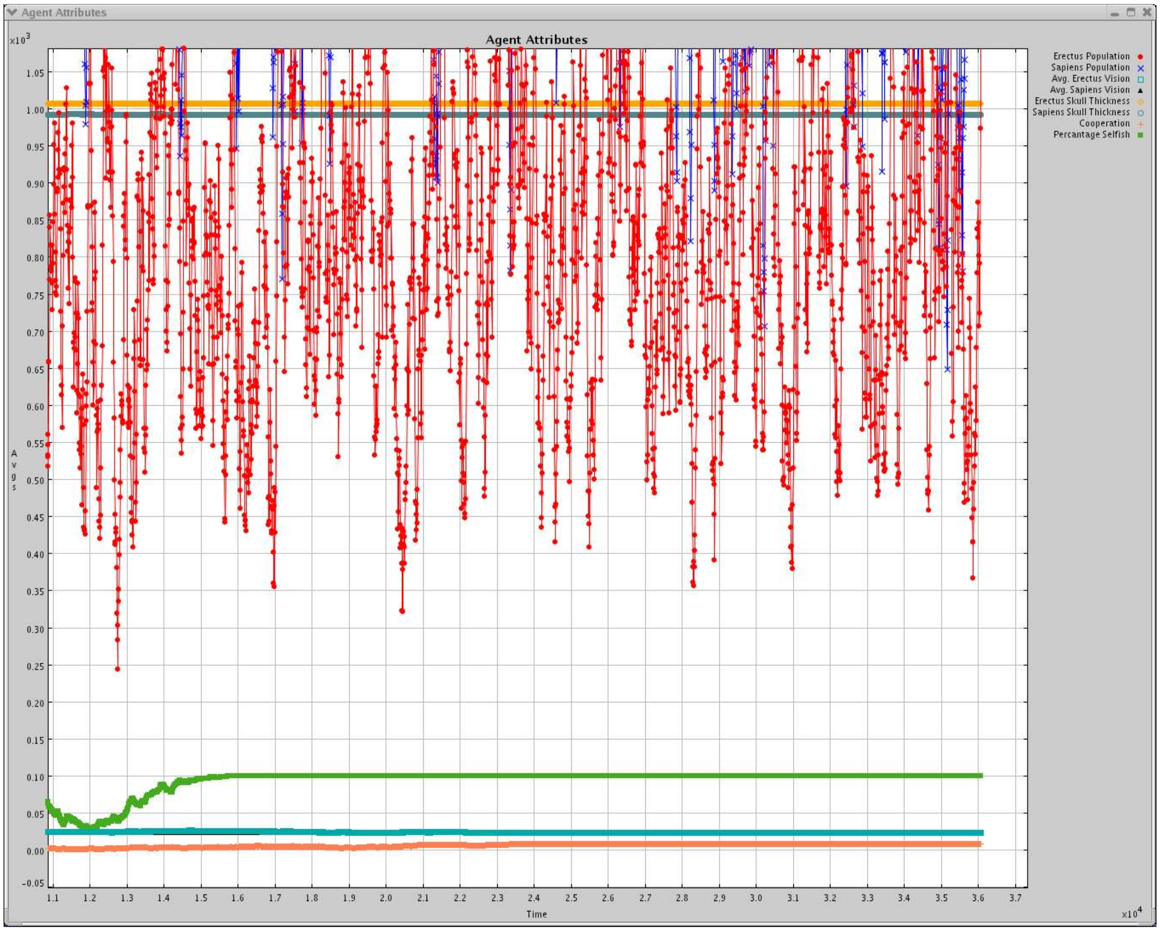


Figure 18

Simulation run # 9:
In this simulation we have lowered the minimum vision to 8 (and maximum to 15) to see if this causes more increase in the agents' vision.
Variables:
Initial population of Homo Erectus : 1000
Initial population of Homo Sapiens: 100
Homo Sapiens Introduced At Turn: 1000
Erectus minimum vision: 8
Sapiens minimum vision: 8
Cooperation maximum: 63
Selfishness: dominant

Figure 19
this shows the initial oscillations

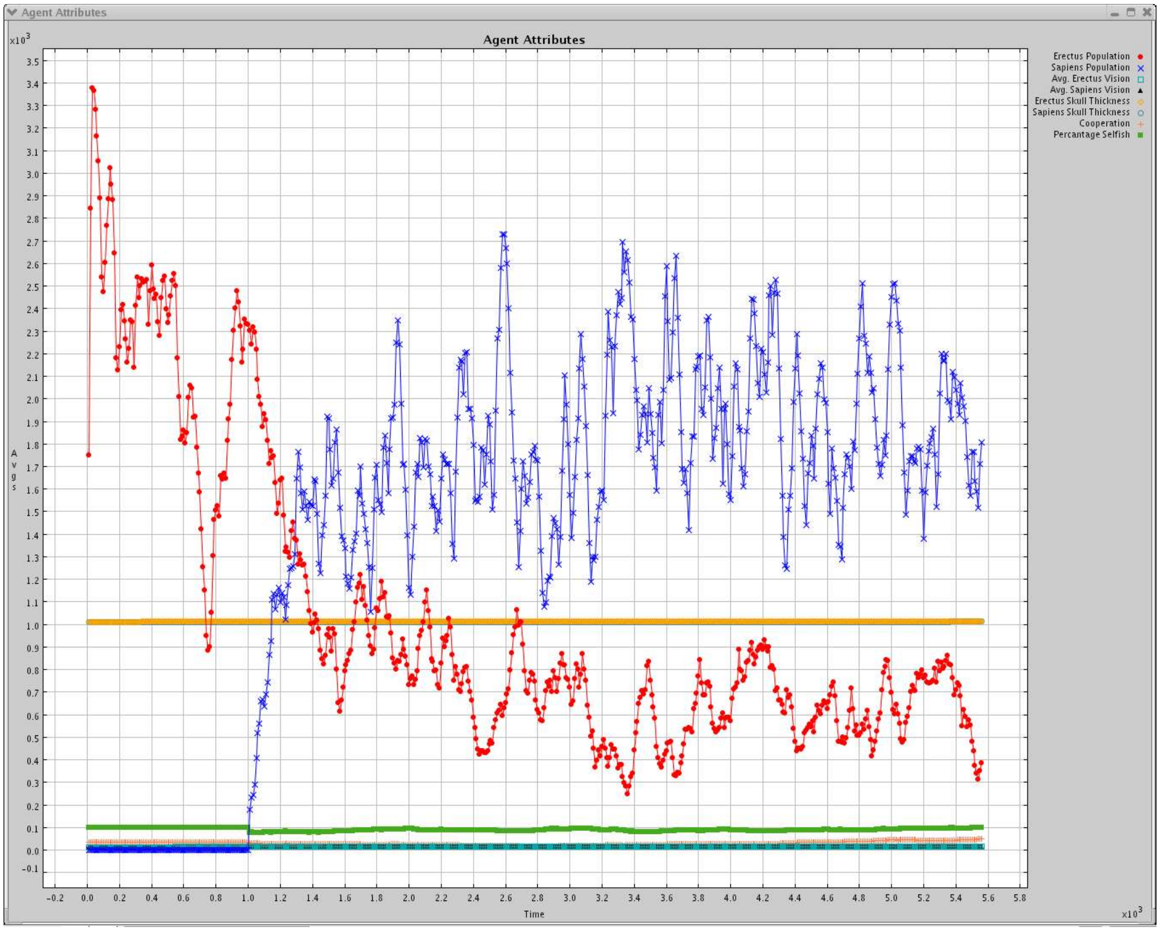
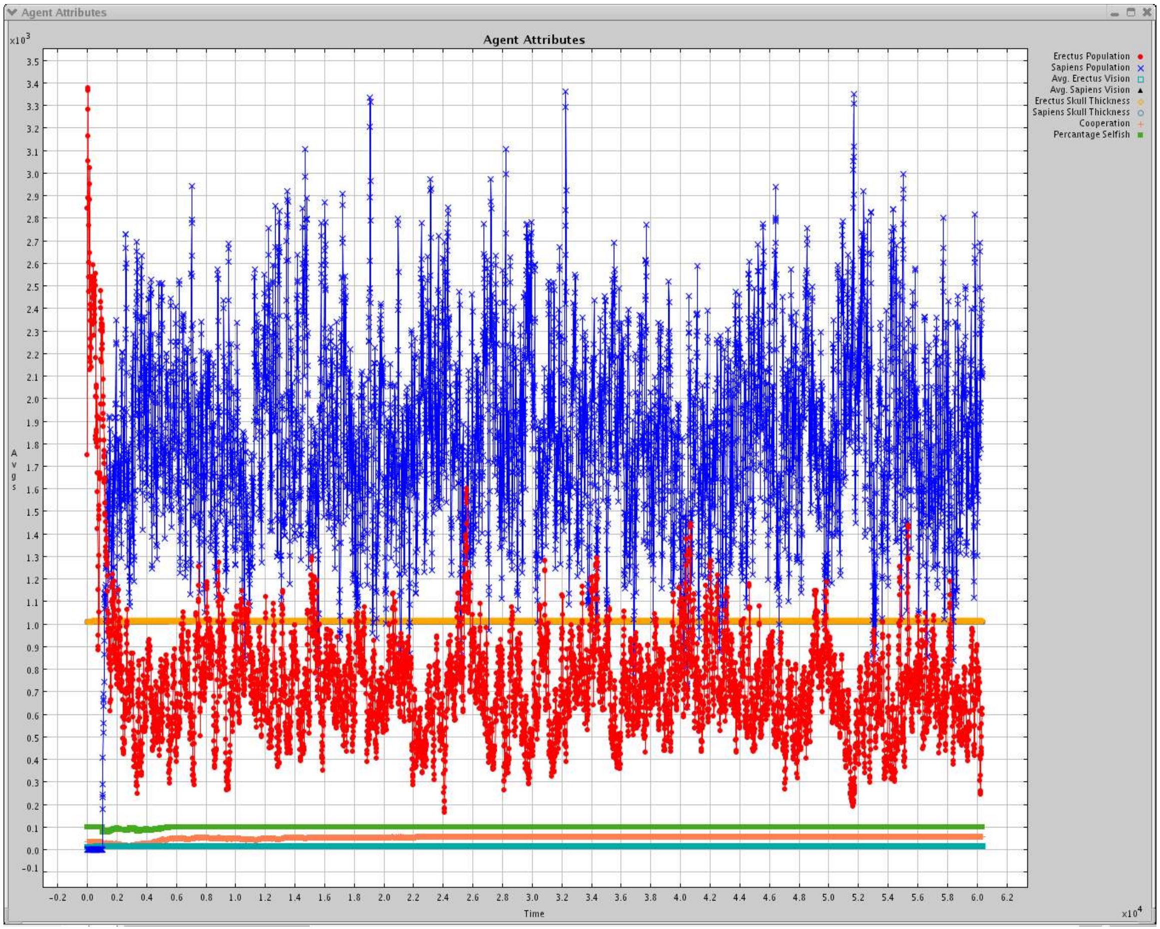


Figure 20
this shows the simulation after it has stabilized
Erectus vision: 19 -> 15
Sapiens vision: 11 -> 15
Cooperation: 31 -> 58
Selfishness(%): 75 -> 100
Erectus skull thickness: 1003-> 1015
Sapiens skull thickness: 1011 -> 1010

Simulation run # 10:
In this simulation we have lowered the minimum vision to 8 and kept the maximum at 31 to allow for more room for agents' vision to grow.


Agent Attributes
Agent Attributes
x10^3
3.5
3.4
3.3
3.2
3.1
3.0
2.9
2.8
2.7
2.6
2.5
2.4
2.3
2.2
2.1
2.0
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-0.1
Avgs
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0
2.2
2.4
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0
4.2
4.4
4.6
4.8
5.0
5.2
5.4
5.6
5.8
6.0
6.2
Time
x10^4
Erectus Population ●
Sapiens Population ×
Avg. Erectus Vision □
Avg. Sapiens Vision ▲
Erectus Skull Thickness ◆
Sapiens Skull Thickness ○
Cooperation +
Percantage Selfish ■

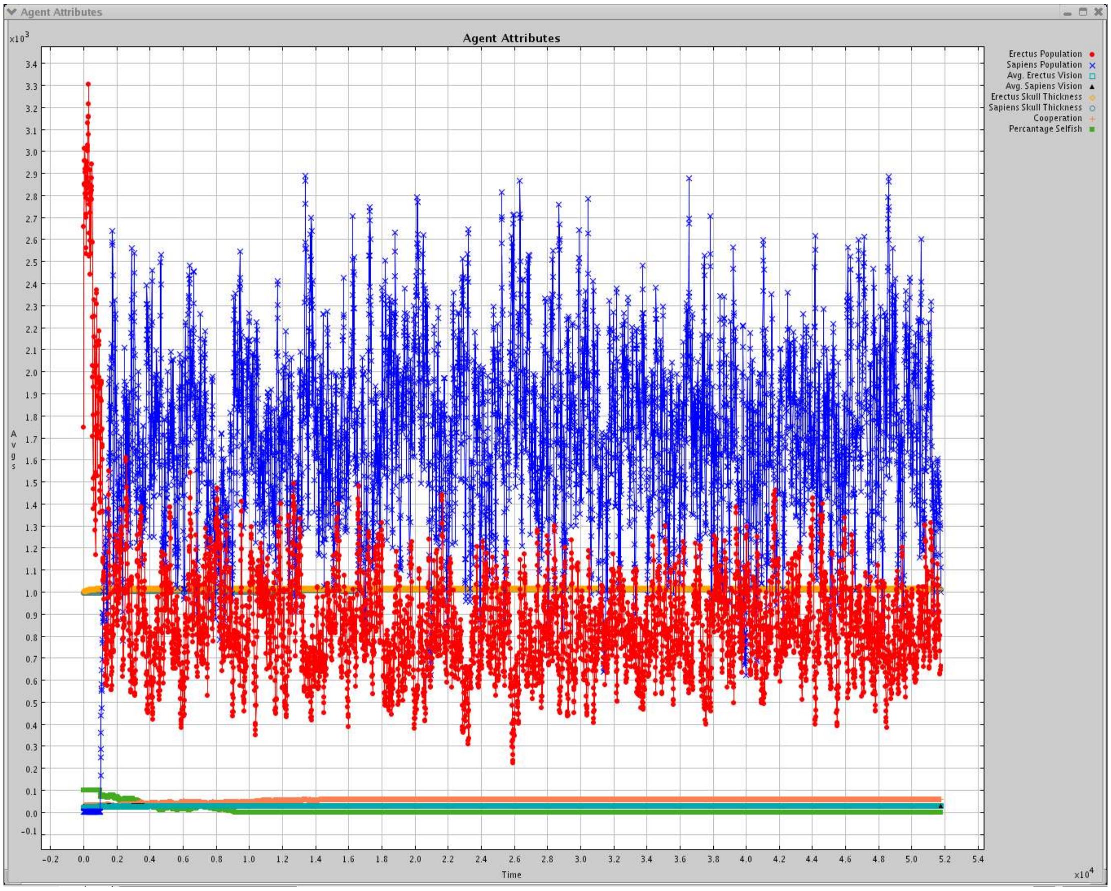


Figure 21
Initial population of Homo Erectus : 1000
Initial population of Homo Sapiens: 100
Homo Sapiens Introduced At Turn: 1000
Erectus minimum vision: 8
Sapiens minimum vision: 8
Erectus maximum vision: 31
Sapiens maximum vision: 31
Cooperation maximum: 63
Selfishness: dominant

Erectus vision: 19 -> 28
Sapiens vision: 19 -> 29
Cooperation: 32 -> 59
Selfishness(%): 74 -> 0
Erectus skull thickness: 1003-> 1015
Sapiens skull thickness: 1003-> 1011

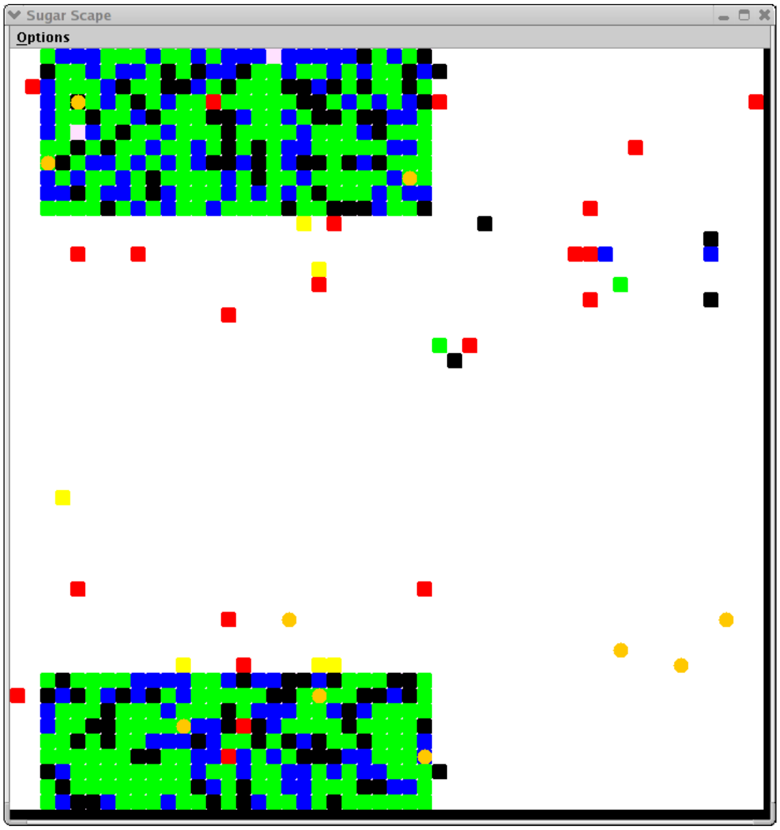


Figure 22. Here is how the map looks at the simulation's end:

Results of applying the Diet Theory:
So, the Erectus skull thickened due to fighting.Would it also have thickened due to their food if there was no fighting? Would it have thickened more in addition to the one due to fighting? We tested these and other questions next.
Simulation run #11:
Requiring Homo Erectus to have thick skulls in order to chew their food.

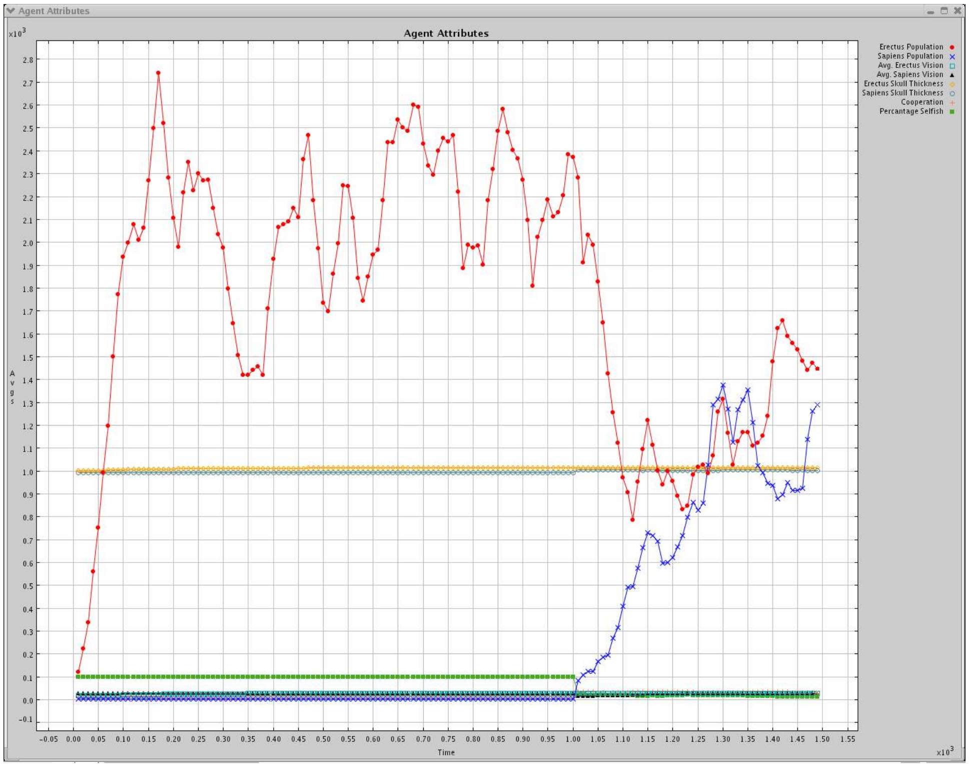


Figure 23. The initial situation. Erectus skull thickness has increased.

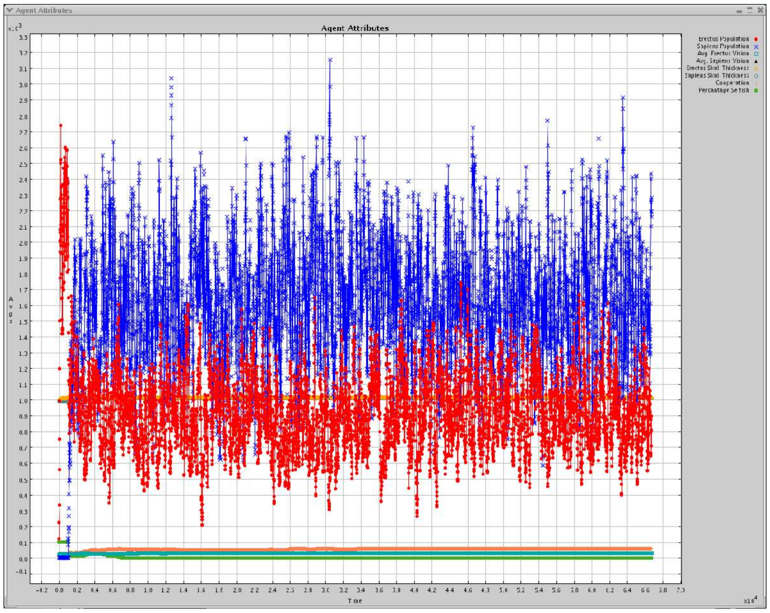


Figure 24. The overall situation.

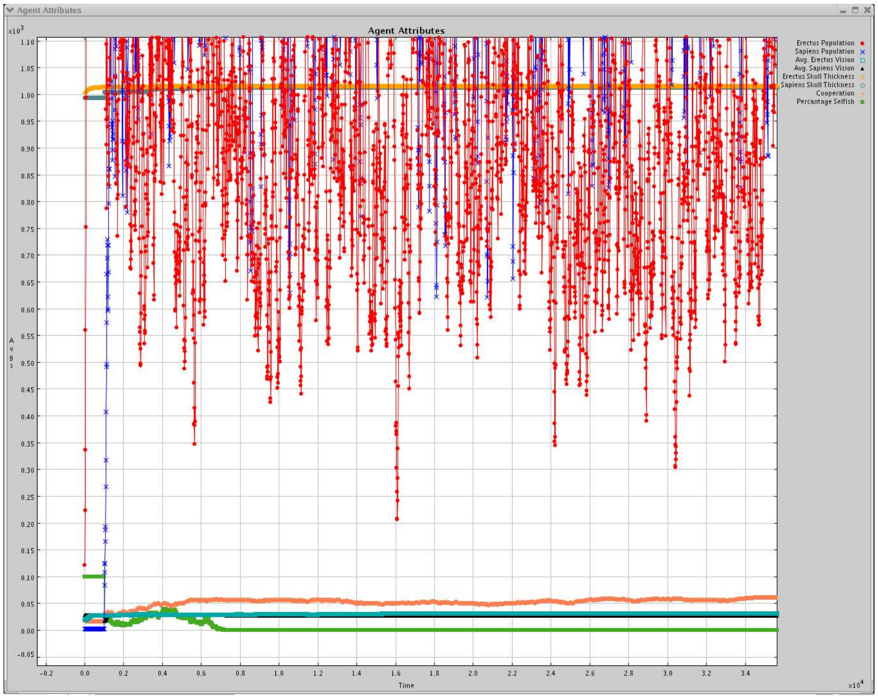


Figure 25. Changes in attributes shown in detail.

Erectus vision: 19 -> 31
Sapiens vision: 17 -> 26
Cooperation: 33 -> 61
Selfishness: 22 ->0
Erectus Skull Thickness: 1003 -> 1015
Sapiens Skull Thickness: 1005 ->1013

Simulation Run # 12:
Next we took out the fighting from the simulation so the only pressure on Erectus skull growth was their diet.

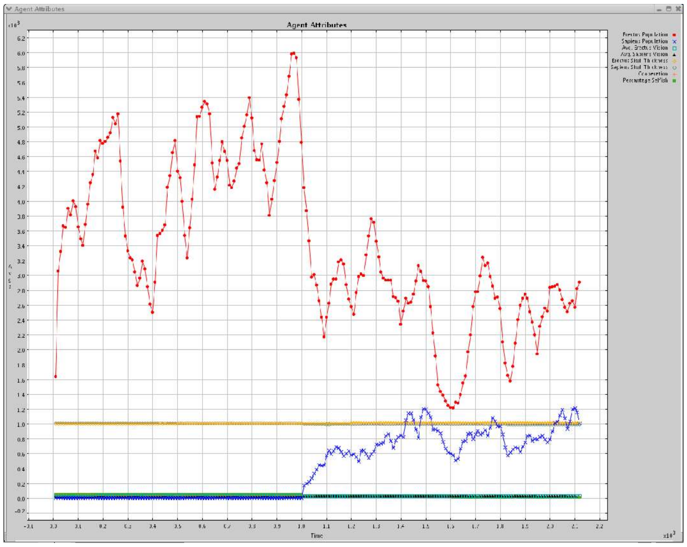


Figure 26. The initial situation.

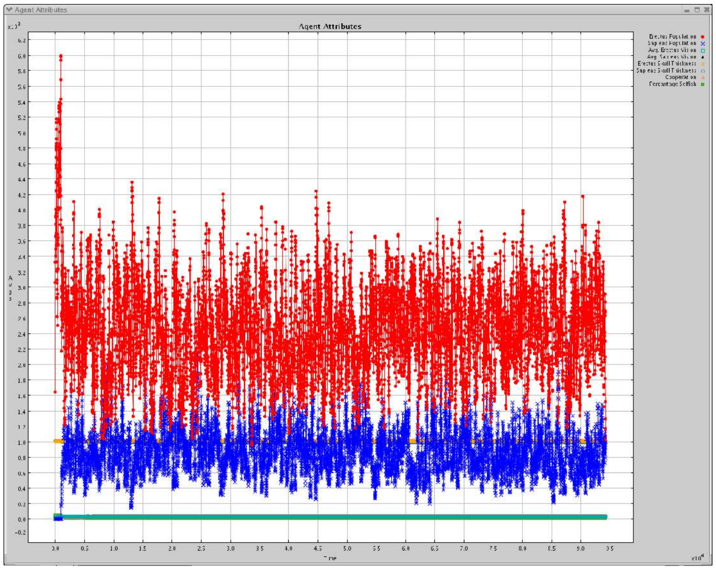


Figure 27. The overall situation. Removing the intra-erectus fighting has given the erectus' a huge advantage.

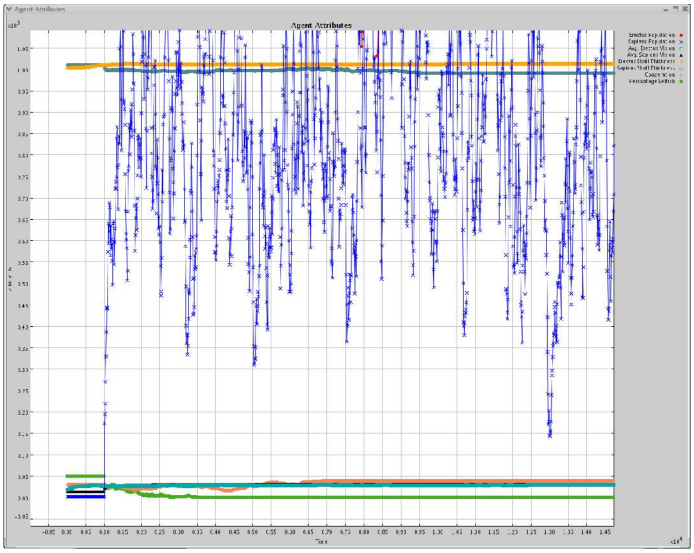


Figure 28. The attributes shown in detail.

Erectus vision: 19 -> 27
Sapiens vision: 19 -> 28
Cooperation: 33 -> 20
Selfishness: 23 ->6
Erectus Skull Thickness: 1003 -> 1012
Sapiens Skull Thickness: 1003 ->995

Simulation Run #13
With the same parameters, we now remove the hard-to-chew diet from the simulation, leaving the Erectus' with only fighting as a means of skull enlargement. So now we will be able to compare the affects of the two separately.

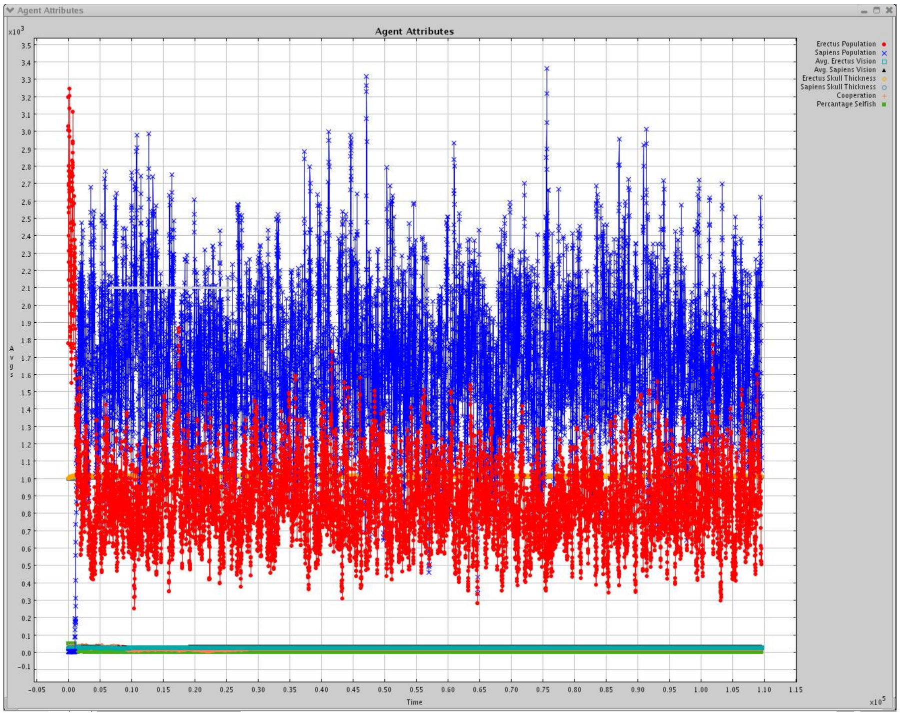


Figure 29. The overall situation.

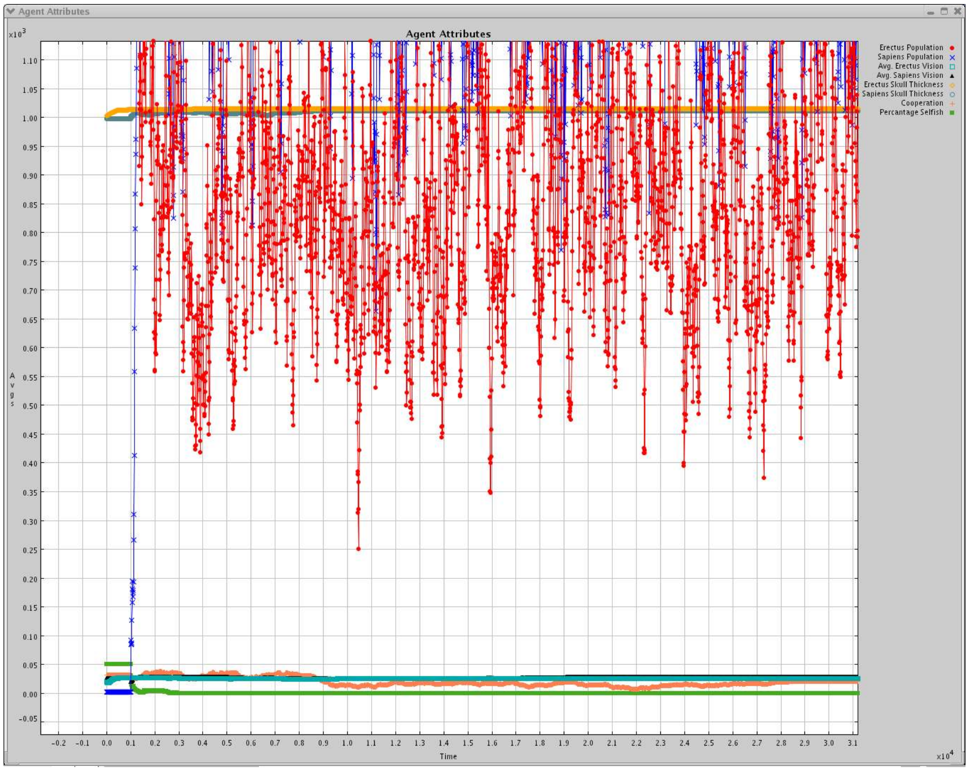


Figure 30. The attributes shown in detail.

Erectus vision: 19 -> 25
Sapiens vision: 19 -> 28

Cooperation: 28 -> 21
Selfishness: 31 ->0
Erectus Skull Thickness: 1003 -> 1015
Sapiens Skull Thickness: 1003 ->1011

Simulation Run # 14
We have seen an increase in the skull thicknesses so far but we have to run a control case to see if these are really due to the fighting or the diet. This is the control run.

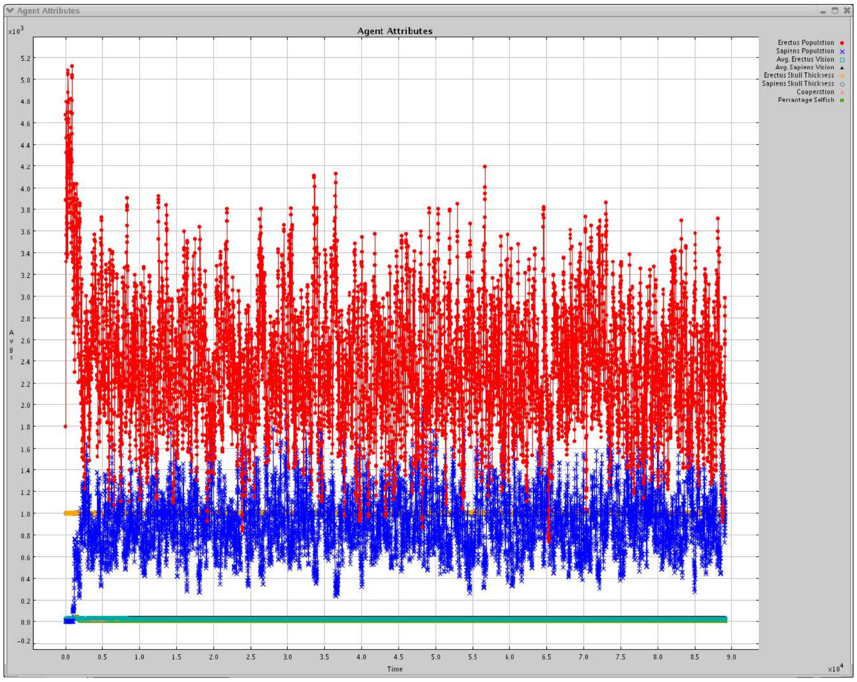


Figure 31. The overall situation. Mark the change of colors. With no handicap, the erectus' never give the upper hand to the sapiens' since the sapiens' start later in a map full of erectus'.

Erectus vision: 19 -> 25
Sapiens vision: 19 -> 28
Cooperation: 30 -> 23
Selfishness: 29 ->0
Erectus Skull Thickness: 1003 -> 1010
Sapiens Skull Thickness: 1003 ->993

Simulation Run # 15
To see if selfishness disappears even when dominant, we repeated our experiment with only the diet as the skull enlarging factor (simulation run # 12) with the selfishness gene made dominant.

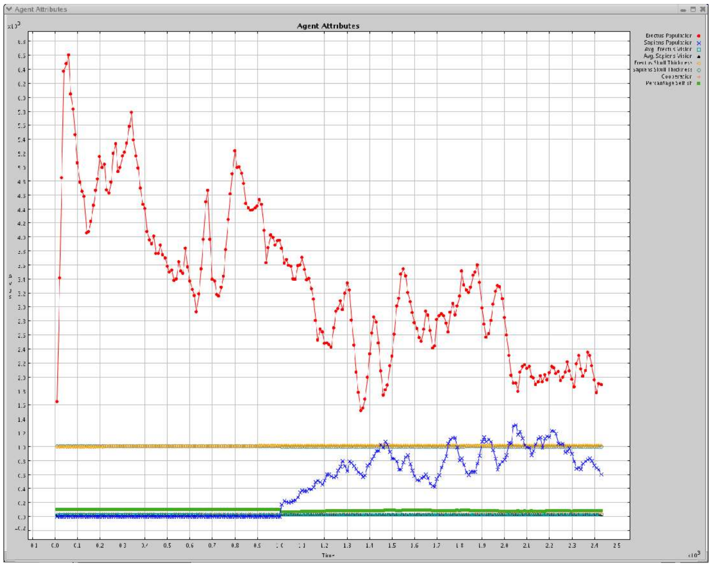


Figure 32. The initial situation.

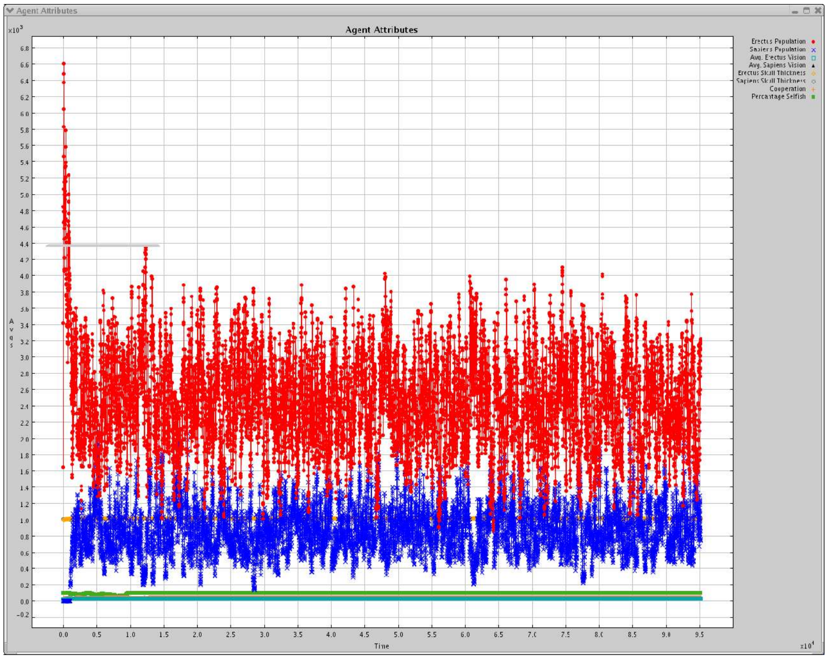


Figure 33. The overall situation. Mark the inability of the sapiens' to surpass the erectus' in number.

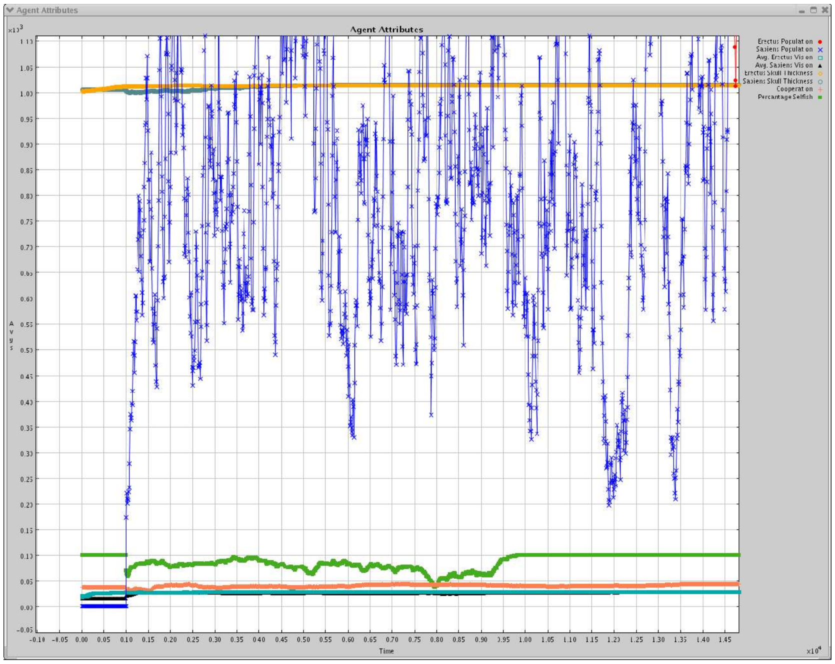


Figure 34. Some of the attributes shown in detail.

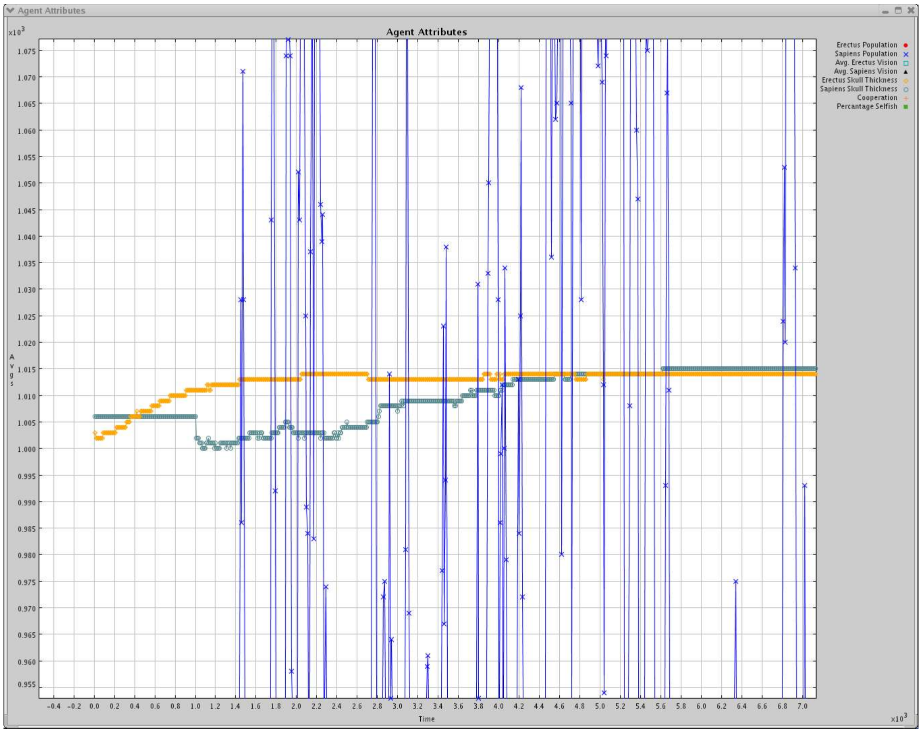


Figure 35. Changes in the skull thicknesses of both populations shown in detail.

Erectus vision: 19 -> 29
Sapiens vision: 20 -> 29
Cooperation: 33 -> 44
Selfishness: 70 ->100

Erectus Skull Thickness: 1003 -> 1014
Sapiens Skull Thickness: 1002 ->1015

DISCUSSION OF THE SIMULATION RESULTS

As we see from the simulation results, in all cases except for the last two, after the introduction of the Sapiens into the environment, a considerable decline in the population of the HomoErectus occurs although the order of the stepping of HomoSapiens and HomoErectus is random which leads to fairness. If we had made the Sapiens step first in all turns, that would make the Sapiens much more advantageous in getting the food as they would be the first to see and go to the sugar spots. In all the simulations we performed, although we introduce fewer Sapiens than Erectus, Sapiens eventually take over HomoErectus and there occurs a huge gap between the number of two kinds. It is also noted that when we decrease the minimum initial sugar of the Sapiens, the rate of increase in the population decreases.

When we remove both handicaps from the Erectus, however, the Sapiens' cannot take over the the Erectus population. This also happened when we removed only the intra-erectus fighting. This means that mortal combat for females was a huge disadvantage for the erectus' population. On the other hand, both the fighting and the diet cause an increase in the erectus' skull thickness. Erectus skull thickness always increases while that of sapiens' either stays the same or decreases.

The fact that the erectus skull thickness has still increased in the control run should not dissuade us from trusting our results because as a general trend, the erectus skull thickness has always increased and that of sapiens' has generally decreased due to the inverse proportion of the vision beginning vision values with skull thickness. We link the unexplainable increase in erectus skull thickness in the control run to the elimination of thin skulled individuals in the beginning of the simulation by chance. They perhaps turned out to be the ones given the least sugar when they were born.

Looking at simulations 11, 12 and 13, we can conclude that fighting has a greater affect on skull enlargement than does diet.

Cooperation values of Homo Sapiens agents grows significantly in almost all simualations, allowing us to conclude that cooperation, which is better for the group and for the individual is selected for.

Another important observation that we made is that the change in the vision of the agents changes the population dynamics. When we decrease the initial minimum vision of the HomoErectus' by half for instance and keep the vision of the Sapiens unchanged, the final population difference between the Sapiens and the Erectus gets much larger when compared to the other simulations where we started the two kinds with the same vision.

It is also observed that the skull thickness of the HomoErectus increases which verifies that fighting for mating favors the increase in skull thickness.

In some simulations, it is also recorded that the cooperation value in the Sapiens does not increase significantly which can be explained with the fact that there is already plenty of sugar sources in the environment so that being able to hunt animals and predators does not effect the fitness of the agents in the current situations. However, it is worth noting that in the simulations in which we decrease the minimum vision or the minimum initial sugar, cooperation increases as Sapiens become more dependent on animals.

When we gave vision more leeway for growth, as in the last simulation run, we have seen that it and cooperation grew considerably. This shows that agents appreciate possibility of more vision if their vision is not close to 25 (which is the maximum they would ever need) in the beginning.

Another result that we got is about the recessive and dominant genes. Selfishness gene can be selected as either recessive or dominant and we observed that when the gene is recessive, in most simulations its value decreased to 0 whereas when it is selected to be dominant it either fluctuated or increased. In simulation 10 where the minimum and maximum vision values were farther apart, selfishness started out at around 75 % but fell down to 0 % and stabilized there, which forbids us from concluding that selfishness is selected. It is actually eliminated even though it was a dominant gene. We have seen it reach 100% when it was dominant and 0% when it was recessive but never have we seen it fall to 0 % when it was dominant. This is the first time we have seen selection *against* this gene. And cooperation clearly increases in this last simulation, which was a result we have seen in other runs too.

FUTURE IMPROVEMENTS

One of the major future improvements that can be made in this project is about the movements of the agents. As it is implemented in the RePast library, the agents can move and see only in the 4-directions which corresponds to lines. However this can be made more realistic by enabling the agents to see the whole area with a radius of their vision value instead of the four lines.

The second development would be for the agents to have a certain speed at which they can move so that instead of being teleported to the point they want to go, they would advance a certain amount in that direction, thus not their vision, but their speed would be the limit on their movement.

Finally, the map could have been made larger rather than a 50*50 square space. This could support more agents and larger sugar maps which could correspond to more realistic results. Moreover, the agents would be able to move more and be more like explorers in the case of a larger map since in the current map, after a point there is no space left for the agents to roam and they crowd together.

Another improvement that can be made is to have inter-agent communication which does not need to be agent to agent per say. If movement can be based on speed as mentioned above, then it could be useful for an agent to send a, say “food!” signal when it has found food and the others to change their directions towards the source of this signal if they get it. This signal can be given to the agents in the perimeter of the signal directly or it can be another pheromone that agents that are close enough can “smell”.

References:

An Agent Based Modelling Toolkit for Java.
<http://repast.sourceforge.net> [12 04 2004]

Boaz, T. & Ciochon, L. (2004). *Natural History Magazine: Headstrong Hominids.*
Retrieved from web site:
<http://www.naturalhistorymag.com/0204/0204_feature.html>
[09 04 2004]

Callan, R. (2003). *Artificial Intelligence.* Ashford Color Press, UK

Champanlard, A. (2003). *A.I. Applications.*
Retrieved from web site:
<http://artificialintelligence.ai-depot.com/Definition/FieldAI-Applications.html>
[14 04 2004]

Epstein, J.M & Axtell, R. (1996*). Growing Artificial Societies: Social Science From the Bottom Up.* Brookings Institution, Washington, D.C., USA.

Grimshaw, D. (2001). *CPS 721 Lecture Notes.*
Retrieved from web site:
<http://www.ryerson.ca/~dgrimsha/courses/cps721/aLifeIntro.html>
[05 04 2004]

Vlassis, N. (2003). *A Concise Introduction to Multiagent Systems and Distributed AI.*
Intelligent Autonomous Systems Informatics Institute, University of Amsterdam.