Knowledge & Agent Systems
Assignment 1

Karolina Kozikowska (s5504503) &
Kajetan Karbowski (s5470161)

September 10, 2024

Question 1

1.1

sugar

It is the agent's amount of sugar and if its level drops below a certain threshold, the agent dies. The higher the variable value, the longer it takes for the agents to exhaust all their resources and start dying.

metabolism

The value of the `metabolism` variable determines how much sugar an agent consumes at once, per tick. It can be observed that the population decreases at a slower rate if metabolism is low.

vision

The value of `vision` determines the agents' ability to notice and move towards the richer patches of sugar in a single tick.

1.2

Number of population after running the experiment: `215, 228, 223, 235, 219, 236, 238, 239, 223, 215, 236, 212, 240, 225, 229, 230, 232, 222, 234, 234`

Number of turtles that can stably survive in the environment using the parameters provided in this exercise:

Mean	228.25
Standard Deviation	8.595
The number of runs performed	20

It is important to aggregate the carrying capacity over multiple runs, as the environment at the beginning of a simulation is not constant. The first aspect to consider would be that the agents do not always start from the same position, so in some runs, they may start with easier access to resources. Moreover, the agents' initial sugar level, metabolism and vision are random values from a certain range, so in every run, the parameters are a bit different. This is why we chose to do so many runs - by redoing the simulation 20 times, we had a chance to see the values of the carrying capacity created from many different starting positions.

Question 2

2.1

The turtles exhibit their proactivity by taking action to reach the patches of the resource and guarantee their prolonged survival (energy not dropping below the threshold). When doing so, they exhibit their reactive behaviours by actively avoiding colliding with other agents, as no two agents can occupy the same patch in a single tick. It can be assumed that agents' behaviours skew more towards their reactive collision avoidance over their proactive search for food. These priorities come from the fact that an agent will never further their plans if they are intruding on other agents. Moreover, agents are not planning any complex moves as well as their plans only revolve around their present state and the next tick, so their behaviour skews more towards the reactive side.

2.2

What are the turtles' BDI in Sugarscape? BDI as Beliefs-Desires-Intentions

Belief

We can understand the beliefs as the state and rules of the world seen by a particular agent. In this case, we can assume those beliefs to be the amount of sugar in patches seen by the agent (in the range of its vision), and the fact that they will bring it the energy it requires.

Desire

The agent's desire is the need to supply enough sugar so that its energy level does not drop below the threshold. Moreover, it does not want to collide with other agents and thus share the resources.

Intention

The agent wants to find themselves at the patch of sugar that guarantees survival. If the agent's sugar level is high enough that after one tick the level is still above the threshold (`if sugar - metabolism` $>$ `0`), the agent will not move. Otherwise, the agent will move to the patch within its area of vision with the higher level of `psugar` (patch's sugar level).

2.3

The turtles in Sugarscape act more like objects, as they always try to fulfil the goal of consuming enough sugar to not let their energy levels drop below the threshold. If they were to act more like agents, they could allow themselves not to follow that desire and thus voluntarily "starve". However, such behaviour does not ever take place and the only possibility of an agent dying is when in its radius of vision there are no patches with enough sugar to sustain them. As such, we cannot assume turtles to have an autonomy significant enough to act like agents.

Question 3

3.1

The grid in NetLogo is 50x50, so the third quadrant has a size of 25 x 25, which is why only 625 turtles can fit on the map without making more of them inhabit the same spot. It means that the code will produce an error if the setup is initialised with more than 625 turtles. The turtles inhabit random squares in the third quadrant and the more of them are initialised at the beginning, the closer they are to each other, which is a result of very limited space for them. Moreover, when the simulation is run for a small population, all turtles stay in one group (with the exception of a few turtles that move to a much farther part of the grid) and the population reaches its carrying capacity while maintaining such a configuration (see Fig.1). However, the bigger the initial population, the more turtles leave the original group and move to the first quadrant. For a very big initial population, after a short time we could see two similarly-sized groups, in the first and third quadrants of the grid. That corresponds to where the patches have the most sugar on them. It means that a large population has to find an additional spot to feed to ensure its survival.

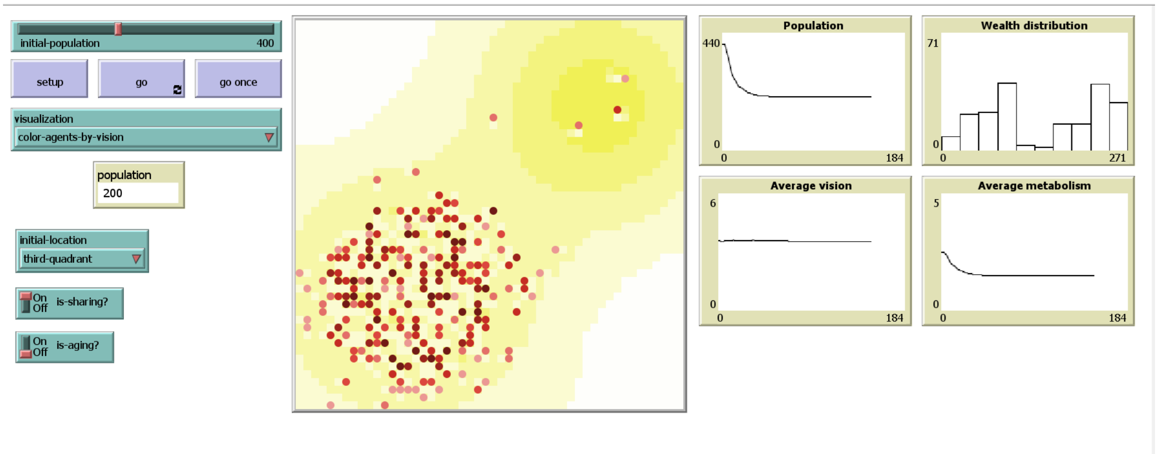

Figure 1: The environment highlights the emergent behaviour

```
to turtle-setup ;; turtle procedure
  set color red
  set shape "circle"
  (ifelse
    initial-location = "all-over-grid" ;; Using a chooser as a global variable for Exercise 3.1
    [
      move-to one-of patches with [not any? other turtles-here]
    ]
    initial-location = "third-quadrant"
    [
      move-to one-of patches with [pxcor < 25 and pycor < 25 and not any? other turtles-here]
    ]
  )
```

3.2

To check if our code works, we created an additional setting of `color-agents-by-sugar`, which is why in the screenshots provided for this exercise the agent colours are different compared to the options given to us in the original code.
We decided to implement the sharing behaviour of agents by making a ”rich” turtle (a turtle with

the highest amount of sugar compared to other turtles in its radius of vision) give away at most 30% of its sugar level or its sugar level - 10 to neighbouring turtles with a lower level of sugar. Such values were chosen after a few trials with percentages ranging from 10 to 30, and the maximum amount of sugar given fluctuating between the turtle's sugar level - 5 and 15. The settings written in the code and explained above resulted in the highest number of agents in the carrying capacity. As a result of our implementation, the changes in the number of population, which always started as 250 agents, were smaller compared to running the code without the sharing function, which means that sharing 30% of the richest turtles' sugar (or its sugar level - 10) is enough to increase the surviving of a bigger part of the population. However, as seen in the wealth distribution histogram (see Fig. 2), there are no turtles with a very low or very high sugar level. Most turtles in the population do not have a moderate amount of sugar, so they have enough to survive

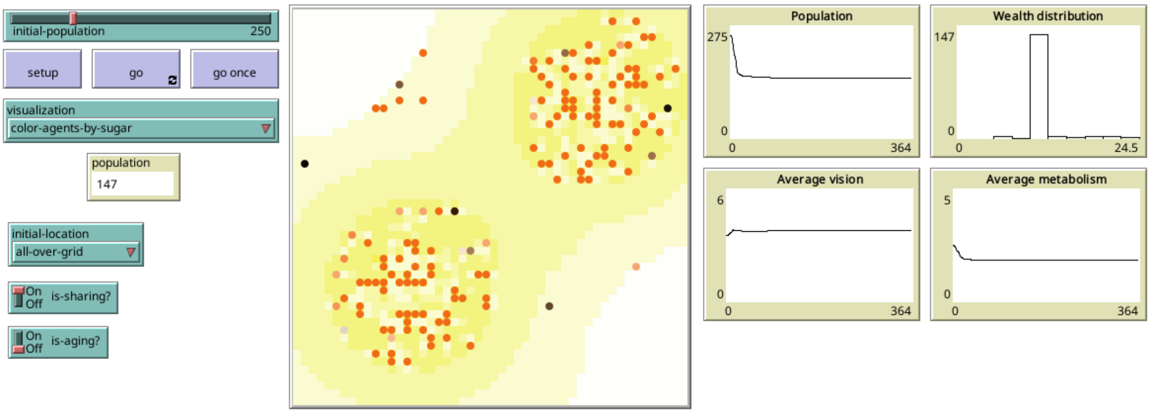

Figure 2: Output obtained from the model after implementing sharing behaviour

```
to turtle-share ;; turtle procedure
  let neighbour-turtles turtles-on patches in-radius vision
  let hungry-turtles neighbour-turtles with [sugar < [sugar] of myself]
  if any? hungry-turtles [
    let sherable-sugar min list (sugar * 0.3) (sugar - 10)
    if sherable-sugar > 0 [
      let sugar-gift sherable-sugar / count hungry-turtles
      ask hungry-turtles [
        set hungry-turtles  (sugar + sugar-gift)
      ]
      set sugar (sugar - sherable-sugar)
    ]
  ]
end
```

3.3

From the initial set of 250 turtles, due to every turtle being reborn after dying, the population stays constant. That is because even if a turtle is born far from a lucrative sugar patch and dies almost immediately, it gets instantly replaced with anew turtle at a new spot on the grid. Since there is constantly the same number of turtles on the grid, the wealth distribution graph shows that most of the agents do not have a lot of sugar and the smallest number of them has the most sugar as the sugar distribution is very skewed to the right (see Fig. 3).

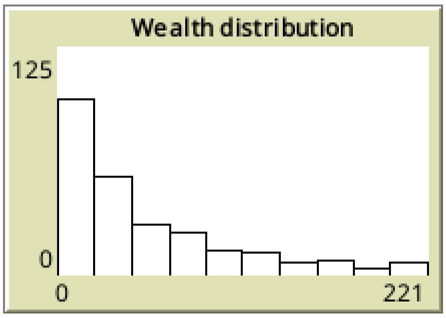

Figure 3: Histogram showing wealth distribution in an ageing population near its end

```
turtles-own [
 age               ;; the number of year this turtle has left

...

 ask turtles [
   turtle-move
   turtle-eat
   if is-aging? ;; Using a switch as a global variable for Exercise 3.3
   [
     if age <= 0
     [
       hatch 1 [ turtle-setup ] ;; new turtle is born
       die                      ;; turtle dies of old age
     ]
     set age (age - 1) ;; turtle ages by one year
   ]
   if sugar <= 0
   [
       hatch 1 [ turtle-setup ] ;; new turtle is born
       die                      ;; turtle dies of old starvation
   ]
   ...
```