Final Assignment: Bee Society Simulation in Netlogo

Multi-Agent Systems 2015-2016

Group 13
Ben de Haan
Erik van den Boogaard
and Tjeerd Schoonderwoerd

1. CONCEPTUAL DESIGN

For our final assignment, we chose to build a multi-agent system that resembles the main activities of a bee society. Our system was build in the Netlogo programming environment (Wilensky, 1999). Our inspiration is partly based on the BeeSmart Hive Finding Model in Netlogo. This Hive Finding model focuses on finding a new living place (i.e. new hive) for one bee colony. Our system however has a much broader focus and introduces many new aspects and challenges (as described below). We based our model upon information from various scientific sources (e.g. [2] and [1]). Although the system is realistic to a certain extent, it is important to note that our aim was *not* to build a one-to-one representation of the real-world, i.e. on natural behaviours of bee societies. That is, we focused on modeling a society of bees, which has to perform certain tasks in order to enable it to survive. Our agents (i.e. bees) show intelligent (flexible) behaviour by reacting on changes in the environment (i.e. reactivity), exhibiting goal-directed behaviour (i.e. proactivity), and by communicating with other agents (i.e. social ability). Details of these behaviours are described later in the report.

Below, we first specify the research question that we want to answer based on a simulation of our multi-agent system. Then, we describe the different types of task that will have to be performed, the type of agents that the system includes, and the environment in which these agents act. These characteristics of the environment and agents are summarized in Figure 1.

1.1 Research question

The main question we want to investigate is how a bee society can successfully solve problems in the environment by performing tasks and communicating with each other. More specifically, we want to know the optimal conditions for a society of bees in order to survive. Our model will include many parameters (see below) about the environment and bee society that can be tuned (due to time constraints, we will do this manually). A set of parameter settings will be successful when this leads to a stable number of bees in the environment (i.e. bees will not systematically decrease in number over time). Thus, with our model we aim to answer the following research question:

Which set of parameter settings lead to a stable living condition for an artificial bee society?

1.2 Overall task and subtasks

The most important and main task of all agents is to survive as long as possible in the environment. We interpret this task as being completed when a stable society of bees has been reached (see above). This overall task will include maintaining the hives in the bee colony, and also the creation of and migration to a new hive on a different location, when the current hive cannot *handle* (i.e. there are too many bees or not enough food) the number of bees. These tasks can further be divided into several subtasks that need to be achieved by the agents. These tasks are:

- Search for food
- Communicate location of food
- Collect food
- Bring food to hive
- Eat
- Reproduce
- Search for potential new nest-site location
- Determine quality of potential new nest-site location
- Communicate potential new nest-site location
- Determine new hive location
- Communicate new hive location
- Set up new hive and colony (migrate)

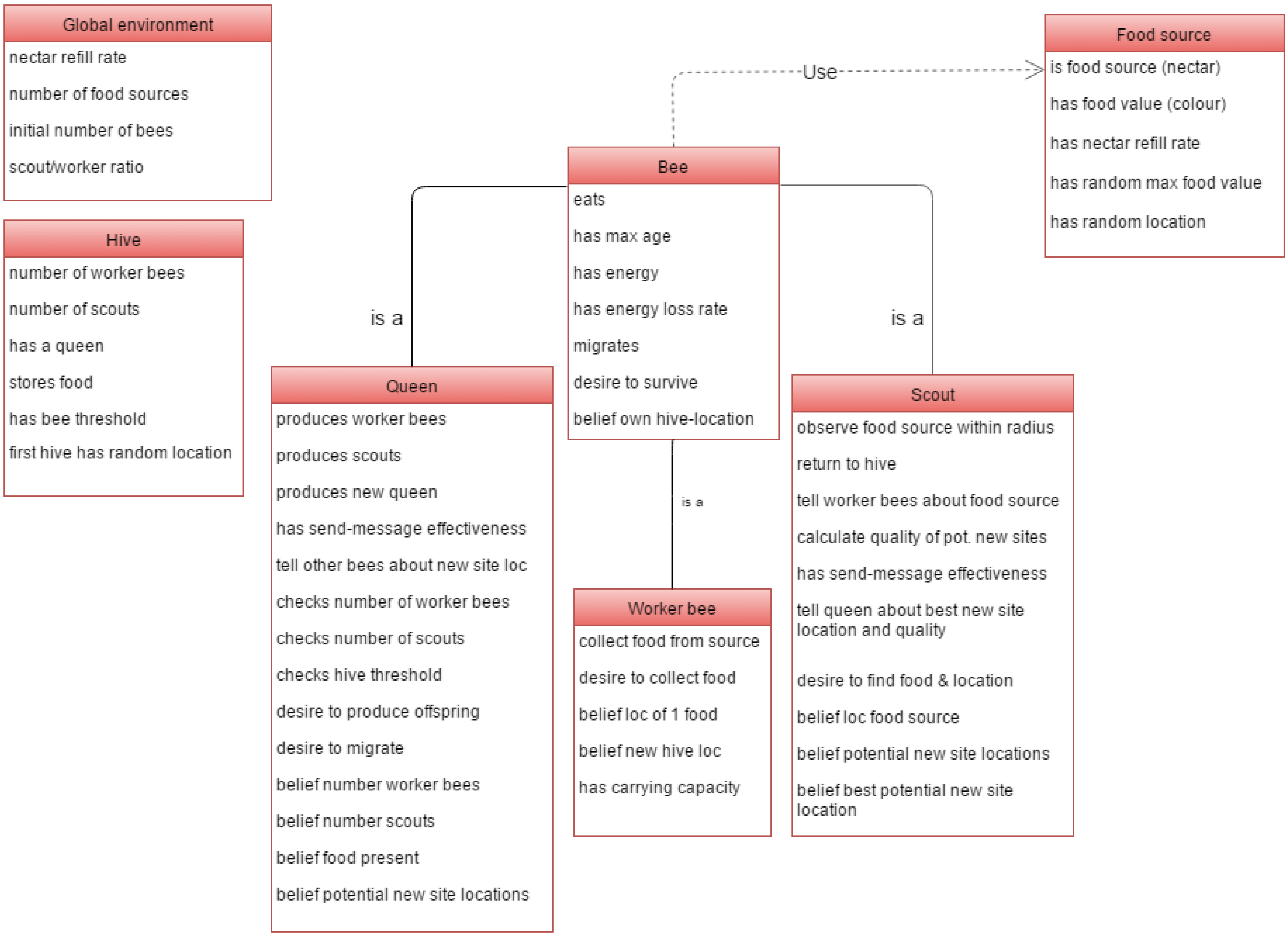

Figure 1: Graphical representation of the main characteristics of the system

1.3 Agent types

In the living environment of bees in real-life, there are three different types of bees, each with a different purpose and function to preserve the colony: a queen produces new bees, a worker-bee collects food, and a scout-bee locates food and new locations for a hive. These three bee types will each be represented by an agent type in our system, and will have to work together (cooperate) in order to be successful in the environment (i.e. maintain the colonies). To summarize, the system will contain the following acting agent types, each with a specific main purpose:

- **Worker bee**: gather food and bring this to hive
- **Scout bee**: scout locations of food sources and a potential new nest-site
- **Queen bee** (one per colony): produce new bees and establish one new hive

1.4 Environment characteristics

The environment in which the agents will act is represented by a two-dimensional plane of *patches* which contain bees, hives, and food sources. The environment is dynamic in the sense that the number and location of agents and hives change during simulation (the number and location of food sources is determined at setup and remains static during runtime). Furthermore, the quality of food sources will vary during simulation as a result of bees extracting nectar from these sources and of nectar replenishment. The environment in which the agents act is in-accessible, i.e. agents do not have a complete representation of the environment. Instead, their representation of the environment is limited to their direct observations, and memories of these observations. Table 1 specifies aspects of the environment and what changes them over time.

1.5 Communication between agents

In our multi-agent system, communication between agents is required in order to successfully maintain the bee society. More specifically, because agents have different main purposes (dependent on their agent-type), they need different kinds of information in order to successfully fulfill their tasks. Some of this information cannot be collected by themselves, so they need to receive this by means of communication from other (types of) agents. In a natural bee colony a scout bee performs a so-called *waggle dance* to communicate the location of a found food source to worker bees in the hive, which in turn fly out to collect nectar from this food source. In our model, communication between agents

Table 1: Environment specification

Entity	**Depends upon:**
Bee agents	number of flowers (nectar) and birth/death rate
Nectar (flowers)	nectar refill rate and number of bees
Hives	number of bees

is implemented by allowing agents to perform 'speech acts'. I.e., we treat communication a specific action that can and will be performed by agents in order to message other bees in their colony. Table 2 on the next page lists the communicating actions that will take place between agents in our system.

1.6 Mental states & Reasoning

As already stated our agents exhibit intelligence in the form of reactive, pro-active and social behavior. An agent chooses and performs its actions based on its mental state and internal reasoning. In order to represent the mental states of agents, the BDI-framework is used. That is, all agents have desires about what they want to achieve, beliefs about the environment and about other agents, and intentions. Based on their beliefs and desire, each agent creates an intention to perform a certain action. Furthermore, based on their beliefs it reasons about whether it does or does not want to perform this action. Executing an action (e.g. eating) will likely change its belief again (e.g. having enough energy), which can in turn change its intention to do something (e.g. collect food or not). In our model, mental states are represented in textual format to increases readability (see Netlogo code).

1.6.1 Mental states

Below we list the mental states (desires, beliefs, and intentions) we used for the agents in our system. As mentioned above, agents can be one of three types: worker, scout, or queen. The mental states of an agent depends on this type. Therefore, the states that are listed below are separated by the type of agent.

Desires

Worker: provide colony with food
Scout: find food and find optimal new hive location
Queen: maintain colony or create new colony

Beliefs

Each bee has beliefs about the location of its hive, its current age and energy level, and about an incoming message from the queen concerning a new nest-site. In addition, an agent can have other beliefs, dependent on its type:

Worker:

- location of food source
- quality of food source
- carrying capacity (food)
- current amount of food carrying
- incoming messages from scout (about food source)

Scout:

- location of new food source
- locations of known food sources
- location and quality of potential new nest-sites
- outgoing message (about food source or potential new nest-site)

Queen:

- total bees in hive
- hive threshold (maximum capacity)
- amount of food in hive
- location and quality of potential new nest-sites
- incoming messages from scouts
- outgoing message to other bees

Intentions

All intentions of an agent depend on the desires and beliefs of an agent. Each agent can have the intention to eat, to move, or to migrate a new nest site. Furthermore, agents can have the following intentions, depending on their agent type:

Worker:

- wait for message from scout about food location
- fly to location of food
- collect food
- fly to hive
- drop food in hive

Scout:

- look around (observe)
- fly to hive
- message workers (about food location)
- message queen (about potential new nest-site locations and qualities)

Queen:

- produce new worker
- produce new scout
- produce new queen
- wait for new potential nest-site
- tell others about a new nest-site to migrate to
- create new hive

1.6.2 Reasoning

Each agents reasoning takes place internally and is based on its current desire, beliefs and intention, and is mostly aimed at deciding what to do next. In our system, each agent

Table 2: Communication between agents

From agent-type:	To agent-type:	Speech act
Scout	Worker	inform about location of food source
Scout	Queen	inform about potential new nest-site location (and quality)
Queen	Scout and Worker	inform about new nest-site

frequently reasons about what action to perform in order to achieve its goal. All reasoning is explicitly programmed into the system, and we will therefore not describe in full detail how all reasoning takes place in this report. However, below we describe the main reasoning processes of agents in a short and simple sentence. Please note that this is a simplification of the reasoning process. See the Netlogo code for the full implementation.

First of all, each agent has simple reasoning capabilities about when to eat (i.e. energy is low), and when to migrate (when the queen tells him to, i.e. it has an incoming message from the queen). Furthermore, there are several different reasoning processes taking place in an agent, again depending on the agent type. Two more complex reasoning processes (patch-evaluation from scouts and queen reasoning about the current hive) will be discussed in more detail below.

Worker:

- if at hive and doesn't know about food location: wait for message
- if location of food is known, and carries no food: fly to food location
- if at food location, there is food and it still can carry food: collect food
- if collected food and not yet home: fly to hive
- if home and carries food: drop food in hive

Scout:

- if no food seen yet: observe and evaluate patches (see below)
- if observed food source and not yet home: fly to hive
- if observed food source and home: message workers about food location
- if observed potential new nest-site(s) and home: message queen about potential nest-site

Queen:

- if hive is full, but doesn't know about a potential new nest-site yet: wait for new potential nest-site (as communicated by scouts)
- if there are too few workers in hive: produce new worker
- if there are too few scouts in hive: produce new scout
- if it believes hive is full and it knows a nest-site location: produce new queen and tell others to migrate
- if it is a the new nest-site location: create new hive

We will now discuss the reasoning of scouts about the quality of potential new nest-sites, and the reasoning of a queen about the maintenance of her hive.

Scouts: utility function
One important reasoning method we used - evaluation - is described in more detail here. The newly produced queen swarms out with her assigned colony to her best known location for a new hive. The knowledge of best new hive location comes from scouts. Scouts not only keep track of new food sources but also of the most optimal new hive location. During flight (wandering over the grid of patches) a scout detects several food sources and hives within its sensing range (determined by a slider, see parameters). For each location (patch) a scout flies over, it *evaluates* how *fit* this location is a hive location. Table 3 shows the Netlogo code concerning this evaluation. Each scout remembers the patch with its *best* score. If an existing hive is within sensing range, then the fitness for this location will be 0. If not a *fitness* score is calculated based on distances to *known* (i.e. those in the scout belief base) food sources. If this fitness score is higher than its best score then it remembers this location (i.e. patch-here) with its concerning fitness score.

Queen: hive-maintenance and migration
The queen has the main task of producing new bees adding to her existing colony. Furthermore, she has to decide whether it is time for part of her colony to migrate to another location. The queen holds beliefs about the number of worker and scout bees in her colony (i.e. the ones with the same home as she), and calculates a ratio between these two types of bees (number of scouts relative to the number of workers). If this ratio between scouts and workers is lower than (or equal to) the ratio as set by the user that sets up the model (see 'Parameter' section below), she reasons that there are too few scouts and intends to create a new scout. When the ratio is higher, she reasons that there are too few workers and intends to birth a new worker. Apart from the ratio between workers and scouts, the queen also holds beliefs about the total number of bees in her colony and, importantly, the number of bees that her hive can support (this relates to the size of a hive in nature). The maximum hive capacity is set by the user. When the total size of her colony reaches this maximum, the queen reasons that her hive is full meaning that part of her colony should migrate. Migration is a set of actions that involves all types of bees (see 'Implementation' section).

2. IMPLEMENTATION

During the development process of the bee-society model, we made several design choices with respect to the implementation of different aspects of the system. In this section, we will discuss some important implementation choices.

Table 3: Utility function

Score = 0
FOR each hive DO
” IF hive is within vision range of scout THEN set Score = -1”
IF Score = 0 *; no hive is within vision range*
THEN FOR each known food source DO *; belief set of food sources of this scout*
” Score = Score + 1/(distance to food source)
IF Score > High-score THEN set New-hive-location to current location AND set High-score to Score

Hives as agents

Apart from the more ’obvious’ scout, worker, and queen agents, we also chose to implement hives (and sensors) as agents. This introduced several advantages, as we were able to associate specific variables to hives. That is, hives own a variable for the total bees inside it, as well as the total amount of food that is stored. Conveniently, as every hive was an agent, these numbers could easily be updated and requested during simulation.

Sensors and vision radius

Scout sensors were also implemented as agents. The functioning of these sensors is similar to the sensors of vacuums that were used in the Netlogo assignments for this course. That is, in order to simulate the perception of scouts, each scout owns ’sensors’ with which they can observe the environment. These sensors are implemented as agents that are connected to a scout and spawn on patches in a radius near the scout whenever this scout performs the action ’look around’. Whenever one of these ’sensor-agents’ of a scout is on a food source, the scout perceives this food source (i.e. it is added to its belief base) *and the sensors disappear (die)*. We chose to immediately let the sensors die after observing, as seeing the sensors of dozens of scouts constantly appearing and disappearing again made the simulation very unclear to watch. Each sensor has a vision radius that determines how far around itself an agent can observe the environment. This vision radius is set by the user in the interface. More specifically, suppose a vision radius of 3. This means that whenever an agent is looking around, sensors (attached to the agent) are spawned on patches that are within 3 patches around this agent.

Food source as patch with food value

Food sources are implemented as purple-colored patches that have a fixed location in the environment, but differ with respect to their maximum food value that they hold (i.e. the food quality). This can be thought of as representing a location with many flowers and/or flowers that are rich with nectar. In our model, the quality of food sources is visualized by the ’darkness’ of the color: the darker purple a food source is, the higher its maximum food value. Furthermore, the food sources slowly replenishes their food values over time, as determined by the nectar refill rate (parameter can be set in interface).

Carrying capacity

Workers have the main task of providing their colony with enough food, stored in their hive. However, whenever they obtain information about the location of food, there is a limit to the amount of food they can carry at once. That is, reaching the food source, a worker decides whether it can still carry food by comparing the amount of food it currently carries with its maximum carrying capacity. This carrying capacity is determined by the user in the interface.

Migration

As described earlier, migration involves multiple agents and is put in action whenever the queen of a hive believes that her hive is full and she knows about at least one new nest-site location. If she has beliefs about several potential nest-sites, she picks the one that has the highest quality (as evaluated and communicated by scouts). After that, the first thing she does is birthing a new queen that can become the queen of the new colony. This new queen immediately has the desire to ’create a new colony’. Then, the old queen communicates the location of the new nest-site that she chose to a part of her colony. The amount of bees that receive this message is dependent upon the ’queen message effectiveness’ parameter that is set in the interface. That is, this parameter directly corresponds to the percent of workers and scouts that will receive the message about the new home. Furthermore, the new queen will always receive this message. Subsequently, part of the colony inside the old hive will migrate to the new location and the new queen will create a hive there, that has now become the home of part of the old colony. The new queen then changes her desire to ’maintain her colony’ and will perform the accompanying tasks.

Communication

In nature, bees communicate to each other through dances and pheromones. That is, scouts communicate the location of food sources through a series of movements known as a ’waggle dance’. As described earlier, communication between bees in this model is implemented as ’speech acts’; specific actions that agents can perform in order to communicate with each other. Most importantly, **as in nature, all communication takes place from within the hive**. Each agent has a list of incoming and/or outgoing messages, which include the information they want to convey to /receive from (an)other agent(s). Whenever an agent wants to send a message, it puts it in its outgoing messages list. Whenever the conditions for communication (e.g. the listener being close to the speaker) are met, a listener can receive the information by 'copying' the message to his incoming messages list. In our model, workers 'choose' to only receive one message about a food source at a time, just as it happens in nature. A queen however, is open to all messages about the location and quality of potential new nest-sites, as communicated by scouts. Scouts communicate their message about a food source only to some workers, just as queens only send their 'migrate' message to part of the colony (as determined by the scout- and queen message effectiveness).

2.1 Parameters

Below is a list of global parameters that have been used in the model, their specifications, and a small description. The values for all global parameters (apart from color-list) can all be set up by the user in the interface. To save space, we chose not list all local variables that we used in the model. Instead, the names and descriptions of the local variables (together with globals) can be found in the code (see attached Netlogo file).

2.1.1 Global parameters

- *initial bees*: initial number of workers and scouts (0-20)
- *scout worker ratio*: desired ratio between scouts and workers (0.05-2.00)
- *number of food sources*: number of food sources in the environment (1-100)
- *nectar refill rate*: the speed at which food sources replenish their food values (0-10)
- *bee capacity*: total number of bees that a hive can hold (0-50)
- *energy loss rate*: speed at which a bee loses energy (0-10)
- *carrying capacity*: maximum amount of food a worker can carry (0-20)
- *gain from food*: amount of energy a bee gets from eating 1 food (0-10)
- *color list*: list of colors to indicate food quality

3. SIMULATION

Figure 2 shows a screenshot of our artificial bee society model running in NetLogo. You can see food sources (colored squares with food values), hives (yellow trees), scouts (red bugs), and worker bees (sticking out of the hive). A queen is only visible during migration from her birth hive to the next optimal new hive location where she creates a new hive. The graph clearly shows that the colony in its environment reaches an equilibrium. The number of hives with accompanying queen, scouts and workers stabilizes with time (ticks).

Each simulation run starts with an evironment with a number (slider) of food sources and 1 hive with a queen and a number (slider) of bees in a certain ratio (slider) of scouts and worker bees. Each bee has some initial beliefs and only one desire and intention at all times. During a run, a sequence of interleaved steps - with resulting environment states - is performed for each agent:

- *Update belief*: depending on environment and past belief
- *Update intention*: based on current belief and some conditions
- *Execute actions*: based on its intention leading to a new environment with new conditions
- *Communicate*: a special action performed by a scout to the queen and/or by a queen to some workers and scouts within the hive
- *Grow food*: a special action where each food source is replenished with an amount of nectar

A typical sequence for a scout is to wander (random flight) away from its hive looking for a new food source and a new optimal hive location. Once it has found a new food source it flies back to its hive where it communicates the new food source to the workers in the hive and the best new hive location to its queen. This sequence repeats until the scout dies of old age or of energy depletion. A typical sequence for a worker is to wait in its hive for instructions until it receives a message from a scout with the location a new food source after which it flies to this food source to collect food, fly back and drop the food in the hive. This sequence repeats until the worker dies of old age or of energy depletion. A typical sequence for the queen is to create a new bee (scout/worker ratio slider) which repeats itself until the hive is full (slider) after which it produces a new queen which swarms together with a number of bees (ratio slider) to a new optimal hive location where the new queen creates a new hive. For flying and producing offspring an amount (slider) of energy is required. Energy is produced in the hive from collected food which each bee can consume (eat) during its stay in the hive until the food suply runs out. Most runs - depending on the set of parameters - reach a balanced environment with a constant number of hives and bees after which nothing special happens anymore. Some runs end in a state where the whole society died (no more bees). This end state occurs mostly with a low number (slider) of food sources and relatively large energy consumption (slider).

4. CONCLUSIONS

The main question we want to investigate was if we could create a stable artificial bee society which is able to survive and procreate in an artificial environment using multi-agent principles like intentional notions (BDI) and communication and several (also manually adjustable) conditions. We wondered if the bee society - starting with only one hive - was able to maintain itself while producing new hives which would also be able to maintain themselves. Generic conclusion about our model based on the research question and on simulations we ran is that the bee society model we created seems very robust. Given enough food the bee society is able to create and maintain several hives with a stable number of bees even when varying other variable conditions to a large extent. The number of hives created and maintained mainly depends on a variable vicinity condition (any hive in the vicinity?) which influences the fitness of a potential new hive location and also the decision whether to create a new hive on that site or not. Our bee model is inspired by nature using birth, increasing age, death and also energy level which increases when eating and decreases when flying. One exception is made for each created queen which in our model is immortal. In nature an aging queen mostly is replaced well in time to be able to maintain the hive which of course sorts the same effect as immortality.

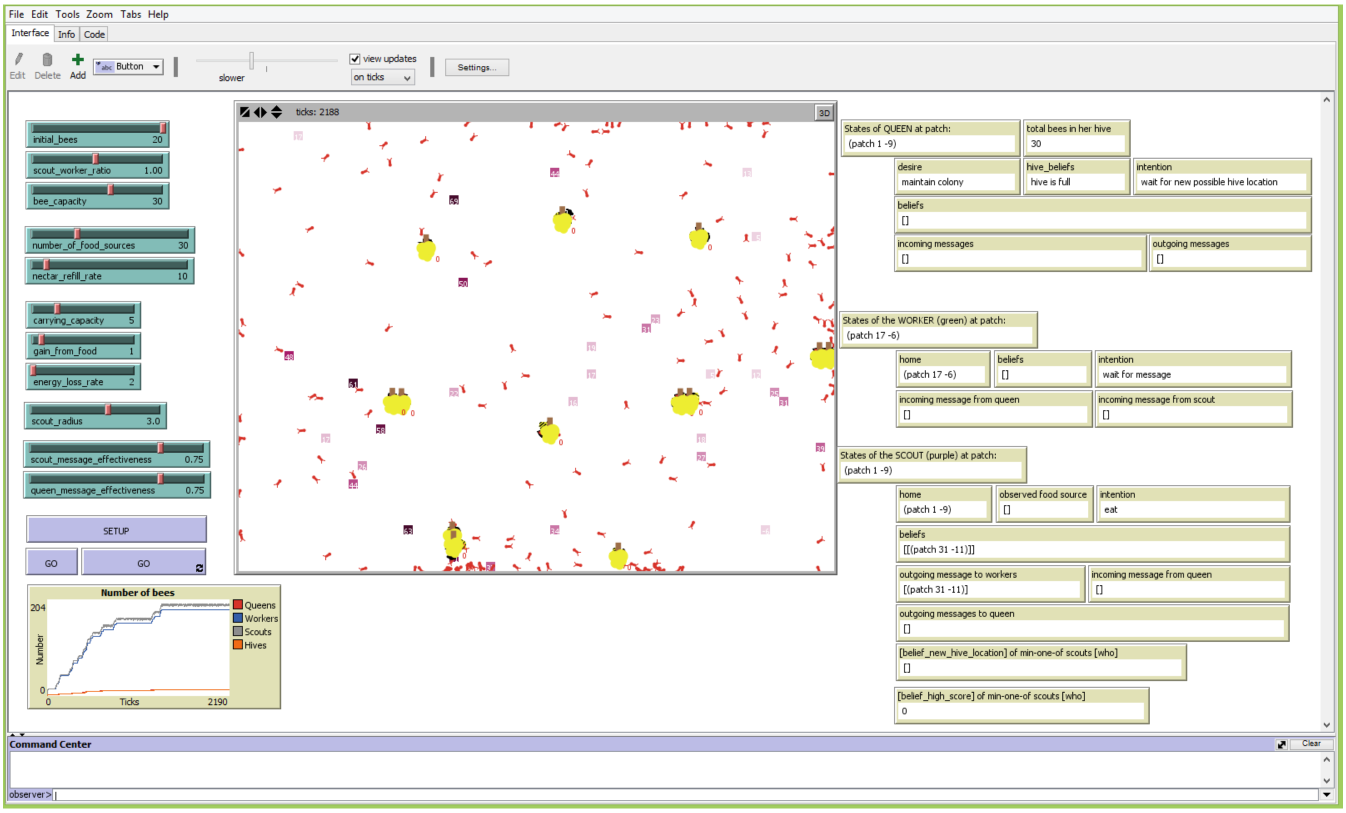

Figure 2: Screenshot of a simulation run with the Artificial Bee Society model in NetLogo

4.1 Limitations and future recommendations

Due to time constraints, we did not model enemies with their behaviour, nor incorporate different types of food that normally exist in a typical bee environment. Finally, none of us are experts in the field of bee behaviour; this may impact how realistic our model is. There are many behaviours that can be added to our model in order to more closely resemble a real-world bee society. Examples are:

Enemies: to make the environment more complex, enemies that attack bees and/or hives can be added. In the natural environment of bees, their greatest enemies are bears and birds, who could also steal food (honey) from the hive, and also eat from food sources. These enemies could be added in our model by implementing them as agents with beliefs (e.g. there is a hive at location X), desires (e.g. wanting to feed their children), and intentions (e.g. steal food from hive at location X) of their own.

Stop signal: in our model, scouts communicate by performing a speech act inside the hive, that can reach workers who are inside the hive and currently listening (in our system implemented with the 'scout message effectiveness' parameter). Scouts always communicate locations of new food sources. However, in nature worker bees can also stop scouts from performing their communicative action (i.e. their 'waggle dance'), by telling them to stop dancing (they do this for example when they believe there is there is no food left, or that the route is unsafe). Our model can be extended by implementing this kind of 'counter-communicative' actions.

Food sources: apart from agents and their communications, another aspect of the natural environment that can be added in our artificial environment is other types of food sources and dynamic locations. In nature, bees get their food from different kinds of sources, and these sources change in location over time. For example, food sources disappear as a result of age (i.e. flowers that die), and spread by depositing their pollen.

Strategies: a last example of an improvement of our system is to include behavioral strategies that bees can choose from, which will result in different outcomes. For example, more defensive/conservative bees could be focusing more on conservation of the current colony. On the other hand, there could be more aggressive/explorative bees that value finding optimal locations for new nest-sites over conservation of the current colony.

Given time, the above improvements could be added to our system, and could potentially lead to a better insight into the dynamics of natural bee societies.

5. REFERENCES

[1] Gould, J. L., Gould, C. G., Von Frisch, K., Frisch, K. v., Wenner, A. M., and Wells, P. H. *The Honey Bee*. Scientific American Library, 1995.

[2] Velthuis, H. Diversiteit en concurrentie bij bijen. *Entomologische berichten 72*, 1 (2012), 99–106.