Using agent based modelling techniques to investigate the effectiveness of the honeybee Waggle dance

Chris MacNeil and David C. Moffat[1]

Abstract.

Computational models of animal cognition may help us to understand the complexities of animal behaviours. Agent based models (ABM) allow us to model individuals that may quite simple in themselves, but who interact in ways that produce coherent behaviours at scale. Animal communication is a key example of a behaviour pattern that can only be understood as a group phenomonon.

In this study we model the famous waggle dance of the honeybee. The software platform Netlogo is used to see how helpful it can be to biologists who may not be expert in computer science. Bees are modeled as finite-state machines, and there can be hundreds of them in the colony. They leave the hive to search for food sources (flowers) and return to communicate their direction and distance to other bees. In order to check the value of the waggle dance, it can be turned on or off in the artificial bee population.

We find that the communication is indeed valuable to helping the bees locate food more quickly, and especially when it is not plentiful. We find that the component of direction is more useful to the bees than distance, although honeybees use both in real life. Netlogo system is shown to be a good platform for such research, allowing biologists to make artificial experiments that can help to clarify the roles and mechanisms of animal cognition.

1 Honey Bee foraging and a role for ABM

Honeybees are simple creatures with tiny brains, but they achieve very clever things with them, by working together. In particular they communicate by gestures — the "waggle dance" — where the richest food sources are. We used agent based modelling (ABM) techniques to create a virtual honeybee colony and surrounding habitat. The bees followed a foraging algorithm to find and collect food around the habitat before returning to the hive to store their resources. In the model that was built, it is possible to control how much the bees are able to communicate with each other, as well as various other control variables. This allows us to conduct virtual experiments to test hypotheses about which factors or bee characteristics are most important for producing their adaptive behaviours. We describe the concept of ABM, and the development of the honeybee model. Then we apply the model to test our hypothesis that the waggle dance of a returning bee will be witnessed by more observer bees, if the habitat is a richer environment for the bees. The model shows itself able to answer such questions, and is moreover usable by scientists of all types, and not only computer scientists. It suggests that such ABM methodology should play a more central role in future, and would be a big help to researchers into animal cognition.

[1] Department of Computing, Glasgow Caledonian University, UK. email: D.C.Moffat@gcu.ac.uk

Honey bee recruitment to food sources has been the centre of wonder for a long time. Gould & Gould [2] summarises how Aristotle first noticed that once a honey bee had discovered a food source, before long she had recruited more of her sisters to that location. Gould follows up by explaining how Reverend Ernst Spitzner used a glass hive in 1788 to inspect the work of a honey bee colony inside the hive and noticed that on return from a successful foraging trip a honey bee would twirl round in the hive almost to let the other bees know of her discovery, possibly to spread the scent of the food source they had just visited to get other bees' attention. Leadbeater & Chittka [5] confirm this initial assumption stating that it was assumed that the bees witnessing the dance would simply follow the dancing bee to the food source. As they quoted Spitzner: "Full of joy, they twirl in circles about those in the hive ... in a few minutes, after these had made it known to the others, they came in great numbers to the place!"

However Gould & Gould (*ibid* p55) clarify it was Maurice Maeterlinck who concluded this wasn't the case when he acquired a bee from its hive and marked it with paint before releasing the bee at a feeding station. Once the bee had fed from the station and embarked on its journey home, Maurice returned to the hive and waited for the bee that he had marked with paint to exit from its home. When Maurice spotted the bee departing he captured it before it was able to lead any other bees to the feeding station. He then ran to the feeding station and saw other bees had arrived at the station without having followed the bee that knew of the location.

It wasn't until 1944 when Austrian zoologist Karl Von Frisch discovered a more detailed explanation of the dance (*ibid* p55). He found that the twirling that Reverend Ernst Spitzner had discovered was part of the process of communicating with the other bees. He established that as the distance to a food source increased, so did the waggle duration of the bees dance. Karl von Frisch also discovered that the angle of the dance performed on a vertical comb translated as the direction required to navigate to the food source.

Throughout this long period of time the theory behind the dance has changed significantly. This highlights the need for further testing to ensure that the hypotheses that are understood at present are correct and are not missing any finer details that would be of important scientific discovery. Gould & Gould (*ibid* p61) highlights this issue stating that there has long been an element of controversy related to the honey bee waggle dance which has included a series of papers that were published in 1967 that challenged the legitimacy of the dance language as well as a university press that refused to publish a book written by Karl von Frisch that documented his theory behind the waggle dance.

Replicating the work of a Honey bee colony in a computational model Agent based modelling (ABM) is a technique that recreates an

environment in which agents are given a set of instructions to follow, and has even been proposed as a way to pursue research questions in archaeology [1] This approach would suit modelling a colony of bees that forage using the waggle dance as each bee could be programmed to perform a waggle dance on return from discovering a profitable food source. Any bees that witness the dance (within a specified radius of the bee) could use this information to forage. Due to the size of bee colonies consisting of tens of thousands of bees they are very hard to keep track of. Using a computer to simulate the behaviour would make the process of studying the bee behaviour much simpler.

In the following we apply ABM techniques to answer a simple research question. Using agent based modelling techniques; how much does the waggle dance behaviour of a honey bee increase the honey yield of its colony in a computational model? The aim is to test the applicability and feasibility of ABM as a research technique in this field of insect cognition.

2 The Netlogo system and methods for ABM

As a case-study applying AMB methods to the domain of bees, we selected the Netlogo system, and we followed a methodical approach to the software development process that has been recommended for ABM, and which we also outline here.

In order to make a computational model of the waggle dance efficacy, we chose the widely used NetLogo simulation tool [11]. It is a software tool that has been predominantly used for teaching social and biological concepts, and has many small models to illustrate biological phenomena in its library. For example, there is a model of a 2D world in which wolves hunt sheep, which eat grass [10]. The student can vary the population sizes and other variables, and can see their consequences for the relative population dynamics [13], [12].

The system has also been used for research purposes in its own right. In the fields of the social sciences, for example, it has become one of the most popular systems, especially since it recently went open-source [9]. Some applications there have included models of worker protest, in times of economic inequality [3], showing that a simple us-and-them categorisation of other workers can be enough to account for observed protesting behaviour. Another example is SimDrink [7], in which young male binge-drinkers and their verbally violent behaviours can bring other troubles in consequence, such as coming to harm in attempting to get home on public transport after heavy drinking in town. Whether such models will be seen as helpful to public policy decision makers remains to be seen, but they represent an interesting development in the social sciences.

In this work, we follow the methodical approach recommended by Railsback & Grimm [6]. It is quite common in AI for researchers to develop their own methods as they go, heading towards a model that they aim to build, but without having much concern to follow a standard method that other researchers in the field have or will also follow. In order to be more methodical, and thus make such research more scientifically comparable, and valuable to the field, we adopt the methods set out by these experienced ABM researchers.

They explain (pp4-7) how a model is generally used to represent real systems, using variables to describe the state of the whole system. An agent based model (ABM) differs in one particular and significant aspect: rather than model the state of the whole system, the state of the individual agents within the system are modelled. They further explain how the agents represent objects such as organisms and animals and explains how they can interact with each other and the computational world they reside in.

Klügl [4] illustrate this view with their own application, detailing how an ABM can simulate agents within their environment, providing an example of using an ABM to determine how long it would take commuters to evacuate an airport terminal during an emergency whilst taking into account commuters travelling in groups or with restricted mobility. Although commuters evacuating an airport is completely unrelated to honey bee foraging, an ABM could be used to model honey bee foraging by simply changing the rules that each agent would follow as well as the surrounding environment. The environment would be changed to simulate a honey bee habitat and rather than having agents of commuters following rules to evacuate an airport the agents would now consist of the foraging behaviour of a honey bee.

Before attempting to write an ABMs code, it is important to identify the ideas, data and problems the developer will need to take into account. Railsback & Grimm (*ibid* pp35-45) emphasise that this is the first step of full model formulation, and until all of the components and interactions of the model have been written down, it is improbable that a comprehensive design of the ABM will be completed. For us, this would involve noting down all of the relevant bee behaviours and interactions with their habitat including how they behave when they have found a food source. The second step of formulating an ABM is for the designer to discuss the work completed in step one with their colleagues or domain experts; and the third step is to formalise the model by writing out simplified (pseudo-)code. These steps are essentially good general practice in software engineering. We did not discuss the models' development directly with entomologists or other domain experts (as we are in a computer science department); but instead relied on their published research. This is a clear issue in all kinds of inter-disciplinary research, and will help to test the feasibility of this research approach in general.

On completion of an AMB's design process, Railsback & Grimm (*ibid* pp47-59) provide advice for beginning to implement the design work into a working model. The first phase of implementation should involve modelling a version of the planned ABM in its simplest form by excluding many of the components and behaviours with a view to completing these elements later in the development. The second phase is to develop the ABM in a hierarchical procedure. This would include ensuring that the general characteristics and structures of the ABM are performing as they should. Once these rules have been implemented correctly, the program will have a solid base and then finer detailed instructions can be included. Their third recommendation is to implement the ABM initially with a simple environment. This is particularly important if the project is intended to be of a realistic nature, consisting of complex interactions. As the first environment will be simpler, this will allow development to focus on the agents and their interactions and, without a complex environment, should allow for easier debugging. The environment can then be developed and tested with each iteration to ensure that the algorithms are working correctly. If an algorithm begins to fail then the developer will know that the development on the last iteration has probably caused this, which will allow for easier identification of what has caused the error. Finally, they emphasise the importance of formatting code correctly and providing informative comments that document the code's intended purpose. This extra time invested in producing an easy to read code segment should reap benefits for reasons such as debugging. It will also be a lot easier to debug when you can determine what functionality is failing in the program and then locate it quickly and easily in the code section.

Again, this advice amounts to good software engineering practice; but it is surprising how often it is ignored, even when people start

out their projects with the best of intentions. The great quantities of broken, badly written software in the world, that we have come to accept, or at least tolerate, is testament to the dangers here; but in a scientific research project we agree with Railsback & Grimm that a stricter emphasis on software quality is needed, if one is to draw credible conclusions from the computer experiment.

There is more advice on how to test the ABM model, that Railsback & Grimm have to offer, but to save space we do not relate it here, and moreover, we have indeed only summarised their methodology so far above. Interested readers are directed to their book which is a manual and proposal for their approach (*ibid*).

3 The ABM system design

As our chief source for what is known about honeybees, we took the fairly recent book by Seeley, one of the world's leading authorities on the subject [8]. That provided us with sufficient information on honeybee characteristics and behaviours, the honeybee waggle dance, and honeybee habitats. For example, honeybees are able to differentiate between colour and scent. They take these senses into account when foraging for food. Honeybees are also known to use the sun when navigating to and from their hive. In this work, we do not model such details as the sun's passage across the sky during the bee's workday, because we wish to focus only on the waggle dance that bees use for communication, so that we can try to ascertain how useful (or vital) it must be to the honeybees.

On discovery of a profitable food source a honeybee will perform a waggle dance on return to the hive to inform her sisters of the find. The direction of the waggle dance translates as the direction required to reach the food source. It has also been suggested that the duration of the dance conveys the distance of the food source from the hive. The longer the duration of the waggle dance, the further the distance required to reach the location. Research has also provided the author with information Regarding the habitat of honeybees, it has been found that they prefer to nest in high trees in woodland so they are safe from ground predators, and sheltered from rain and wind. They also prefer to build their nests in habitats that are surrounded by a steady supply of nourishment.

3.1 The representation of a single honeybee

In order to model a single bee, to be placed within the simulated habitat, we chose to use a simple agent-modeling technique that is often used in robotics, say, or in video games: namely, the FSM (finite state machine).

It has the advantage that a visual sketch of it conveys all the essential points, even to people who do not understand computer code. An FSM is only suitable for relatively simple systems, but as bees are generally assumed to be simple it is a good idea to at least start the attempt to model them with an FSM. As can be seen from Figure 1, the simplified FSM already has nine states, which makes for some complexity in code, but it is still manageable.

To illustrate how the FSM can then be converted into computer code, in this case into the Netlogo language, here is what an artificial bee does to collect nectar from a flower. When the bee lands on a flower it makes a transition to the feed state. This corresponds to arrow 14 in Figure 1. State changes are signified by the bees changing colour, so that the experimenter can see which bees are in which state at any time.

```
to feed
```

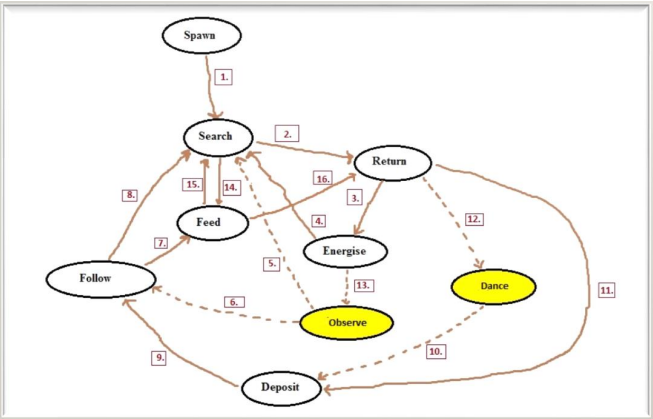

Figure 1. The Finite State machine for a simple honeybee

```
  ifelse pollen > 20
    [ set color blue
      return-to-hive ]
    [ set pollen pollen + 1
      ask flowers-on patch-here
          [set pollen pollen - 1] ]
end
```

The algorithm checks if the energy of the bee (representing pollen collected) is above 20, and if so its colour changes to signifiy the change to another state, corresponding to arrow 16 in Figure 1. The bee in its new state will then return to the hive. If however the energy isn't yet high enough, the bee takes one unit of energy from the flower, and remains in the same state so that it will continue feeding in the next time step.

When the bee returns to the hive later it may dance to show other bees the way to this flower, making transition 12 in Figure 1. Here is how the state **Observe** allows a bee to watch a dance and pick up the direction and distance information. As we use colours to reveal what state a bee is in, observing bees are green when they execute this piece of code.

```
to check-for-dance
  ifelse distance-to-travel > 1
    [ set color orange
      set time 0]
    [set time time + 1]
  if time > time-to-wait-for-dance
    [ set color red
      set time 0]
end
```

While in the **Observe** state, a bee can be directed by any dancing bee on the dancefloor in the hive, to adopt the communicated information about a food source. In the syntax of Netlogo, a procedure is defined by the `to` keyword. A Boolean logic branch is given by the `ifelse` keyword, which is followed first by the Boolean test, then a branch to execute if the test proves true (in `[square brackets]`), and then the branch to follow if the test proved false.

This procedure would observe any dance by placing itself in a state that would allow a dancing bee to pass the relevant information across to it. It does this by monitoring a variable named

distance-to-travel. If this variable changed then the program would know that the bee had received the set of instructions. The algorithm would then change the colour of the bee to orange and reset the time to 0 for the next procedure. Changing the colour to orange would allow the go procedure to change the state of the bee to **Follow**. If however the bee had not witnessed a dance then the time would increment per frame and if this time became higher than the variable time-to-wait-for-dance defined by the user / experimenter in the user interface, then the bee would change colour to red in order to return to the **Search** state. The time would also be reset to 0 for future procedures. In order for this procedure to work there would need to be bees that performed the dance. This was accomplished with the following procedure.

```
to dance
  ifelse time > time-to-dance
    [deposit]
    [ set time time + 1
      ask turtles with [color = green]
                  in-radius 5
        [ set heading
             [heading - 180] of myself
          set distance-to-travel
             [travelled-distance] of myself
        ]
    ]
end
```

Bees immediately perform a dance before depositing any of their resources in the hive. The model accommodates this by ensuring that the dance procedure occurs before the deposit procedure.

The implemented timer would increment every frame where the time was less than the user defined variable time-to-dance. This would allow the bee to perform the dance for a user defined set amount of time. Once the desired time had been reached the bee would then move to the deposit state to deposit its collected resources. If the desired time had not been reached then the bee would ask any green bees (as mentioned before bees in the observe state would change to colour green) to set their heading to the reversed heading of the asking bee and also to set their distance-to-travel variable to the travelled-distance of the dancing bee. This would effectively allow the observing bee to receive the heading and distance required to travel to the food source from the dancing bee. Note that we do not simulate the waggle dance itself, in this model; but only the theoretical communicative value of it.

We have also approximated real honeybee behaviour in other ways. Tyically honeybees will try to deposit their payload (of nectar, pollen or water) on arriving at the hive, and only then will they dance. This would be important for modeling the likelihood that bees will communicate their foodsources, but is beyond the scope of the current model. The code here then simplifies by having the bees dance first, and without choice, so that the model can focus on the efficacy of the dance to any observer bees then thence to the entire colony. In fact honeybee behaviour is fairly richly known, and there is a lot more scope for modeling than we have explored so far.

3.2 Running the model for many bees

Once the behaviour patterns and variables of a single bee are specified in code, as above, then it becomes a simple matter to create the artificial world in which there are a number of food sources, and a number of bees in the hive. The model can then be run at a convenient simulation speed, showing the states and movement of all the bees, as in Figure 2, to give a good impression of how the bees are behaving collectively. Charts such as the one in the Figure are also drawn in real time as the simulation run progresses.

The sliders and other controls in the interface allow the experimenter to play with the3 values of the various control variables, and see what effects they may have on the bees, in their individual behaviours and also in the global, collective behaviour patters of the whole colony.

When the model is working as desired, the simulations can be run at full speed, end even with the display turned off, so that the results can be collected from many runs with variables set at different values. In the end it is the *differences* in behaviour patterns, that result from different variable settings, that yield the scientific value from the "artificial experiment."

4 Results

The model is shown with its user-interface in Figure 2.

In the central area is the 2D view of the bees' world. Each little bee is run by the same code as outlined above, but bees can be in different states, as shown by their colour at the time. The hive is in the centre, and bees can be seen out foraging, fetching nectar from the flowers that they have so far discovered. The bees are shown as little arrows moving around the space. They are red when searching for new flower sources, and so they are pointing in all directions. But when they know where to go they head out directly from the hive, shown in the Figure as a black colour; and when returning to the hive laden with nextar they are blue.

In the simulated environment, the bees can inhabit one of three habitats; where the food sources are "few" (with only two flowers), or "intermediate" with four flowers, or "most" with eight flowers (as in Figure 2).

There are other key variables to specify the bee behaviours, as the program code concults their values at various points. The experimenter can manipulate these control variables, using the slider controls at the side. These include the variables time-to-dance and time-to-wait-for-dance that appear in the code above. There are many other variables that the experimenter can use to change the characteristics of the bees and habitat, such as how fast they eat, how much energy the flowers have, and how many there are at different locations, and so on, but we have not varied those for this experiment.

One crucial variable is the Boolean variable for waggle? which controls whether the bees are able to communicate using the waggle dance or not. The controls for distance? and for direction? allow the experimenter to specify what kind of information the bees are able to convey with the waggle dance.

A convenience of the netlogo system is that it allows charts to be drawn in real-time as the simulation progresses. The total energy the bees have accumulated in the hive is shown in the Figure. In this example, the bees expended a lot of energy until they found some flowers and were able to start collecting nectar.

4.1 Testing the hypothesis

Our hypothesis envisaged that the effectiveness of the waggle dance would differ over varying habitats consisting of different quantities of flowers, particularly with environments containing small amounts

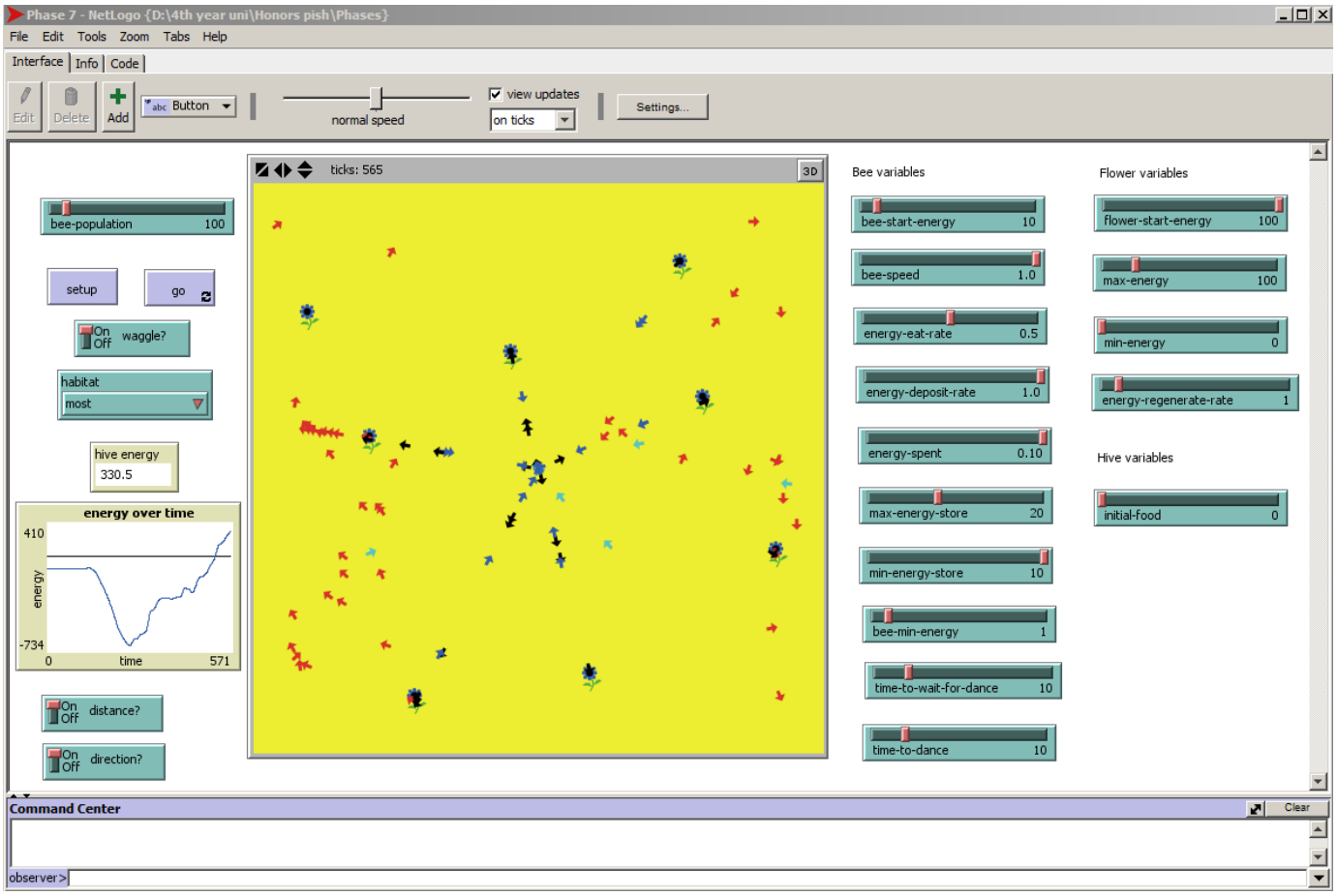

Figure 2. The model as seen by the experimenter

of food. We expected to see a reduced effectiveness for the waggle dance in habitats that contain a large supply of food. The reasoning behind this hypothesis was that as long as there were enough bees foraging, there would be a higher probability of bees finding flowers on their own, even if not told about them. We had originally planned to test this theory by using the net food content of the hive as a measurement. However it became apparent that the foraging success of the hive would not determine the effectiveness of the waggle dance as the food in the hive could have been contributed by bees that were not relying on the dance to find their food. A method had to be created that allowed the system to calculate the percentage of unsuccessful foraging bees that witnessed a dance on returning to the hive to feed. This was done by adapting the method `check-for-dance` to keep track of every time a bee had successfully witnessed a dance and every time it was unsuccessful (through incrementing respective variables). This allowed a value to be calculated that represented the percentage chance that observing bees were witnessing a dance. This was displayed through a monitor in the ABM to provide instant feedback to the user.

In order to evaluate the hypothesis, three separate experiments were then set up with the waggle dance switch activated. Each experiment would differ through the habitat selected by the `habitat` selector in the user interface. It was important that all of the slider variables remained the same throughout each experiment otherwise they could have had an effect that would have invalidated the results. Each experiment was executed for 1000 ticks (time units) before the result was recorded, in order to allow sufficient time for the bees to forage and improve the legitimacy of the experiment, as experiments running for different times would not have produced a fair comparison. We also increased validity by repeating each experiment three times to achieve an average result. This decreased the chance of an improbable result occurring that would provide a false representation of how the model behaves.

Table 1. The chance that a bee will see a waggle dance in poorer and richer habitats

Habitat	Test 1	Test 2	Test 3	Average	increase
few	48%	60%	57%	55%	
intermediate	69%	56%	67%	64%	9%
most	78%	89%	81%	83%	19%

Table 1 displays the results that were recorded during the experiments. The column on the left indicates the habitat that was being tested ("few" was the habitat containing the least flowers and increasing through "intermediate" to "most" having the most flowers in the habitat) and the first, second and third columns represent the

test number for that habitat before the final column calculates the average value over each habitat.

The results show that the chance of a bee witnessing a dance increases as the number of flowers increases. Another point to note is that this percentage chance seems to grow as the number of flowers increases. This is highlighted by an increase of only 9% from the “few” habitat to the “intermediate” habitat. However the increase from “intermediate” to “most” more than doubles to 19% supporting the hypothesis that the success of the waggle dance increases as the number of flowers increases. If we had come up with mathematical equations to model all this bee behaviour with some analytic solution, then it would be possible to prove such hypotheses mathematically, and with precision limited only by the modeling assumptions. However, the computational approach has its own advantages, which are illustrated in this example of an agent based model (ABM).

4.2 Testing the waggle dance direction

In order to investigate the efficacy of the waggle dance, we ran the simulation with it turned on or off; and with the direction component on or off. Bees can communicate both direction and distance to the food sources, but in a simulation we can control which information the artificial bees can communicate to each other. We experimented with this by allowing the bees to communicate both direction and ditance (so full waggle dance on); just direction; just distance; and neither direction nor distance (so equivalent to having no waggle dance at all).

Without the waggle dance the hive should clearly perform less effectively, as we know it is useful to real honeybees in the wild. Where a real colonoy could well die however, in this simulation we show the efficacy of the dance by measuring time to break even. As the bees expend energy while out searching and foraging, they need to eat honey when they return to the hive. therefore the energy level of the hive starts to fall soon after the simulation begins; and continutes to fall until enough flowers have been found by enough bees to be able to replenish the food stocks faster than they were being depleted. As the energy level starts at zero, by our convention, this means the the hive goes into netagive energy, or an energy debt. We do not "kill" the hive, but let it continue in a state of deepening negative energy until it manages to repay the debt. We take the time required in simulation "steps" to bring the hive back up to its original zero energy level, plus a small value of 100 energy units, as the measure of its success, and then stop the simulation. The quicker the hive can recover its energy then, the more successful it is.

Table 2. The time steps needed before the colony manages to earn energy to take it above its starting level

waggling habitat	on	distance	direction	off
few	1793	3818	6445	9131
intermediate	1031	1665	2889	3430
most	726	758	1109	988

In Table 2 the effectiveness of the different components of the waggle dance are shown in the simulation. Each cell in the table results from three runs that were averaged, as the simulations are all slightly different every time they run. It can be seen that the waggle dance is indeed effective in the simulator, as the hive repays its energy debt sooner when it is on. Of the two components, it is apparent that direction is more important information to convey than distance. But even with the waggle dance turned off, the colony can still recover its debt eventually. This is because the flowers are not depleted in the simulations, and so eventually anough bees will find them on their own, and then all bees are earning more eergy than they consume.

When the environment provides rich food resources, the experiment shows that the waggle dance brings less benefit to the colony. This is understandable as the bees can find the flowers more easily even if they have no help. This suggests that the waggle dance may have developed as a solution to help colonies that fell on hard times, such as having to face harsh environmental conditiouns brought about by drought. We make no further comment on evolution, save to say that modeling approaches such as the present one may well offer helpful paths to investigation that would not be open to biologists working with real creatures in the wild.

In reality a colony would not start with unlimited energy resources of course, and could fail to survive long enough for the bees to find the flowers. The artificial bees here do not die either; but in reality honeybees can forage effectively for only a few weeks, before their flight muscles give out, and they fall to earth where they are, to be eaten by birds, or die of exhaustion. Furthermore, the waggle dance for real bees doe not precisely convey the information about distance and direction; but is rather approximate. The simulator does demonstrate however, within its scope of simulation, and with its assumtpions of relatively precise information transfer, that the potential benefit to the colony of the waggle dance is great.

5 Conclusion

We have shown how the ABM (agent based modelling) research method can be applied to biological systems in which the individual agents are simple enough. Whether the artificial results are generally useful remains to be seen. In this case, honeybees could be modelled as FSMs (finite state machines), at least for the present simulation purposes. In reality, honeybees have more intelligence that is simulated here, and their behaviours would in principle be simulated with more complex FSMs, but beyond a certain level of complexity some other techniques would be more appropriate. While FSMs can in principle simulate anything, they are not so convenient for more complex systems. Their suitability for this level of description, however, has been shown to be good.

The scientific interest of the model comes in the way that the agents then interact with each other, producing much greater complexity, and emergent behaviour patterns that would not be obvious from consideration of the structure of an individual agent herself. Such complexity precludes any premature mathematical treatment, and we suggest that the first approach these days should be with a computational model, and any precise analytical solutions can be developed later, once the main ideas are explored and established.

While it is understood that many phenomena in the natural world are "emergent" and especially so in biology, it remains a struggle to understand the nature of this emergence in any given case. It is to fill this kind of conceptual gap that we feel ABM is suited. The experimental approach that is encouraged by the system developed here allows the experimenter to develop intuitions about how the whole system behaves. Then any apparent lawlike behaviour that emerges may be modeled mathematically, with analytical precision, and to that extent may be confirmed by other means. Even if such formalisation is not achieved however, the ABM approach will still have provided a good if rough-and-ready understanding; and it will still allow predictions to be made that can be tested in principle with real world systems.

To show how experimentation can proceed, we ran several experimental runs, placing the artificial bees in richer and poorer environments. The hypothesis that richer environments would make the waggle dance more effective was borne out in these experiments, although more comprehensive runs would been to be done to see how far the effect would go before reaching saturation.

Other experiments are also clearly possible using the developed model, but we have only sought to make the case for doing this kind of ABM work in general for animal cognition, and not only for bees. ABM can be applied to anything from slime moulds to great apes, and we may expect it to give us some insights during development as well as when the model is tested under different conditions. The Netlogo system used here has proved to be a capable platform which could feasibly be used by biologists say, without deep programming knowledge. The emphasis in the user interface is on interactive controls, allowing the experimenter to play around with the variables and develop intuitions, before going on to conduct more formal computer experiments.

There remains a final question as to the scientific status of such ABM models in biology. They are not traditional, as are systems of mathematical equations; but we hope to have shown that is only because the technology to make such a research path more negotiable for anybody but computer scientists has taken a long time coming. But now that it has arrived we can look forward to a bright future for zoology in general and especially for animal cognition, with this additional research method. The ABM model may take its place in the general scientific method, alongside the observation, the natural experiment, the hypothesis, the laboratory controlled experiment, the statistical analysis technique, and the mathematical equation. We may one day start to talk of "artificial experiments" to emphasise the similar role that ABM methods have in animal and other cognition, as we have done here. While it might be controversial to speak in such a way for some time, there will proobably be some researchers in AI and zoology that find it quite a comfortable way to think of what this kind of work does.

REFERENCES

[1] Jared M. Diamond, 'Archaeology: Life with the artificial anasazi', *Nature*, **419**(6907), 567–569, (2002).

[2] James L Gould, Carol Grant Gould, et al., *The honey bee.*, Scientific American Library, 1988.

[3] Jae-Woo Kim and Robert Hanneman, 'A computational model of worker protest', *Journal of Artificial Societies and Social Simulation*, **14**(3), 1, (2011).

[4] Franziska Klügl and Ana LC Bazzan, 'Agent-based modeling and simulation', *AI Magazine*, **33**(3), 29, (2012).

[5] Ellouise Leadbeater and Lars Chittka, *Social Information Use in Foraging Insects*, 135–146, Food Exploitation By Social Insects, Informa UK Limited, 2009.

[6] Steven F Railsback and Volker Grimm, *Agent-based and individual-based modeling: a practical introduction*, Princeton university press, 2011.

[7] Nick Scott, Michael Livingston, Aaron Hart, James Wilson, David Moore, and Paul Dietze, 'Simdrink: An agent-based netlogo model of young, heavy drinkers for conducting alcohol policy experiments', *Journal of Artificial Societies and Social Simulation*, **19**(1), 10, (2016).

[8] Thomas D Seeley, *The wisdom of the hive: the social physiology of honey bee colonies*, Harvard University Press, 2009.

[9] Jan C. Thiele, Winfried Kurth, and Volker Grimm, 'Agent-based modelling: Tools for linking netlogo and r', *Journal of Artificial Societies and Social Simulation*, **15**(3), 8, (2012).

[10] U Wilensky, 'Netlogo wolf sheep predation model', *Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL*, (1997).

[11] Uri Wilensky and I Evanston, 'Netlogo: Center for connected learning and computer-based modeling', *Northwestern University, Evanston, IL*, 49–52, (1999).

[12] Uri Wilensky and Kenneth Reisman, 'Connectedscience: Learning biology through constructing and testing computational theories-an embodied modeling approach', *Learning*, **3**, 50, (1909).

[13] Uri Wilensky and Kenneth Reisman, 'Thinking like a wolf, a sheep, or a firefly: Learning biology through constructing and testing computational theories-an embodied modeling approach', *Cognition and Instruction*, **24**(2), 171–209, (2006).