Rigorous Analysis of Idealised Pathfinding Ants in Higher-Order Logic

Rigorous Analysis of Idealised Pathfinding Ants in Higher-Order Logic

Marco Maggesi, Cosimo Perini Brogi

Rigorous Analysis of Idealised Pathfinding Ants in Higher-Order Logic

Marco Maggesi[1[0000−0003−4380−7691]]
Cosimo Perini Brogi[2[0000−0001−7883−5727]]

[1] University of Florence, Italy
marco.maggesi@unifi.it
[2] IMT School for Advanced Studies Lucca, Italy
cosimo.perinibrogi@imtlucca.it

Abstract. This paper presents a modelling framework for an idealised system of foraging ants, using Higher-Order Logic (HOL) and implemented in the HOL Light proof assistant. Exploiting the expressive capabilities of HOL, we create a detailed, principled model that describes individual ant behaviours to explore long-term dynamics and formally verify the emergent property of the colony we are interested in, namely shortest path finding.
Using HOL Light guarantees rigorous verification of the model, also confirming the simulation accuracy. We present our results as highlights for the potential of formal computer mathematics in studying complex adaptive systems. By merging formal methods with complex systems science, we aim to explore emergent behaviours in biological and artificial systems with mathematical precision and reliability.

Keywords: Logical verification · Pathfinding ants · HOL Light · Emergent behaviours · Agent-based modelling · Formal methods

1 Introduction

A colony of ants at work captures the interest of observers of any age. What fascinates us is a certain level of *order* and *regularity*, almost a method or deliberate coordination, which progressively emerges automatically from the seemingly disordered bustle that initially grabs our primate attention.

As human beings with scientific backgrounds, some call this order the *self-organisation of a distributed natural system*, and ponder the principles underlying such a phenomenon.

The behaviour of an individual ant appears simple, and myrmecology confirms it to be purely instinctive, without the mediation of any form of reasoning. Indeed, we know that ants lack the sufficient nervous system structure to develop complex behaviours typically attributed to, for example, mammals and some species of birds.

Nevertheless, the colony manifests complex and structured collective behaviours that allow ants to brilliantly accomplish tasks essential for the colony's survival and the species' as well. These tasks often exceed the capabilities of an individual ant and likely surpass the mere sum of the colony's abilities. Among them are brood sorting, cooperative transport, division of labour, nest building (sewing leaves together), fungus farming, and aphid husbandry.[3]

In this article, we focus on one collective behaviour typical to a large part of the *Formicidae*: solving the problem of finding the shortest path to a food source, thereby foraging efficiently for the nest.

Their ability to coordinate through indirect communication mediated by modifications to their surroundings enables ants to excel in this task (and in others mentioned previously). In biology [22,46,21], and more recently in computer science [14,48,8,47,34], this method is referred to as *stigmergy* [27,28,4].

Biologists have demonstrated that ants, like other eusocial insects, achieve the levels of self-organisation necessary for these and other remarkable feats solely through stigmergy.

For ants, communication between individuals and their understanding of the environment occurs through releasing pheromones. For our purposes, the relevant pheromone is the so-called trail pheromone, which some species, such as *Iridomyrmex humilis*, use to mark paths on the ground during foraging. By chemically sensing these pheromone trails, foragers can follow the path other ants (the scouts) laid down to the food source. By detecting the pheromone left by the latter, they probabilistically choose paths marked by higher concentrations of the pheromone [2,13,43,5,3].

Double bridge experiment. Biologists have observed this pheromone-trail-and-following behaviour not only in nature but also in controlled experiments.

Here, we are interested in the classic double bridge experiment [21,12].

A double bridge connects a nest N of *Iridomyrmex humilis* to a sugary food source F. Biologists experiment by varying the ratio $r = \frac{l}{s}$ between the lengths of the two branches of the double bridge, where l is the length of the longer branch and s that of the shorter branch.

When $r = 1$, the ants can move between N and F. Although choices are initially random, all ants use the same branch over time. This phenomenon occurs because, at time t_0, there is no pheromone on the branches, so the ants cannot prefer one branch over the other: only due to random fluctuations a small number of ants will choose one branch over the other. Given their instinct to deposit trail pheromones as they travel, this random imbalance results in more pheromones on the branch favoured by chance. This increase, in turn, stimulates other ants to choose that branch again, and so on, until the foragers *converge* on a single path. In literature, this phenomenon is referred to as *positive feedback* (among computer scientists) or *auto-catalysis* (among biologists).

When $r = 2$, however, most trials show that after some time, all the ants start using only the short branch. In this scenario, at time t_0, the ants continue

[3] Different ant species undertake these tasks; not all species perform *all* these activities. We collected this information from the classic and thought-provoking monograph [29], to which we refer for further myrmecological details.

to choose randomly between the two branches, but since one is half the length of the other, the ants that choose it reach the food first and return more quickly to the nest. Thus, when they again face the choice between the two branches during foraging, the higher level of pheromone biases their choice towards the shorter branch, where even more pheromone accumulates more quickly until all the ants converge on this shortcut thanks to the auto-catalysis mechanism discussed in the previous scenario. In this case, however, the initial random fluctuation plays no role in the experiment's outcome.[4]

Our contribution. In this paper, we model a discrete, abstract, and idealised version of the second scenario in the double bridge experiment previously discussed. Our model is illustrated in the Figure 1.

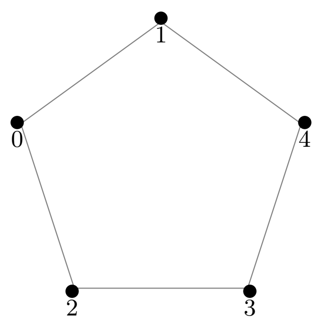

Fig. 1. The discrete model for the idealised version of the double bridge experiment

The environment is represented by a simple graph shaped like a regular pentagon, with the node labelled 0 as the nest and the node labelled 4 as the food source. We assume that ants move discretely between adjacent nodes in unit steps, such that the ratio r between the path 0–1–4 and 0–2–3–4 is 2, mirroring the laboratory experiment.

We model this idealised system using *higher-order logic* and implement the *modelling, simulation, and verification* of the colony's dynamics within the proof assistant HOL Light [24,23,25].

Ours is a new and principled approach, applied here for the first time to this well-known case study in biological, computer science, and engineering literature, aiming for the most rigorous verification of the model's fundamental properties. It suggests using proof assistants as a unified environment for the formalisation

[4] It is noteworthy that if the colony, which has converged on the shorter branch, is provided with an additional, quicker shortcut between N and F, the new path is only occasionally chosen, and the ants remain on the previously discovered, now sub-optimal, path. Biology explains this phenomenon in terms of pheromone evaporation rates, which are too slow on the path selected through auto-catalysis to allow subsequent exploration of alternatives.

and verification of behaviours in this and other complex adaptive systems and for conducting functional simulations of their dynamics.

The methodology we propose here intends to supplement the rigorous engineering of complex (adaptive) systems with the exactitude, reliability, and expressiveness of mathematical logic (more precisely, type theory), which underpins the main proof assistants currently used in academic and industrial settings for software and hardware verification [26,15,1].

Our approach is based on the principle of compositionality, which has become prominent in various disciplines dealing with complex adaptive systems, including ecology and economics. At the same time, we have endeavoured to maintain the bottom-up methodology initiated by [7,8,9,10,11], focusing on modelling the logic/behaviour/actions of individual components of the system. We then describe the essential collective dynamics based on these individual elements, explore the emergence of more complex behaviours through recursive functional simulation of these basic dynamics, and verify the expected collective behaviour – i.e., shortest path finding – in *colonies of any size* under reasonable preconditions.

Unlike the existing literature we know, our verification of this specific form of colony self-organisation is presented as formal proof of a precise mathematical theorem. This theorem demonstrates the emergence and stability of this behaviour for any ant colony in the idealised environment described, regardless of the number of agents involved, despite having been described in terms of the individual dynamics of those agents.

Source code. Each source code snippet in this paper includes a link – marked by the word 'sources' on the right – to a copy of the code stored in our GitHub repository.[5]

In the main body of the paper, we present several code fragments to illustrate our development and provide additional documentation for readers interested in technical details. We have partially edited the code to enhance the readability of mathematical formulas. For instance, we have replaced HOL Light's purely ASCII notations with conventional graphical notation. We report HOL terms and types enclosed in backquotes, as in `` `1+1` `` and `` `:bool` ``. Furthermore, we report theorems with their associated name (the name of its associated OCaml constant), and we write their statement prefixed with the turnstile symbol ($\vdash$).

In the expository style, we omit formal proofs, but the meaning of definitions, lemmas, and theorems in natural language is clear.

Paper structure. The paper is structured as follows. In Section 2, we introduce the basic formalisations for the environment (i.e., five possible positions in the discrete model of Figure 1) and for the individual ants (described as a pair of attributes, for their position and direction during the foraging, resp.). In Section 3, we develop our formal specifications for the system made of ants

[5] This may be particularly useful for the reader who wishes to examine the raw code and understand how it integrates into its original context.

and the stigmergy, for the behaviour of each ant composing the colony, and for the abstract dynamic of the colony itself in terms of its components. Next, in Section 4, we explain how to explore the dynamics of a fixed-size colony by implementing a simulation mechanism in HOL Light performing iterated conversions of the terms representing single-step evolution and system configurations. Finally, in Section 5, we formally prove the convergence of the foraging ants on the shortest path under specific conditions, assessing the emergence of the expected collective behaviour of the colony. Section 6 provides concluding remarks and discusses related work.

2 Basic formalisations

HOL Light is a proof assistant based on classical higher-order logic, crucial for rigorous mathematical and computational reasoning. It uses polymorphic type variables and treats equality as its only primitive notion. Its logical engine comprises term-conversions and inference rules, enhanced by an infinity axiom, function extensionality, and Hilbert's choice operator.

Practically, HOL Light adopts a procedural approach to proof construction, using tactics to break down goals into manageable subgoals. These steps are refined into concise proof scripts through tacticals, allowing efficient and rigorous machine verification.[6]

Terms in HOL Light denote values, functions, or predicates and are basic ingredients to proof development. Function application and lambda abstraction are common operations for constructing terms in this typed functional system.

Types classify terms, ensuring logical consistency in expressions. They define the values a term can take and how functions can be applied, following rules based on the system's inference principles.[7]

2.1 Environment

For instance, to model the discrete environment we set the experiment on, we introduce a type `` `:position` ``, which is inhabited by the nodes of the pentagon in Figure 1, denoted in the formal setting by `P0`, `P1`, `P2`, `P3`, `P4`.

On the contrary, we model the stigmergy of the environment using natural numbers: the environmental information `STI` is given by a vector of three natural numbers `num^3`, which represent the pheromone levels in each of the intermediate nodes between the nest `P0` and the food source `P4`.

[6] For further information, refer to the official documentation [23,24].

[7] These principles create conservative extensions of the fundamental system, defining new entities by modelling them within the existing theory, ensuring new constants (terms or types) are merely definitional extensions of the preexisting structures. For more on this methodology, see [19,20].

2.2 Ants

To model ants, we adopt the attribute-based approach proposed in [9,10], that we rephrase in our specific formal setting.

Henceforth, an ant is determined by a pair $(\mathtt{p}, \mathtt{d})$ consisting of a position `` `p:position` `` (the node that the ant is currently occupying in the graph) and a boolean `d:bool` (denoting the direction it is moving on the pentagon: from the nest to the food source, or vice versa).

The type `` `:ant` `` abbreviates then the Cartesian product `` `:position×bool` `` of the attributes qualifying an agent as an ant.

3 Specification

In this section, we present our mathematical description of the discrete model. We will use either the declarative or the algorithmic style for the system specification, depending on which is more transparent and concise.

3.1 System description and pheromone release

Let us start by introducing the minimal system description that is given by a new type `` `:system` `` consisting of an `N`-ary vector of ants (where `` `:N` `` is a type parameter counting the number of ants inhabiting the colony) and a vector of three natural numbers (for the stigmergy).

Next, we define the code computing the pheromone levels for each intermediate node in the graph. In mathematical terms, we write

$$s'(a, s)_p = s_p + \sum_{i=1}^{N} \delta_{\mathsf{pos}(\mathsf{a_i}),\mathsf{PP}(\mathsf{p})}$$

where s_p is the level of pheromones at the position p, $\mathsf{pos}(a_i)$ is the position of the ant a_i and $\delta_{x,y}$ (the Kroneker symbol) is 1 if $x = y$ and 0 otherwise.

In HOL Light, this is formally rendered as

```
NEW_STI_COMPONENT                                        (source)
⊢ ∀p. 1 ≤ p ∧ p ≤ 3
      ⟹ NEW_STI (System ant sti)$p =
          sti$p +
          nsum (1..dimindex(:N))
               (i ↦ if FST(ant$i) = PP(p) then 1 else 0)
```

where `` `nsum s f` `` is the sum of all `` `f(i)` `` for all `` `i` `` in `` `s` ``, `` `v$i` `` is the i-th component of the vector `` `v` ``, and `` `dimindex(:N)` `` is the cardnality of the type `` `:N` ``.

This specification encodes the pheromone release by each ant in each of the intermediate nodes ($\forall p.\ 1 \leq p \wedge p \leq 3$): in our idealised context, we thus assume that the level is locally risen discretely, by adding one unit of pheromone for each ant on the specific node.[8]

3.2 Behaviour of an ant

We can now write the formal specification describing the individual logic/behaviour/dynamics of each ant in the colony:

```
NEW_ANT                                                     (source)
⊢ NEW_ANT sti (pos,dir) =
  (if pos = P1 then {((if dir then P4 else P0),dir)} else
   if pos = P2 then {((if dir then P3 else P0),dir)} else
   if pos = P3 then {((if dir then P4 else P2),dir)} else
   if pos = P0
   then {pos,T | sti$2 ≤ sti$1 ∧ pos = P1 ∨
                 sti$1 ≤ sti$2 ∧ pos = P2}
   else {pos,F | sti$3 ≤ sti$1 ∧ pos = P1 ∨
                 sti$1 ≤ sti$3 ∧ pos = P3})
```

The listing states that an ant cannot change its direction halfway during foraging (first three nested guards). However, when it is on the nest node or has reached the food source node, it chooses the next position (for reaching the food, or bringing it back to the nest, resp.) based on the levels of pheromone on each potential next node (last two nested guards), opting for the one with higher concentration.[9] Notice that here we handle the non-deterministic choice between the two potential next positions using set-theoretic constructions that enable us to describe the general system dynamics and its alternative evolutions in terms of this individual behaviour of the single ants.

3.3 Compositional dynamics

The specification of the evolution of the whole system according to the individual dynamics specified by `NEW_ANT` is naturally stated: a colony evolves in any possible system (i.e., any potential distribution of the ants inhabiting the colony and correlated stigmergy) that complies with the individual specification of the foraging ants. The corresponding code formally defines the set of systems obtained after such an evolution:

```
NEW_SYSTEM                                                  (source)
⊢ NEW_SYSTEM sys =
  {System ant' (NEW_STI sys) | ant' |
       ∀i. 1 ≤ i ∧ i ≤ dimindex(:N)
           ⟹ ant'$i ∈ NEW_ANT (STI sys) (ANT sys$i)}
```

[8] The same assumption is made in [11, Listing 7], and a similar, though more refined, one is in [10, Listing 4].

[9] Contrary to [11], we do not consider a parametric difference in the pheromone concentrations of the potential destinations.

4 Simulation

We are now interested in simulating the previously defined dynamics and exploring the possibility of the emergence of the expected collective behaviour of the system, namely the convergence of a given colony to the shortest path for foraging.

We aim to perform such a simulation without recurring to external tools and check whether it is possible to manage this task effectively in HOL Light.

First, we must translate the previous declarative specification (intuitive for the human reader) into a completely algorithmic description. We do that semi-automatically.

We start by proving some theorems that make the computational nature of the previous general, intuitive and abstract definitions explicit. Then, we combine the "semi-procedural" specifications, which are obtained via conversion mechanisms that implement a definite call-by-value evaluation involving both the logical constructions of the proof assistant and the specific results for our scenario. The resulting code provides a system description that the logical engine of HOL Light can manipulate (with different levels of automation) as a purely functional expression.

4.1 Exploring the behaviour of a minimal colony

To make our simulation methodology more transparent, we show how to explore in HOL Light the evolution of a minimal system consisting of two foraging ants in the pentagon.

```
NEW_SYSTEM_2                                                  (source)
⊢ NEW_SYSTEM (System (vector[pos1,dir1; pos2,dir2])
                     (vector[s1; s2; s3])) =
    IMAGE (a ↦
      System (vector a)
             (vector[s1 + (if pos1 = P1 then 1 else 0) +
                          (if pos2 = P1 then 1 else 0);
                     s2 + (if pos1 = P2 then 1 else 0) +
                          (if pos2 = P2 then 1 else 0);
                     s3 + (if pos1 = P3 then 1 else 0) +
                          (if pos2 = P3 then 1 else 0)]))
    (SETBIND
       (l ↦
          if pos1 = P1 then
            {CONS ((if dir1 then P4 else P0),dir1) l}
          else if pos1 = P2 then
            {CONS ((if dir1 then P3 else P0),dir1) l}
          else if pos1 = P3 then
            {CONS ((if dir1 then P4 else P2),dir1) l} else
          if pos1 = P0 then
```

```
          (if s2 ≤ s1 then {CONS (P1,T) l} else ∅) ∪
          (if s1 ≤ s2 then {CONS (P2,T) l} else ∅)
        else
          (if s3 ≤ s1 then {CONS (P1,F) l} else ∅) ∪
          (if s1 ≤ s3 then {CONS (P3,F) l} else ∅))
    (if pos2 = P1 then {[(if dir2 then P4 else P0),dir2]} else
     if pos2 = P2 then {[(if dir2 then P3 else P0),dir2]} else
     if pos2 = P3 then {[(if dir2 then P4 else P2),dir2]} else
     if pos2 = P0
     then (if s2 ≤ s1 then {[P1,T]} else ∅) ∪
          (if s1 ≤ s2 then {[P2,T]} else ∅)
     else (if s3 ≤ s1 then {[P1,F]} else ∅) ∪
          (if s1 ≤ s3 then {[P3,F]} else ∅)))
```

In this code, we are “explaining” the operations that are performed by the function NEW_SYSTEM transforming an input system into the collection of potential next configurations of it according to the individual dynamics of the two ants in the colony.[10]

4.2 Simulation run

The lemma NEW_SYSTEM_2 provides the central equation for translating the intuitive specification of the two-ant colony dynamics, as embodied by the abstract definition NEW_SYSTEM, into an algorithmic definition of the system update. By running a recursive call of NUM_SYSTEM over an initial state of the colony, the machine can evaluate the concrete values that describe the potential states of the colony at a specific point in the dynamics by converting into NEW_SYSTEM_2 each occurrence of NEW_SYSTEM within the (intermediate terms specifying the) colony evolution. Thus, the simulation output reduces to a (potentially large) collection of results of if-then-else statements.

To show how the simulation is performed, we call 30 iterations of the update function NEW_SYSTEM on an initial configuration where the stigmergy is null overall, an ant is on P1 moving towards the food source, and the second one is on P2, returning to the nest:

```
 |- ITER 30 (SETBIND NEW_SYSTEM)
            {System (vector[(P1,T); (P2,F)])
                    (vector[0; 0; 0])} =
    {System (vector[P1,F; P1,T]) (vector[29; 1; 0]),
     System (vector[P1,F; P0,F]) (vector[28; 2; 1]),
     System (vector[P4,T; P0,F]) (vector[27; 3; 2]),
     System (vector[P4,T; P1,F]) (vector[25; 4; 3]),
```

[10] In a sense, we are revealing the operational nature of the intensional and conceptual definition of the system dynamics given in Section 3.3, bridging our intuitive description, formally rendered by NEW_SYSTEM, and the effective procedure that we expect the machine has to perform to simulate that description.

```
System (vector[P1,T; P1,F]) (vector[23; 5; 4]),
System (vector[P1,T; P4,T]) (vector[22; 6; 5]),
System (vector[P0,F; P4,T]) (vector[21; 7; 6]),
System (vector[P0,F; P1,T]) (vector[19; 8; 7]),
System (vector[P1,F; P1,T]) (vector[17; 9; 8]),
System (vector[P1,F; P0,F]) (vector[16; 10; 9]),
System (vector[P4,T; P0,F]) (vector[15; 11; 10]),
System (vector[P4,T; P1,F]) (vector[13; 12; 11]),
System (vector[P3,T; P1,F]) (vector[12; 13; 11])}`
```

The right-hand side of the equation represents the collection of possible system configurations after 30 steps of evolution. This equality is *formally proven* in approximately 7 seconds of automated computation in HOL Light, running on a mid-level personal computer, without needing external resources.

5 Logical verification

This section discloses the most relevant potential of our methodology for the rigorous engineering of adaptive systems based on proof assistants.

In the following pages, we show that we can prove formally that the foraging ants converge to the shortest path between the nest and the food source *independently of the colony size* whenever a reasonable precondition is met. This way, verifying the emergence of the expected collective behaviour reduces to checking that its preconditions – as identified by our theorems and less demanding from the perspective of reachability analysis – are met by the system under consideration.

5.1 Minimal invariant: stigmergy preservation

Given an ant system of any size, let us consider the following property: the pheromone concentration of the top node 1 of the pentagon is higher than the pheromone levels of the remaining intermediate nodes 2 and 3. We name it ‘stigmergy imbalance property’ and define it formally as

```
INVARIANT_STI                                              (source)
⊢ ∀sti. INVARIANT_STI sti ⇔ sti$1 > MAX (sti$2) (sti$3)
```

Next, we consider a sort of “2-step conservation principle” for that property, namely that if the stigmergy of a given system satisfies `INVARIANT_STI`, then the stigmergy of any system evolving out of it still satisfies that property, and so does any system evolving out of the latter. Formally, we write

```
INVARIANT                                                  (source)
⊢ ∀sys. INVARIANT sys ⇔
         ∀s t. s ∈ NEW_SYSTEM sys ∧
               t ∈ NEW_SYSTEM s
               ⟹ INVARIANT_STI (STI sys) ∧
```

```
                    INVARIANT_STI (STI s) ∧
                    INVARIANT_STI (STI t)
```

Then, we can prove the following theorem:

> The higher pheromone concentration on the shortest path is preserved by the evolution of any system of foraging ants satisfying the 2-step conservation principle for stigmergy.

The formal statement for that theorem in HOL Light is

```
INVARIANT_THM                                                    (source)
⊢ ∀sys sys'. INVARIANT sys ∧
             sys' ∈ NEW_SYSTEM sys
             ⟹ INVARIANT sys'
```

5.2 Collective convergence

We are finally ready to prove that the foraging ants find the shortest path between the nest and the food source, two evolution steps after the stigmergy imbalance property is met.

The convergence on the shortest path is formally rendered as

```
INVARIANT_ANT                                                    (source)
⊢ ∀ant. INVARIANT_ANT ant ⇔
        (!i. 1 ≤ i ∧ i ≤ dimindex(:N) ⟹ FST (ant$i) ∈ {P0, P1, P4})
```

Henceforth, the main theorem states the following:

> The convergence of the ants on the shortest foraging path emerges after two evolution steps from any foraging ant system satisfying the stigmergy imbalance and evolving in one step only into systems that preserve that property.

The formal counterpart in HOL Light is given by

```
INVARIANT_ANT_THM                                                (source)
⊢ ∀sys sys' sys''.
       sys'  ∈ NEW_SYSTEM sys ∧
       sys'' ∈ NEW_SYSTEM sys' ∧
       INVARIANT_STI (STI sys) ∧
       INVARIANT_STI (STI sys')
       ⟹ INVARIANT_ANT (ANT sys'')
```

whose formal proof is tweaked from that for the stigmergy invariant theorem.

6 Conclusions and related work

The ability of foraging ants to find the shortest path between a food source and the nest is a simple example of bio-inspired problem-solving. It has incepted new optimisation methods and (meta-)heuristic design [14,47]. It also provides a case of adaptive self-organisation typical of complex natural systems that we have observed long before the advent of that field of study [10,30,31,2,48].

In this paper, we have modelled, simulated, and verified the emergence of this collective behaviour in colonies of arbitrary size within a discrete, abstract, and idealised environment, using the proof assistant HOL Light [25].

Using this specific case study, we have introduced a new framework for formalising and analysing complex adaptive systems centred on modern proof assistants' capabilities. Based on the linguistic expressiveness and deductive robustness of type theory, these tools can be used uniformly and effectively in this research area.

In our work, we followed a principle of compositionality inspired by the recent papers [7,8,9,10,11], adapting their methodology to the formal tools we have chosen to use for similar purpose.

As mentioned, our analysis of pathfinding ants has been conceived as an initial experiment in the logical verification of complex adaptive systems. We propose that our approach to modelling, simulating, and verifying complex adaptive systems through the rigorous tools of mathematical logic and proof assistants is, in principle, feasible. There is, of course, ample scope for further work and refinement of these initial results, which we plan to pursue in at least two directions:

- On the methodological side, we plan to better integrate the compositionality of our modelling with the more rigorous bottom-up approach embodied in [7,8,9,10,11]. This involves first developing a precise and sophisticated formalisation in HOL Light of the decentralised data structure corresponding to the virtual stigmergy of [8], along with a detailed library of mathematical results to be used for handling stigmergy at the automated and interactive level within our framework.
- On the technical and programming side, it is possible to enhance the performance of the conversion function used to simulate the long-term dynamics of colonies of fixed size. This improvement can be achieved within the proof assistant itself. However, we also find it interesting to experiment with a potential interface with SMT-based theorem provers, such as Z3 [35].This interface could distribute the workflow between a formal platform more oriented towards exploration and simulation (i.e., the automated theorem prover) and one dedicated to the mathematically rigorous verification of the expected properties of such simulations (i.e., the proof assistant). Such a division of labour within a uniform logical verification philosophy would facilitate extending these initial results to more complex and realistic phenomena of spontaneous self-organisation in other natural and artificial scenarios.

Related work. The idealised and discrete version of the original double bridge experiment we considered in this paper is derived from the paper [11], where they analyse this simple scenario as one of many examples of applying a bottom-up methodology for the specification, simulation, and verification of complex adaptive systems (CASs). This methodology is implemented through a high-level formal language (LAbS) for specifications, accompanied by a tool (SLiVER) for the automatic translation of these specifications into sequential imperative programs, which can then be subjected to advanced techniques of reachability analysis and (bounded) model checking [8,11]. Their approach seeks to harness the advantages offered by the inherent compositional nature of process algebras for providing an intuitive and high-level set of primitives and constructs for specifying collective systems, keeping the rigour of formal validations for the system dynamics.

Our modelling style is inspired by their bottom-up methodology, which we aim to translate into our working environment fully, incorporating the refinements mentioned earlier to handle better the distributed nature of their virtual stigmergy within HOL Light. We do not rule out replicating in our framework the transition from their discrete model in [11] to the more general and realistic one in their subsequent [10], where the foraging environment is represented by a grid delimiting a two-dimensional tape.

However, our analysis technique eschews the ingenious sequential emulation of specifications proposed in these works, instead verifying emergent behaviours through the formal interactive proof of mathematical theorems rather than automated reachability analysis and model checking.

The scenario of bumping ants on a bar studied in [16] in terms of causal chains and later reproduced in [11] within the "LAbS+SLiVER" paradigm, can be easily formalised within our framework. We consider the rigorous analysis of this scenario, particularly the precise deduction of the collision order among the ants as a meta-property of the system's dynamics, to be the next benchmark for our approach to studying CASs with a proof assistant. Once the previously mentioned extensions and refinements are made, it seems natural to compare our methodology with other works on similar systems, starting with the revised version [17] of the techniques (based on Symbolic Petri nets) used in [16] for the bumping ants on a bar.

Among further related works, it is worth noting that the literature abounds of agent-based mathematical models for foraging, as in [44,31]; works based on process-algebras techniques include [42,40] (using WSCSS [41]) and [33] (using Bio-PEPA [6]). At the same time, for simulation, we recall [36] (based on the programming language MASON [32]) and, in particular, [38] as an example of the noteworthy capabilities of LOGO and its derivatives, StarLogo and NetLogo [18,39,37,45], for system biology, simulation of self-organising natural systems, and scientific education of the child as well.

Acknowledgments. This work was partially funded by: the MIUR project PRIN 2017FTXR7S IT MATTERS (Methods and Tools for Trustworthy Smart Systems); the MIUR project PRIN 2017JZ2SW5 "Real and Complex Manifolds: Topology, Geometry and holomorphic dynamics"; the project SERICS PE00000014 (SEcurity and RIghts in the CyberSpace) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU; GNSAGA of INdAM.

Disclosure of Interests. The authors have no competing interests to disclose.

References

1. Baanen, A., Bentkamp, A., Blanchette, J., Hölzl, J., Limperg, J.: The Hitchhiker's Guide to Logical Verification (2024)
2. Bernstein, R.A.: Foraging strategies of ants in response to variable food density. Ecology **56**(1), 213–219 (1975)
3. Bonabeau, E.: Flexibility at the edge of chaos: a clear example from foraging in ants. Acta Biotheoretica **45**(1), 29–50 (1997)
4. Bonabeau, E., Dorigo, M., Theraulaz, G.: Inspiration for optimization from social insect behaviour. Nature **406**(6791), 39–42 (2000)
5. Bonabeau, E., Theraulaz, G., Deneubourg, J.L., Aron, S., Camazine, S.: Self-organization in social insects. Trends in ecology & evolution **12**(5), 188–193 (1997)
6. Ciocchetta, F., Hillston, J.: Bio-PEPA: A framework for the modelling and analysis of biological systems. Theoretical Computer Science **410**(33-34), 3065–3084 (2009)
7. De Nicola, R., Di Stefano, L., Inverso, O.: Toward formal models and languages for verifiable multi-robot systems. Frontiers Robotics AI **5**, 94 (2018). https://doi.org/10.3389/FROBT.2018.00094, https://doi.org/10.3389/frobt.2018.00094
8. De Nicola, R., Di Stefano, L., Inverso, O.: Multi-agent systems with virtual stigmergy. Sci. Comput. Program. **187**, 102345 (2020). https://doi.org/10.1016/J.SCICO.2019.102345, https://doi.org/10.1016/j.scico.2019.102345
9. De Nicola, R., Di Stefano, L., Inverso, O., Valiani, S.: Modelling flocks of birds from the bottom up. In: Margaria, T., Steffen, B. (eds.) Leveraging Applications of Formal Methods, Verification and Validation. Adaptation and Learning - 11th International Symposium, ISoLA 2022, Rhodes, Greece, October 22-30, 2022, Proceedings, Part III. Lecture Notes in Computer Science, vol. 13703, pp. 82–96. Springer (2022). https://doi.org/10.1007/978-3-031-19759-8_6, https://doi.org/10.1007/978-3-031-19759-8_6
10. De Nicola, R., Di Stefano, L., Inverso, O., Valiani, S.: Intuitive modelling and formal analysis of collective behaviour in foraging ants. In: Pang, J., Niehren, J. (eds.) Computational Methods in Systems Biology - 21st International Conference, CMSB 2023, Luxembourg City, Luxembourg, September 13-15, 2023, Proceedings. Lecture Notes in Computer Science, vol. 14137, pp. 44–61. Springer (2023). https://doi.org/10.1007/978-3-031-42697-1_4, https://doi.org/10.1007/978-3-031-42697-1_4
11. De Nicola, R., Di Stefano, L., Inverso, O., Valiani, S.: Modelling flocks of birds and colonies of ants from the bottom up. Int. J. Softw. Tools Technol. Transf. **25**(5), 675–691 (2023). https://doi.org/10.1007/S10009-023-00731-0, https://doi.org/10.1007/s10009-023-00731-0
12. Deneubourg, J.L., Aron, S., Goss, S., Pasteels, J.M.: The self-organizing exploratory pattern of the Argentine ant. Journal of insect behavior **3**, 159–168 (1990)

13. Deneubourg, J.L., Pasteels, J.M., Verhaeghe, J.C.: Probabilistic behaviour in ants: a strategy of errors? Journal of theoretical Biology **105**(2), 259–271 (1983)
14. Dorigo, M., Stützle, T.: Ant colony optimization: overview and recent advances. Springer (2019)
15. Ferguson, W.E., Bingham, J., Erkök, L., Harrison, J.R., Leslie-Hurd, J.: Digit serial methods with applications to division and square root. IEEE Transactions on Computers **67**(3), 449–456 (2017)
16. Fettke, P., Reisig, W.: Discrete models of continuous behavior of collective adaptive systems. In: Margaria, T., Steffen, B. (eds.) Leveraging Applications of Formal Methods, Verification and Validation. Adaptation and Learning - 11th International Symposium, ISoLA 2022, Rhodes, Greece, October 22-30, 2022, Proceedings, Part III. Lecture Notes in Computer Science, vol. 13703, pp. 65–81. Springer (2022). https://doi.org/10.1007/978-3-031-19759-8_5, https://doi.org/10.1007/978-3-031-19759-8_5
17. Fettke, P., Reisig, W.: A causal, time-independent synchronization pattern for collective adaptive systems. Int. J. Softw. Tools Technol. Transf. **25**(5), 659–673 (2023). https://doi.org/10.1007/S10009-023-00733-Y, https://doi.org/10.1007/s10009-023-00733-y
18. Feurzeig, W., Papert, S., Bloom, M., Grant, R., Solomon, C.: Programming-languages as a conceptual framework for teaching mathematics. ACM SIGCUE Outlook **4**(2), 13–17 (1970)
19. Gordon, M., Milner, R., Wadsworth, C.: Edinburgh LCF, Lecture Notes in Computer Science, vol. 78. Springer-Verlag (1979)
20. Gordon, M.J., Melham, T.F.: Introduction to HOL: A theorem proving environment for higher order logic. Cambridge University Press (1993)
21. Goss, S., Aron, S., Deneubourg, J.L., Pasteels, J.M.: Self-organized shortcuts in the argentine ant. Naturwissenschaften **76**(12), 579–581 (1989)
22. Grassé, P.P.: La reconstruction du nid et les coordinations interindividuelles chez bellicositermes natalensis et cubitermes sp. la théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs. Insectes sociaux **6**, 41–80 (1959)
23. Harrison, J.: The HOL Light System Reference. https://www.cl.cam.ac.uk/~jrh13/hol-light/reference.pdf (2016)
24. Harrison, J.: HOL Light tutorial. http://www.cl.cam.ac.uk/~jrh13/hol-light/tutorial.pdf (2017)
25. Harrison, J.: The HOL Light Theorem Prover. Available at https://github.com/jrh13/hol-light (2024)
26. Harrison, J., Urban, J., Wiedijk, F.: History of Interactive Theorem Proving. In: Computational Logic. vol. 9, pp. 135–214 (2014)
27. Heylighen, F.: Stigmergy as a universal coordination mechanism i: Definition and components. Cognitive Systems Research **38**, 4–13 (2016)
28. Heylighen, F.: Stigmergy as a universal coordination mechanism ii: Varieties and evolution. Cognitive Systems Research **38**, 50–59 (2016)
29. Hölldobler, B., Wilson, E.: The Ants. Belknap Press of Harvard University Press (1990), https://books.google.it/books?id=R-7TaridBX0C
30. Levin, S.: Complex adaptive systems: exploring the known, the unknown and the unknowable. Bulletin of the American mathematical Society **40**(1), 3–19 (2003)
31. Li, L., Peng, H., Kurths, J., Yang, Y., Schellnhuber, H.J.: Chaos–order transition in foraging behavior of ants. Proceedings of the National Academy of Sciences **111**(23), 8392–8397 (2014). https://doi.org/10.1073/pnas.1407083111, https://www.pnas.org/doi/abs/10.1073/pnas.1407083111

32. Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., Balan, G.: MASON: A Multiagent Simulation Environment. SIMULATION **81**(7), 517–527 (2005). https://doi.org/10.1177/0037549705058073
33. Massink, M., Latella, D.: Fluid analysis of foraging ants. In: Coordination Models and Languages: 14th International Conference, COORDINATION 2012, Stockholm, Sweden, June 14-15, 2012. Proceedings 14. pp. 152–165. Springer (2012)
34. de Mingo López, L.F., Gómez Blas, N., Morales Lucas, C.: Ant colony systems optimization applied to bnf grammars rule derivation (acord algorithm). Soft Computing **24**(5), 3141–3154 (2020)
35. de Moura, L.M., Bjørner, N.S.: Z3: an efficient SMT solver. In: Ramakrishnan, C.R., Rehof, J. (eds.) Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29-April 6, 2008. Proceedings. Lecture Notes in Computer Science, vol. 4963, pp. 337–340. Springer (2008). https://doi.org/10.1007/978-3-540-78800-3_24, https://doi.org/10.1007/978-3-540-78800-3_24
36. Panait, L.A., Luke, S.: Ant Foraging Revisited. In: Artificial Life IX: Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems. The MIT Press (09 2004). https://doi.org/10.7551/mitpress/1429.003.0096, https://doi.org/10.7551/mitpress/1429.003.0096
37. Papert, S.A.: Mindstorms: Children, computers, and powerful ideas. Basic books (2020)
38. Perna, A., Granovskiy, B., Garnier, S., Nicolis, S.C., Labédan, M., Theraulaz, G., Fourcassié, V., Sumpter, D.J.T.: Individual rules for trail pattern formation in argentine ants (linepithema humile). PLOS Computational Biology **8**(7), 1–12 (07 2012). https://doi.org/10.1371/journal.pcbi.1002592, https://doi.org/10.1371/journal.pcbi.1002592
39. Resnick, M.: Turtles, termites, and traffic jams: Explorations in massively parallel microworlds. Mit Press (1997)
40. Sumpter, D.J., Blanchard, G.B., Broomhead, D.S.: Ants and agents: a process algebra approach to modelling ant colony behaviour. Bulletin of Mathematical Biology **63**, 951–980 (2001)
41. Tofts, C.: A synchronous calculus of relative frequency. In: International Conference on Concurrency Theory. pp. 467–480. Springer (1990)
42. Tofts, C.: Describing social insect behaviour using process algebra. TRANSACTIONS-SOCIETY FOR COMPUTER SIMULATION **9**, 227–227 (1992)
43. Traniello, J.F.A.: Foraging strategies of ants. Annual review of entomology **34**(1), 191–210 (1989)
44. Vittori, K., Talbot, G., Gautrais, J., Fourcassié, V., Araujo, A.F.R., Theraulaz, G.: Path efficiency of ant foraging trails in an artificial network. Journal of theoretical biology **239 4**, 507–15 (2006), https://api.semanticscholar.org/CorpusID:35239013
45. Wilensky, U.: Modeling nature's emergent patterns with multi-agent languages (2001)
46. Wilson, E.: Sociobiology: The New Synthesis, Twenty-Fifth Anniversary Edition. Harvard University Press (2000), https://books.google.it/books?id=v7lV9tz8fXAC

47. Zaidman, D., Wolfson, H.J.: PinaColada: peptide–inhibitor ant colony ad-hoc design algorithm. Bioinformatics **32**(15), 2289–2296 (03 2016). https://doi.org/10.1093/bioinformatics/btw133, https://doi.org/10.1093/bioinformatics/btw133
48. Zedadra, O., Jouandeau, N., Seridi, H., Fortino, G.: Multi-agent foraging: state-of-the-art and research challenges. Complex Adaptive Systems Modeling **5**, 1–24 (2017)