Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)

# Verifying Emergence of Bounded Time Properties in Probabilistic Swarm Systems

**Alessio Lomuscio, Edoardo Pirovano**
Department of Computing
Imperial College London, UK
{a.lomuscio,e.pirovano17}@imperial.ac.uk

## Abstract

We introduce a parameterised semantics for reasoning about swarms as unbounded collections of agents in a probabilistic setting. We develop a method for the formal identification of emergent properties, expressed in a fragment of the probabilistic logic PCTL. We introduce algorithms for solving the related decision problems and show their correctness. We present an implementation and evaluate its performance on an ant coverage algorithm.

## 1 Introduction

Robotic swarms have been put forward as an alternative to single-robot systems for a wide variety of applications including automated surveillance and maintenance of industrial plants. Individual physical agents in a swarm usually follow simple protocols, in line with the real-life limitations of their sensing, communication, and computation capabilities. Even in the presence of these limitations, it is possible for the agents in the swarm to interact in a way that leads to the overall system displaying sophisticated global behaviours such as flocking [Şahin, 2005; Şahin and Winfield, 2008].

In applications, swarms are composed of varying numbers of agents and their number typically varies at runtime, e.g., due to malfunctions. To validate a swarm before deployment, it is therefore desirable to assess whether the intended swarm specifications hold in the system regardless of how many agents the swarm is composed of at runtime. This cannot be achieved by traditional model checking techniques, which only address systems composed of a fixed number of agents. Indeed, this problem (known as the parameterised model checking problem) is undecidable in general [Apt and Kozen, 1986]. Nonetheless, decidable fragments can be obtained by limiting the modalities of interaction and specification languages [Aminof *et al.*, 2018; John *et al.*, 2013; Clarke *et al.*, 1989; Kouvaros and Lomuscio, 2016].

The problem we address here is related to parameterised model checking. It consists of establishing whether or not a swarm displays a *global emergent behaviour* [Bonabeau *et al.*, 1999] as a result of the sophisticated patterns of interactions among the agents. Specifically, we intend to establish whether or not a certain emergent property is always displayed in a swarm, as long as the number of agents composing the swarm is beyond a certain threshold. For example, we may wish to establish that a simple local protocol, governing the behaviours of the agents, results in an overall flocking pattern, as long as the number of the agents in the formation is larger than a certain number.

As discussed below, the formal verification of emergent properties in swarms has received some attention recently [Kouvaros and Lomuscio, 2015b]. But, differently from existing approaches, we here fully embrace the probabilistic aspects of swarms. This opens the way for analysing protocols where statistical behaviour is essential in the modelling and cannot trivially be replaced by non-determinism, such as opinion formation protocols [de Oca *et al.*, 2011]. To achieve the above, we provide a parameterised semantics for swarms where agents' behaviour is governed by probability distributions, and specifications are expressed by a probabilistic temporal logic. We provide algorithms and implementations both for identifying whether or not a specification is an emergent property and, if this is the case, identifying a sufficient number of agents for this behaviour to be displayed.

**Related Work.** We are not aware of any previous work addressing the question of proving that a property holds for all probabilistic swarm systems above of a certain size. Existing research in verification of probabilistic swarms [Gainer *et al.*, 2016; Konur *et al.*, 2012; Winfield *et al.*, 2008; Wan *et al.*, 2013] can only address systems of a predetermined number of agents; so, it can give no guarantees regarding whether the specifications hold for systems with more agents. Our approach is parameterised; hence not comparable to these.

Closer to this research are the parameterised model checking approaches in [Kouvaros and Lomuscio, 2013a; Kouvaros and Lomuscio, 2013b; Kouvaros and Lomuscio, 2015b; Kouvaros and Lomuscio, 2015a]. However, these approaches do not cater for probabilistic swarms or probabilistic specifications, as we do here.

Also related to this paper are techniques that have been put forward to verify probabilistic network protocols with an unbounded number of agents against probabilistic specifications [Graham, 2008; Fournier, 2015]; however, the semantics used has a communication pattern that is not amenable to modelling swarm systems.

The rest of this paper is organised as follows. In Section 2, we introduce a novel semantics for representing and reason-

ing about swarms of unbounded size and identify a suitable fragment of probabilistic temporal logic as a language for expressing swarm properties. In Section 3 we introduce a procedure, based on abstraction, for solving the decision problems of determining whether a specification is emergent in a swarm, and in Section 4 we address the related problem of identifying an adequate threshold of agents for the emergent property to be shown by a swarm. We present an implementation of these methods in Section 5 and evaluate it against an ant coverage algorithm [Koenig *et al.*, 2001]. We conclude and point to some topics for possible future work in Section 6.

## 2 Probabilistic Swarm Systems

In this section we introduce a novel semantics for swarms which, as we show in the next section, allows us to study the notion of *emergence* with respect to properties expressed in a probabilistic temporal logic.

**Semantics.** We modify the non-probabilistic parameterised semantics presented in [Kouvaros and Lomuscio, 2015b]. To encode arbitrarily many agents acting stochastically in a system we introduce the concept of *probabilistic agent template* which interacts with a deterministic environment. Any concrete system will be composed of a finite number of instantiations of the agent template and an environment. The results below are given for the case of a single agent template. They can be extended to a finite number of agent templates; we do not pursue this here for simplicity.

**Definition 1.** (Probabilistic Agent Template) A *probabilistic agent template* is a tuple $T = \langle S, \iota, Act, P, t\rangle$ where:

- The set $S$ represents a non-empty (possibly infinite) set of agent local states.
- $\iota \in S$ is a distinguished initial state.
- The set $Act \neq \emptyset$ is a finite set of possible agent actions.
- The agent's protocol function $P : S \times Act \to [0,1]$ is such that for each $s \in S$ we have $\sum_{a \in Act} P(s,a) = 1$ giving the probability distribution for the next action of an agent given its current state.
- The agent's transition function $t : S \times S_E \times \mathcal{P}(Act) \times Act \to S$ returns the agent's next state given its current state, the environment's current state, the set of actions performed by all the agents and the action performed by this agent at this time step.

Notice that while we assume a unique initial state for ease of presentation, our results could be extended to a probability distribution on the initial state.

**Definition 2.** (Environment) An *environment* is a tuple $E = \langle S_E, \iota_E, t_E\rangle$ where $S_E$ is a non-empty (possibly infinite) set of environment states with a distinguished initial state $\iota_E \in S_E$. The environment's transition function $t_E : S_E \times \mathcal{P}(Act) \to S_E$ gives the environment's next state given its current state and the actions performed by the agents.

Notice that while the agents behave probabilistically, the environment evolves deterministically based on the actions of the agents. This is a simplifying assumption often made when considering probabilistic multi-agent systems [Huang and Luo, 2013] that still allows us to capture noteworthy scenarios, as we show below. A swarm system consists of an agent template, an environment and a labelling function.

**Definition 3.** (Probabilistic Swarm System) A *probabilistic swarm system* is a tuple $\mathcal{S} = \langle T, E, \mathcal{V}\rangle$, where $T$ is a probabilistic agent template, $E$ is an environment and $\mathcal{V} : S \times S_E \to \mathcal{P}(AP)$ is a labelling function on a set of atomic propositions $AP$.

A swarm system $\mathcal{S}$ gives an abstract description of an infinite number of different swarm systems, which can be obtained by instantiating it with a different number of agents. We now define how to obtain such a concrete model of a swarm system, which we will encode in a *discrete-time Markov chain* (DTMC) [Kemeny *et al.*, 1976].

**Definition 4.** (Concrete Model) Given a swarm system $\mathcal{S}$, a *concrete model* of $n$ agents for system $\mathcal{S}$ is a discrete-time Markov chain $\mathcal{S}(n) = \langle G_n, \iota_n, P_n, L_n\rangle$, representing the behaviour of a global system composed of $n$ agents and the environment, where:

- The set of *global states* $G_n$ is the set of $(n+1)$-tuples of the form $S \times \cdots \times S \times S_E$. Given a global state $g$ we write $g.i$ to denote the local state of agent $i$ and $g.E$ to denote the local state of the environment.
- The global state $\iota_n = (\iota, \ldots, \iota, \iota_E) \in G_n$ is the initial global state of the model $\mathcal{S}(n)$.
- The set $ACT(n) = Act \times \cdots \times Act$ is the set of global actions, i.e., $n$-tuples representing the instantaneous actions for each of the $n$ agents in $\mathcal{S}(n)$. Given a global action $a$, we write $a.i$ to denote the action of agent $i$. The transition probability function $P_n : G_n \times G_n \to [0,1]$ is defined by:

$$(g, g') \mapsto \sum_{a \in A_{g,g'}} \left( \prod_{i=1}^{n} P(g.i, a.i) \right)$$

  where $A_{g,g'} \subseteq ACT(n)$ is defined such that $a \in A_{g,g'}$ iff:

  1. $t(g.i, g.E, \{a.1, \ldots, a.n\}, a.i) = g'.i$ for all $i = 1, \ldots, n$
  2. $t_E(g.E, \{a.1, \ldots, a.n\}) = g'.E$.

  The set $A_{g,g'}$ denotes the set of global actions which could lead to a transition from $g$ to $g'$.
- The labelling function $L_n : G_n \to \mathcal{P}(AP \times \{0, \ldots, n\})$ is defined by:

$$g \mapsto \{(p, i) : p \in \mathcal{V}(g.i, g.E)\}.$$

Observe from the above that the probability of transitioning from $g$ to $g'$ is thus given by summing, over each possible action, the probability of it occurring, which in turn is obtained by taking the product of the probabilities of each agent choosing their respective local action. Further note that the labelling function creates $n$ copies of each atomic proposition based on whether it holds for each agent at its local state.

**Specifications.** We express properties of probabilistic swarms by means of a fragment of PCTL [Hansson and Jonsson, 1994], which we define below.

**Definition 5.** For $a \in AP$ and $i, k \in \mathbb{N}$, the logic P(CTL$^k$) is the set of formulas $\chi$ defined by the following BNF:

$$\chi ::= P_{\geq x}[\psi], \text{ for } x \in [0,1]$$
$$\psi ::= X^k \phi \mid \phi \, U^{\leq k} \phi \qquad \text{(path formulas)}$$
$$\phi ::= \top \mid (a,i) \mid \neg\phi \mid \phi \wedge \phi \qquad \text{(state formulas)}$$

Definition 5 identifies the fragment of PCTL with a single probabilistic operator at the start of the formula (while we use $\geq$, our method can easily be adapted to the others) without unbounded operators (thus, our logic expresses properties on finite traces). We can also express some derived operators in P(CTL$^k$), such as $F^{\leq k}\phi \equiv \top U^{\leq k}\phi$.

Notice that a number of properties of interest in swarm contexts can be expressed in the logic above. For example, in aggregation protocols [Nembrini, 2005], a property of interest is whether an agent, say agent 1, is connected to the swarm within a certain time bound, say 10 time steps, with at least a certain probability, e.g., 0.9. This can be expressed by the P(CTL$^k$) formula $P_{\geq 0.9}[F^{\leq 10}(\mathit{connected}, 1)]$, where *connected* is an atomic proposition holding when an agent is connected. Other properties of interest can similarly be represented.

**Definition 6.** (Time Bound) The time bound $tb(\psi)$ for a path formula $\psi$ is defined as

$$tb(X^k\phi) \triangleq k \qquad tb(\phi_1 U^{\leq k} \phi_2) \triangleq k$$

We say a formula is $m$-indexed if it refers to agents with index at most $m$. For example, our specification above for connectedness is a 1-indexed formula with time bound 10.

**Observation 1.** Notice that, intuitively, a bounded time property $P_{\geq x}[\psi]$ holds in a concrete system $\mathcal{S}(n)$ precisely if it holds by considering only the first $tb(\psi)$ steps of its behaviour.

We define satisfaction of an $m$-indexed formula on concrete models of the swarm system $\mathcal{S}(n)$ (with $n \geq m$) as usual in PCTL. Informally, $\mathcal{S}(n) \models P_{\geq x}[X^k\phi]$ if, starting at the initial state, the probability that $\phi$ holds after $k$ time steps is at least $x$. Similarly, $\mathcal{S}(n) \models P_{\geq x}[\phi_1 \, U^{\leq k} \phi_2]$ if, starting at the initial state, the probability that $\phi_2$ holds within $k$ time steps and $\phi_1$ holds until then is at least $x$.

Formally, as in [Hansson and Jonsson, 1994], let Path$(g)$ denote the set of all infinite paths originating from a global state $g$ (each with an associated probability) and for $X \subseteq$ Path$(g)$ denote by Prob$(X)$ the sum of the probabilities of the paths in $X$. For an $\omega \in$ Path$(g)$ and $n \in \mathbb{N}$, denote by $\omega(n)$ the $n$th global state in $\omega$.

**Definition 7.** (Satisfaction) Given a swarm system $\mathcal{S}(n)$, a global state $g \in G_n$, and a formula $\chi \in P(CTL^{\mathrm{k}})$, the satisfaction relation $\models$ is defined by induction as follows.

- $g \models P_{\geq x}[X^k\phi]$ iff Prob$(\{\omega \in \text{Path}(g) : \omega(k) \models \phi\}) \geq x$
- $g \models P_{\geq x}[\phi_1 \, U^{\leq k} \phi_2]$ iff Prob$(\{\omega \in \text{Path}(g) : \exists k' \leq k$ such that $\omega(k') \models \phi_2 \wedge \forall k'' < k' \; k'' \models \phi_1\}) \geq x$,
- $g \models \top$ for all $g \in G_n$
- $g \models (a,i)$ iff $(a,i) \in L_n(g)$
- $g \models \neg\phi$ iff $g \not\models \phi$
- $g \models \phi_1 \wedge \phi_2$ iff $g \models \phi_1$ and $g \models \phi_2$.

If $\iota_n \models P_{\geq x}[\psi]$, we write $\mathcal{S}(n) \models P_{\geq x}[\psi]$. We can now formalise the model checking problem.

**Definition 8.** (Model Checking) Given a probabilistic swarm system $\mathcal{S}$, an $m$-indexed formula $P_{\geq x}[\psi]$, and an $n \geq m$, the model checking problem involves determining whether $\mathcal{S}(n) \models P_{\geq x}[\psi]$.

Note that model checking DTMCs against PCTL specification, a problem more general than the one above, is widely studied in the literature and efficient implementations exist, e.g., PRISM [Kwiatkowska *et al.*, 2011]. For more details of PCTL, we refer the reader to [Baier and Katoen, 2008].

## 3 Emergent Property Identification

We now formalise the notion of emergence that will be used throughout the paper when discussing emergent behaviours.

**Definition 9.** (Emergence) Given a probabilistic swarm system $\mathcal{S}$, an $m$-indexed formula $P_{\geq x}[\psi]$ is said to be an *emergent property* of $\mathcal{S}$ if there is some $t \geq m$ such that for all $t' \geq t$ we have $\mathcal{S}(t') \models P_{\geq x}[\psi]$. If this is the case we call $t$ an *emergence threshold*.

We now address the actual identification of emergence properties in a swarm system. To do so we define the corresponding decision problem (Definition 10) and an abstract model (Definition 11), which can be used to determine whether a specification is an emergent property in a swarm (Theorems 1 to 3).

**Definition 10.** (EPI Decision Problem) Given a probabilistic swarm system $\mathcal{S}$ and an $m$-indexed formula $P_{\geq x}[\psi]$, the emergent property identification (EPI) decision problem concerns deciding whether or not $P_{\geq x}[\psi]$ is an emergent property of the swarm system $\mathcal{S}$.

Before defining a suitable abstract model, observe that it is the case with probability arbitrarily close to 1 that any action which can occur (with some probability larger than 0) at a time step $k' \leq tb(\psi)$ does occur in the system *as long as a sufficiently large number of agents is considered*. So, when identifying emergent properties, we can assume that every action that may occur does, in fact, occur. Hence, we define an abstract model which captures the first $k$ time steps of the behaviour of $m$ agents and an environment that have interacted in every possible way with arbitrarily many other agents.

**Definition 11.** (Abstract Model) Given a probabilistic swarm system $\mathcal{S} = \langle T, E, \mathcal{V} \rangle$ and some $k, m \in \mathbb{N}$, an *abstract swarm* (or *abstract model*) is a discrete-time Markov Chain $D_{\mathcal{S},k,m} = \langle G_{k,m}, \iota_{k,m}, P_{k,m}, L_{k,m} \rangle$, where the various components are defined as follows:

- The set $G_{k,m} = (G_m \times \{0, \ldots, k\}) \cup \{\perp\}$ denotes the set of states of the model, where $(g,k)$ represents that the $m$ agents and the environment are in global state $g$ after $k$ actions; $\perp$ is a new state which is used as a sink state after $k$ steps.

- The element $\iota_{k,m} = (\iota_m, 0) \in G_{k,m}$ is the initial state of the model.
- For any $m$, the transition probability function $P_{k,m} : G_{k,m} \times G_{k,m} \to [0,1]$ is defined by induction on $k$. For $k = 0$, it is given as follows:

$$P_{0,m}(s, s') = \begin{cases} 1 \text{ if } s' = \bot \\ 0 \text{ otherwise} \end{cases}$$

This results in a system with precisely two reachable states: the initial state, and the sink state. From the initial state the system moves to the sink state and remains there indefinitely with probability 1.

Now, given $P_{k,m}$ we need to define $P_{k+1,m}$. To do this, we first define the set of actions that could be performed by some agent at time step $k$, denoted as $A_k \subseteq Act$, as the set of actions $a$ such that there is some $l \in S$ and $l_E \in S_E$ with $P(l, a) > 0$ and $((l, l_E), k)$ reachable in $D_{\mathcal{S},k,1}$.

Now we can construct $P_{k+1,m}$ as follows:

$$\begin{aligned} P_{k+1,m}((g,k'),s) &= P_{k,m}((g,k'),s) \text{ if } k' < k \\ P_{k+1,m}((g,k),\bot) &= 0 \\ P_{k+1,m}((g,k),(g',k')) &= 0 \text{ if } k' \leq k \\ P_{k+1,m}((g,k),(g',k+1)) &= \sum_{a \in A'_{g,g'}} \left( \prod_{i=1}^{m} P(g.i, a.i) \right) \\ P_{k+1,m}((g,k+1),s') &= P_{k+1,m}(\bot, s') \\ &= \begin{cases} 1 \text{ if } s' = \bot \\ 0 \text{ otherwise} \end{cases} \end{aligned}$$

where we define $A'_{g,g'} \subseteq ACT(m)$ by $a \in A'_{g,g'}$ iff:

1. $t(g.i, g.E, A_k, a.i) = g'.i$ for all $i = 1, \ldots, m$
2. $t_E(g.E, A_k) = g'.E$

This behaves like $P_{k,m}$ for the first $k$ time steps. Then, at time step $k+1$ it behaves like a system in which *every* possible action that could have been performed by some agent does occur in the model. This comes from the fact that in our abstract model we assume that there are sufficiently many other agents to perform all possible actions. After this, it transitions to the sink state and remains there with probability 1.

- The labelling function is defined by:

$$\begin{aligned} L_{k,m}(g,k) &= L_m(g) \\ L_{k,m}(\bot) &= \emptyset \end{aligned}$$

This labels each state with the atomic propositions that hold at the corresponding global state, discarding the time step information. No atomic propositions hold in the sink state.

This completes the definition of the abstract model $D_{\mathcal{S},k,m}$ for any $k, m \in \mathbb{N}$, and leads us to our main technical results.

**Theorem 1.** Given a probabilistic swarm system $\mathcal{S}$ and an $m$-indexed formula $P_{\geq x}[\psi]$ then if $D_{\mathcal{S},tb(\psi),m} \models P_{>x}[\psi]$ we have that $P_{\geq x}[\psi]$ is an emergent property of $\mathcal{S}$.

**Algorithm 1** EPI Decision Procedure

**Input:** Swarm system $\mathcal{S}$, $m$-indexed formula $\chi = P_{\geq x}[\psi]$
**Output:** Whether or not $\chi$ is an emergent property of $\mathcal{S}$

1: **if** `ModelCheck` $(D_{\mathcal{S},tb(\psi),m}, P_{>x}[\psi])$ **then**
2: **return** true
3: **end if**
4: **if** `ModelCheck` $(D_{\mathcal{S},tb(\psi),m}, P_{<x}[\psi])$ **then**
5: **return** false
6: **end if**
7: **return** $\mathcal{S}(m) \models \phi$

*Proof sketch.* Notice that, by Observation 1, we only need to consider the first $tb(\psi)$ time steps of the swarm system's behaviour to verify the satisfaction of $\psi$. The assumption made in the abstract model (that every possible action occurs) can be achieved with an arbitrarily high probability $P < 1$ in a large enough concrete system. We expand on how high we need this probability to be and how many agents are needed for this to be the case in the next section. □

The result above provides a sufficient condition for establishing whether a property is an emergent behaviour of a swarm. We can also similarly show the following.

**Theorem 2.** Given a probabilistic swarm system $\mathcal{S}$ and an $m$-indexed formula $P_{\geq x}[\psi]$ then if $D_{\mathcal{S},tb(\psi),m} \models P_{<x}[\psi]$ we have that $P_{\geq x}[\psi]$ is not an emergent property of $\mathcal{S}$.

In the case where both $D_{\mathcal{S},tb(\psi),m} \not\models P_{>x}[\psi]$ and $D_{\mathcal{S},tb(\psi),m} \not\models P_{<x}[\psi]$ the probability of $\psi$ holding in the abstract model is precisely $x$. Notice that intuitively concrete systems with high number of agents behave "similarly" to those encoded by their corresponding abstract model; however an abstract model will include only behaviours resulting from the assumption that every possible action is present at every step. If it is possible for this to not be the case and affect the satisfaction of our formula then it will remain the case, with a small probability, regardless of how many agents we add to our concrete system. This leads to the following.

**Theorem 3.** Given a probabilistic swarm system $\mathcal{S}$ and an $m$-indexed formula $P_{\geq x}[\psi]$ then if $D_{\mathcal{S},tb(\psi),m} \not\models P_{>x}[\psi]$ and $D_{\mathcal{S},tb(\psi),m} \not\models P_{<x}[\psi]$ we have that $P_{\geq x}[\psi]$ is an emergent property of $\mathcal{S}$ iff $\mathcal{S}(m) \models P_{\geq x}[\psi]$.

Together, these theorems give us a sound and complete decision procedure for EPI shown in Algorithm 1.

**Corollary 1.** Algorithm 1 is a sound and complete decision procedure for EPI.

*Proof.* Soundness for the return values of lines 2, 5 and 7 follows from Theorems 1 to 3 respectively. Completeness is clear. □

## 4 Emergence Threshold Identification

We now define the *emergence threshold identification* (ETI) decision problem and outline a sound and complete decision procedure for it. Given a swarm and a specification, the procedure for ETI will return a number of agents for the specification to be satisfied in any concrete swarm of size equal or over this threshold.

**Definition 12.** (ETI Decision Problem) Given a probabilistic swarm system $\mathcal{S}$ and an $m$-indexed formula $P_{\geq x}[\psi]$, determine an emergence threshold for $P_{\geq x}[\psi]$ in the swarm system $\mathcal{S}$ if it exists, or determine its non-existence.

Notice we do not require the emergence threshold returned by ETI to be a minimal one. We now introduce and study the decision procedure for ETI outlined in Algorithm 2.

**Theorem 4.** If Algorithm 2 returns an emergence threshold $m$, then $m$ is an emergence threshold for $P_{\geq x}[\psi]$ in $\mathcal{S}$. If it returns non-existence, then $P_{\geq x}[\psi]$ is not an emergent property of $\mathcal{S}$.

*Proof.* The procedure begins by computing $D_{\mathcal{S},tb(\psi),m}$. We then evaluate using existing techniques for *quantitative model checking* of DTMCs [Kwiatkowska *et al.*, 2010] the probability of $\psi$ occurring in $D_{\mathcal{S},tb(\psi),m}$, which we will denote as $y$.

**Failure.** The cases for non-existence of an emergence threshold (line 3 of the algorithm) follow directly from the theorems for EPI in the previous section.

**Success.** Suppose we have that $y > x$. Given a number of agents $M \in \mathbb{N}$, let $P_M$ denote the probability that at each time step $k \leq tb(\psi)$, every action that can occur will be performed by at least one agent.

Now, if we pick a path of length $tb(\psi)$ in the concrete system of $M$ agents $D_{\mathcal{S},M}$, with probability at least $P_M$ there is a path in the abstract model $D_{\mathcal{S},tb(\psi),m}$ such that at each time step up to $tb(\psi)$, the first component of the state is the state of the first $m$ agents and the second is the time step.

So, if we pick $M \in \mathbb{N}$ such that $y \cdot P_M > x$, $M$ will be an emergence threshold. Now, notice that if at least one agent follows every possible path of length $tb(\psi)$ then every possible action at each time step $k' \leq tb(\psi)$ will be performed.

To obtain a sufficient number of agents for this to be the case, enumerate all paths of length $tb(\psi)$ in $D_{\mathcal{S},tb(\psi),1}$ and denote the probabilities of each path occurring by $p_1, \ldots, p_n$. Notice that since at each time step the agent chooses its action independently, the paths followed by the agents are independent of one another. By using a generalisation of the *coupon collector's problem* [Anceaume *et al.*, 2015], where we take the objects being drawn as the paths of length $tb(\psi)$, it follows that $P_M \geq P'(M, p_1, \ldots, p_n)$ where we define:

$$P'(M, p_1, \ldots, p_n) \triangleq 1 - \sum_{i=1}^{n-1} \left( (-1)^{n-1-i} \sum_{J \in S_i} (P_J)^M \right)$$

where $S_i = \{J \subseteq \{1, \ldots, n\} : |J| = i\}$ and $P_J = \sum_{j \in J} p_j$

This equation can be solved by numerical methods to get a sufficiently large $M$. Then, this is an emergence threshold which can be returned (line 7 of the algorithm). Notice that this need not be minimal, since while having an agent following every path is a sufficient condition to ensure that the abstract model has a path corresponding to the path followed by the concrete system, it is certainly not a necessary one.

For the final success case (line 9 of the algorithm), notice that if $y = x$ and $\mathcal{S}(m) \models \phi$, then certainly $m$ is an emergence threshold since adding more agents can only make the concrete system behave more like the abstract model. □

**Algorithm 2** ETI Decision Procedure

**Input:** Swarm system $\mathcal{S}$, $m$-indexed formula $P_{\geq x}[\psi]$
**Output:** An emergence threshold or non-existence

1: $y \leftarrow \texttt{QuantModelCheck}(D_{\mathcal{S},tb(\psi),m}, P_{=?}[\psi])$
2: **if** $y < x$ or $(y = x$ and $\mathcal{S}(m) \not\models \phi)$ **then**
3: **return** non-existent
4: **end if**
5: **if** $y > x$ **then**
6: $p_1, \ldots, p_n \leftarrow$ probability of each distinct path of length $tb(\psi)$ in $D_{\mathcal{S},tb(\psi),1}$ occurring
7: **return** least $M$ such that $P'(M, p_1, \ldots, p_n) \leq \frac{x}{y}$
8: **end if**
9: **return** $m$

```
agent
  xCoord : [0..2] init 0;
  ...
  [moveLeft] (observedLeft=observedRight
              & observedRight=observedUp
              & observedUp=observedDown
              & state=2) : (1/4);
  ...
  update
    (moveLeft, xCoord>0, {}) ->
        (xCoord'=xCoord-1) & (state'=0);
    ...
  endupdate
endagent
environment
  uValue00 : int init 0;
  ...
  update
    (true, {visit00}) ->
        (uValue00'=uValue00+1);
    ...
  endupdate
endenvironment
label "allVisited" = uValue00_E>0 & ...;
```

Figure 1: A fragment of the code modelling the ant coverage algorithm in a $3 \times 3$ grid with the node counting value update rule.

Theorem 4 is one of the key results in this paper. It gives guarantees that Algorithm 2 provides a correct threshold, if this exists, for all swarm systems larger than the threshold to satisfy a probabilistic emergent property under study.

## 5 Implementation and Evaluation

We implemented the EPI and ETI procedures in a Java tool called PSV-BD (**P**robabilistic **S**warm **V**erifier for **B**ounde**D** time properties), built on top of PRISM 4.0 [Kwiatkowska *et al.*, 2011], which provides the relevant model checking and parsing procedures. The sources and the models (including the one we evaluate below) are released as open source [PSV-BD, 2018].

The modelling language we use (an example of which can be seen in Figure 1) is based on PRISM's modelling language. Agent templates are made up of three parts: variables, actions and update rules. Variables have a domain and an initial value. Actions have a name, a guard (an expression on the agent's local state) which defines when they are enabled and a probability that they are chosen when enabled. Notice we can define an action with the same name multiple times to

| n | p | k | threshold | EPI time (s) | ETI time (s) |
|---|---|---|---|---|---|
| 3 | 0.25 | 7 | 16 | 2.1 | 2.3 |
| 3 | 0.50 | 7 | 20 | 2.1 | 2.3 |
| 3 | 0.75 | 7 | 26 | 2.1 | 2.3 |
| 3 | 0.95 | 7 | 38 | 2.1 | 2.3 |
| 3 | 0.99 | 7 | 51 | 2.1 | 2.3 |
| 5 | 0.25 | 13 | 70 | 2.2 | 81 |
| 5 | 0.50 | 13 | 85 | 2.2 | 81 |
| 5 | 0.75 | 13 | 105 | 2.2 | 81 |
| 5 | 0.95 | 13 | 145 | 2.2 | 81 |
| 5 | 0.99 | 13 | 183 | 2.2 | 81 |
| 10 | 0.25 | 31 | - | 3.9 | *timeout* |

Table 1: For different values of $n$ and $p$, the least $k$ such that $P_{\geq p}[F^{\leq k}\text{allVisited}]$ is an emergent property in an $n \times n$ grid and the emergence threshold for this property, along with the time needed to run the EPI and ETI procedures (to two significant figures).

have it occur with different probabilities in different states (if multiple guards for the same action are true in some local state, then the probability of that action occurring will be the sum of the probabilities in the respective definitions). Updates have a guard which is a triple and a rule that defines what variable changes occur if the guard is enabled. An update guard $(a, e, S)$ is enabled if the agent performed action $a$, the expression $e$ (which can express conditions on both the local state of the agent and the environment) was true before *any* updates and $S$ is a *subset* of the actions performed by all the agents. Notice more than one update rule may have its guard enabled, in which case the enabled update rules are applied in sequence from first to last. This is slightly different from our formal definitions in the previous sections but allows us to express models more concisely.

Similarly, the environment is made of variables (defined exactly as for the agent template) and update rules (as for the agent but without the first component of the guard).

To evaluate the proposed algorithm, we modelled the *ant coverage* algorithm described in [Koenig *et al.*, 2001], using the *node counting* value update rule. We also modelled the *LRTA** value update rule, but found that for the properties that we were considering the two are equivalent; so we omit this analysis here.

In the ant coverage problem agents on a graph visit every node repeatedly infinitely often. This problem arises in a number of real-world scenarios, such as surveillance and automated cleaning or maintenance tasks.

They do this by storing a value $u$ on each node, and at each time step moving to the neighbour with the lowest value for $u$ and then incrementing this by one. When two neighbours are tied for the lowest value of $u$, they make a random choice between the tied neighbours with equal probability for each.

In our case, we take the graph as being an $n \times n$ grid where every cell is connected to the four cells left, right, above and below of it. The grid is allowed to loop around so that the cell $(1, 1)$ is one position to the right of the cell $(1, n)$.

We model the agents as all starting in the top-left corner of the grid, and repeatedly performing a sequence of three actions: incrementing the $u$ value of their current cell by one, observing the $u$ values of the neighbours, and moving to a random neighbouring cell with a minimal $u$ value. As in [Koenig *et al.*, 2001] the incrementing and observing steps are atomic. Notice that precisely three time steps in our model correspond to one movement action. A fragment of the code for the model is reported in Figure 1. The full code is at [PSV-BD, 2018].

We consider the property that with probability at least $p$, every cell is visited at least once within $k$ time steps, which we will write as $P_{\geq p}[F^{\leq k}\text{allVisited}]$ where $allVisited$ is an atomic proposition that holds precisely when every cell has been visited at least once. This property is known as *cover time*, and is of interest in the real-world scenarios this algorithm is applied to [Koenig *et al.*, 2001].

For different values of $n$ and $p$ we first establish, by calling EPI on $P_{\geq p}[F^{\leq k}\text{allVisited}]$ for increasing values of $k$, the least value for which this specification is an emergent property. Then, we call ETI on the corresponding specification to compute the emergence threshold for this.

Table 1 shows the results obtained by running the tool with OpenJDK 1.8 (64-bit version, 4GB heap size) on a machine running Ubuntu 17.10 (Linux kernel 4.13.0-38) and an Intel i7-6700 processor. No comparison is offered as, to the best of our knowledge, no other method or toolkit supports the verification of the emergence of properties in probabilistic swarm systems. The results show that the toolkit correctly identified the property in question to be an emergent property for the swarm with various thresholds depending on the parameters.

It was found that most computation is dedicated to the ETI procedure, with calls to EPI being comparatively lighter. This was expected as the ETI procedure is much more expensive than the EPI one as it requires explicitly computing all paths and constructing and solving a large numerical equation based on these.

## 6 Conclusions

We have introduced a semantics for reasoning about the emergence of behaviours in probabilistic swarm systems and given sound and complete decision procedures for the EPI and ETI decision problems on a fragment of specifications called bounded time properties. We have also implemented our technique into a toolkit and evaluated it against an example swarm algorithm. To our knowledge, while approaches for formally establishing emergence of properties in a swarm do exist [Kouvaros and Lomuscio, 2015b], they do not cater for properties expressing probabilistic temporal properties; yet, probabilistic aspects are of considerable importance.

In future work we plan to apply the method and tool here described to more swarm algorithms in order to better understand its applicability and scalability. We would also like to consider richer specifications using probabilistic knowledge and strategies [Huang and Luo, 2013]. Also, it is of interest in applications to find the *minimal* threshold for a property to be displayed by a swarm; we leave this for further research.

## Acknowledgements

The authors would like to thank Benjamin Walker and the anonymous referees for valuable comments and suggestions.

## References


[Aminof *et al.*, 2018] B. Aminof, S. Rubin, I. Stoilkovska, J. Widder, and F. Zuleger. Parameterized model checking of synchronous distributed algorithms by abstraction. In *Proceedings of VMCAI18*, pages 1–24. Springer International Publishing, 2018.

[Anceaume *et al.*, 2015] E. Anceaume, Y. Busnel, and B. Sericola. New results on a generalized coupon collector problem using markov chains. *Journal of Applied Probability*, 52(2):405–418, 2015.

[Apt and Kozen, 1986] K.R. Apt and D. C. Kozen. Limits for automatic verification of finite-state concurrent systems. *Information Processing Letters*, 22(6):307–309, 1986.

[Baier and Katoen, 2008] C. Baier and J. P. Katoen. *Principles of Model Checking*. The MIT Press, 2008.

[Bonabeau *et al.*, 1999] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm intelligence*. Oxford University Press, 1999.

[Clarke *et al.*, 1989] E.M. Clarke, O. Grumberg, and M.C. Browne. Reasoning about networks with many identical finite state processes. *Information and Computation*, 81(1):13–31, 1989.

[de Oca *et al.*, 2011] M. de Oca, , E. Ferrante, A. Scheidler, Carlo C. Pinciroli, M. Birattari, and M. Dorigo. Majority-rule opinion dynamics with differential latency: a mechanism for self-organized collective decision-making. *Swarm Intelligence*, 5(3-4):305–327, 2011.

[Fournier, 2015] P. Fournier. *Parameterized verification of networks of many identical processes*. PhD thesis, Université de Rennes 1, 2015.

[Gainer *et al.*, 2016] P. Gainer, C. Dixon, and U. Hustadt. Probabilistic model checking of ant-based positionless swarming. In *Proceedings of TAROS16*, pages 127–138. Springer, 2016.

[Graham, 2008] D. Graham. *Parameterised verification of randomised distributed systems using state-based models*. PhD thesis, University of Glasgow, 2008.

[Hansson and Jonsson, 1994] H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6(5):512–535, 1994.

[Huang and Luo, 2013] X. Huang and C. Luo. A logic of probabilistic knowledge and strategy. In *Proceedings of AAMAS2013*, pages 845–852. IFAAMAS, 2013.

[John *et al.*, 2013] A. John, I. Konnov, U. Schmid, H. Veith, and J. Widder. Parameterized model checking of fault-tolerant distributed algorithms by abstraction. In *Formal Methods in Computer-Aided Design (FMCAD)*, pages 201–209. IEEE, 2013.

[Kemeny *et al.*, 1976] J. G. Kemeny, J. Laurie Snell, and A. W. Knapp. *Denumerable Markov Chains*. Graduate Texts in Mathematics. Springer-Verlag, 2 edition, 1976.

[Koenig *et al.*, 2001] S. Koenig, B. Szymanski, and Y. Liu. Efficient and inefficient ant coverage methods. *Annals of Mathematics and Artificial Intelligence*, 31(1):41–76, 2001.

[Konur *et al.*, 2012] S. Konur, C. Dixon, and M. Fisher. Analysing robot swarm behaviour via probabilistic model checking. *Robotics and Autonomous Systems*, 60(2):199–213, 2012.

[Kouvaros and Lomuscio, 2013a] P. Kouvaros and A. Lomuscio. Automatic verification of parametrised interleaved multi-agent systems. In *Proceedings of AAMAS13*, pages 861–868. IFAAMAS, 2013.

[Kouvaros and Lomuscio, 2013b] P. Kouvaros and A. Lomuscio. A cutoff technique for the verification of parameterised interpreted systems with parameterised environments. In *Proceedings of IJCAI13*, pages 2013–2019. AAAI Press, 2013.

[Kouvaros and Lomuscio, 2015a] P. Kouvaros and A. Lomuscio. A counter abstraction technique for the verification of robot swarms. In *Proceedings of AAAI15*, pages 2081–2088. AAAI Press, 2015.

[Kouvaros and Lomuscio, 2015b] P. Kouvaros and A. Lomuscio. Verifying emergent properties of swarms. In *Proceedings of IJCAI15*, pages 1083–1089. AAAI Press, 2015.

[Kouvaros and Lomuscio, 2016] P. Kouvaros and A. Lomuscio. Parameterised verification for multi-agent systems. *Artificial Intelligence*, 234:152–189, 2016.

[Kwiatkowska *et al.*, 2010] M. Kwiatkowska, G. Norman, and D. Parker. Advances and challenges of probabilistic model checking. In *Proceedings of Allerton 2010*, pages 1691–1698. IEEE Press, 2010.

[Kwiatkowska *et al.*, 2011] M. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: Verification of probabilistic real-time systems. In *Proceedings of CAV11*, pages 585–591. Springer, 2011.

[Nembrini, 2005] J. Nembrini. *Minimalist Coherent Swarming of Wireless Networked Autonomous Mobile Robots*. PhD thesis, University of the West of England, 2005.

[PSV-BD, 2018] PSV-BD. Probabilistic Swarm Verifier for BoundeD time properties `http://vas.doc.ic.ac.uk/software/psv-bd/`, 2018.

[Şahin and Winfield, 2008] E. Şahin and A. Winfield. Special issue on swarm robotics. *Swarm Intelligence*, 2(2):69–72, 2008.

[Şahin, 2005] E. Şahin. Swarm robotics: From sources of inspiration to domains of application. In *Proceedings of SAB04*, volume 3342 of *Lecture Notes in Computer Science*, pages 10–20. Springer, 2005.

[Wan *et al.*, 2013] W. Wan, J. Bentahar, and A. Ben Hamza. Model checking epistemic-probabilistic logic using probabilistic interpreted systems. *Knowledge-Based Systems*, 50(Supplement C):279–295, 2013.

[Winfield *et al.*, 2008] A. Winfield, W. Liu, J. Nembrini, and A. Martinoli. Modelling a wireless connected swarm of mobile robots. *Swarm Intelligence*, 2(2-4):241–266, 2008.