HAL open science

Fluid Analysis of Foraging Ants

Mieke Massink, Diego Latella

To cite this version:

Mieke Massink, Diego Latella. Fluid Analysis of Foraging Ants. 14th International Conference on Coordination Models and Languages (COORDINATION), Jun 2012, Stockholm, Sweden. pp.152-165, 10.1007/978-3-642-30829-1_11 . hal-01529603

HAL Id: hal-01529603

https://inria.hal.science/hal-01529603v1

Submitted on 31 May 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Fluid Analysis of Foraging Ants

Mieke Massink and Diego Latella

Istituto di Scienza e Tecnologie dell'Informazione 'A. Faedo', CNR, Italy

Abstract. Workers of the Argentine ant, Iridomyrmex humilis, are known to be capable to find efficiently the shortest route from their nest to a food source. Their approach is based on a simple pheromone trail-laying and following behaviour accessing only local information. In this note we explore the modelling and analysis of foraging ants in Bio-PEPA [8, 6]. The simple case study concerns ants that need to cross a bridge with two branches of different length to reach food and carry the food home and is based on empirical data described by Goss and Deneubourg et al. [13, 10]. We explore the conditions for which the shortest path emerges as the preferred one by the ants. The analysis is based on stochastic simulation and fluid flow analysis. The behaviour of ant colonies has inspired the development of an interesting class of optimisation algorithms ranging from alternative shortest path algorithms to new scheduling and routing algorithms, algorithms to solve set partition problems and for distributed information retrieval. Process algebraic fluid flow analysis may be an important additional technique to the analysis of such algorithms in a computationally efficient way.

Keywords: Fluid flow, process algebra, collective dynamics, self-organisation, emergence.

1 Introduction

The behaviour of ant colonies has inspired the development of an interesting class of optimisation algorithms ranging from alternative shortest path algorithms to new scheduling and routing algorithms, algorithms to solve set partition problems and for distributed information retrieval, see for example work by Dorigo et al. [11]. The formal specification and scalable analysis of such systems, that in general consist of a large number of autonomous entities, is still a challenging problem. Such analysis is however essential to assure functional and non-functional properties of such systems, especially when they are employed in safety critical applications. Algorithms based on ant colony behaviour are also inspiring the development of techniques to contribute to the solution of difficult problems of self-organisation, self-awareness, autonomous and collective behaviour, and resource optimisation in a complex system setting such as those that can be found in collaborative swarms of robots, see for example the ASCENS project [1]. In this preliminary work we address the formal modelling and analysis of a colony of foraging ants in Bio-PEPA and in particular we study the emergent behaviour of this model using a process algebra based fluid flow approach.

2 The Problem of Foraging Ants

To study the problem of foraging ants the following experimental set up is considered from the literature [13, 10]. A nest of ants is situated at site `nest` and some food is situated at site `food`. Ants can reach the food from the nest through two alternative branches of a bridge, a long branch passing by locations `L1` and `L2`, and a short branch, passing by locations `S1` and `S2`. The ants choose a branch based on the relative quantity of *pheromone* present at the start of the two branches, i.e. in `L1` and `S1` for ants leaving choice point `A` moving towards site `food`, and `L2` and `S2` for ants leaving choice point `B` and going towards site `nest`. Pheromone is a chemical substance that is released by the ants as they move. It is assumed that once a branch is chosen the ant's decision is not reverted, i.e. the ant eventually arrives at destination once it starts moving along a particular branch. After the ant has found the food it returns to the nest following the same procedure. Pheromone is assumed *not* to be subject to decay in the present set up since the behaviour is studied over a time interval of ca. 40 minutes, which is compatible with the lifetime of pheromone. However, such a feature could be easily added when longer time intervals are of interest. The aim of this modelling exercise is to find a high-level, process algebraic, agent based model that can be used to determine under which conditions which of the two branches is emerging as the preferred one when ants apply the above described local, pheromone based, algorithm. The site `nest` and choice point `A` are represented as two different locations. Similarly for site `food` and choice point `B`. A constant flow of ants leaving the nest is assumed, with, on average, one ant leaving the nest every two seconds. The choice behaviour of the ants depends on the relative amounts of pheromone present at the beginning of each branch. In [10] a simple general choice function is used, which quantifies the way in which a higher concentration of pheromone on one branch gives a higher probability of choosing that branch, depending on the absolute and relative amounts on the beginning of these branches. The particular choice function suggested for the probability to choose the short branch at a choice point is of the form:

$$prob_S = (k + P_S_i)^d / ((k + P_S_i)^d + (k + P_L_i)^d)$$

where P_S_i represents the amount of pheromone in location S_i and P_L_i the amount of pheromone in location L_i, for $i \in \{1, 2\}$. In a similar way $prob_L$ can be defined. The parameter k quantifies the degree to which ants are sensitive to the difference in amounts of pheromone marking at the two branches when there is still relatively little pheromone present. For example, for $k = 1$ the probability for the first ant passing choice point `A` to choose one of the branches is $k/2k = 1/2$ because there is no pheromone on either branch yet. Suppose this ant chooses the long branch, then it lays a unit of pheromone at site `L1`. So, the next ant that passes choice point `A` in the same direction (assuming that no other ants have passed) has now a probability of $1/3$ to choose the short branch and $2/3$ to choose the long branch. For larger values of k the initial amounts of pheromones laid by the first ants produce a less accentuated difference in the probabilities.

For example, for $k = 20$ and $d = 1$ the probability that the second ant chooses the short branch reduces to $21/41 = 0.51$. Empirical results described in [10] have shown that $k = 20$ provides a realistic value. The parameter d in the formula determines the degree of non-linearity of the choice. A higher value of d amplifies the effect of the difference between the amounts of pheromone present on the two branches at a choice point. Empirical results showed that $n = 2$ provides a good fit between empirical data and the function modelling the choice. Note that $prob_S + prob_L = 1$. This simple mechanism, in which each ant that passes a choice point during exploratory recruitment modifies the following ant's probability to choose a particular branch by adding pheromone on the chosen branch, forms a positive feedback system in which, after some initial fluctuation, one branch emerges as being "selected", which is usually the shortest branch.

3 Bio-PEPA Briefly Recalled

Bio-PEPA [8, 6, 7], is a process algebraic language that originally was developed for the stochastic modelling and analysis of biochemical systems. Bio-PEPA models consist of two main kinds of components. The first kind is called the *"species" component*, describing the behaviour of individual entities. The second kind is the *model component*, describing the interactions between the various species. In the context of the paper, the individual entities are the robots, and the model component defines how they interact.

The syntax of Bio-PEPA components is defined as:

$$S ::= (\alpha, \kappa)\ \mathtt{op}\ S \mid S + S \mid C \quad \text{with } \mathtt{op} = \downarrow \mid \uparrow \mid \oplus \mid \ominus \mid \odot \quad \text{and } P ::= P \underset{\mathcal{L}}{\bowtie} P \mid S(x)$$

where S is a *species component* and P is a *model component*.

The *prefix combinator* "op" in the prefix term $(\alpha, \kappa)\ \mathtt{op}\ S$ represents the impact that action α has on species S. Specifically, $\downarrow$ indicates that the number of entities of species S reduces when α occurs, and $\uparrow$ indicates that this number increases. The amount of the change is defined by the *stoichiometry* coefficient κ. This coefficient captures the multiples of an entity involved in an occurring action. We will see an example of its use in the next section. The default value of κ is 1 in which case we simply write α instead of (α, κ). Action durations are assumed to be random variables (RVs) with negative exponential distributions, characterised by their *rates*. The rate of action α is defined by a so called functional rate or kinetics rate. Action rates are defined in the context section of a Bio-PEPA specification. The symbol $\oplus$ denotes an *activator*, $\ominus$ an *inhibitor* and $\odot$ a generic *modifier*, all of which play a role in an action without being produced or consumed and have a defined meaning in the biochemical context. The operator "+" expresses the choice between possible actions, and the constant C is defined by an equation $C{=}S$. The process $P \underset{\mathcal{L}}{\bowtie} Q$ denotes synchronisation between components P and Q, the set $\mathcal{L}$ determines those actions on which the components P and Q are forced to synchronise. The shorthand $P \overset{*}{\bowtie} Q$ denotes synchronisation on all actions that P and Q have in common. In $S(x)$, the parameter $x \in \mathbb{R}$ represents the initial amount of the species. A Bio-PEPA *system*

with *locations* consists of a set of species components, a model component, and a context containing definitions of locations, functional/kinetics rates, parameters, etc.. The prefix term (α, κ) `op` $S@l$ is used to specify that the action is performed by S in location l.

Bio-PEPA is given a formal operational semantics [8] which is based on Continuous Time Markov Chains (CTMCs). An alternative semantics for Bio-PEPA is also given in [7, 8] where Bio-PEPA models are mapped into sets of ordinary differential equations (ODEs) which allow for *fluid flow* approximation. As we have seen above, a Bio-PEPA model consists of a number of sequential components each of which represents a number of entities in a distinct state. The result of an action is to increase the number of some entities and decrease the number of others, these adjustments reflecting the stoichiometry with respect to the action. Thus we can represent the total state of the system at any point in time as a vector whose elements store the counts of each species component. This gives rise to a discrete state system which undergoes discrete events. Intuitively, the idea of fluid flow approximation is to *approximate* these discrete jumps by continuous flows between the states of the system. This approximation becomes good when entities are present in such high numbers as to make the frequency of actions high and the relative change from each single event small. In this case we can derive a set ODEs capturing the continuous approximation of system jumps, the solution of which is a vector of functions of time, which approximate the average behaviour of the CTMC; there is one such a function per each species and its value at time t gives the expected fraction, over the total population, of entities of that species at time t.

Bio-PEPA is supported by a suite of software tools which automatically process Bio-PEPA models and generate internal representations suitable for different types of analysis [8, 5]. These tools include mappings from Bio-PEPA to ODEs, supporting fluid flow approximation [14], to stochastic simulation models [12], to CTMCs with levels [7] and to PRISM models [16, 17] amenable to (statistical) model-checking. Consistency of the analyses is supported by a rich theory including process algebra, and the relationships between CTMCs and ODE.

4 A Bio-PEPA Model of Foraging Ants

To model the behaviour of foraging ants in Bio-PEPA [8, 6] the following parameters are used:

- N denotes the total, and constant, number of worker ants in the ants colony.
- *move* denotes the constant rate at which ants leave site `nest` to look for food. The rate at which ants leave site `food` depends also on the number of ants present on site `food` and is modelled by a mass action law.
- *walk_long* (*walk_short*) is the rate related to the average time an ant needs to traverse a section of the long (short, respectively) branch of the bridge.

Both the long and the short branch are composed of the same number of sections, i.e. 2. The length of a branch is modelled by the average time it takes

an ant to traverse a section on that branch. This time is longer for the long branch than for the short branch and is specified by the two parameters walk_long and walk_short. An alternative solution would be to vary the number of sections on the paths and keep the average walking time for each section the same.

The following compartments (locations) are introduced (apart from the default location 'top'):

- `nest` is the location of the ants' nest from which ants start initially.
- A is the location where they choose between the long and the short path when leaving the nest.
- S1 and S2 are the locations where ants lay pheromone after they selected the short branch. They do this every time they pass by these locations.
- L1 and L2 are the locations where ants lay pheromone after they selected the long branch. They do this every time they pass by these locations.
- `food` is the place where they collect food.
- B is the location where the ants decide which branch to take when returning from the food to the `nest`.

In Figure 1 the locations as well as the names of the transitions modelling the movement of individual ants are indicated. The direction of movement of the ants is modelled as part of the name of the species.

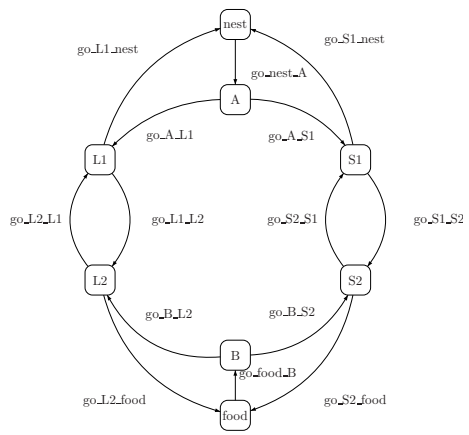


Fig. 1. Locations and transitions of ants in the model

In the following the behaviour of an *individual* ant is modelled. The focus is on how its behaviour effects the population of ants and the amount of pheromones in the relevant locations. It is assumed that all ants are in location `nest` initially. From there they move with a rate of one ant every 2 seconds towards the food. This rate is associated with the action *go_nest_A* labelling the transition of an ant moving from location `nest` to choice point `A`. In `A` ants choose between the long (*go_A_L*1) and the short (*go_A_S*1) branch. The total exit rate of an ant at a

choice point is 1, i.e. the sum of the probabilities given by the choice function to choose a branch. In this paper, we assume that, initially, there is no pheromone in locations `S1` and `L1`, so an ant chooses either way with approximately equal probability for $k = 20$. When there is significantly more pheromone in one of the locations, the rate with which that branch is chosen increases and the race condition between the rates for the long and the short branch model a probabilistic choice between them. So, the larger the difference between the amounts of pheromone at the beginning of a path, the higher the probability that an ant chooses the branch with the highest amount of pheromone. The Bio-PEPA fragment below models this initial behaviour of the ant:

$$\begin{aligned} Ant_nest = {} & go_nest_A{\downarrow}Ant_nest@nest+ \\ & go_S1_nest{\uparrow}Ant_nest@nest+ \\ & go_L1_nest{\uparrow}Ant_nest@nest; \end{aligned}$$

$$\begin{aligned} Ant_A = {} & go_A_S1{\downarrow}Ant_A@A+ \\ & go_A_L1{\downarrow}Ant_A@A+ \\ & go_nest_A{\uparrow}Ant_A@A; \end{aligned}$$

After an ant selects a branch, it lays some pheromone in the location at the beginning of a branch in `S1` or `L1` as a side-effect of the actions go_A_S1 and go_A_L1, respectively. This part of the behaviour is modelled as processes Ant_S1_NtoF and Ant_L1_NtoF, where the suffix 'NtoF' indicates that the ant's travelling direction is from `nest` to `food`. Pheromone is also left in the locations along the branch when ants pass by them on their way back. This part of the behaviour is modelled by the processes with suffix 'FtoN'. Below an excerpt from the model for an ant moving to and returning via $S1$ is shown. The behaviour involving $S2$, $L1$ and $L2$ is similar and not shown.

$$\begin{aligned} Ant_S1_NtoF = {} & go_A_S1{\uparrow}Ant_S1_NtoF@S1+ \\ & go_S1_S2{\downarrow}Ant_S1_NtoF@S1; \end{aligned}$$

$$\begin{aligned} Ant_S1_FtoN = {} & go_S2_S1{\uparrow}Ant_S1_FtoN@S1+ \\ & go_S1_nest{\downarrow}Ant_S1_FtoN@S1; \end{aligned}$$

After traversing the short or long branch, ants end up in location `food`. From there they go eventually back to the nest, passing by choice point `B`, as specified in the fragment below:

$$\begin{aligned} Ant_food = {} & go_food_B{\downarrow}Ant_food@food+ \\ & go_S2_food{\uparrow}Ant_food@food+ \\ & go_L2_food{\uparrow}Ant_food@food; \end{aligned}$$

$$\begin{aligned} Ant_B = {} & go_B_S2{\downarrow}Ant_B@B+ \\ & go_B_L2{\downarrow}Ant_B@B+ \\ & go_food_B{\uparrow}Ant_B@B; \end{aligned}$$

The increment in pheromone level takes place as a side-effect of an ant passing by a particular location on its path. This is modelled by the change in pheromone

level at each of the locations and occurs when a go_X_Y action happens on which the pheromone process is synchronised, for example:

$$P_S1 = go_A_S1{\uparrow}P_S1@S1 + go_S2_S1{\uparrow}P_S1@S1;$$

The system model below shows the initial number of ants in each location using cooperation with synchronisation on shared actions:

$$\begin{array}{l}
Ant_Nest@Nest[N] \underset{*}{\bowtie} Ant_A@A[0] \underset{*}{\bowtie} \\
Ant_S1_NtoF@S1[0] \underset{*}{\bowtie} Ant_S1_FtoN@S1[0] \underset{*}{\bowtie} P_S1@S1[0] \underset{*}{\bowtie} \\
Ant_S2_NtoF@S2[0] \underset{*}{\bowtie} Ant_S2_FtoN@S2[0] \underset{*}{\bowtie} P_S2@S2[0] \underset{*}{\bowtie} \\
Ant_L1_NtoF@L1[0] \underset{*}{\bowtie} Ant_L1_FtoN@L1[0] \underset{*}{\bowtie} P_L1@L1[0] \underset{*}{\bowtie} \\
Ant_L2_NtoF@L2[0] \underset{*}{\bowtie} Ant_L2_FtoN@L2[0] \underset{*}{\bowtie} P_L2@L2[0] \underset{*}{\bowtie} \\
Ant_Food@Food[0] \underset{*}{\bowtie} Ant_B@B[0]
\end{array}$$

The rates of the actions in the model reflect those found empirically by Goss et al. [13].

Ants leave the nest at a rate of one ant every 2 seconds. This is modelled by the constant rate *move*. To avoid that a reaction takes place when there are no ants present in the nest, the rate is multiplied with a factor that makes sure that there is a positive number of ants in the nest (H(Ant_nest@nest)). Ants returning to their nest from location `food` are assumed to leave that location with a rate depending on the number of ants present in that location.

$$\begin{array}{r}
kineticLawOf\,go_nest_A : move * H(Ant_nest@nest); \\
kineticLawOf\,go_food_B : move * Ant_food@food;
\end{array}$$

The rates at which ants choose a branch at a choice point are defined using the choice function of degree 2, for example for ants leaving choice point `A`:

$$\begin{array}{r}
kineticLawOf\,go_A_S1 : \frac{(k+P_S1@S1)^2}{(k+P_S1@S1)^2+(k+P_L1@L1)^2} * Ant_A@A; \\
kineticLawOf\,go_A_L1 : \frac{(k+P_L1@L1)^2}{(k+P_S1@S1)^2+(k+P_L1@L1)^2} * Ant_A@A;
\end{array}$$

It takes ants more time to traverse a section on the long branch than on the short branch. This is simply modelled by two different rates; `walk_long` and `walk_short`. For example:

$$\begin{array}{r}
kineticLawOf\,go_L1_L2 : walk_long * Ant_L1_NtoF@L1; \\
kineticLawOf\,go_S1_S2 : walk_short * Ant_S1_NtoF@S1;
\end{array}$$

5 Emerging Paths

Despite the simplicity of the model of foraging ants there are a number of interesting aspects to analyse. In the present paper we focus on whether or under what conditions stochastic simulation and fluid flow analysis confirm that the simple trail laying and selection mechanism indeed are enough to lead to the

emergence of the shortest path in most cases. In other words, we want to know whether or under what conditions our model shows a similar emergent behaviour as has been observed in empirical research, so that we can validate it.

In order to be able to compare results obtained from the model with those from empirical research presented in Goss et al. [13] the model is analysed for the following values of the parameters taken from [13]:

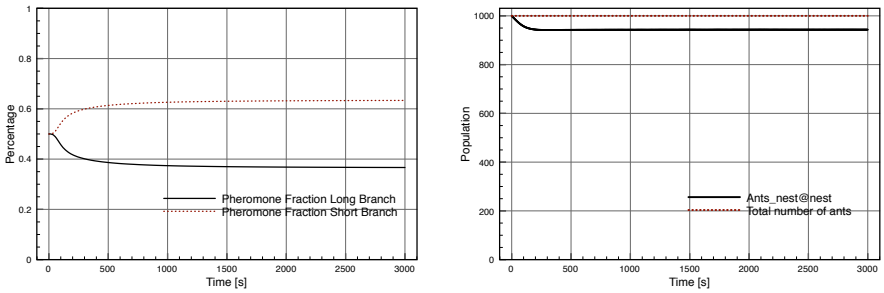
- $N = 1000$: number of workers in the ant colony
- $move = 0.5$: one ant every two seconds leaves the nest
- $r = 2$: relation between length of long branch w.r.t. short branch
- $walk_short = 0.05$: the short branch takes on average 20 seconds to be traversed
- $walk_long = 0.05/r$: traversing the long branch takes r times much time as traversing the short branch
- $k = 20$: factor k in the choice function

One way to visualize the relative preference of ants for one branch w.r.t. the other is to show how the fraction of pheromone present on each branch changes over time. This is shown in Fig. 2(a) for a stochastic simulation with 10000 independent runs over a time period of 3000 seconds (50 min.), covering the duration of the experiments by Goss et al.. The figure shows that, after a brief initial time interval, on average, there is more pheromone on the short branch than on the long branch. Inspection of single simulation runs reveals that the behaviour tends to two different stable states: one in which all ants use the short branch (Fig. 3(a)) and one in which all ants use the long branch (Fig. 3(b)). Two such simulations are shown in Fig. 3. However, on average the short branch emerges more often than the long branch for the given parameter values, which explains the results in Fig. 2(a).

Fig. 2(b) shows the total number of ants in the colony and the number of ants that are in the nest over time, which are at least 942 at any time.

Let us now turn to a fluid flow analysis of the same model and same parameter values as shown in Fig. 5(a). It is immediately clear that the fluid results are quite different from the simulation results of Fig. 2(a). Fig. 5(a) seems to indicate that the short branch emerges much earlier and more often than results from a stochastic simulation of the same model with 10000 independent runs.

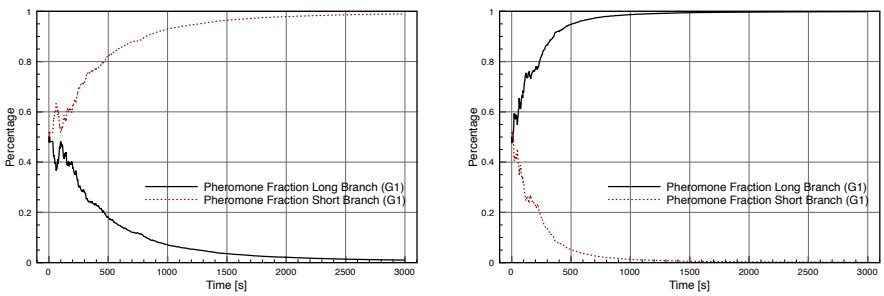
An important result by Kurtz [15] on continuous approximation of stochastic processes says that the fluid approximation can be obtained as the limit of a sequence of CTMC models for increasing population levels. In particular, it has been shown that such approximation works well if the rates of the model can be shown to be density dependent. But let us first consider what it would mean to generate a sequence of models with increasing population levels for the ant colony model. If a system has a total number of components that does not change when the system evolves this total could be taken as the system size. However, in the ant colony model the number of ants is constant, but the amount of pheromone in the various locations grows unlimited over time. So, let us consider as the system size the sum of the *initial* populations of ants and pheromone and



(a) Fraction of pheromone amounts at beginning of short (`S1`) and long (`L1`) branch over time

(b) Number of ants present in the nest

Fig. 2. Stochastic simulation, 10000 independent runs

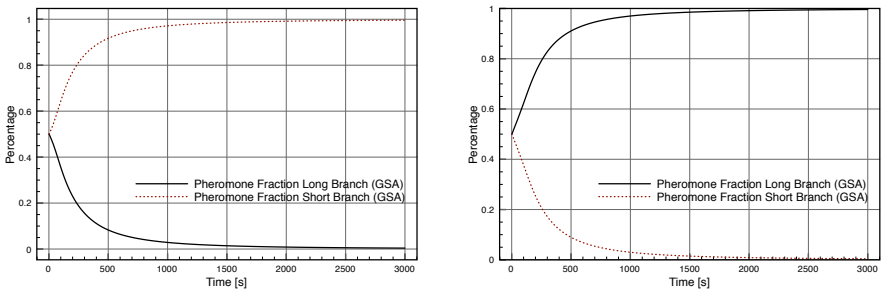


(a) Fraction of pheromone amounts in `S1` and `L1`

(b) Fraction of pheromone amounts in `S1` and `L1`

Fig. 3. Two different single simulation runs

consider a sequence of systems where this populations are scaled with a factor $E \in \mathbb{N}$, so $N = 1000 * E$. Unfortunately, this alone does not mean that there are really more ants active (i.e. out of the nest) in our system. This is due to the fact that ants leave the nest with a constant rate of one ant every two seconds. We can work around this by scaling the exit rate as well by factor E, i.e. define $move = 0.5 * E$. Fig. 4(a) shows the result of a single simulation of the model scaled by a factor $E = 100000$ and renormalised afterwards. The shape of the graph seems to get closer to the fluid flow results of Fig. 5(a). However, a second simulation, shown in Fig. 4(b) of exactly the same model and parameters shows that it is also possible that the *long* branch emerges, giving a simulation trace that differs completely from that obtained by fluid flow analysis. In other words the simulation results are very unstable despite the large population considered.



(a) Fraction of pheromone amounts in `S1` and `L1`

(b) Fraction of pheromone amounts in `S1` and `L1`

Fig. 4. Two different single simulation runs for model scaled by E=100000

Let us consider the rate function of the actions go_A_S1 and go_A_L1 for the choice between two branches. The factor k in that function could also be interpreted as the initial amount of pheromone present at both branches. Viewed that way, it would be interesting to consider a model in which also k is scaled in a similar way as N, i.e. $k = 20 * E$. We found that, in this case, single simulation traces, such as the one shown in Fig. 5(b), give indeed results that are very similar to the fluid flow results in Fig. 5(a); in the sequel we give a formal justification to this intuitive interpretation. A closer inspection of the rate function involving the choice between branches showed that it is indeed easy to verify that it is density dependent when also k is scaled, thus the hypothesis of Kurtz' theorem are fulfilled and the stochastic process associated with the underlying CTMC—which the simulation is based on—*coincides*, in the limit, when scaling the population to infinite, with the solution of the ODEs. If k is not scaled, it is not immediate how a remaining dependency on the population size in the formula could be dealt with.

More formally, consider the rate function for the transition go_A_S1 for the choice of the short branch at choice point A:

$$f_{go_A_S1}(\bar{X}) = \frac{(k+P_S1@S1)^2}{(k+P_S1@S1)^2+(k+P_L1@L1)^2} * Ant_A@A;$$

where $\bar{X}$ denotes the population vector of the model containing the values for $P_S1@S1$, $P_L1@L1$, $Ant_A@A$ etc. To show that this rate function is density dependent we need to find a function g such that:

$$E \cdot A_0 \cdot g(\frac{1}{E \cdot A_0}(\bar{X})) = f(\bar{X})$$

where A_0 is the initial value of ants in the nest and E is the multiplication factor.

Let us focus on transition go_A_S1. Consider a function g such that its component for transition go_A_S1, which we denote by $g_{go_A_S1}$, is defined as below, assuming that also k is scaled and is part of the population vector $\bar{X}$. Let us also assume that the arguments of $g_{go_A_S1}$ for the values related to $P_S1@S1$, $P_L1@L1$, $Ant_A@A$, k etc. are indicated by Y_{P_S1}, Y_{P_L1}, Y_{Ant_A}, Y_k etc., resp.:

$$g_{go_A_S1}(\bar{Y}) = \frac{(Y_k + Y_{P_S1})^2}{((Y_k + Y_{P_S1})^2 + (Y_k + Y_{P_L1})^2)} \cdot Y_{Ant_A}$$

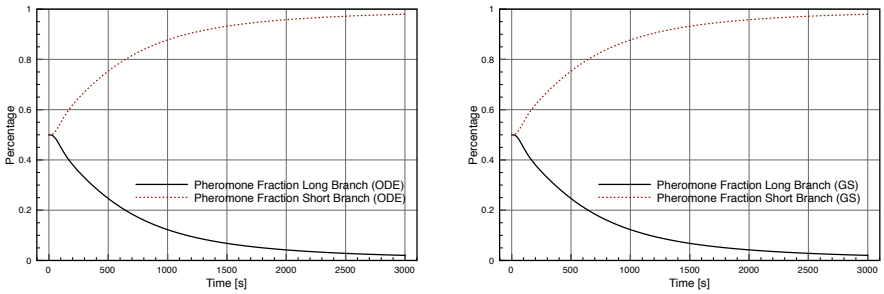
Now we need to show that $E \cdot A_0 \cdot g(\frac{1}{E \cdot A_0}(\bar{X})) = f(\bar{X})$ for the transition from A to the short branch:

$$\begin{aligned}
& E \cdot A_0 \cdot g_{go_A_S1}(\tfrac{1}{E \cdot A_0}(\bar{X})) \\
= & \\
& E \cdot A_0 \cdot \frac{(\frac{k}{E \cdot A_0} + \frac{P_S1@S1}{E \cdot A_0})^2}{((\frac{k}{E \cdot A_0} + \frac{P_S1@S1}{E \cdot A_0})^2 + (\frac{k}{E \cdot A_0} + \frac{P_L1@L1}{E \cdot A_0})^2)} \cdot \frac{Ant_A@A}{E \cdot A_0} \\
= & \\
& \frac{(\frac{k}{E \cdot A_0} + \frac{P_S1@S1}{E \cdot A_0})^2}{((\frac{k}{E \cdot A_0} + \frac{P_S1@S1}{E \cdot A_0})^2 + (\frac{k}{E \cdot A_0} + \frac{P_L1@L1}{E \cdot A_0})^2)} \cdot Ant_A@A \\
= & \\
& \frac{\frac{1}{(E \cdot A_0)^2} \cdot (k+P_S1@S1)^2}{\frac{1}{(E \cdot A_0)^2} \cdot (k+P_S1@S1)^2 + \frac{1}{(E \cdot A_0)^2} \cdot (k+P_L1@L1)^2} * Ant_A@A \\
= & \\
& \frac{(k+P_S1@S1)^2}{(k+P_S1@S1)^2+(k+P_L1@L1)^2} * Ant_A@A
\end{aligned}$$

It is easy to see that g in this case is indeed the function we were looking for. Other rate functions in the model are simple mass action functions or similar to the one shown above. So, the model can be shown to be density dependent when scaling the ant population, the pheromones and the factor k. This explains the good correspondence found in that case between fluid flow results and renormalised stochastic simulation runs of a scaled model with $E = 100000$ as shown in Fig. 4.

6 Related Work

It is not the first time that formalisms of a process algebraic nature are used to model and analyse the social behaviour of insects. In the work by Tofts in the



(a) Fraction of pheromone amounts in `S1` and `L1` (ODE)

(b) Fraction of pheromone amounts in `S1` and `L1` (G1 scaled)

Fig. 5. Fluid flow analysis and scaled simulation

early nineties [20] the calculus Weighted Synchronous CCS (WSCCS), an extension of Milner's SCCS [18], was used to describe a probabilistic synchronisation algorithm assumed to underly the observed auto-synchronisation behaviour of ants. In that work the behaviour of individual ants has been modelled and the global behaviour has been analysed by studying the related transition matrices and by performing model based simulation. Later work by Sumpter et al. [19] extends Tofts' work on modelling ant colony behaviour by further types of analyses of the WSCCS models, like Markov chain analysis and mean field methods. Mean field approximation is a technique which is similar to fluid flow analysis but it is usually applied to Discrete Time Markov Chains (DTMCs) rather than CTMCs. The use of CTMCs instead of DTMCs adds to the model the dimension of (continuous) *time*, still preserving the probabilistic behaviour. Consequently, it makes it natural to model issues like the fact that it takes *longer* to ants to traverse the long path.

7 Discussion and Further Work

The emergent effect of ants choosing the short branch can be explained by the positive feedback information that returning ants provide. The ants that choose the short branch arrive before those that chose the long branch and therefore more pheromone builds up at the beginning of the short branch. This only works if pioneer ants have time to return to the nest before many of their fellow ants start to look for food too. Choosing a constant rate for ants leaving the nest leads to a model that better reflects such experimental observations.

The small exploratory study of ants described in the present note illustrates some of the progress that has been made in recent years that facilitates considerably the analysis of process algebraic models of collective behaviour. Among such progress is the development of automated tools for the generation of stochastic simulations and *continuous* fluid flow approximations from process algebraic

models of large collectives of interacting agents. The inclusion of explicit notions of locality and context dependent flexible definition of activity rates in the formalism may further broaden the kind of dynamical systems that can be modelled and analysed using a formal modelling approach. Such an approach has the additional advantage that it provides a mathematical underpinning of the collective behaviour under study and the related underlying distributed and stochastic algorithms. Furthermore, fluid flow methods exhibit a high degree of *scalability* in the numbers of individuals of each species or, more generally, in the number of processes in a certain state. In fact, such numbers are just the initial condition of the initial value problem for the associated ODEs. On the other hand, the approach is more sensitive to the number of different species, since the number of differential equations in the system grows (linearly) with it.

As we have seen, scaling k made it easy to prove density dependence. This fact brought us to the conjecture that if k is considered a constant of the model, then the scaling conditions of Kurtz' theorem might not be fulfilled. The full consequences of these findings are topic of further study.

There are several other issues which remain to be addressed and which we would like to study further, in the future. We have observed that which of the two branches emerges depends critically on the initial conditions of the model. Such effects can be studied by imposing artificially a larger initial amount of pheromone on the long branch than on the short branch and perform a fluid flow analysis. It would be interesting to study this phenomenon and establish a critical value for such a "phase shift" for different conditions of sensitivity of ants to pheromone differences and its relation to the value k, i.e. the sensitivity of ants to differences in relatively small amounts of pheromone. Another parameter to which sensitivity of the model could be studied is the ratio r between the long and the short path (set to 2 in this paper). It would also be interesting to investigate the effect of choice functions with higher or lower degrees and, of course, to investigate whether and how the model could be extended for more complicated topologies.

Finally, in this work we have used Bio-PEPA as modelling language and the related Bio-PEPA toolset for fluid flow analysis and simulation. However, a number of other interesting formal languages for stochastic modelling have been proposed, among which SCCP [2–4], a Stochastic Concurrent Constraint language, fluid flow PEPA [14] and StoKLAIM [9], a stochastic variant of a Linda-like language based on tuple spaces and with asynchronous coordination. These languages reflect different modelling concepts, while for all of them, analysis techniques such as stochastic simulation, e.g. in PRISM [16], and fluid flow approximation have been implemented or are under development. We plan to consider the various approaches taking variants of the foraging ant colony as a central modelling theme.

Acknowledgments. The authors would like to thank Stephen Gilmore, Allan Clark and Adam Duguid (University of Edinburgh) for their support with the Bio-PEPA plug-in and other features. We also would like to thank Jane Hillston

(University of Edinburgh) for making us aware of the earlier work on modelling ant colony behaviour and Mirco Tribastone (LMU Munich), Luca Bortolussi (University of Trieste) and Michele Loreti (University of Florence) for taking up ant modelling as well in different formalisms and sharing with us their findings and results which contributed to improvements in the model and analysis presented in the present paper. This research has been partially funded by the EU-IP FET-Open project ASCENS (nr. 257414) and Project TRACE-IT - PAR FAS 2007-2013 - Regione Toscana.

References

1. Ascens Project: http://www.ascens-ist.eu/
2. Bortolussi, L.: Stochastic concurrent constraint programming. In: Proceedings of the 4th International Workshop on Quantitative Aspects of Programming Languages (QAPL 2006). vol. 164-3. ENTCS (2006)
3. Bortolussi, L., Policriti, A.: Modeling biological systems in concurrent constraint programming. Constraints 13(1-2), 66–90 (2008)
4. Bortolussi, L., Policriti, A.: Dynamical systems and stochastic programming — from ordinary differential equations and back. Transactions of Computational Systems Biology XI, 216–267 (2009)
5. Ciocchetta, F., Duguid, A., Gilmore, S., Guerriero, M.L., J., H.: The Bio-PEPA Tool Suite. In: Proc. of the 6th Int. Conf. on Quantitative Evaluation of SysTems (QEST 2009). pp. 309–310 (2009)
6. Ciocchetta, F., Guerriero, M.L.: Modelling biological compartments in Bio-PEPA. ENTCS 227, 77–95 (2009)
7. Ciocchetta, F., Hillston, J.: Bio-PEPA: An extension of the process algebra PEPA for biochemical networks. ENTCS 194(3), 103–117 (2008)
8. Ciocchetta, F., Hillston, J.: Bio-PEPA: A framework for the modelling and analysis of biological systems. TCS 410(33-34), 3065–3084 (2009)
9. De Nicola, R., Katoen, J., Latella, D., Loreti, M.: StoKLAIM: A stochastic extension of KLAIM (2006), CNR-ISTI Technical Report number ISTI-2006-TR-01
10. Deneubourg, J.L., Aron, S., Goss, S., Pasteels, J.M.: The self-organizing exploratory pattern of the argentine ant. Journal of Insects Behaviour 3(2) (1990)
11. Dorigo, M., Stützle, T.: Ant Colony Optimization. The MIT Press (2004)
12. Gillepie, D.T.: Exact stochastic simulation of coupled chemical reactions. The Journal of Physical Chemistry 81(25), 2340–2361 (1977)
13. Goss, S., Aron, S., Deneubourg, J.L., Pasteels, J.M.: Self-organized shortcuts in the Argentine Ant. Naturwissenschaften 76, 579–581 (1989)
14. Hillston, J.: Fluid flow approximation of PEPA models. In: Proceedings of QEST'05. pp. 33–43. IEEE Computer Society (2005)
15. Kurtz, T.G.: Solutions of ordinary differential equations as limits of pure Markov processes. Journal of Applied Probability 7(1), 49–58 (1970)
16. Kwiatkowska, M., Norman, G., Parker, D.: PRISM: Probabilistic model checking for performance and reliability analysis. ACM SIGMETRICS Performance Evaluation Review (2009)
17. Kwiatkowska, M., Norman, G., Parker, D.: PRISM 4.0: Verification of probabilistic real-time systems. In: Gopalakrishnan, G., Qadeer, S. (eds.) Proc. 23rd International Conference on Computer Aided Verification (CAV'11). LNCS, vol. 6806, pp. 585–591. Springer (2011)

18. Milner, R.: Calculi for synchrony and asynchrony. Theoretical Computer Science 25(3), 267–310 (1983)
19. Sumpter, D.J.T., 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), doi: 10.1006/bulm.2001.0252
20. Tofts, C.: The autosynchronisation of *leptothorax acervorem* (fabricius) described in WSCCS. Tech. Rep. ECS-LFCS-90-128, LFCS, University of Edinburgh (1990)