Edinburgh Research Explorer

Automatic Moment-Closure Approximation of Spatially Distributed Collective Adaptive Systems

Automatic Moment-closure Approximation of Spatially Distributed Collective Adaptive Systems

CHENG FENG, University of Edinburgh
JANE HILLSTON, University of Edinburgh
VASHTI GALPIN, University of Edinburgh

Spatially distributed collective adaptive systems are an important class of systems, which pose significant challenges to modelling due to the size and complexity of their state spaces. This problem is acute when the dynamic behaviour of the system must be captured, for example in order to predict system performance. In this paper we present an abstraction technique which automatically derives a moment-closure approximation of the dynamic behaviour of a spatially distributed collective adaptive system from a discrete representation of the entities involved. The moment-closure technique is demonstrated to give accurate estimates of dynamic behaviour, although the number of ordinary differential equations generated for the second order joint moments can grow large in some cases. For these cases, we propose a rigorous model reduction technique and demonstrate its use to substantially reduce the computational effort with only limited impact on the accuracy if the reduction threshold is set appropriately. All the techniques reported in this paper are implemented in a tool which is freely available for download.

CCS Concepts: •**Computing methodologies** → **Model development and analysis;** •**Mathematics of computing** → *Mathematical analysis;* •**Theory of computation** → Formal languages and automata theory;

Additional Key Words and Phrases: Collective adaptive systems, moment-closure approximation, model reduction

1. INTRODUCTION

Collective adaptive systems, consisting of many communicating entities who react according to their local knowledge, without centralised control, and which nevertheless achieve coherent results at the global level, are becoming pervasive. Capturing the *local* nature of knowledge means that the location of an entity is a key characteristic which may deeply influence the pattern and scope of communication. But recording this aspect of each entity adds to the complexity of modelling and analysing such systems. This problem is exacerbated by the fact that collective systems necessarily consist of many, many distributed entities.

In this paper we are concerned with predicting the behaviour of collective adaptive systems. As outlined above, the major challenge to such analysis is the size of the state space representation of the system, when the state encompasses not only logical

behaviour but also location. Numerical techniques traditionally used for performance analysis, based on a Markovian approach, are entirely infeasible. Discrete event simulation is feasible but extremely costly in terms of computational resources and may not scale as the size of the system increases. Here we propose an approach, based on a formal modelling language to support the description of a collective adaptive system as a discrete event system (amenable to simulation), which makes an approximation of the system as a set of ordinary differential equations (ODEs). Unlike earlier fluid approximation techniques of process algebras such as PEPA [Hillston 2005b; Tribastone et al. 2012], our approach is not limited to the expectation or first moment characterisation of system behaviour. Importantly, our moment-closure-based approach also incorporates second order moments supporting the analysis of the compliance of a system to service level agreements and other performance requirements.

The context in which we develop our approach is the stochastic process algebra PALOMA. This language was introduced in [Feng and Hillston 2014] and provides a high-level modelling language for the multi-message multi-class Markovian agent modelling framework of [Cerotti et al. 2010]. PALOMA has been revised since [Feng and Hillston 2014] in order to enhance the expressiveness of the language whilst also making model descriptions more compact. The language is equipped with both individual-based and population-level semantics. Here our focus is on the population level as we consider the emergent performance of the system when large numbers of entities interact. Thus we explain how to derive a set of ODEs describing the evolution of the moments of the population-level dynamics of an arbitrary PALOMA model. We apply moment-closure techniques to close the moment ODEs at the second order. The obtained ODEs are not, in general, amenable to analytical solution but can nevertheless be solved efficiently by numerical simulation. Importantly, the structure of the set of ODEs is independent of the number of agents in the model, making the approach scalable even in the face of very large populations. We illustrate the applicability of the approach on three case studies including a classic epidemic model, a wireless sensor network model and a bike-sharing model taken from the domain of smart urban transport, such as that found in London and many other European cities [Midgley 2009].

The number of ODEs characterising the evolution of the moments, whilst independent of the size of the populations of entities, does depend on the number of locations and local state space of each entity type. In some circumstances this can lead to a prohibitive number of ODEs, or to a slow-down in analysis. Thus we also propose a model reduction technique which generates a reduced set of ODEs on the basis of a formally defined neighbourhood relation that is defined at the level of process algebra description and can be automatically applied. We demonstrate through the three examples that this can significantly improve the efficiency and scalability of moment analysis whilst still retaining high accuracy.

The paper is structured as follows. We present the syntax and semantics of PALOMA in the next section. This is followed by the section of moment analysis of PALOMA models. In Section 4, three case studies will be presented. Finally, Sections 5 and 6 discuss related work and draw final conclusions.

2. PALOMA

The Process Algebra for Located Markovian Agents (PALOMA) is a stochastic process algebra, specifically designed to support the construction of formal models of large collective adaptive systems in which agents are distributed over a discrete set of named locations, $\mathcal{L}$. Agents are parameterised by a *location*, denoted by ℓ, $\ell \in \mathcal{L}$. Each individual agent is a finite state machine, and the language is *conservative* in the sense that no agents are created or destroyed during the evolution of the model. There is

a finite set of action types $\mathcal{A}$, and actions may be undertaken spontaneously or may be induced by a message of the same type, sent by another agent in the system. All spontaneous actions are assumed to have a duration governed by an exponential distribution and characterised by a rate r. A model P consists of a number of agents composed in parallel. There is no direct communication between agents, e.g. in the style of CSP [Hoare 1985] or PEPA [Hillston 2005a], but synchronisation between agents is achieved through message passing.

The language has the following grammar:

$$\pi \ ::= \ !(\alpha,r)@\mathbf{IR}\{\overrightarrow{\ell}\} \ | \ ?(\alpha,p)@\mathbf{Pr}\{v\} \ | \ !!(\alpha,r)@\mathbf{IR}\{\overrightarrow{\ell}\} \ | \ ??(\alpha,p)@\mathbf{Wt}\{v\} \ | \ (\alpha,r)$$
$$S(\ell) \ ::= \ \pi.S'(\ell') \ | \ S_1(\ell) + S_2(\ell) \ | \ C$$
$$P \ ::= \ S(\ell) \ | \ P \parallel P$$

Agents can change their states and locations by different actions:

Spontaneous action with broadcast message emission: $!(\alpha,r)@\mathbf{IR}\{\overrightarrow{\ell}\}$ describes that the agent performs an action α, $\alpha \in \mathcal{A}$, *spontaneously* with rate r. During the occurrence of the action, a broadcast message, also typed α, is emitted. The *influence range* of the broadcast is defined by the location vector $\overrightarrow{\ell}$, which gives a list of locations where agents can *potentially* be influenced by this message. $\overrightarrow{\ell}$ can be defined both statically and dynamically. For example, $\overrightarrow{\ell} = [\ell_1, \ell_2, \ell_3]$ means that the influence range of the broadcast is locations ℓ_1, ℓ_2 and ℓ_3, whereas $\overrightarrow{\ell} = range(d)$ denotes that the influence range is a set of locations whose distance from the location of the sender agent is less than a specific threshold d. Some other frequently used definitions of influence range are $\overrightarrow{\ell} = local$ and $\overrightarrow{\ell} = all$, which represent that the influence range of the broadcast message is restricted to the location of the sender agent or consists of all the locations in the model, respectively.

Spontaneous action with unicast message emission: $!!(\alpha,r)@\mathbf{IR}\{\overrightarrow{\ell}\}$ also describes a spontaneous action of type α, rate r and influence range $\overrightarrow{\ell}$. The difference is that here the message is a unicast, meaning that at most one agent can receive the message.

Action induced by a broadcast message: $?(\alpha,p)@\mathbf{Pr}\{v\}$ describes that the agent performs an action α *immediately* after receiving and accepting a broadcast message of type α. Whether the agent receives the broadcast message is decided by two factors. Firstly, the agent must be located within the influence range of the message; otherwise, the message will be ignored. Secondly, the value $v \in [0,1]$ gives the probability that the message is received by the agent given that it is within the influence range of the broadcast. v can be defined as a constant. For example, $v = 0.5$ means that the agent has 50% chance of receiving the message. v can also be defined dynamically. For instance, $v = 1/|S(\ell)|$ denotes that the message reception probability is dependent on the number of agents in state S in location ℓ, where $| \cdot |$ is an operator which gives the number of agents in a particular state and location. Formally, the definition of v follows this grammar:

$$v \ ::= \ c \quad | \quad dist(\ell_1,\ell_2) \quad | \quad |S(\ell)| \quad | \quad v \ (op) \ v$$

where c is a constant real number, $dist(\ell_1,\ell_2)$ is the distance between locations ℓ_1 and ℓ_2, (op) is a basic arithmetic operator. This means that v defines a function that can be dependent on the number of agents, and distance between locations. Once the message has been received, the agent decides whether to accept it. Here, a constant value $p \in [0,1]$ encodes the probability that the agent will accept the message. This can be thought of as the agent choosing to respond to a spontaneous action of the given type with probability p. The definition of v and p support a rich set of possible interaction patterns between agents.

Action induced by a unicast message: $??(\alpha, p)@\mathbf{Wt}\{v\}$ describes that the agent performs an action α immediately after receiving and accepting a unicast message of type α. Here, $v \in \mathbb{R}^+$ gives the *weight* of the agent to be the receiver of this unicast message. The definition of v follows the same grammar as previously, but with a different value domain. The weights are used to resolve between several potential receiver agents: suppose there are n agents denoted by $S_1(\ell_1), S_2(\ell_2), ..., S_n(\ell_n)$, which can potentially receive the unicast message, with weights $v_1, v_2, ..., v_n$. Then, the probability that agent $S_1(\ell_1)$ receives the message is v_1/Σ, where Σ denotes $\sum_{i=1}^n v_i$, the sum of the associated weights of all potential receivers. The calculation of this probability can be seen as a new function w_{v_1} obtained from the functions v_i, and it may also involve agents counts. If there is no potential receiver, the message is simply discarded. The value $p \in [0,1]$ is a distinct probability deciding whether a received message is accepted or not. Note that if the selected agent does not accept the unicast message, the message is discarded; it cannot be passed to any other potential receiver agent.

Spontaneous action without message emission: (α, r) denotes that the agent performs a spontaneous action named α with a rate r governed by a negative exponential distribution. No message is sent out during the firing of this action. Thus it remains an individual action solely of this agent.

Alternative behaviours are represented by the standard choice operator, $+$. A choice between spontaneous actions is resolved via the race policy, based on their corresponding rates. We assume that there is never a choice between two induced actions of the same type within a single component. C denotes a constant name for an agent. The symbol $\|$ means parallel composition of agents.

2.1. Semantics

In this section, we present the population-level semantics of PALOMA which provides the theoretical foundation for population-level stochastic simulation of PALOMA models. Simulating a PALOMA model at the population level can dramatically reduce the computational cost compared with traditional individual-based (agent-based) simulation when the model contains a large number of symmetric agents. More importantly, the population-level semantics also serves as an intermediate tool for the generation of the moment ODEs which we will see later.

In PALOMA, as agents in the same state and location are indistinguishable, it is advantageous to derive a population model in which symmetric agents located in the same location are aggregated through a counting abstraction. Specifically, we first construct a state vector $\mathbf{S}$ whose size is $|\mathbf{S}|$, where each element S_i denotes an agent state variable that appears in the definition of the PALOMA model. Then, a location vector $\mathbf{L}$ whose size is $|\mathbf{L}|$ is also constructed, in which each element ℓ_i denotes a location variable that appears in the PALOMA model ($\mathbf{S}$ and $\mathbf{L}$ can be initialized by simply traversing the model definition and listing all the distinct states and locations in the

$$S(\ell)[n] \equiv \underbrace{S(\ell) \| \ldots \| S(\ell)}_{n \text{ copies}}$$

$$S(\ell) \equiv S(\ell)[1]$$

$$S(\ell)[n_1] \| S(\ell)[n_2] \equiv S(\ell)[n_1 + n_2]$$

$$S(\ell) \| S'(\ell') \equiv S'(\ell') \| S(\ell)$$

$$\Big(S(\ell) \| S'(\ell')\Big) \| S''(\ell'') \equiv S(\ell) \| \Big(S'(\ell') \| S''(\ell'')\Big)$$

Fig. 1. Structural congruence in PALOMA

model). Furthermore, we use a $|\mathbf{S}| \times |\mathbf{L}|$ numerical matrix $\boldsymbol{\xi}$ to represent the current count of agents in all possible states and locations. Specifically, the element $\xi_{i,j}$ at the ith row and jth column of the matrix denotes the current number of agents in state S_i and location ℓ_j (we set $\xi_{i,j} = -1$ if $S_i(\ell_j)$ does not exist).

Before introducing the semantics, we define a structural congruence in Figure 1, which will allow us to define the semantics and the models in a more compact and straightforward way. Moreover, we let $\mathrm{pf}(S_i(\ell_j))$ denote the set of prefixes of an agent in state S_i and location ℓ_j such that:

$$\mathrm{pf}(S_i(\ell_j)) \; = \; \{\forall \, \pi_n.S_{i_n}(\ell_{j_n}) \mid S_i(\ell_j) = \sum_{n \in N} \pi_n.S_{i_n}(\ell_{j_n})\}$$

Now, we formally define the population-level structured operational semantics with rules for the derivation of population-level transitions for PALOMA in Figure 2. The rule NoMsg infers a population-level transition from a spontaneous action with no message emission of a single agent with rate r. The idea is that if there are $\xi_{i,j}$ copies of agents in state S_i and location ℓ_j at any given time instant ($\xi_{i,j} > 0$), then the total rate at which a spontaneous action with no message emission in the premise fires is $r \times \xi_{i,j}$.

The rule BrPair infers a set of population-level transitions from an induced action coupled with a spontaneous action with broadcast message emission with the same action name. Suppose there are $\xi_{i,j}$ copies of agents $S_i(\ell_j)$ who can do a spontaneous action with broadcast message emission at rate r at any time instant, then the total emission rate of the broadcast message α from those agents is $r \times \xi_{i,j}$. For an agent in state S_m and located in ℓ_n which is within the influence range of the broadcast message given by $\overrightarrow{\ell}$, the probabilities of receiving and accepting the message are v and p, respectively. Then, suppose a spontaneous action α with broadcast message emission is fired by an agent in state S_i and location ℓ_j, in the population level, the number of agents in state S_m and location ℓ_n who actually fire the corresponding induced action caused by the broadcast message is a random variable (denoted as K) following a Binomial distribution, $K \sim Binomial(\boldsymbol{\xi}_{m,n}, v \times p)$. Thus, we can infer $\boldsymbol{\xi}_{m,n} + 1$ population-level transitions denoted as $\rightarrow_k$ each with rate $r \times \xi_{i,j} \times \mathrm{Pr}(k, \boldsymbol{\xi})$, and during which a copy of $S_i(\ell_j)$ goes to $S_{i'}(\ell_{j'})$ and k copies of $S_m(\ell_n)$ go to $S_{m'}(\ell_{n'})$, where:

$$\mathrm{Pr}(k, \boldsymbol{\xi}) = \binom{\xi_{m,n}}{k}(p \times v(\boldsymbol{\xi}))^k(1 - p \times v(\boldsymbol{\xi}))^{\xi_{m,n}-k} \quad \forall k, \; 0 \le k \le \boldsymbol{\xi}_{m,n}$$

As there can be multiple agents in different states or locations which enable actions induced by the same broadcast message, we use a further rule BrCombo to infer a population-level transition from a spontaneous action with broadcast message coupled with multiple agent types. As different agents choose to receive or respond to the message independently, the probabilities of the number of actions of the different agents induced by the message can be multiplied to obtain the total probability.

The rule UniPair infers a population-level transition from an induced action coupled with a spontaneous action with unicast message emission. This requires the use of the function w_v to transform the weights of all agents involved into a probability. Given n weight functions v_i with $v_i(\boldsymbol{\xi}) \in \mathbb{R}^+$ then $w_{v_j}(\boldsymbol{\xi}) = v_j(\boldsymbol{\xi})/\sum_{i=1}^{n} v_i(\boldsymbol{\xi})$.

The above four rules can be used to infer basic population-level transitions for any PALOMA model.

In the rules Choice and Parallel, we use expressions E and F to denote a component which consists of a set of agents which together enable a basic population-level transition. As can be seen in the rule Choice, a choice between two basic population-level transitions is resolved by a race policy, where the associated agents can either behave

NoMsg $\dfrac{(\alpha,r).S_{i'}(\ell_{j'}) \in \mathrm{pf}(S_i(\ell_j))}{S_i(\ell_j) \xrightarrow{(\alpha,r \times \xi_{i,j})} S_{i'}(\ell_{j'})}$

BrPair $\dfrac{!(\alpha,r)@\mathbf{IR}\{\overrightarrow{\ell}\}.S_{i'}(\ell_{j'}) \in \mathrm{pf}(S_i(\ell_j)) \quad ?(\alpha,p)@\mathbf{Pr}\{v\}.S_{m'}(\ell_{n'}) \in \mathrm{pf}(S_m(\ell_n)) \quad \ell_n \in \overrightarrow{\ell}}{\textbf{For } 0 \le k \le \boldsymbol{\xi}_{m,n} \quad S_i(\ell_j) \parallel S_m(\ell_n)[k] \xrightarrow{(\alpha,r \times \xi_{i,j} \times \mathrm{Pr}(k,\boldsymbol{\xi}))}_k S_{i'}(\ell_{j'}) \parallel S_{m'}(\ell_{n'})[k]}$

BrCombo $\dfrac{P \parallel Q_1[k_1] \xrightarrow{(\alpha,r(\boldsymbol{\xi}) \times \mathrm{Pr}(k_1,\boldsymbol{\xi}))}_{k_1} P' \parallel Q_1'[k_1] \;\cdots\; P \parallel Q_n[k_n] \xrightarrow{(\alpha,r(\boldsymbol{\xi}) \times \mathrm{Pr}(k_n,\boldsymbol{\xi}))}_{k_n} P' \parallel Q_n'[k_n]}{P \parallel Q_1[k_1] \parallel \ldots \parallel Q_n[k_n] \xrightarrow{(\alpha,r(\boldsymbol{\xi}) \times \prod_{z=1,\ldots,n} \mathrm{Pr}(k_z,\boldsymbol{\xi}))} P' \parallel Q_1'[k_1] \parallel \ldots \parallel Q_n'[k_n]}$

UniPair $\dfrac{!!(\alpha,r)@\mathbf{IR}\{\overrightarrow{\ell}\}.S_{i'}(\ell_{j'}) \in \mathrm{pf}(S_i(\ell_j)) \quad ??(\alpha,p)@\mathbf{Wt}\{v\}.S_{m'}(\ell_{n'}) \in \mathrm{pf}(S_m(\ell_n)) \quad \ell_n \in \overrightarrow{\ell}}{S_i(\ell_j) \parallel S_m(\ell_n) \xrightarrow{(\alpha,r \times \xi_{i,j} \times p \times w_v(\boldsymbol{\xi}) \times \xi_{m,n})} S_{i'}(\ell_{j'}) \parallel S_{m'}(\ell_{n'})}$

Choice $\dfrac{E \xrightarrow{(\alpha,r(\boldsymbol{\xi}))} E'}{E + F \xrightarrow{(\alpha,r(\boldsymbol{\xi}))} F'}$ $\qquad\qquad$ $\dfrac{F \xrightarrow{(\alpha,r(\boldsymbol{\xi}))} F'}{E + F \xrightarrow{(\alpha,r(\boldsymbol{\xi}))} F'}$

Parallel $\dfrac{E \xrightarrow{(\alpha,r(\boldsymbol{\xi}))} E'}{E \parallel F \xrightarrow{(\alpha,r(\boldsymbol{\xi}))} E' \parallel F}$ $\qquad\qquad$ $\dfrac{F \xrightarrow{(\alpha,r(\boldsymbol{\xi}))} F'}{E \parallel F \xrightarrow{(\alpha,r(\boldsymbol{\xi}))} E \parallel F'}$

Constant $\dfrac{P \xrightarrow{(\alpha,r(\boldsymbol{\xi}))} P'}{C \xrightarrow{(\alpha,r(\boldsymbol{\xi}))} P'} C \triangleq P$ $\qquad\qquad$ Congruence $\dfrac{P \xrightarrow{(\alpha,r(\boldsymbol{\xi}))} P'}{Q \xrightarrow{(\alpha,r(\boldsymbol{\xi}))} P'} P \equiv Q$

Fig. 2. The Population-level Structured Operational Semantics of PALOMA (all rules for arbitrary α; $r \in \mathbb{R}^+$, $p \in [0,1]$, $v(\boldsymbol{\xi}) \in [0,1]$, $w_v(\boldsymbol{\xi}) \in [0,1]$.)

as E or F. Parallel components proceed independently as shown in the rule Parallel. Note that the rate of any population-level transition can be thought as a function of the numerical population matrix $\boldsymbol{\xi}$.

Constants are agents whose meaning is given by a defining equation such as $C \triangleq P$ which gives the constant C the behaviour of the agent P.

2.1.1. The PCTMC Model. With the population-level structured operational semantics, any PALOMA model can be treated as a population continuous time Markov chain (PCTMC) represented as a tuple $\mathcal{P} = (\mathbf{X}, \mathcal{T}, \mathbf{x_0})$, where:

— $\mathbf{X} = (x_1, ..., x_n)$ maps the population numerical matrix $\boldsymbol{\xi}$ to a vector format, where each vector element is the count variable of agents in a specific state and location (we assume that there is an implicit function f such that $f(\xi_{i,j}) = x_k$ if $\xi_{i,j} \neq -1$). We will use $\mathbf{X}$ as short for $\mathbf{X}(t)$ hereafter to represent the current state of the model in terms of the population level at a time instant.
— $\mathcal{T} = \{\tau_1, ..., \tau_m\}$ is the set of enabled population-level transitions, of the form $\tau_k = (r_k(\mathbf{X}), \mathbf{d}_k)$, where:
  (1) $r_k(\mathbf{X}) \in \mathbb{R} \geq 0$ is the rate function of transition τ_k depending on the current population-level state of the system. Note that $r_k(\mathbf{X})$ is not restricted to polynomial functions.
  (2) $\mathbf{d}_k \in \mathbb{Z}^n$ is the update vector in which each element is a constant which gives the net change for a specific element of $\mathbf{X}$ caused by transition τ_k.
— $\mathbf{x_0}$ is the initial state of the model.

The tuple $\mathcal{P}$ contains all the information that is needed for the population-level discrete event simulation of a PALOMA model using the standard stochastic simulation algorithm [Gillespie 1977].

3. MOMENT ANALYSIS OF PALOMA MODELS

As collective adaptive systems usually consist of a few hundreds, or perhaps many more entities, discrete event simulation of the associated PCTMC models can be computationally expensive even with the use of a counting abstraction and simulation at the population level. Moreover, deriving performance measures from stochastic simulation often requires us to simulate the model a large number of times, and then obtain the measures of interest such as mean, variance and covariance from the trajectories of those simulation runs. This means that analysing large-scale collective adaptive systems may become extremely inefficient. Thus, in this section, we show how to automatically derive an ODE approximation for a PCTMC model expressed in PALOMA. Specifically, we will describe the evolution of the first moment, second moment and second-order joint moment of population variables in an arbitrary PALOMA model by a set of coupled ODEs. As these ODEs contain additional moment variables (moment variables higher than the second order), we apply moment-closure techniques to close the moment ODEs at the second order. Obtaining performance measures by solving these moment ODEs using numerical techniques is computationally much more efficient than stochastic simulation since we only need to solve the ODEs once to get our performance measures and the associated cost is independent of the number of agents in the model.

3.1. The Derivation of Moment ODEs

We have mentioned that the population-level semantics can serve as an intermediate tool for the generation of the moment ODEs. Specifically, the evolution of the moments of the underlying population-level stochastic process of an arbitrary PCTMC model can be approximated by the following system of ODEs [Engblom 2006]:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[M(\mathbf{X}(t))] = \sum_{\tau \in \mathcal{T}} \mathbb{E}[(M(\mathbf{X}(t) + \mathbf{d}_\tau) - M(\mathbf{X}(t)))r_\tau(\mathbf{X}(t))]$$

where $M(\mathbf{X})$ denotes the moment to be calculated, $\mathbf{d}_\tau$ and $r_\tau(\mathbf{X}(t))$ represent the update vector and rate of a population-level transition τ, respectively.

By substituting $M(\mathbf{X})$ with x_i, $x_i{}^2$ and $x_i x_j$, we get the following ODEs to describe the first moment, second moment and second-order joint moment respectively, of population variables in an arbitrary PALOMA model (we use the same notations as in Section 2.1.1):

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[x_i] = \sum_{k=1}^{m} \mathbb{E}[(x_i + d_{ki} - x_i)r_k] = \sum_{k=1}^{m} \mathbb{E}[d_{ki} \times r_k]$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[x_i{}^2] = \sum_{k=1}^{m} \mathbb{E}[((x_i + d_{ki})^2 - x_i{}^2)r_k] = \sum_{k=1}^{m} \mathbb{E}[2d_{ki} \times x_i \times r_k + d_{ki}{}^2 \times r_k]$$

$$= 2\sum_{k=1}^{m} \mathbb{E}[d_{ki} \times x_i \times r_k] + \sum_{k=1}^{m} \mathbb{E}[d_{ki}{}^2 \times r_k]$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[x_ix_j] = \sum_{k=1}^{m}\mathbb{E}[((x_i+d_{ki})(x_j+d_{kj})-x_ix_j)r_k]$$

$$= \sum_{k=1}^{m}\mathbb{E}[d_{ki}\times x_j\times r_k+d_{kj}\times x_i\times r_k+d_{ki}\times d_{kj}\times r_k]$$

$$= \sum_{k=1}^{m}\mathbb{E}[d_{ki}\times x_j\times r_k]+\sum_{k=1}^{m}\mathbb{E}[d_{kj}\times x_i\times r_k]+\sum_{k=1}^{m}\mathbb{E}[d_{ki}\times d_{kj}\times r_k]$$

where d_{ki} is the ith element in $\mathbf{d}_k$ representing the update on the population variable x_i caused by transition τ_k. Furthermore, we denote any moment variable in the above equations as $\mathbb{E}[f(\mathbf{X})]$ because both d_{ki} and r_k can be thought of as a function of the population vector $\mathbf{X}$.

In order to be able to solve the above moment ODEs, we face three problems. First of all, consider a vector of population-level transitions inferred from the rule BrPair, we can write the transitions in the PCTMC as follows ($0 \le k \le x_m$):

$$x_i,x_j,x_m,x_n \xrightarrow{r\times x_i\times\binom{x_m}{0}(p\times v(\mathbf{X}))^0(1-p\times v(\mathbf{X}))^{x_m}}_{0} \quad x_i-1,x_j+1,x_m-0,x_n+0$$

$$\vdots$$

$$x_i,x_j,x_m,x_n \xrightarrow{r\times x_i\times\binom{x_m}{k}(p\times v(\mathbf{X}))^k(1-p\times v(\mathbf{X}))^{x_m-k}}_{k} \quad x_i-1,x_j+1,x_m-k,x_n+k$$

$$\vdots$$

$$x_i,x_j,x_m,x_n \xrightarrow{r\times x_i\times\binom{x_m}{x_m}(p\times v(\mathbf{X}))^{x_m}(1-p\times v(\mathbf{X}))^0}_{x_m} \quad x_i-1,x_j+1,x_m-x_m,x_n+x_m$$

Clearly, there will be no meaning for the *Choose* function $\binom{x_m}{k}$ in the rate function of these transitions, given that we treat the populations of agents as continuous variables in moment analysis. Thus, in order to deal with this problem, we combine the above vector of transitions for each possible update into a single probabilistic population transition:

$$x_i,x_j,x_m,x_n \xrightarrow{r\times x_i} x_i-1,x_j+1,x_m-p\times v(\mathbf{X})\times x_m,x_n+p\times v(\mathbf{X})\times x_m$$

since:

$$p\times v(\mathbf{X})\times x_m = \sum_{k=0,\ldots,x_m} k\times\binom{x_m}{k}(p\times v(\mathbf{X}))^k(1-p\times v(\mathbf{X}))^{x_m-k}$$

Intuitively, this means that we get a combined population-level transition for a vector of population-level transitions inferred from BrPair where the rate of the combined transition is the rate of the spontaneous action with broadcast message emission, the update vector of the combined transition is the expected number of agents to actually change their state. Furthermore, we can write a combined population-level transition inferred from BrCombo in the PCTMC as follows:

$$x_i,x_j,\ldots,x_{m_k},x_{n_k},\ldots \xrightarrow{r\times x_i} x_i-1,x_j+1,\ldots,x_{m_k}-p_k\times v_k(\mathbf{X})\times x_{m_k},x_{n_k}+p_k\times v_k(\mathbf{X})\times x_{m_k},\ldots$$

where $r\times x_i$ is also the rate of the broadcast message emission, $p_k\times v_k(\mathbf{X})\times x_{m_k}$ is the expected number of agents in a state and location with associated transitions induced by the message to actually change their state. With the combined transitions, we are able to circumvent the intractable *Choose* function.

A second point to note is that, the above ODEs are not closed as $\mathbb{E}[f(\mathbf{X})]$ may contain moment variables with arbitrary orders.

Lastly, when there are many population variables in the model, a problem we call *ODE explosion* (so many coupled ODEs that traditional machines do not have enough memory and computational power to numerically simulate them) may also emerge.

Before we show how to close the moment ODEs, in the next subsection, we first introduce a neighbourhood relation between population variables which can be used to significantly mitigate the *ODE explosion* problem.

3.2. Neighbourhood Relation

Describing the evolution of expected population-level dynamics by moment ODEs can be dramatically improved our ability to analyse large scale collective adaptive systems. However, *ODE explosion* may significantly reduce the scalability of the moment analysis. Specifically, consider there are n elements in the population vector $\mathbf{X}$. Then there will be n ODEs to describe the evolution of all $\mathbb{E}[x_i]$, n ODEs for all $\mathbb{E}[x_i^2]$, and $(n^2 - n)/2$ ODEs for all $\mathbb{E}[x_i x_j]$. Obviously, the problem of ODE explosion is mostly caused by the number of ODEs for $\mathbb{E}[x_i x_j]$. Fortunately, the number of moment ODEs for $\mathbb{E}[x_i x_j]$ can be significantly reduced if we can approximate $\mathbb{E}[x_i x_j] \approx \mathbb{E}[x_i]\mathbb{E}[x_j]$ when the degree of correlation between x_i and x_j is less than a specific threshold.

Concretely, we estimate the degree of correlation between two population variables x_i and x_j by their neighbourhood relation. We say x_i, x_j are one-hop neighbours if one of them can directly influence the evolution of the other. Specifically, we define:

$$(x_i, x_j) \in \mathcal{R}^{(1)} \iff \exists\, \tau_k, \quad (d_{ki} \neq 0 \wedge \delta_k^j = 1) \vee (d_{kj} \neq 0 \wedge \delta_k^i = 1)$$

where δ_k^j is an indicator which equals to 1 only if x_j is updated after transition τ_k ($d_{kj} \neq 0$) or x_j appears in the rate function (r_k) of τ_k. Intuitively, this means there exists a transition, in which one of the two population variables is updated, and the other is also involved. Moreover, we can infer two-hop neighbours by:

$$\exists\, k \notin \{i, j\} \quad (x_i, x_j) \notin \mathcal{R}^{(1)} \wedge (x_i, x_k) \in \mathcal{R}^{(1)} \wedge (x_k, x_j) \in \mathcal{R}^{(1)} \Rightarrow (x_i, x_j) \in \mathcal{R}^{(2)}.$$

More generally, $\mathcal{R}^{(n)}$ is the smallest relation that satisfies

$$\exists\, k \notin \{i, j\} \quad (x_i, x_j) \notin \mathcal{R}^{(1)}, \ldots, \mathcal{R}^{(n)} \wedge (x_i, x_k) \in \mathcal{R}^{(1)} \wedge (x_k, x_j) \in \mathcal{R}^{(n)} \Rightarrow (x_i, x_j) \in \mathcal{R}^{(n+1)}.$$

After calculating the neighbourhood relation between two arbitrary population variables, we can deal with moment ODE explosion by approximating

$$\mathbb{E}[x_i x_j] \approx \mathbb{E}[x_i]\mathbb{E}[x_j] \quad \text{if } (x_i, x_j) \in \mathcal{R}^{(d')} \wedge d' > d$$

where $d \geq 0$ is a threshold chosen by the modeller. Note when $d = 0$, it means we assume all the population variables are independent; when $d = \infty$, it means we keep all the correlation between population variables, and no reduction is actually applied. With smaller values of d, we can use fewer moment ODEs to describe the first and second order dynamics of a PALOMA model at the cost of losing some accuracy, and *the optimal value of d can be chosen whenever increasing the value of d will not make any observable difference in the results*. In our case studies, we will show this reduction method can substantially reduce the number of moment ODEs, but still produces very good results.

3.3. Moment-closure Method

After the reduction of moment ODEs, we show how to close the moment ODEs (note that we can use the following moment-closure method regardless of whether we are working with the full set or a reduced set of moment ODEs).

First of all, when the derived moment ODEs only contain moment variables that are no higher than the third order (these are cases when the probability of receiving

a broadcast message and the weight of receiving a unicast message are constants), we can apply the lognormal moment closure [Singh and Hespanha 2006a] which assumes that the dependence of a higher-order moment on lower order ones is consistent with the population being joint lognormally distributed, to close the moment ODEs at the second-order. The lognormal moment closure provides a pure multiplicative way of closing higher-order moments. This can provide more robust analysis than traditional additive moments, such as normal moment-closure [Whittle 1957], which inevitably also assign probability for negative values that are not possible in PALOMA models [Keeling 2000; Singh and Hespanha 2006b]. Concretely, suppose we want to approximate $\mathbb{E}[\mathbf{x}^{(\mathbf{m})}]$ ($\mathbf{x}$ and $\mathbf{m}$ are both vectors with size n) where

$$\mathbf{x}^{(\mathbf{m})} := x_1^{(m_1)} x_2^{(m_2)} \cdots x_n^{(m_n)}$$

using only moments of order up to $o(\mathbf{m}) - 1$, where $o(\mathbf{m}) = \sum_1^n m_i$. Let $\mathbb{M} = \{\mathbf{m_1}, \ldots, \mathbf{m_k}\}$ be the moment order set containing all the moments up to the order $o(\mathbf{m}) - 1$, then we can approximate $\mathbb{E}[\mathbf{x}^{(\mathbf{m})}] \approx \prod_{p=1}^k (\mathbb{E}[\mathbf{x}^{\mathbf{m_p}}])^{\gamma_p}$, where $\gamma = (\gamma_1, \ldots, \gamma_k)$ is the unique solution to the following system of linear equations

$$\mathbf{C}_{(\mathbf{m}_s)}^{(\mathbf{m})} = \sum_{p=1}^k \gamma_p \mathbf{C}_{(\mathbf{m}_s)}^{(\mathbf{m}_p)} \quad \forall s = \{1, \ldots, k\} \quad \text{where} \quad \mathbf{C}_{(\check{\mathbf{m}})}^{(\hat{\mathbf{m}})} = C_{(\check{m}_1)}^{(\hat{m}_1)} C_{(\check{m}_2)}^{(\hat{m}_2)} \ldots C_{(\check{m}_n)}^{(\hat{m}_n)}$$

where $\hat{m}$ and $\check{m}$ are vectors with size n, C_h^l is defined as: $C_h^l = \begin{cases} \frac{l!}{(l-h)!h!} & \text{if } l \geq h \\ 0 & \text{if } l < h \end{cases}$.

Specifically, using the lognormal closure technique, we obtain:

$$\mathbb{E}[x_i x_j^2] = \frac{\mathbb{E}[x_j^2]\mathbb{E}[x_i x_j]^2}{\mathbb{E}[x_i]\mathbb{E}[x_j]^2} \quad \mathbb{E}[x_i x_j x_k] = \frac{\mathbb{E}[x_i x_j]\mathbb{E}[x_i x_k]\mathbb{E}[x_j x_k]}{\mathbb{E}[x_i]\mathbb{E}[x_j]\mathbb{E}[x_k]}$$

By substituting the above equations into the moment ODEs, we get a closed system of ODEs for the first, second and joint moment of population variables which can be numerically solved.

However, when the derived moment ODEs also contain moment variables that are higher than the third order, we use a heuristic algorithm to reduce the order of moment variables to the third order. We utilise the neighbourhood relation defined in the previous subsection to estimate the degree of correlation of a population variable with a list of other population variables. Specifically, we let

$$(x_i, [x_{j_1}, \ldots, x_{j_n}]) \in \mathcal{R}^{(N)} \iff (x_i, x_{j_k}) \in \mathcal{R}^{(d_k)}, \quad 1 \leq k \leq n \wedge \sum_{k=1}^n d_k = N.$$

The above is the basis for reducing the order of moment variables in $\mathbb{E}[f(\mathbf{X})]$. Concretely, we let $\mathbb{E}[f(\mathbf{X})] \approx \frac{\mathbb{E}[f_1(\mathbf{X})]}{\mathbb{E}[f_2(\mathbf{X})]}$ if $f(\mathbf{X}) = \frac{f_1(\mathbf{X})}{f_2(\mathbf{X})}$. When $\mathbb{E}[f_i(\mathbf{X})], i \in \{1, 2\}$ contains moment variables that are higher than third-order, we reduce the order of its moment variables also, by approximating $\mathbb{E}[f_i(\mathbf{X})] \approx \mathbb{E}[x_k]\mathbb{E}[g(\mathbf{X})]$, if $f_i(\mathbf{X}) = x_k g(\mathbf{X})$, and x_k is the least correlated population variable in $f_i(\mathbf{X})$. We apply the above algorithm until the maximum order of moment variables is three. After that, the lognormal moment-closure can be applied.

Figure 3 illustrates the procedure for the moment analysis of PALOMA models. By following the procedure, an arbitrary PALOMA model can be analysed by solving a set of ODEs. The accuracy and efficiency of our moment analysis will be demonstrated by examples in the next section.
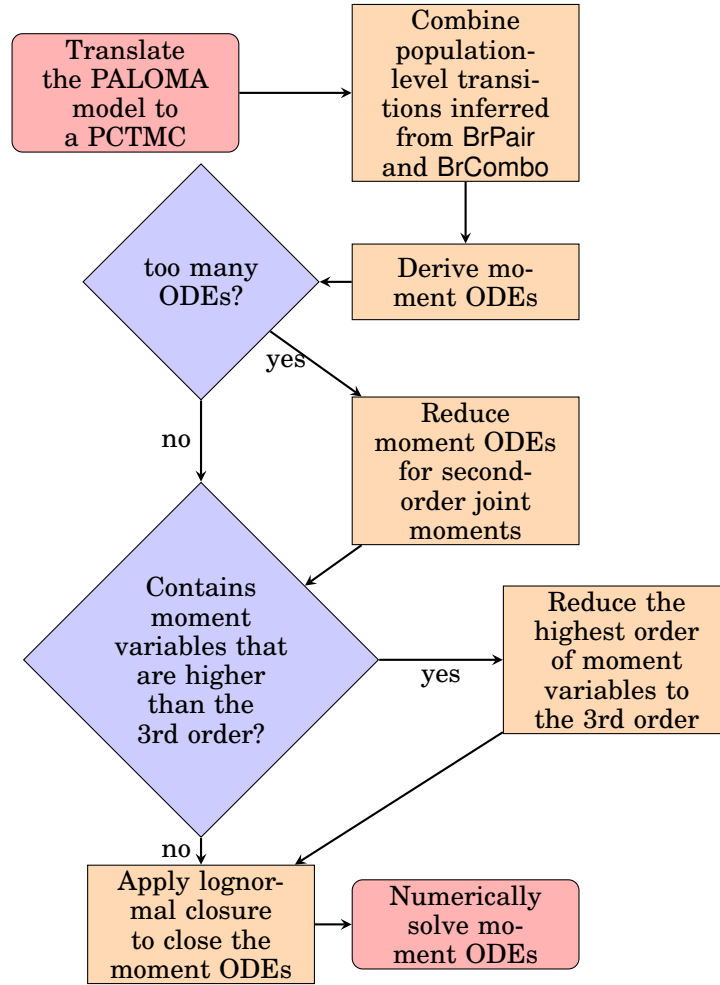
Fig. 3. The procedure for the moment analysis of PALOMA models

4. TOOL AND CASE STUDIES

We have developed a tool for parsing and analysing PALOMA models. The tool is developed as an Eclipse plug-in which is available on https://github.com/cfeng783/paloma/wiki#the-paloma-eclipse-plug-in. It can support both population-level stochastic simulation and automatic moment ODE script generation of PALOMA models written in the syntax introduced earlier in this paper. The generated moment ODE script is directly executable in Matlab without any further modification.

In this section, we show three PALOMA models capturing three collective adaptive systems in different areas to show both the expressiveness of the language and the correctness and efficiency of the various techniques we use in the moment analysis.

4.1. An Epidemiological SIS Model

We first consider a classical epidemiological SIS model of individuals partitioned into communities, where individuals move between communities but infections only take

place within communities. Each individual is considered to be susceptible (S) or infected (I) with respect to the disease. More specifically, consider a total fixed population of N individuals partitioned into m communities in a ring topology, each of which contains n individuals ($N = n \times m$). A continuous-time SIS epidemiological model is then applied to this population as follows: each individual, regardless of susceptible or infected, can move to his/her neighbour communities with rate r. Each infected individual contacts and attempts to infect others in the same community at rate λ. Each contact is with a randomly chosen individual. When an infected individual contacts a susceptible individual, the latter becomes infected as well. Finally, infected individuals independently recover to the susceptible state at rate μ.

The individuals in susceptible and infected states can be represented in PALOMA by the following agents:

$$S(\ell_i) = ??(contact, 1)@\mathbf{Wt}\{1\}.I(\ell_i) + \sum_{j \in \text{nearby}(i)} (move_{ij}, r).S(\ell_j)$$

$$I(\ell_i) = !!(contact, \lambda)@\mathbf{IR}\{local\}.I(\ell_i) + ??(contact, 1)@\mathbf{Wt}\{1\}.I(\ell_i)$$
$$+ (recover, \mu).S(\ell_i) + \sum_{j \in \text{nearby}(i)} (move_{ij}, r).I(\ell_j)$$

where $S(\ell_i)$ and $I(\ell_i)$ denote an individual in the susceptible and infected state currently in community i respectively, $\text{nearby}(i) = \{(i+1) \mod m, (i-1+m) \mod m\}$ is the index of nearby communities of community i.

The initial population of agents are given in the following definition:

$$S(\ell_1)[n - I_1] \parallel I(\ell_1)[I_1] \parallel \ldots \parallel S(\ell_i)[n - I_i] \parallel I(\ell_i)[I_i] \parallel \ldots \parallel S(\ell_m)[n - I_m] \parallel I(\ell_i)[I_m]$$

where I_i denotes the number of initial infected individuals in community i.

In the simulation, we randomly choose 5 out of 50 communities as the source of the epidemic. There are 5 individuals in the 5 chosen source communities who are infected initially. All the other individuals in the model are in the susceptible state initially. Table I gives the simulation configuration of the SIS model. The values of the parameters in the model are chosen to make the model close to realistic scenarios. The number of simulation runs is chosen to make the first moment and the second moment observable. The stop time of a simulation run is chosen to let the first moment and the second moment converge. The same standard is also applied in the next two examples.

Table I. SIS Model Simulation Configuration

m	50
n	50
r	1
λ	2
μ	1
I_i (ℓ_i is a source community)	5
I_i (ℓ_i is not a source community)	0
Stop time of a simulation run	20
Number of simulation runs	10,000

The analysis of interest in the SIS model is the number of infected individuals over all the communities. We apply moment-closure analysis on the SIS model with different reduction thresholds based on the neighbourhood relation of population variables, and then compare the results with the stochastic simulation. Figure 4 shows the trajectories of the first and second moments of the infectious population in the SIS model
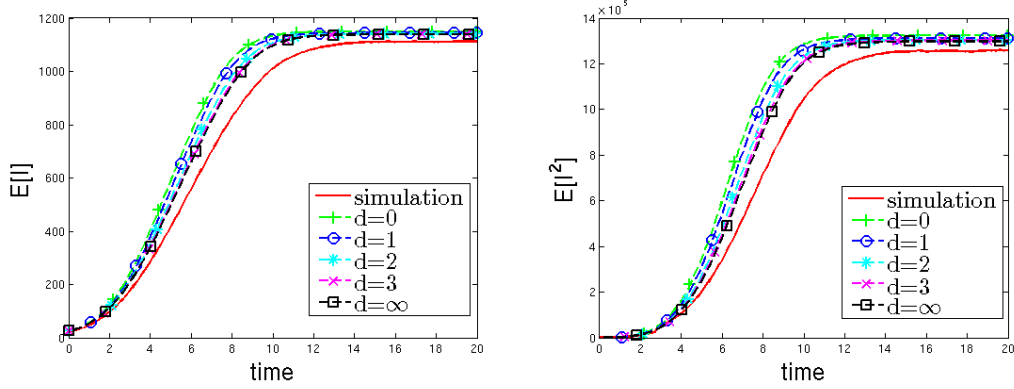
Fig. 4. The first moment and the second moment of infected population

in the 10,000 runs of stochastic simulation as well as moment analysis. It can be seen that with a stricter reduction standard (the larger value of d), the result of moment analysis is closer to the stochastic simulation. Nevertheless, the result of moment analysis when $d = 3$ is almost the same as that with $d = \infty$, which means that the second-order joint moments between population variables when their neighbourhood relation is larger than three hops gives no extra information than their first moments. Therefore, we can ignore the correlation between those population variables without loss of accuracy but with the gain of reduced solution time for the moment ODEs. More evidence is shown in Table II, where the error ratio is calculated by averaging the difference between the stochastic simulation and the moment analysis over 200 data points evenly selected in the trajectories along the simulation time. We can see that the number of ODEs and solution time for moment analysis can be significantly reduced by our reduction method with only limited loss of accuracy compared with the full moment analysis ($d = \infty$) as long as the optimal value of d is chosen.

Table II. Simulation V.S. moment analysis of the SIS model

SIS model	ODE number	Solution time	Error ratio	
			1st moment	2nd moment
Stochastic simulation (10,000 runs)	N/A	10.39 hrs	N/A	N/A
Moment analysis with $d = 0$	200	0.31 secs	14.01%	22.78%
Moment analysis with $d = 1$	350	0.83 secs	11.27%	18.37%
Moment analysis with $d = 2$	550	1.1 secs	8.61%	14.16%
Moment analysis with $d = 3$	750	1.34 secs	7.42%	12.3%
Moment analysis with $d = \infty$	5150	31.99 secs	7.16%	11.34%

Furthermore, it is clear that moment analysis can enormously reduce the computational cost of analysing a PALOMA model compared with stochastic simulation.

4.2. A wireless sensor network model

Here, we discuss a spatial model that represents the spread of pheromone in a multi-hop Wireless Sensor Network (WSN). In nature, pheromone is a hormone laid down by colony-based insects, to indicate popular routes to food sources or new nest sites. In a similar manner pheromone gradients have been adapted in the WSN literature as an abstract means of studying the evolution of routes from source to sink nodes. Several models have been built to investigate the spread of pheromone in such networks

[Bruneo et al. 2012; Guenther et al. 2013]. We show how to capture those models in PALOMA. Figure 5 visualises the topology of the WSN model, where there is a sink node in cell 13 and there is a sensor deployed in each cell.
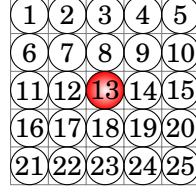

Fig. 5.   The topology of the WSN model

Assuming we are only interested in the spread of pheromone, then the sink node which is also the source of the pheromone spread broadcasts a message containing the maximum pheromone level to all sensor nodes in the network at rate λ_{sink}. Thus, the sink node is represented as follows:

$$Sink(\ell_{13}) \ = \ !(ph_{\max}, \lambda_{sink})@\mathbf{IR}\{all\}.Sink(\ell_{13})$$

The pheromone level in a sensor node is denoted by an integer in the range 0 to max. A sensor node can update its pheromone level to max once it receives a broadcast message from the sink node. However, the probability of a message from the sink node being received by a sensor node depends on the physical distance between the sink node and sensor node. The sensor nodes can also exchange pheromone information with their neighbourhood sensor nodes using a Manhattan style communication pattern. The pheromone level in each sensor node is assumed to decrease exponentially at rate μ. Moreover, sensor nodes can also enter an off state at rate r_{off}. Sensor nodes in the off state can do nothing but sleep for a while and return to the on state with rate r_{on}. When a sensor node re-enters the on state, its pheromone level is set to 0.

Thus, the sensor nodes can be represented as follows:

$$
\begin{aligned}
Sensor_{ph_k}(\ell_i) \ = \ & \sum_{k<j\leq\max} ?(ph_j,1)@\mathbf{Pr}\{\frac{1}{1+dist(\ell_s,\ell_i)}\}.Sensor_{ph_j}(\ell_i) \\
& + \ !(ph_k, \lambda_{sensor})@\mathbf{IR}\{range(1)\}.Sensor_{ph_k}(\ell_i) \\
& + \ (evaporate, \mu).Sensor_{ph_{k-1}}(\ell_i) \quad (k > 1) \\
& + \ (\mathrm{off}, r_{\mathrm{off}}).Sensor_{\mathrm{off}}(\ell_i) \\
Sensor_{\mathrm{off}}(\ell_i) \ = \ & (\mathrm{on}, r_{\mathrm{on}}).Sensor_{ph_0}(\ell_i)
\end{aligned}
$$

where $Sensor_{ph_k}(\ell_i)$ denotes a sensor node in cell i currently with pheromone level k. The sensor node can receive a message containing a higher level pheromone and then update its pheromone level. $dist(\ell_s, \ell_i)$ is the distance between ℓ_i and the location of the message sender (can be either a sink node or a sensor node). It can also broadcast a message containing its current pheromone level to its neighbourhood at rate λ_{sensor}. The other two actions capture the evaporation of pheromone and the sleep of nodes.

Table III summarises the simulation configuration of the model, where we set the maximum pheromone level to 5. The analysis of interest is the spread of pheromone: the number of sensor nodes with different pheromone levels and the expected pheromone level in each node. Figures 6 and 7 show the trajectories of the first moment and second moment of the number of sensor nodes with different pheromone levels. Figure 8 shows the expected pheromone level of the sensor nodes in each cell at time 100 which is the stop time of a simulation run. In both cases, we can see that

moment analysis with $d = 1$ gives much closer results to stochastic simulation than moment analysis with $d = 0$. Table IV compares the moment analysis with different reduction thresholds with stochastic simulation. In this case, moment analysis cannot be applied without our reduction method since the number of ODEs is too large for Matlab to solve when $d > 1$. Moreover, again we can see that moment analysis with $d = 1$ gives much better results than moment analysis with $d = 0$. This is because fluid limit analysis neglects all the correlations between population variables and can only give a good result at the first moment when the population is large [Tribastone et al. 2012]. However, in this case, the population is too small for this kind of analysis.

Table III. WSN Model Simulation Configuration

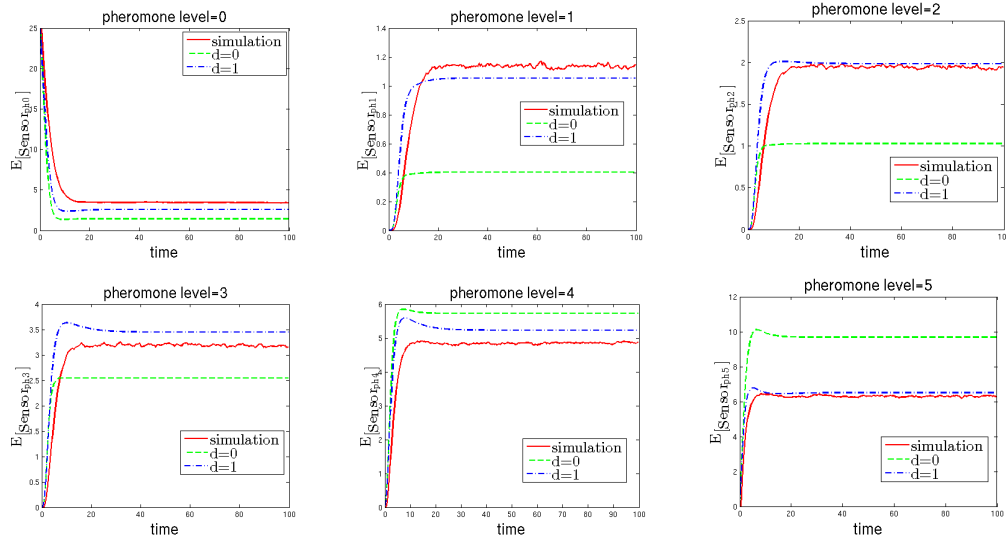| max	5
λ_{sink}	0.35
λ_{sensor}	0.35
μ	0.5
r_{off}	0.05
r_{on}	0.25
Stop time of a simulation run	100
Number of simulation runs	10,000

Fig. 6.   The first moment of number of sensor nodes with different pheromone level

Table IV. Simulation V.S. moment analysis of the sensor network model

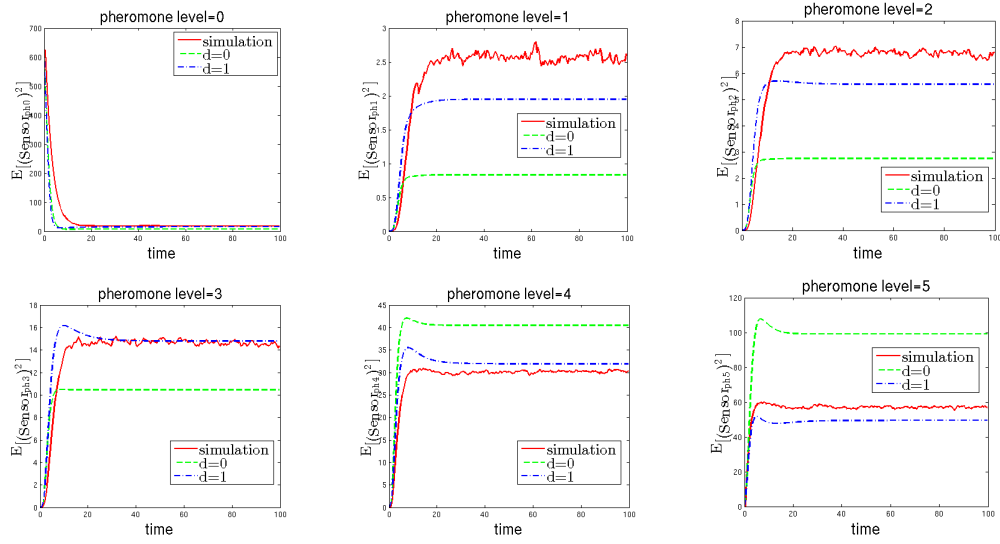| sensor network model	ODE number	Solution time	Error ratio	
			1st moment	2nd moment
Stochastic simulation (10,000 runs)	N/A	15.55 mins	N/A	N/A
Moment analysis with $d = 0$	352	0.21 secs	33.34%	45.42%
Moment analysis with $d = 1$	2427	25.94 secs	8.72%	12.97%
Moment analysis with $d = 2$	12332	out of memory	N/A	N/A

Fig. 7. The second moment of number of sensor nodes with different pheromone level
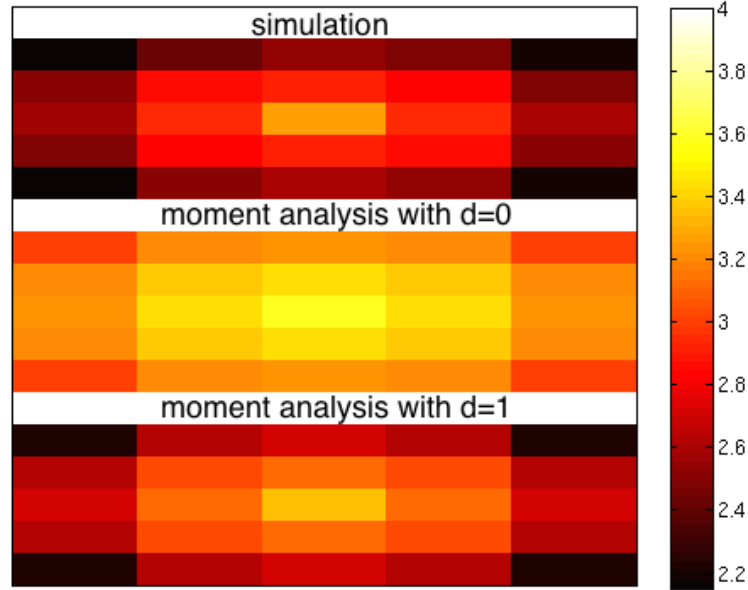

Fig. 8. The expected pheromone level in each cell at time 100.

4.3. A City Bike-sharing Model

The last example we discuss is a PALOMA model for a city bike-sharing system. Suppose that the city is divided into 16 zones in a 4×4 grid topology, and each zone has a bike station with a number of available bikes and slots. Therefore, we represent the

available bikes and slots in Station i by agents as follows:

$$Slot(\ell_i) =??(return,1)@\mathbf{Wt}\{1\}.Bike(\ell_i) \quad Bike(\ell_i) =??(borrow,1)@\mathbf{Wt}\{1\}.Slot(\ell_i)$$

Both $Slot(\ell_i)$ and $Bike(\ell_i)$ are passive. They can only be induced to make a $return$ (a bike is returned to this station) or $borrow$ (a bike is borrowed from this station) action by a unicast message, and when this happens they switch role.

The agents to represent the bike stations are defined as:

$$Station(\ell_i) = !(SlotAvailable_i,\gamma)@\mathbf{IR}\{range(d)\}.Station(\ell_i) +$$
$$!(BikeAvailable_i,\gamma)@\mathbf{IR}\{range(d)\}.Station(\ell_i)$$

A bike station performs both a $BikeAvailable_i$ and a $SlotAvailable_i$ self-jump spontaneous actions with broadcast message emission at the rate of γ. The influence range of the broadcast messages is defined by the function $range(d)$, which means that only agents in locations whose distance to the location of the sender station is less than d can potentially be influenced by this message.

The agents representing bike users are given as follows:

$$Pedestrian(\ell_i) = (seekb_i,b_i).SeekBike(\ell_i) + \sum_{j\neq i}(walk_{ij},w_{ij}).Pedestrian(\ell_j)$$

$$SeekBike(\ell_i) = \sum_{j=1}^{m}?(BikeAvailable_j,1)@\mathbf{Pr}\{v_1\}.Walk2Station_j(\ell_i)$$

$$Walk2Station_j(\ell_i) = (W2S_{ij},w2s_{ij}).CheckBikeNum(\ell_j)$$
$$CheckBikeNum(\ell_i) = ?(BikeAvailable_i,1)@\mathbf{Pr}\{v_2\}.BorrowBike(\ell_i)$$
$$BorrowBike(\ell_i) = !!(borrow,o)@\mathbf{IR}\{local\}.Biker(\ell_i)$$

$$Biker(\ell_i) = (seeks_i,s_i).SeekSlot(\ell_i) + \sum_{j\neq i}(ride_{ij},r_{ij}).Biker(\ell_j)$$

$$SeekSlot(\ell_i) = \sum_{j=1}^{m}?(SlotAvailable_j,1)@\mathbf{Pr}\{v_3\}.Ride2Station_j(\ell_i)$$

$$Ride2Station_j(\ell_i) = (R2S_{ij},r2s_{ij}).CheckSlotNum(\ell_j)$$
$$CheckSlotNum(\ell_i) = ?(SlotAvailable_i,1)@\mathbf{Pr}\{v_4\}.ReturnBike(\ell_i)$$
$$ReturnBike(\ell_i) = !!(return,o)@\mathbf{IR}\{local\}.Pedestrian(\ell_i)$$

where

$$v_1 = \theta_0 + \theta_1\frac{d - dist(\ell_i,\ell_j)}{d} + \theta_2\frac{|Bike(\ell_j)|}{|Bike(\ell_j)| + |Slot(\ell_j)|} \tag{1}$$

$$v_2 = \frac{|Bike(\ell_i)|}{|Bike(\ell_i)| + \sigma} \tag{2}$$

$$v_3 = \theta_0 + \theta_1\frac{d - dist(\ell_i,\ell_j)}{d} + \theta_2\frac{|Slot(\ell_j)|}{|Bike(\ell_j)| + |Slot(\ell_j)|} \tag{3}$$

$$v_4 = \frac{|Slot(\ell_i)|}{|Slot(\ell_i)| + \sigma} \tag{4}$$

As can be seen from the definition, when the user agent is in the $Pedestrian$ state, it travels from location ℓ_i to location ℓ_j at the rate of w_{ij} by doing a spontaneous action $walk_{ij}$ without message emission. It may also seek a bike at the rate of b_i, and goes into the $SeekBike$ state.

The user agent in the $SeekBike(\ell_i)$ state can do a $BikeAvailable_j$ action induced by a broadcast message sent by a station agent in location ℓ_j and goes to the $Walk2Station_j(\ell_i)$ state, which represents the user walking from location ℓ_i to the bike station in location ℓ_j. The probability of receiving a bike available message from the

Table V. Bike-sharing Model Simulation Configuration

d	1
θ_0	0
θ_1	0.5
θ_2	0.5
$Pedestrian_i$	25
$Biker_i$	5
$Slot_i$	5
$Bike_i$	10
Stop time of a simulation run	150
Number of simulation runs	10,000

station in location ℓ_j is defined in Equation (1). It can be interpreted as follows: the users tend to borrow a bike from a closer bike station with more available bikes, and θ_1, θ_2 are associated weights of those factors, θ_0 is the noise term (imagine that the user checks the bike numbers in nearby stations using a smart phone application). The user in the $Walk2Station_j(\ell_i)$ state can do a spontaneous action $W2S_{ij}$ at the rate of $w2s_{ij}$, where $1/w2s_{ij}$ is the expected time to walk from ℓ_j to the bike station in ℓ_i.

The user in the $CheckBikeNum(\ell_i)$ state can only do a $BikeAvailable_i$ action induced by a broadcast message sent by the station in ℓ_i. The probability of receiving the message is defined in Equation (2), where σ is a very small real number to avoid a zero denominator. This ensures that the user can only go to the $BorrowBike(\ell_i)$ if the bike station is not empty. The borrow bike action $borrow$ is fired at the rate of o. Meanwhile, a unicast message $borrow$ is sent out, and the user becomes a $Biker$.

A user agent in the $Biker$ state can perform actions and become a $Pedestrian$ again in a similar fashion.

Finally, the initial population of agents are given in the following definition:

$$\ldots \parallel Pedestrian(\ell_i)[Pedestrian_i] \parallel Biker(\ell_i)[Biker_i]$$
$$\parallel Slot(\ell_i)[Slot_i] \parallel Bike(\ell_i)[Bike_i] \parallel Station(\ell_i) \parallel \ldots$$

where $Pedestrian_i$, $Biker_i$, $Slot_i$ and $Bike_i$ are numbers indicating the initial count of pedestrians, bikers, available slots and bikes in location i or station i, respectively.

Table V gives the simulation configuration of the bike-sharing model (only the values of key parameters are listed). The analysis of interest is the number of available bikes in each station over time. Here, we only give the trajectories of the first and second moments of the number of available bikes in the central station in Figure 9. Table VI gives the detailed comparison between moment closure with different reduction thresholds and stochastic simulation whereas the error ratio takes into the account of trajectories of available number of bikes in all the 16 stations. Here, we can observe that moment analysis with $d = 1$ gives almost the same accuracy as moment analysis with any higher thresholds, but with much less solution time needed. Moreover, moment analysis without reduction is again infeasible which shows the usefulness of our method.

Table VI. Simulation V.S. moment analysis of the bike-sharing model

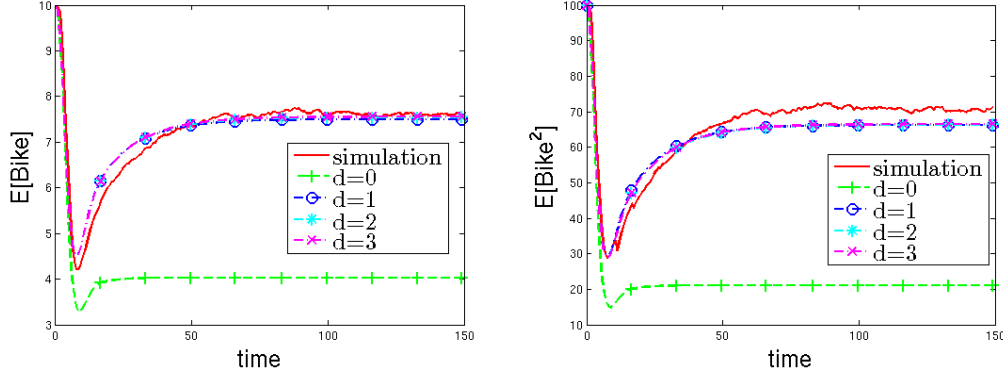| bike-sharing model model	ODE number	Solution time	Error ratio	
			1st moment	2nd moment
Stochastic simulation (10,000 runs)	N/A	19.96 hrs	N/A	N/A
Moment analysis with $d = 0$	608	2.84 secs	42.5%	64.91%
Moment analysis with $d = 1$	1952	35.14 secs	2.74%	6.39%
Moment analysis with $d = 2$	7342	3.36 mins	2.44%	6.16%
Moment analysis with $d = 3$	18934	7.9 mins	2.38%	6.1%
Moment analysis with $d = \infty$	46664	out of memory	N/A	N/A

Fig. 9.   The first moment and the second moment of number of available bikes in the central station

4.3.1. Service Level Agreement Estimation. Finally, we take the bike-sharing example to show how to efficiently conduct service level agreement estimation using the moment analysis of a PALOMA model. Concretely, suppose there is a service level requirement that at any time instant, the probability of a bike station being empty or full should be less than p. Then, we can compute the $1 - p$ prediction interval $[L, U]$ of the number of available bikes in a station at time instant t (which gives the lower and upper limits of an interval that the next observation of the number of available bikes at time t lies in with probability $1 - p$) using the moment ODE model:

$$L = \mathbb{E}[x(t)] - z_{1-p}\sqrt{\mathbb{E}[x(t)^2] - (\mathbb{E}[x(t)])^2} \quad U = \mathbb{E}[x(t)] + z_{1-p}\sqrt{\mathbb{E}[x(t)^2] - (\mathbb{E}[x(t)])^2}$$

where $x(t)$ is the population variable representing the number of available bikes in the station at time t, z_{1-p} is the z-score for $1 - p$ coverage of a normally distributed population (we assume that the available number of bikes in a station at time instant t over many simulation runs follows a normal distribution). Therefore, we can check whether $L \geq 1 \wedge U \leq C - 1$ holds to see if the service level requirement is met, where C is the capacity of the station.

In our experiment, we first compute the 95% prediction interval ($z_{0.95} = 1.96$) for the number of available bikes for the bike-sharing example using our moment ODE model, and then check if the data sampled from our 10,000 discrete event simulation runs agree with it. More specifically, we evenly sample 200 data points from each simulation run for the available number of bikes in each station, and calculate the average percentage of the data points lying in the 95% prediction interval. Our result shows that 95.58% of the data points we sampled from discrete-event simulation are in its 95% prediction interval. Thus, estimating service level agreement of large scale collective adaptive systems by using the moment analysis of a PALOMA model is beneficial because of its low computational cost and has reasonably high accuracy.

5. RELATED WORK

Moment-closure techniques have been studied for many years in different scientific areas. The goal is to achieve a closed form of an infinite set of coupled differential equations by expressing higher-order moments in terms of lower-order moments. Many different closure techniques have been introduced. For example, pair approximation which approximates the density of triplets (third-order moments) by counting certain link densities (second-order moments) that form the triplet, is typically used in spatial epidemic models [Keeling 1999; Hiebeler 2006] and self-organization of adap-

tive networks such as opinion formation [Nardini et al. 2008; Pugliese and Castellano 2009]. Another approach is to use a prior distribution assumption on the population variables of a stochastic process. Typical examples are normal closure [Whittle 1957], beta-binomial closure [Krishnarajah et al. 2005], and lognormal closure [Singh and Hespanha 2006a]. Which assumption is the most valid will depend greatly on the problem being investigated; however, in our context, lognormal closure has the strong advantage of probabilities only being defined for positive values, thus is most suitable for PALOMA. The central moment truncation method follows a similar pattern, which uses the Taylor expansion to approximate moment variables, and then closes the moment equations by assuming the moments of order higher than a certain threshold around the mean to be zero [Engblom 2006; Andreychenko et al. 2015]. However, since the form of moment variables can vary significantly in different PALOMA models, automatically deriving the Taylor expansion of moment variables is nearly infeasible for PALOMA. Unlike normal, beta-binomial, lognormal closure and the central moment truncation method which all assume a particular correlation structure of population variables, the maximum-entropy moment-closure, which is a classic tool in kinetic theory [Singer 2004; Rangan and Cai 2006], makes no assumption on the correlation structure of population variables and chooses the distribution of maximum entropy subject to the constraints based on the knowledge of some lower dimensional marginals. In [Rogers 2011], the maximum-entropy closure has also been applied in spatial epidemic models. Both the pair approximation and maximum-entropy moment-closure have the disadvantage that they require one ODE to capture the density or the population of an agent class in each possible value. This is impossible for PALOMA models, since the possible value of the population of an agent class and the number of agent classes can both be very large, thus the derived ODEs can simply exceed the memory limit.

In most cases, fluid analysis of process algebra models merely focuses on the first-order moment approximation. For example, [Hillston 2005b] showed how to translate a PEPA model to a system of ODEs describing the evolution of the mean population of processes by fluid-flow approximation in an informal way. Later, [Tribastone et al. 2012] formally justified the fluid-flow approximation using Kurtz's fluid limit theorem [Kurtz 1970]. [Cardelli 2008a; 2008b] presented translations from some stochastic process algebras, namely stochastic $\pi-$calculus and stochastic interacting processes, to systems of chemical reactions.

Due to the fact that only investigating the first-order moment cannot provide enough information to reveal the true trace for a stochastic process, recently, there has been some work on higher moment analysis of process algebra models. [Hayden and Bradley 2010] showed how to derive ODEs describing higher moments of PEPA models by letting $\mathbb{E}[\min(P_1(t), P_2(t))] \approx \min(\mathbb{E}[P_1(t)], \mathbb{E}[P_2(t)])$, where $P_1(t)$, $P_2(t)$ are two stochastic processes. [Guenther and Bradley 2011] applied normal moment closure for higher moment analysis of MASSPA models. However, neither PEPA nor MASSPA are expressive enough to model collective adaptive systems; PALOMA is a much more complex modelling formalism, and doing higher moment analysis of PALOMA models is much more difficult in terms of its high nonlinearity and large model size. Among the other suitable process algebras for the modelling of collective adaptive systems are StocS [Latella et al. 2014], a stochastic extension of SCEL [Nicola et al. 2014] and CARMA [Bortolussi et al. 2015]. StocS also supports attribute-based communication among components which is an important feature of collective adaptive systems. However, StocS models can only be analysed by discrete-event simulation, thus the language suffers from the lack of scalable analysis techniques to model large scale collective adaptive systems. CARMA, inspired by PALOMA and SCEL, is a more general and powerful language particularly tailored for the modelling of collective adaptive sys-

tems. Although CARMA is a more expressive language for collective adaptive systems, currently, only its discrete semantics has been introduced. Thus, this paper can be a guide for developing scalable analysis techniques for CARMA.

6. CONCLUSION

In this paper we have presented an approach to the quantitative analysis of collective adaptive systems that is scalable and able to tackle systems which are beyond the scope of traditional analysis techniques. The discrete system is approximated by a system of ODEs that characterize the moments of the dynamic behaviour, and these ODEs are automatically derived from a high-level system description. Here we have used the stochastic process algebra PALOMA but the approach could be readily adapted to other formal description languages for PCTMCs. Although previous work has derived higher moments from process algebra models [Hayden and Bradley 2010; Guenther and Bradley 2011], the system and language we are tackling have more complexity. Moreover, our work is embodied in an implementation[1]. Depending on the structure of the entities that make up the collective system, the number of ODEs, particularly for the second-order joint moments, can grow large, impeding the efficiency of the solution. We have demonstrated a rigorous technique for model reduction which approximates these joint moments when they are likely to have less impact on the overall behaviour, and which reduces the number of ODEs that must be simulated. We have shown that the results of the moment approximation technique have acceptably low levels of error even when the model reduction technique is applied.

REFERENCES

Alexander Andreychenko, Linar Mikeev, and Verena Wolf. 2015. Model Reconstruction for Moment-Based Stochastic Chemical Kinetics. *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 25, 2 (2015), 12. DOI:http://dx.doi.org/10.1145/2699712

Luca Bortolussi, Rocco De Nicola, Vashti Galpin, Stephen Gilmore, Jane Hillston, Diego Latella, Michele Loreti, and Mieke Massink. 2015. CARMA: Collective Adaptive Resource-sharing Markovian Agents. In *Proceedings Thirteenth Workshop on Quantitative Aspects of Programming Languages and Systems, QAPL 2015, London, UK, 11th-12th April 2015*. 16–31. DOI:http://dx.doi.org/10.4204/EPTCS.194.2

Dario Bruneo, Marco Scarpa, Andrea Bobbio, Davide Cerotti, and Marco Gribaudo. 2012. Markovian Agent Modeling Swarm Intelligence Algorithms in Wireless Sensor Networks. *Performance Evaluation* 69, 3 (2012), 135–149. DOI:http://dx.doi.org/10.1016/j.peva.2010.11.007

Luca Cardelli. 2008a. From processes to ODEs by chemistry. In *Fifth Ifip International Conference On Theoretical Computer Science–Tcs 2008*. Springer, 261–281. DOI:http://dx.doi.org/10.1007/978-0-387-09680-3_18

Luca Cardelli. 2008b. On process rate semantics. *Theoretical Computer Science* 391, 3 (2008), 190–215. DOI:http://dx.doi.org/10.1016/j.tcs.2007.11.012

Davide Cerotti, Marco Gribaudo, Andrea Bobbio, Carlos T Calafate, and Pietro Manzoni. 2010. A Markovian Agent Model For Fire Propagation in Outdoor Environments. In *Computer Performance Engineering*. Springer, 131–146. DOI:http://dx.doi.org/10.1007/978-3-642-15784-4_9

Stefan Engblom. 2006. Computing the moments of high dimensional solutions of the master equation. *Appl. Math. Comput.* 180, 2 (2006), 498–515. DOI:http://dx.doi.org/10.1016/j.amc.2005.12.032

Cheng Feng and Jane Hillston. 2014. PALOMA: A Process Algebra for Located Markovian Agents. In *Quantitative Evaluation of Systems*. Springer, 265–280. DOI:http://dx.doi.org/10.1007/978-3-319-10696-0_22

Daniel T Gillespie. 1977. Exact stochastic simulation of coupled chemical reactions. *The journal of physical chemistry* 81, 25 (1977), 2340–2361. DOI:http://dx.doi.org/10.1021/j100540a008

Marcel C Guenther and Jeremy T Bradley. 2011. Higher Moment Analysis of a Spatial Stochastic Process Algebra. In *Computer Performance Engineering*. Springer, 87–101. DOI:http://dx.doi.org/10.1007/978-3-642-39408-9_9

---

[1]https://github.com/cfeng783/paloma/wiki#the-paloma-eclipse-plug-in

Marcel C Guenther, Anton Stefanek, and Jeremy T Bradley. 2013. Moment closures for perfor-
mance models with highly non-linear rates. In *Computer Performance Engineering*. Springer, 32–47.
DOI:http://dx.doi.org/10.1007/978-3-642-36781-6_3

Richard A Hayden and Jeremy T Bradley. 2010. A fluid analysis framework for a Markovian process algebra.
Theoretical Computer Science 411, 22 (2010), 2260–2297. DOI:http://dx.doi.org/10.1016/j.tcs.2010.02.001

David Hiebeler. 2006. Moment equations and dynamics of a household SIS epidemiological model. *Bulletin
of mathematical biology* 68, 6 (2006), 1315–1333. DOI:http://dx.doi.org/10.1007/s11538-006-9080-1

Jane Hillston. 2005a. *A Compositional Approach to Performance Modelling*. CUP.

Jane Hillston. 2005b. Fluid Flow Approximation of PEPA Models. In *2nd International Conference on Quan-
titative Evaluation of Systems*. IEEE, 33–42. DOI:http://dx.doi.org/10.1109/QEST.2005.12

Charles Antony Richard Hoare. 1985. *Communicating sequential processes*. Vol. 178. Prentice-hall Engle-
wood Cliffs.

Matthew J Keeling. 1999. The effects of local spatial structure on epidemiological invasions. *Pro-
ceedings of the Royal Society of London B: Biological Sciences* 266, 1421 (1999), 859–867.
DOI:http://dx.doi.org/10.1098/rspb.1999.0716

Matthew J Keeling. 2000. Multiplicative moments and measures of persistence in ecology. *Journal of Theo-
retical Biology* 205, 2 (2000), 269–281. DOI:http://dx.doi.org/10.1006/jtbi.2000.2066

Isthrinayagy Krishnarajah, Alex Cook, Glenn Marion, and Gavin Gibson. 2005. Novel moment closure
approximations in stochastic epidemics. *Bulletin of mathematical biology* 67, 4 (2005), 855–873.
DOI:http://dx.doi.org/10.1016/j.bulm.2004.11.002

Thomas G Kurtz. 1970. Solutions of ordinary differential equations as limits of pure jump Markov processes.
Journal of applied Probability 7, 1 (1970), 49–58. DOI:http://dx.doi.org/10.2307/3212147

Diego Latella, Michele Loreti, Mieke Massink, and Valerio Senni. 2014. Stochastically timed
predicate-based communication primitives for autonomic computing. *EPTCS* 154 (2014), 1–16.
DOI:http://dx.doi.org/10.4204/EPTCS.154.1

Peter Midgley. 2009. The role of smart bike-sharing systems in urban mobility. *Journeys* 2 (2009), 23–31.

Cecilia Nardini, Balázs Kozma, and Alain Barrat. 2008. Whos talking first? consensus or lack
thereof in coevolving opinion formation models. *Physical review letters* 100, 15 (2008), 158701.
DOI:http://dx.doi.org/10.1103/PhysRevLett.100.158701

Rocco De Nicola, Michele Loreti, Rosario Pugliese, and Francesco Tiezzi. 2014. A formal approach to au-
tonomic systems programming: The SCEL language. *ACM Transactions on Autonomous and Adaptive
Systems (TAAS)* 9, 2 (2014), 7. DOI:http://dx.doi.org/10.1145/2619998

Emanuele Pugliese and Claudio Castellano. 2009. Heterogeneous pair approximation
for voter models on networks. *EPL (Europhysics Letters)* 88, 5 (2009), 58004.
DOI:http://dx.doi.org/10.1209/0295-5075/88/58004

Aaditya V Rangan and David Cai. 2006. Maximum-entropy closures for kinetic theo-
ries of neuronal network dynamics. *Physical review letters* 96, 17 (2006), 178101.
DOI:http://dx.doi.org/10.1103/PhysRevLett.96.178101

Tim Rogers. 2011. Maximum-entropy moment-closure for stochastic systems on networks.
Journal of Statistical Mechanics: Theory and Experiment 2011, 05 (2011), P05007.
DOI:http://dx.doi.org/10.1088/1742-5468/2011/05/P05007

A Singer. 2004. Maximum entropy formulation of the Kirkwood superposition approximation. *The Journal
of chemical physics* 121, 8 (2004), 3657–3666. DOI:http://dx.doi.org/10.1063/1.1776552

Abhyudai Singh and Joao Pedro Hespanha. 2006a. Lognormal moment closures for biochem-
ical reactions. In *45th IEEE Conference on Decision and Control*. IEEE, 2063–2068.
DOI:http://dx.doi.org/10.1109/CDC.2006.376994

Abhyudai Singh and Joao Pedro Hespanha. 2006b. Moment closure techniques for stochas-
tic models in population biology. In *American Control Conference, 2006*. IEEE, 47304735.
DOI:http://dx.doi.org/10.1109/ACC.2006.1657468

Mirco Tribastone, Stephen Gilmore, and Jane Hillston. 2012. Scalable Differential Analysis of
Process Algebra Models. *IEEE Transactions on Software Engineering* 38, 1 (2012), 205–219.
DOI:http://dx.doi.org/10.1109/TSE.2010.82

Peter Whittle. 1957. On the use of the normal approximation in the treatment of stochastic processes. *Jour-
nal of the Royal Statistical Society. Series B (Methodological)* (1957), 268–281.