Model checking single agent behaviours by fluid approximation

Jane Hillston
LFCS, University of Edinburgh

26th October 2016

Outline

Outline

Overview

Fluid or mean field methods have been very successful as an approximate description of the collective (average) behaviour of models made up of many (simple) interacting agents, e.g. stochastic process algebra models of populations.

Overview

Fluid or mean field methods have been very successful as an approximate description of the collective (average) behaviour of models made up of many (simple) interacting agents, e.g. stochastic process algebra models of populations.

Can we use them to query stochastic models and estimate more complex stochastic properties?

Overview

Fluid or mean field methods have been very successful as an approximate description of the collective (average) behaviour of models made up of many (simple) interacting agents, e.g. stochastic process algebra models of populations.

Can we use them to query stochastic models and estimate more complex stochastic properties?

Stated otherwise:

Can we do fluid model checking?

The goal of this work

We consider collective systems, composed of many interacting agents of one or more classes, with behaviour over discrete states in continuous time.

The goal of this work

We consider collective systems, composed of many interacting agents of one or more classes, with behaviour over discrete states in continuous time.

We focus on questions related to the behaviour of an individual agent in the system.

The goal of this work

We consider collective systems, composed of many interacting agents of one or more classes, with behaviour over discrete states in continuous time.

We focus on questions related to the behaviour of an individual agent in the system.

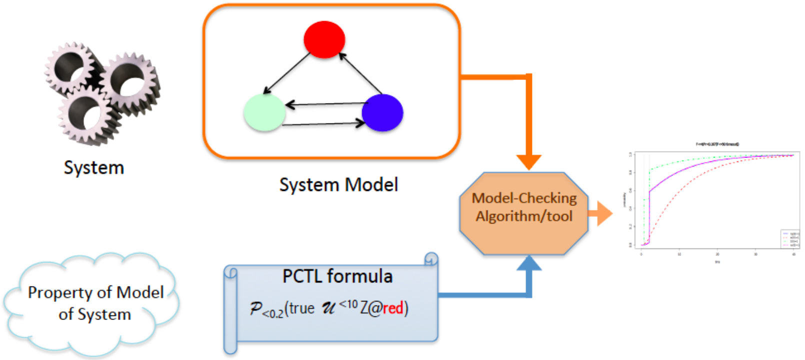
Examples

There are many examples in which this can be interesting:

- Estimate performance metrics in network models, from the point of view of a single user/single server.
- Ecological models, when one is interested in the survival chances of an individual.
- ...

Model checking

Model checking: automatically querying the behaviour of an automata-based model with respect to a property expressed in a suitable logic.



Model checking

Model checking requires two inputs:

- a description of the system, usually given in some high-level modelling formalism, which can be used to automatically generate a state-based representation; CTMC (Continuous Time Markov Chain)

Model checking

Model checking requires two inputs:

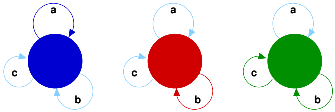
- a description of the system, usually given in some high-level modelling formalism, which can be used to automatically generate a state-based representation; CTMC (Continuous Time Markov Chain)
- a specification of one or more desired properties of the system, normally using temporal logics such as CTL (Computational Tree Logic), LTL (Linear-time Temporal Logic) CSL (Continuous Stochastic Logic).

Model checking

There are two broad approaches to model checking:

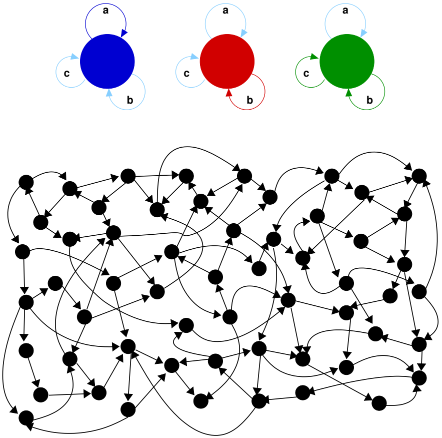
- Explicit state model checking (exhaustive exploration for all possible states/executions): exact results obtained via numerical computation.
- Statistical model-checking (discrete event simulation and sampling over multiple runs): approximate results.

Solving discrete state models



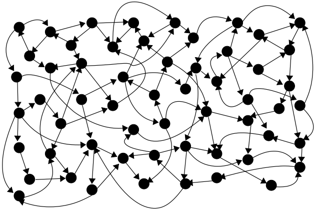
The CTMC will have global states determined by the interleaving of the local states of all the participating agents.

Solving discrete state models



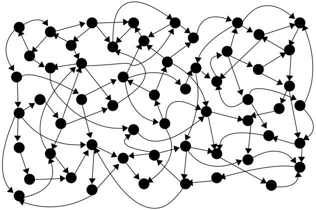
The CTMC will have global states determined by the interleaving of the local states of all the participating agents.

Solving discrete state models



When the size of the state space is not too large they are amenable to numerical solution (linear algebra) to determine a steady state or transient probability distribution.

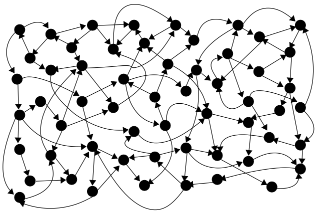
Solving discrete state models



When the size of the state space is not too large they are amenable to numerical solution (linear algebra) to determine a steady state or transient probability distribution.

$$Q = \begin{pmatrix} q_{1,1} & q_{1,2} & \cdots & q_{1,N} \\ q_{2,1} & q_{2,2} & \cdots & q_{2,N} \\ \vdots & \vdots & & \vdots \\ q_{N,1} & q_{N,2} & \cdots & q_{N,N} \end{pmatrix}$$

Solving discrete state models



When the size of the state space is not too large they are amenable to numerical solution (linear algebra) to determine a steady state or transient probability distribution.

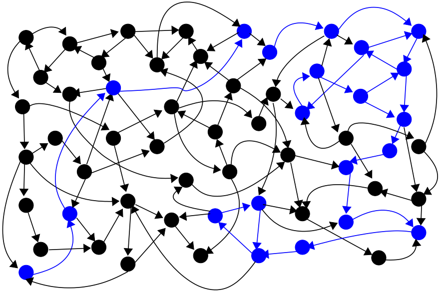
$$Q = \begin{pmatrix} q_{1,1} & q_{1,2} & \cdots & q_{1,N} \\ q_{2,1} & q_{2,2} & \cdots & q_{2,N} \\ \vdots & \vdots & & \vdots \\ q_{N,1} & q_{N,2} & \cdots & q_{N,N} \end{pmatrix}$$

$$\pi(t) = (\pi_1(t), \pi_2(t), \ldots, \pi_N(t))$$

$$\pi(\infty)Q = 0$$

Solving discrete state models

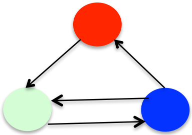
Alternatively they may be studied using stochastic simulation. Each run generates a single trajectory through the state space. Many runs are needed in order to obtain average behaviours.



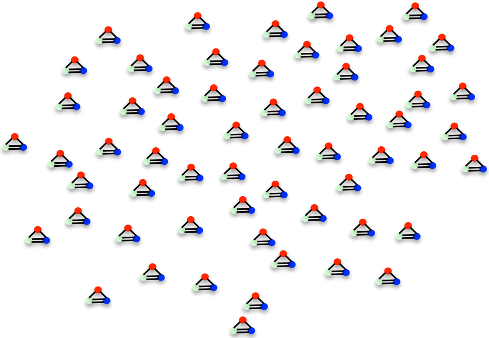
State space explosion

As the size of the state space becomes large it becomes infeasible to carry out numerical solution and extremely time-consuming to conduct stochastic simulation.

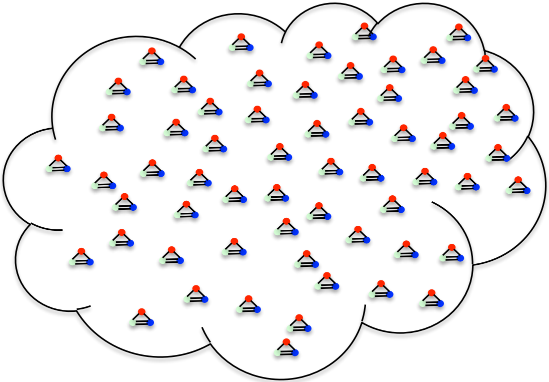
Mean Field Approximation



Mean Field Approximation

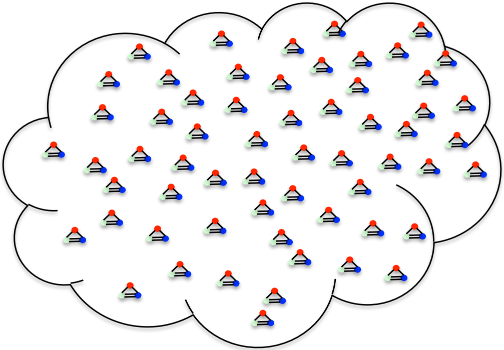


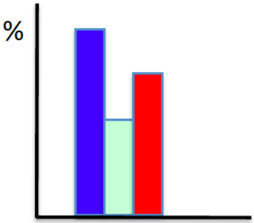
Mean Field Approximation



We view the population of agents more abstractly, assuming that individuals are indistinguishable.

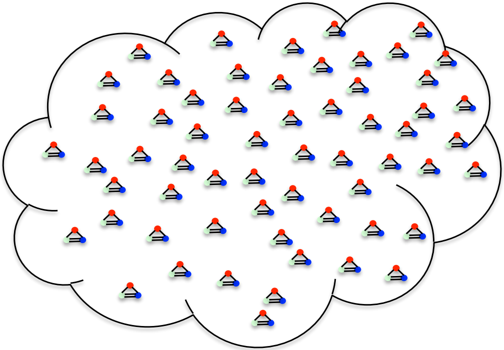
Mean Field Approximation

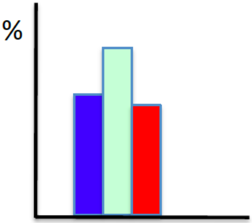




An occupancy measure records the proportion of agents that are currently exhibiting each possible behaviour.

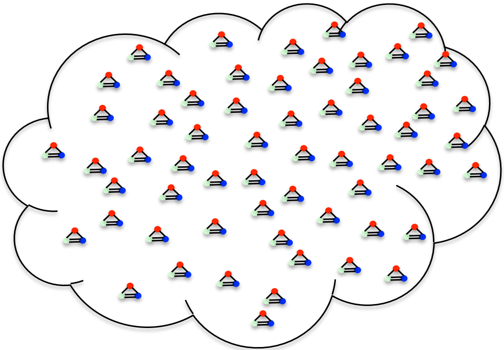
Mean Field Approximation

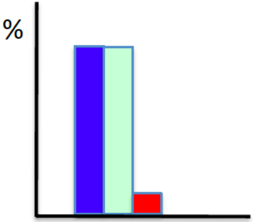




An occupancy measure records the proportion of agents that are currently exhibiting each possible behaviour.

Mean Field Approximation





For a large class of models, just as the size of the state space becomes unmanageable, the models become amenable to an efficient, scale-free approximation in terms of the occupancy measure.

Continuous Approximation: Intuition

Although in reality all state transitions are discrete, we can see that as the size of the population grows, the impact of each state change becomes smaller, and the error introduced by continuous approximation decreases.

Continuous Approximation: Intuition

Although in reality all state transitions are discrete, we can see that as the size of the population grows, the impact of each state change becomes smaller, and the error introduced by continuous approximation decreases.

Continuous Approximation: Intuition

Although in reality all state transitions are discrete, we can see that as the size of the population grows, the impact of each state change becomes smaller, and the error introduced by continuous approximation decreases.

Continuous Approximation: Intuition

Although in reality all state transitions are discrete, we can see that as the size of the population grows, the impact of each state change becomes smaller, and the error introduced by continuous approximation decreases.

Continuous Approximation: Intuition

Although in reality all state transitions are discrete, we can see that as the size of the population grows, the impact of each state change becomes smaller, and the error introduced by continuous approximation decreases.

Continuous Approximation: Intuition

Although in reality all state transitions are discrete, we can see that as the size of the population grows, the impact of each state change becomes smaller, and the error introduced by continuous approximation decreases.

Continuous Approximation: Intuition

Although in reality all state transitions are discrete, we can see that as the size of the population grows, the impact of each state change becomes smaller, and the error introduced by continuous approximation decreases.

Continuous Approximation: Intuition

Although in reality all state transitions are discrete, we can see that as the size of the population grows, the impact of each state change becomes smaller, and the error introduced by continuous approximation decreases.

Continuous Approximation: Intuition

Although in reality all state transitions are discrete, we can see that as the size of the population grows, the impact of each state change becomes smaller, and the error introduced by continuous approximation decreases.

Continuous Approximation: Intuition

Although in reality all state transitions are discrete, we can see that as the size of the population grows, the impact of each state change becomes smaller, and the error introduced by continuous approximation decreases.

Continuous Approximation: Intuition

Although in reality all state transitions are discrete, we can see that as the size of the population grows, the impact of each state change becomes smaller, and the error introduced by continuous approximation decreases.

Continuous Approximation: Intuition

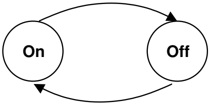
Although in reality all state transitions are discrete, we can see that as the size of the population grows, the impact of each state change becomes smaller, and the error introduced by continuous approximation decreases.

Fluid or mean field approximation

We approximate the discrete steps on the original state space by continuous evolution of the occupancy measures. A system of ordinary differential equations is constructed to approximate the behaviour of the system.

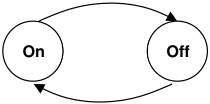
Outline

Population models — intuition



$Y(t)$

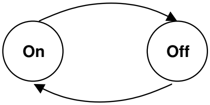
Population models — intuition

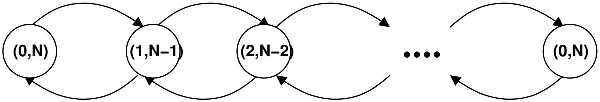


$Y(t)$

N copies: $Y_i^{(N)}$

Population models — intuition



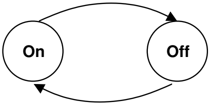


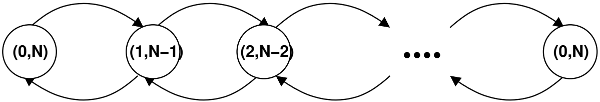
$Y(t)$

N copies: $Y_i^{(N)}$

$\mathbf{X}^{(N)}(t)$

Population models — intuition





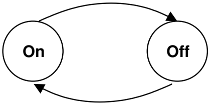
$Y(t)$

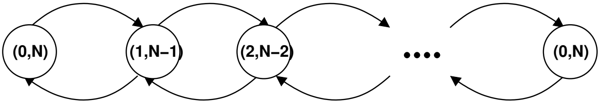
N copies: $Y_i^{(N)}$

$\mathbf{X}^{(N)}(t)$

$$X_j^{(N)} = \sum_{i=1}^{N} \mathbf{1}\{Y_i^{(N)} = j\}$$

Population models — intuition





$Y(t)$

N copies: $Y_i^{(N)}$

$\mathbf{X}^{(N)}(t)$

$$X_j^{(N)} = \sum_{i=1}^{N} \mathbf{1}\{Y_i^{(N)} = j\}$$

- $Y(t)$, $Y_i^{(N)}(t)$ and $\mathbf{X}^{(N)}(t)$ are all CTMCs;
- As N increases we get a sequence of CTMCs, $\mathbf{X}^{(N)}(t)$

Population state space

- The population process $\mathbf{X}^{(N)} = (X_1^{(N)}, \ldots, X_n^{(N)})$ has the dimension of the state space of $Y(t)$.

Population state space

- The population process $\mathbf{X}^{(N)} = (X_1^{(N)}, \ldots, X_n^{(N)})$ has the dimension of the state space of $Y(t)$.
- Importantly, its dimensions are independent of N.

Population state space

- The population process $\mathbf{X}^{(N)} = (X_1^{(N)}, \ldots, X_n^{(N)})$ has the dimension of the state space of $Y(t)$.
- Importantly, its dimensions are independent of N.
- Essentially we are making a counting abstraction and aggregation of the state space.

Population transitions

- The dynamics of the population models is expressed in terms of a set of possible transitions, $\mathcal{T}^{(N)}$.

Population transitions

- The dynamics of the population models is expressed in terms of a set of possible transitions, $\mathcal{T}^{(N)}$.
- Transitions are stochastic, and take an exponentially distributed time to happen.

Population transitions

- The dynamics of the population models is expressed in terms of a set of possible transitions, $\mathcal{T}^{(N)}$.
- Transitions are stochastic, and take an exponentially distributed time to happen.
- Their rate may depend on the current global state of the system.

Population transitions

- The dynamics of the population models is expressed in terms of a set of possible transitions, $\mathcal{T}^{(N)}$.
- Transitions are stochastic, and take an exponentially distributed time to happen.
- Their rate may depend on the current global state of the system.
- Each transition is specified by a rate function $r_{\tau}^{(N)}$, and by an update vector $\mathbf{v}_{\tau}$, specifying the impact of the event on the population vector.

Population transitions

- The dynamics of the population models is expressed in terms of a set of possible transitions, $\mathcal{T}^{(N)}$.
- Transitions are stochastic, and take an exponentially distributed time to happen.
- Their rate may depend on the current global state of the system.
- Each transition is specified by a rate function $r_\tau^{(N)}$, and by an update vector $\mathbf{v}_\tau$, specifying the impact of the event on the population vector.
- The infinitesimal generator matrix $Q^{(N)}$ of $\mathbf{X}^{(N)}(t)$ is defined as:

$$q_{\mathbf{x},\mathbf{x}'} = \sum \{ r_\tau(\mathbf{x}) \mid \tau \in \mathcal{T},\ \mathbf{x}' = \mathbf{x} + \mathbf{v}_\tau \}.$$

Population models — summary of notation

Individuals

We have N individuals $Y_i^{(N)} \in S$, $S = \{1, 2, \ldots, n\}$ in the system (can have multiple classes).

Population models — summary of notation

Individuals

We have N individuals $Y_i^{(N)} \in S$, $S = \{1, 2, \ldots, n\}$ in the system (can have multiple classes).

System variables

$$X_j^{(N)} = \sum_{i=1}^{N} \mathbf{1}\{Y_i^{(N)} = j\}, \quad \text{and} \quad \mathbf{X}^{(N)} = (X_1^{(N)}, \ldots, X_n^{(N)})$$

Population models — summary of notation

Individuals

We have N individuals $Y_i^{(N)} \in S$, $S = \{1, 2, \ldots, n\}$ in the system (can have multiple classes).

System variables

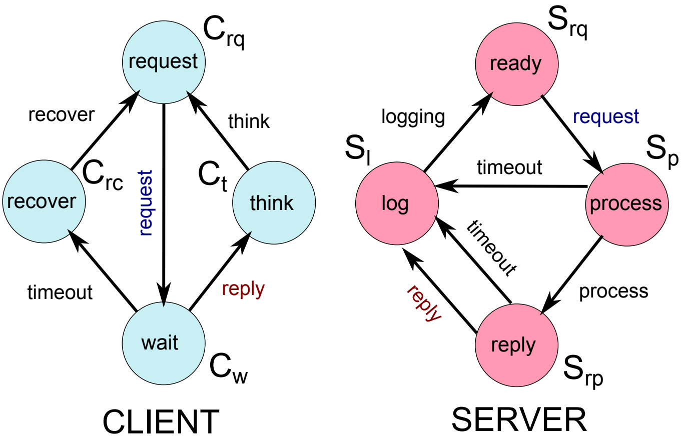
$X_j^{(N)} = \sum_{i=1}^{N} \mathbf{1}\{Y_i^{(N)} = j\}$, and $\mathbf{X}^{(N)} = (X_1^{(N)}, \ldots, X_n^{(N)})$

Dynamics (system level)

$\mathbf{X}^{(N)}$ is a CTMC with transitions $\tau \in \mathcal{T}$:

$$\tau\colon\ \mathbf{X}^{(N)} \text{ to } \mathbf{X}^{(N)} + \mathbf{v}_\tau \text{ at rate } r_\tau^{(N)}(\mathbf{X})$$

Example: client server interaction



Example: client server interaction

Variables

- 4 variables for the client states: C_{rq}, C_w, C_{rc} C_t.
- 4 variables for the server states: S_{rq}, S_p, S_{rp} S_l.

Transitions

There are 7 transition in total. Rates based on hand-shaking.

- request: $(\mathbf{1}_{C_w,S_p} - \mathbf{1}_{C_{rq},S_{rq}}, kr \cdot min(C_{rq}, S_{rq}))$
- reply: $(\mathbf{1}_{C_t,S_l} - \mathbf{1}_{C_w,S_{rp}}, min(k_w C_w, k_{rp} S_{rp}))$
- timeout: $(\mathbf{1}_{C_{rc}} - \mathbf{1}_{C_w}, k_{to} C_w)$
- ...

Scaling Conditions

Scaling assumptions

- We have a sequence $\mathbf{X}^{(N)}$ of population CTMCs.
- We normalise such models, dividing variables by N:

$$\hat{\mathbf{X}} = \frac{\mathbf{X}}{N}$$

Scaling Conditions

Scaling assumptions

- We have a sequence $\mathbf{X}^{(N)}$ of population CTMCs.
- We normalise such models, dividing variables by N:

$$\hat{\mathbf{X}} = \frac{\mathbf{X}}{N} \qquad \text{occupancy measures}$$

Scaling Conditions

Scaling assumptions

- We have a sequence $\mathbf{X}^{(N)}$ of population CTMCs.
- We normalise such models, dividing variables by N:

$$\hat{\mathbf{X}} = \frac{\mathbf{X}}{N} \qquad \text{occupancy measures}$$

- for each $\tau \in \mathcal{T}^{(N)}$
  - the normalised update is $\hat{\mathbf{v}} = \mathbf{v}/N$
  - there is a normalised rate function $\hat{r}_{\tau}^{(N)}(\hat{\mathbf{X}})$

Scaling Conditions

Scaling assumptions

- We have a sequence $\mathbf{X}^{(N)}$ of population CTMCs.
- We normalise such models, dividing variables by N:

$$\hat{\mathbf{X}} = \frac{\mathbf{X}}{N} \qquad \text{occupancy measures}$$

- for each $\tau \in \mathcal{T}^{(N)}$
  - the normalised update is $\hat{\mathbf{v}} = \mathbf{v}/N$
  - there is a normalised rate function $\hat{r}_\tau^{(N)}(\hat{\mathbf{X}})$
- $\forall \tau$ assume there exists a bounded and Lipschitz continuous function $f_\tau(\hat{\mathbf{X}})$, the limit rate function on normalised variables, independent of N, such that $\frac{1}{N}\hat{r}_\tau^{(N)}(\mathbf{x}) \to f_\tau(\mathbf{x})$ uniformly.

Drift

Drift

The drift $F^{(N)}(\hat{\mathbf{X}})$ — the mean instantaneous increment of model variables in state $\hat{\mathbf{X}}$ — is defined as

$$F^{(N)}(\hat{\mathbf{X}}) = \sum_{\tau \in \hat{\mathcal{T}}} \frac{1}{N} \mathbf{v}_\tau \, \hat{r}_\tau^{(N)}(\hat{\mathbf{X}})$$

Drift

Drift

The drift $F^{(N)}(\hat{\mathbf{X}})$ — the mean instantaneous increment of model variables in state $\hat{\mathbf{X}}$ — is defined as

$$F^{(N)}(\hat{\mathbf{X}}) = \sum_{\tau \in \hat{\mathcal{T}}} \frac{1}{N} \mathbf{v}_\tau \, \hat{r}_\tau^{(N)}(\hat{\mathbf{X}})$$

Limit Drift

Let f_τ be the limit rate functions.

The limit drift of the model is

$$F(\hat{\mathbf{X}}) = \sum_{\tau \in \hat{\mathcal{T}}} \mathbf{v}_\tau f_\tau(\hat{\mathbf{X}}),$$

and $F^{(N)}(\mathbf{x}) \to F(\mathbf{x})$ uniformly as $N \longrightarrow \infty$.

Fluid ODE and Fluid approximation theorem

Fluid ODE

The fluid ODE is

$$\frac{d\mathbf{x}}{dt} = F(\mathbf{x}), \quad with\ \mathbf{x}(0) = \mathbf{x_0} \in S.$$

Since F is Lipschitz (all f_τ are), this ODE has a unique solution $\mathbf{x}(t)$ starting from $\mathbf{x_0}$.

Deterministic Approximation Theorem (Kurtz)

Assume that $\exists\, \mathbf{x_0} \in S$ such that $\hat{\mathbf{X}}^{(N)}(0) \to \mathbf{x_0}$ in probability. Then, for any finite time horizon $T < \infty$, it holds that as $N \longrightarrow \infty$:

$$\mathbb{P}\left\{ \sup_{0 \leq t \leq T} ||\hat{\mathbf{X}}^{(N)}(t) - \mathbf{x}(t)|| > \varepsilon \right\} \to 0.$$

T.G.Kurtz. *Solutions of ordinary differential equations as limits of pure jump Markov processes.* Journal of Applied Probability, 1970.

Focusing on one individual

- We focus on a single individual $Y_h^{(N)}(t)$, a (Markov) process on the state space $S = \{1, \ldots, n\}$, conditional on the global state of the complete population $\hat{\mathbf{X}}^{(N)}(t)$.

Focusing on one individual

- We focus on a single individual $Y_h^{(N)}(t)$, a (Markov) process on the state space $S = \{1, \dots, n\}$, conditional on the global state of the complete population $\hat{\mathbf{X}}^{(N)}(t)$.
- Rate $q_{ij}(\hat{\mathbf{X}})$ depends on the global system state and $\hat{\mathbf{X}}^{(N)}(t)$.

Focusing on one individual

- We focus on a single individual $Y_h^{(N)}(t)$, a (Markov) process on the state space $S = \{1, \ldots, n\}$, conditional on the global state of the complete population $\hat{\mathbf{X}}^{(N)}(t)$.
- Rate $q_{ij}(\hat{\mathbf{X}})$ depends on the global system state and $\hat{\mathbf{X}}^{(N)}(t)$.
- Its transition rates q_{ij}, are obtained projecting on a single agent the rate of global transitions that induce a change of state of at least one agent from i to j.

Focusing on one individual

- We focus on a single individual $Y_h^{(N)}(t)$, a (Markov) process on the state space $S = \{1, \ldots, n\}$, conditional on the global state of the complete population $\hat{\mathbf{X}}^{(N)}(t)$.
- Rate $q_{ij}(\hat{\mathbf{X}})$ depends on the global system state and $\hat{\mathbf{X}}^{(N)}(t)$.
- Its transition rates q_{ij}, are obtained projecting on a single agent the rate of global transitions that induce a change of state of at least one agent from i to j.
- However, by the theorem, as $N \longrightarrow \infty$, the stochastic fluctuations of $\hat{\mathbf{X}}^{(N)}(t)$ tend to vanish, and the stochastic behaviour of $Y_h^{(N)}(t)$ can be approximated by making it dependent only on the fluid limit $\mathbf{x}(t)$.

Focusing on one individual

- We focus on a single individual $Y_h^{(N)}(t)$, a (Markov) process on the state space $S = \{1, \ldots, n\}$, conditional on the global state of the complete population $\hat{\mathbf{X}}^{(N)}(t)$.
- Rate $q_{ij}(\hat{\mathbf{X}})$ depends on the global system state and $\hat{\mathbf{X}}^{(N)}(t)$.
- Its transition rates q_{ij}, are obtained projecting on a single agent the rate of global transitions that induce a change of state of at least one agent from i to j.
- However, by the theorem, as $N \longrightarrow \infty$, the stochastic fluctuations of $\hat{\mathbf{X}}^{(N)}(t)$ tend to vanish, and the stochastic behaviour of $Y_h^{(N)}(t)$ can be approximated by making it dependent only on the fluid limit $\mathbf{x}(t)$.
- Thus we construct the time-inhomogeneous CTMC $z(t)$ with state space S and rates $q_{ij}(\mathbf{x}(t))$.

Client-Server example

Single client

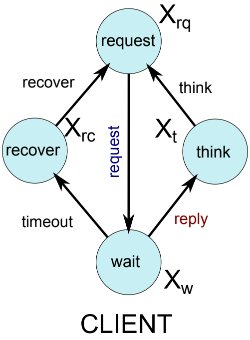
$Y^{(N)} \in \{rq, w, t, rc\}$

Rates of $Y_1^{(N)}$

- request: $\frac{1}{C_{rq}^{(N)}} k_r \min(C_{rq}^{(N)}, S_{rq}^{(N)})$
- reply: $\frac{1}{C_w^{(N)}} \min(k_w C_w^{(N)}, k_{rp} S_{rp}^{(N)})$
- timeout: k_{to}; recover: k_{rc}

Rates of z_1

- request: $k_r \min(1, \frac{s_{rq}(t)}{c_{rq}(t)})$
- reply: $\min(k_w, k_{rp} \frac{s_{rp}(t)}{c_w(t)})$
- timeout: k_{to}; recover: k_{rc}



Fluid Approximation ODEs

The fluid approximation ODEs can be interpreted in two different ways:

- as an approximation of the average of the system (usually a first order approximation). This is often termed a mean field approximation.

Fluid Approximation ODEs

The fluid approximation ODEs can be interpreted in two different ways:

- as an approximation of the average of the system (usually a first order approximation). This is often termed a mean field approximation.
- as an approximate description of system trajectories for large populations.

Fluid Approximation ODEs

The fluid approximation ODEs can be interpreted in two different ways:

- as an approximation of the average of the system (usually a first order approximation). This is often termed a mean field approximation.
- as an approximate description of system trajectories for large populations.

We focus on the second interpretation — a functional version of the Law of Large Numbers.

Fluid Approximation ODEs

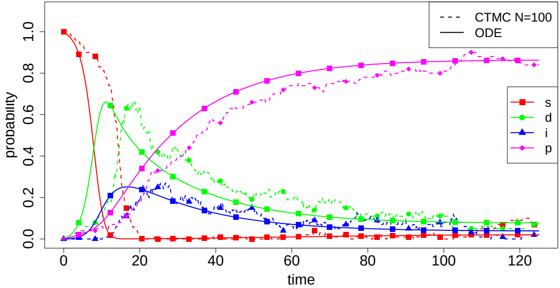
The fluid approximation ODEs can be interpreted in two different ways:

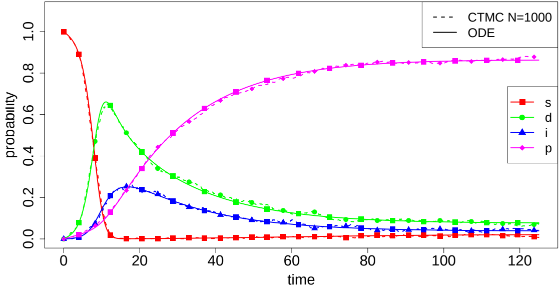
- as an approximation of the average of the system (usually a first order approximation). This is often termed a mean field approximation.
- as an approximate description of system trajectories for large populations.

We focus on the second interpretation — a functional version of the Law of Large Numbers.

Instead of having a sequence of random variables, converging to a deterministic value, here we have a sequence of CTMCs for increasing population size, which converge to a deterministic trajectory, the solution of the fluid ODE.

Illustrative trajectories

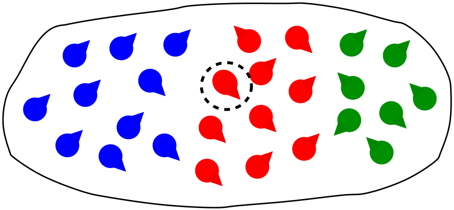




Comparison of the limit fluid ODE and a single stochastic trajectory of a network epidemic example, for total populations $N = 100$ and $N = 1000$.

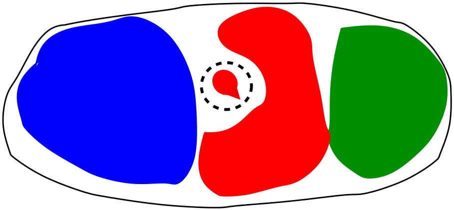
Implications of the Deterministic Approximation Theorem

The Theorem implies that in the limit the dynamics of a single agent becomes independent of other agents — it will sense them only through the collective system state, or mean field, described by the fluid limit.



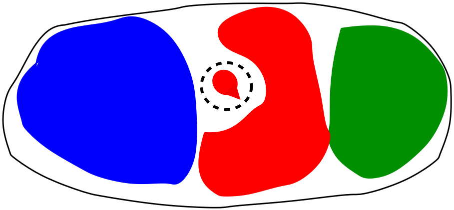
Implications of the Deterministic Approximation Theorem

The Theorem implies that in the limit the dynamics of a single agent becomes independent of other agents — it will sense them only through the collective system state, or mean field, described by the fluid limit.



Implications of the Deterministic Approximation Theorem

The Theorem implies that in the limit the dynamics of a single agent becomes independent of other agents — it will sense them only through the collective system state, or mean field, described by the fluid limit.



This asymptotic decoupling allows us to find a simple, time-inhomogenous, Markov chain for the evolution of the single agent, a result often known as fast simulation.

The idea

Approximate the behaviour of an agent $Y_h^{(N)}$ in the system using the time-inhomogeneous Markov chain z.

Model check temporal logic formulae on z.

The idea

Approximate the behaviour of an agent $Y_h^{(N)}$ in the system using the time-inhomogeneous Markov chain z.

Model check temporal logic formulae on z.

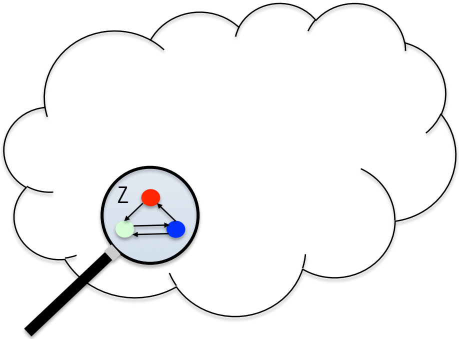
Outline of results

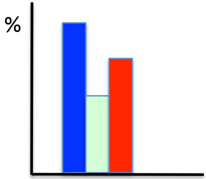
- A model checking algorithm for CSL on time-inhomogeneous CTMC (ICTMC).
- Investigation of its decidability.
- Convergence results (asymptotic correctness for large N).

L. Bortolussi, J. Hillston. Information & Computation 2015.

Fluid Model Checking

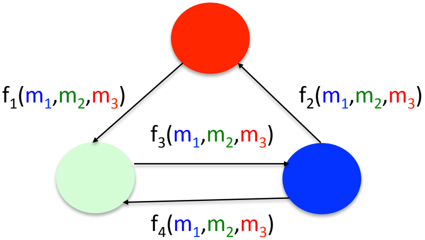
Properties related to a single agent are expressed in CSL,
e.g. *agent Z is in the blue state until it enters the red state and this must occur within time 1.7.*

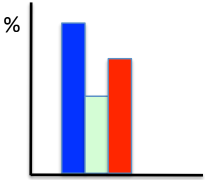




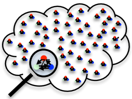
Fluid Model Checking

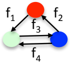
- This agent is considered in the mean field created by the rest of the system.
- The rates of its transitions become dependent on the state of the rest of the system and so vary over time.
- This is represented as a time-inhomogeneous CTMC.



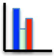


Fluid Model Checking

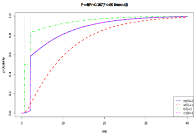




$$\frac{dx_1(t)}{dt} = F_1(x_1(t), x_2(t), x_3(t))$$
$$\frac{dx_2(t)}{dt} = F_2(x_1(t), x_2(t), x_3(t))$$
$$\frac{dx_3(t)}{dt} = F_3(x_1(t), x_2(t), x_3(t))$$



Model-Checking Algorithm/tool



Property of object Z in System

CSL formula

$\mathcal{P}_{<0.2}(Z@\text{blue}\ \mathcal{U}^{<1.7}\ Z@\text{red})$

Outline

Model checking the ICTMC

Care is needed to model check the ICTMC, which proceeds by explicitly calculating the reachability probabilities for states of interest (analogously to CSL model checking on CTMCs).

Model checking the ICTMC

Care is needed to model check the ICTMC, which proceeds by explicitly calculating the reachability probabilities for states of interest (analogously to CSL model checking on CTMCs).

The inhomogeneous time within the model means that truth values may change with respect to time.

Time-bounded Continuous Stochastic Logic (CSL)

The syntax of CSL is as follows:

$$\varphi \quad ::= \quad \texttt{true} \mid a \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathbf{P}_{\sim p}[\varphi \, \mathbf{U}^{I} \, \varphi] \mid \mathbf{S}_{\sim p}[\varphi]$$

where a is an atomic proposition, $\sim \in \{<, \leq, \geq, >\}$, $p \in [0, 1]$, I is an interval of $\mathbb{R}^{\geq 0}$ and $r, t \in \mathbb{R}^{\geq 0}$.

Time-bounded Continuous Stochastic Logic (CSL)

The syntax of CSL is as follows:

$$\varphi \quad ::= \quad \texttt{true} \mid a \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathbf{P}_{\sim p}[\varphi\, \mathbf{U}^{I}\, \varphi] \mid \mathbf{S}_{\sim p}[\varphi]$$

where a is an atomic proposition, $\sim \in \{<, \leq, \geq, >\}$, $p \in [0, 1]$, I is an interval of $\mathbb{R}^{\geq 0}$ and $r, t \in \mathbb{R}^{\geq 0}$.

P and **S** are probabilistic operators which include a probabilistic bound $\sim p$.

Probabilistic operators

A until formula $\varphi_1 \ \mathbf{U}^I \ \varphi_2$ is true of a path ω through the state space if, for some time instant $t \in I$, at time t the CSL subformula φ_2 is true and the subformula φ_1 is true at all preceding time instants.

Probabilistic operators

A until formula $\varphi_1 \; \mathbf{U}^I \; \varphi_2$ is true of a path ω through the state space if, for some time instant $t \in I$, at time t the CSL subformula φ_2 is true and the subformula φ_1 is true at all preceding time instants.

A formula $\mathbf{P}_{\sim p}[\varphi \; \mathbf{U}^I \; \varphi]$ is true in a state s if the probability of the formula $(\varphi \; \mathbf{U}^I \; \varphi)$ being satisfied from state s meets the bound $\sim p$.

Probabilistic operators

A until formula $\varphi_1 \ \mathbf{U}^I \ \varphi_2$ is true of a path ω through the state space if, for some time instant $t \in I$, at time t the CSL subformula φ_2 is true and the subformula φ_1 is true at all preceding time instants.

A formula $\mathbf{P}_{\sim p}[\varphi \ \mathbf{U}^I \ \varphi]$ is true in a state s if the probability of the formula $(\varphi \ \mathbf{U}^I \ \varphi)$ being satisfied from state s meets the bound $\sim p$.

A formula $\mathbf{S}_{\sim p}[\varphi]$ is true in state s if the probability that the formula φ being satisfied in a steady state reached from state s meets the bound $\sim p$.

CSL model checking for CTMC

Consider a CTMC with state space S and rates given by $Q = Q(t)$. Focus on the formula

$$\mathcal{P}_{\bowtie p}\left(\varphi_1 \, U^{[T_1,T_2]} \, \varphi_2\right)$$

Time-homogeneous CTMC

We check this formula by computing, for each state $s \in S$, the probability of paths satisfying $\varphi_1 \, U^{[T_1,T_2]} \, \varphi_2$ and then comparing this probability $\bowtie p$.

This is done via transient analysis on the chain in which $\neg\varphi_1$ and φ_2 states are made absorbing.

Model checking CSL formula on CTMC

- All φ_2 states are made absorbing, because once a state in this set has been reached the future evolution does not matter; this set of goal states is known as G.
- All states in $\neg(\varphi_1 \vee \varphi_2)$ are also made absorbing, because if one of these states is entered it is no longer possible to satisfy the formula; this set of unsatisfactory states is known as U.

Model checking CSL formula on CTMC

- All φ_2 states are made absorbing, because once a state in this set has been reached the future evolution does not matter; this set of goal states is known as G.
- All states in $\neg(\varphi_1 \vee \varphi_2)$ are also made absorbing, because if one of these states is entered it is no longer possible to satisfy the formula; this set of unsatisfactory states is known as U.

Once the modified CTMC is constructed, standard techniques are used to find the transient probabilities with respect to the time interval $[T_1, T_2]$.

C. Baier, B. Haverkort, H. Hermanns, J. Katoen Model checking CTMCs by transient analysis, CAV 2000

Model checking CSL formula on CTMC

- All φ_2 states are made absorbing, because once a state in this set has been reached the future evolution does not matter; this set of goal states is known as G.
- All states in $\neg(\varphi_1 \vee \varphi_2)$ are also made absorbing, because if one of these states is entered it is no longer possible to satisfy the formula; this set of unsatisfactory states is known as U.

Once the modified CTMC is constructed, standard techniques are used to find the transient probabilities with respect to the time interval $[T_1, T_2]$.

C. Baier, B. Haverkort, H. Hermanns, J. Katoen Model checking CTMCs by transient analysis, CAV 2000

Time-homogeneity $\Rightarrow$ we can run each transient analysis from time $T_1 = 0$ even if we have nested until formulae.

CSL model checking for ICTMC

Again consider a CTMC with state space S and rates given by $Q = Q(t)$ and the formula $\mathcal{P}_{\bowtie p}(\varphi_1 \, U^{[T_1, T_2]} \, \varphi_2)$.

CSL model checking for ICTMC

Again consider a CTMC with state space S and rates given by $Q = Q(t)$ and the formula $\mathcal{P}_{\bowtie p}(\varphi_1 \, U^{[T_1,T_2]} \, \varphi_2)$.

Time-inhomogeneous CTMC

We check this formula by computing, for each state $s \in S$, the probability of paths satisfying $\varphi_1 \, U^{[T_1,T_2]} \, \varphi_2$ and then comparing this probability $\bowtie p$.

Again this is done via transient analysis, now based on the Kolmogorov equations, in which $\neg\varphi_1$ and φ_2 states are made absorbing.

CSL model checking for ICTMC

Again consider a CTMC with state space S and rates given by $Q = Q(t)$ and the formula $\mathcal{P}_{\bowtie p}(\varphi_1 \, U^{[T_1,T_2]} \, \varphi_2)$.

Time-inhomogeneous CTMC

We check this formula by computing, for each state $s \in S$, the probability of paths satisfying $\varphi_1 \, U^{[T_1,T_2]} \, \varphi_2$ and then comparing this probability $\bowtie p$.

Again this is done via transient analysis, now based on the Kolmogorov equations, in which $\neg\varphi_1$ and φ_2 states are made absorbing.

But:

The truth value of φ in a state s depends on the time t at which we evaluate it!

This causes problems when we consider nested until formulae.

Time-dependent truth

- When computing the truth value of an until formula, we obtain a time dependent value $\mathbf{true}(\varphi, s, t)$ in each state.

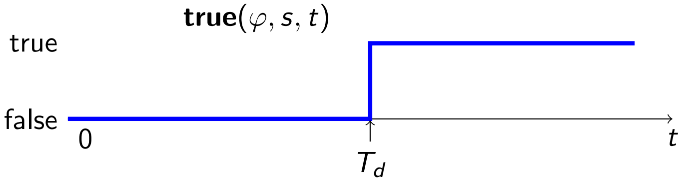
Time-dependent truth

- When computing the truth value of an until formula, we obtain a time dependent value $\mathbf{true}(\varphi, s, t)$ in each state.
- When we consider nested temporal operators, we need to take this into account.

Time-dependent truth

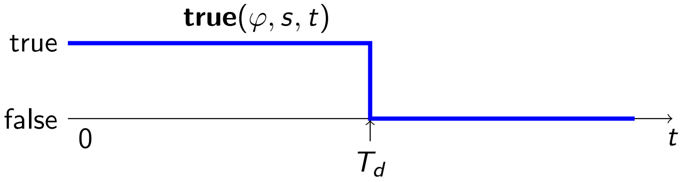
- When computing the truth value of an until formula, we obtain a time dependent value $\mathbf{true}(\varphi, s, t)$ in each state.
- When we consider nested temporal operators, we need to take this into account.
- The problem is that in this case the topology of goal and unsafe states in the CTMC can change in time.

Time dependent truth example: $F^{\leq T}\varphi = (\text{true } U^{[0,T]}\varphi)$



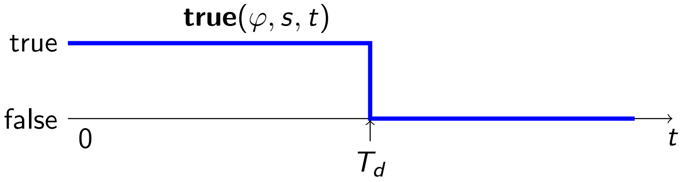
- State s becomes a goal state at time T_d.
- If we are in state s at time T_d^- (without having reached a φ state before), then we are suddenly in a φ-state at time T_d^+.
- At time T_d we need to add $\pi_{s',s}(t, T_d)$ to the reachability probability from each state s'.
- This introduces discontinuties in the reachability probability.
- At each discontinuity event, we also have to appropriately change the sets G and U.

Time dependent truth



At discontinuity times, changes in topology introduce discontinuities in the probability values.

Time dependent truth



At discontinuity times, changes in topology introduce discontinuities in the probability values.

Fortunately

Discontinuities happen at specific and fixed time instants.

We can carry out the transient solution, using Kolmogorov equations, piecewise.

At each discontinuity event, we also have to appropriately change the absorbing structure of the Q matrix.

The Algorithm (sketched)

Proceed bottom-up on the parse tree of a formula.

Case $\mathbf{true}(\mathcal{P}_{\bowtie p}(\varphi_1 U^{[0,T]} \varphi_2), t)$:

- Compute $\mathbf{true}(\varphi_1, t)$ and $\mathbf{true}(\varphi_2, t)$
- Let $T_1, \ldots, T_m$ be all the discontinuity points of $\mathbf{true}(\varphi_1, t)$ and $\mathbf{true}(\varphi_2, t)$ up to a final time T_f.
- Compute $\Pi(T_i, T_i + 1)$ for each i
- Compute $\Pi(0, T)$ using generalized Chapman-Kolmogorov equations
- Integrate $\frac{d}{dt}\Pi(t, t+T)$ up to T_f.
- Return $\mathbf{true}(\mathcal{P}_{\bowtie p}(\varphi_1 U^{[0,T]} \varphi_2), t) = \Pi(t, t+T) \bowtie p$.

The Algorithm (sketched)

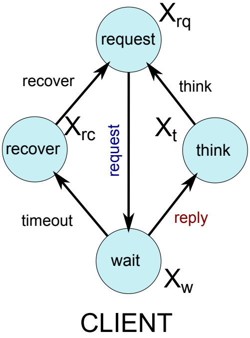
Proceed bottom-up on the parse tree of a formula.
Case $\mathbf{true}(\mathcal{P}_{\bowtie p}(\varphi_1 U^{[0,T]} \varphi_2), t)$:

- Compute $\mathbf{true}(\varphi_1, t)$ and $\mathbf{true}(\varphi_2, t)$
- Let $T_1, \ldots, T_m$ be all the discontinuity points of $\mathbf{true}(\varphi_1, t)$ and $\mathbf{true}(\varphi_2, t)$ up to a final time T_f.
- Compute $\Pi(T_i, T_i + 1)$ for each i
- Compute $\Pi(0, T)$ using generalized Chapman-Kolmogorov equations
- Integrate $\frac{d}{dt}\Pi(t, t+T)$ up to T_f.
- Return $\mathbf{true}(\mathcal{P}_{\bowtie p}(\varphi_1 U^{[0,T]} \varphi_2), t) = \Pi(t, t+T) \bowtie p$.

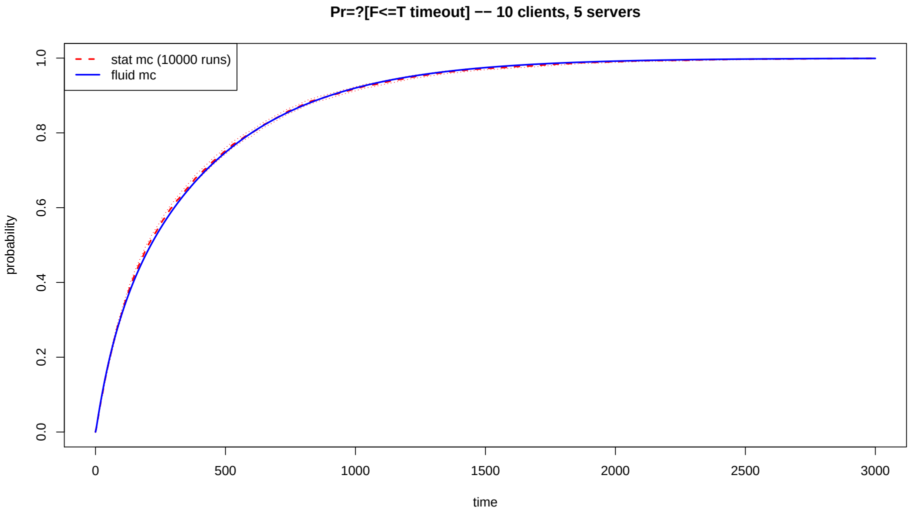
Use of Kolmogorov equations is feasible if the state space is small. This is usually the case for single agent mean field models.

Outline

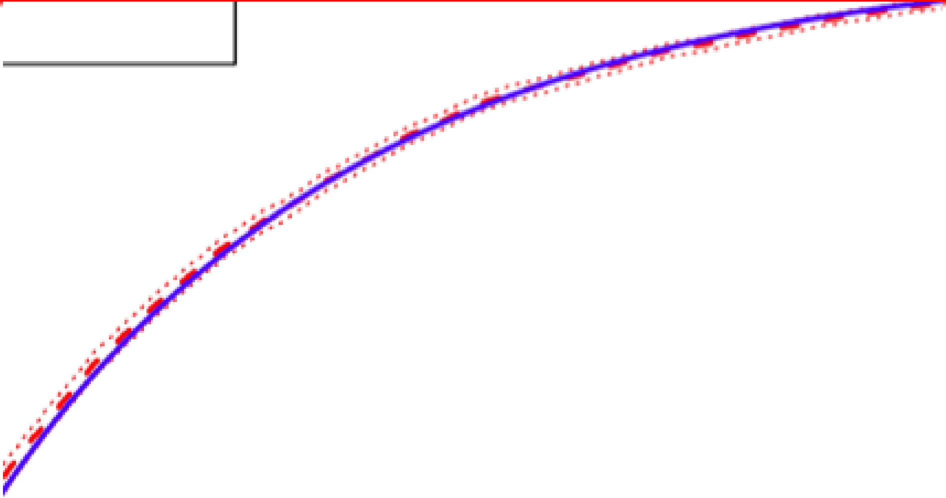
Client-Server example



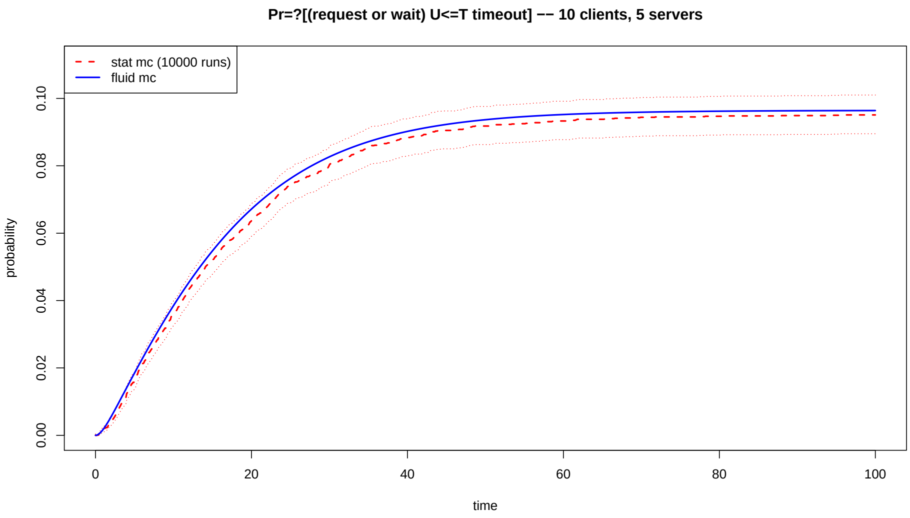
Client-Server: $\mathcal{P}_{=?}(F^{\leq T} a_{timeout}) = \mathcal{P}_{=?}(\text{true } U^{[0,T]} a_{timeout})$



Client-Server: $\mathcal{P}_{=?}(F^{\leq T} a_{timeout})$ (Zoom-in)



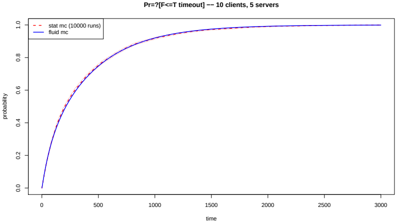
Client-Server: $\mathcal{P}_{=?}(a_{request} \vee a_{wait}\, U^{\leq T}\, a_{timeout})$

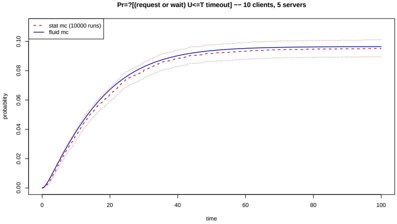


Client-Server: computational cost

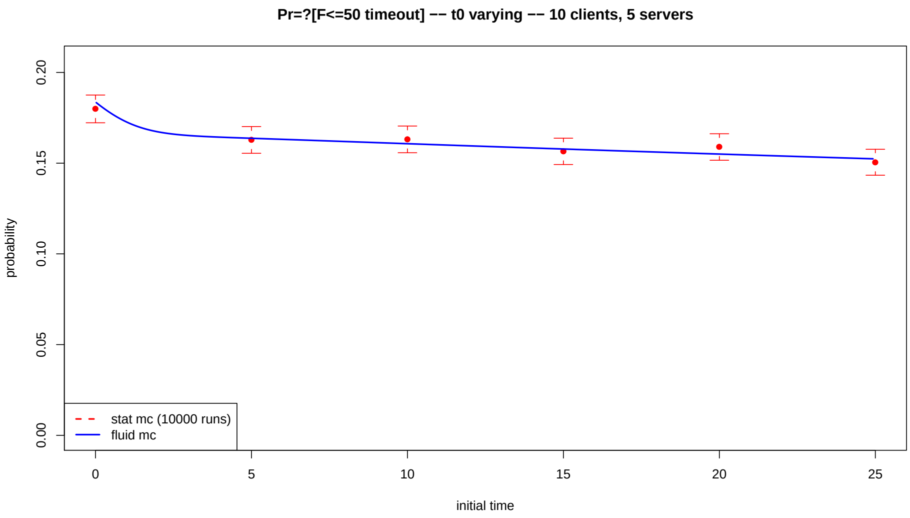
Computational cost

The cost of the fluid system is independent of N.
For this example (10 clients - 5 servers) it is $\sim$100 times faster than the simulation-based approach (which increases linearly with N).

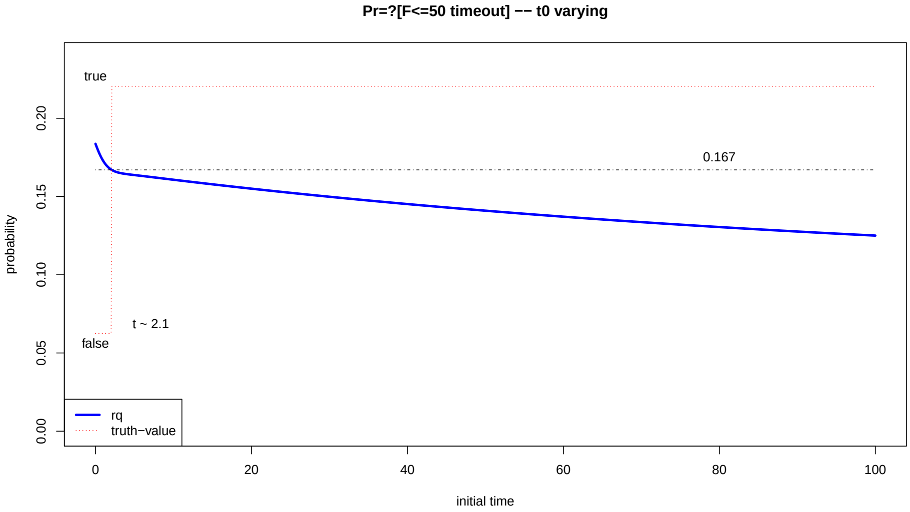




Client Server: $\mathcal{P}_{=?}F^{\leq T}a_{timeout}$ as a function of t_0



Client-Server: $F^{\leq T}(P_{<0.167}(F^{\leq 50}\, timeout))$



$P_{<0.167}(F^{\leq 50}\, timeout)$ from state rq of client.

Outline

Number of zeros of $P(s,t) - p$

When we are testing probability corresponding to a path formula against a probability bound we need solve equations of the form $P(s,t) - p = 0$.

Number of zeros of $P(s,t) - p$

- We want that this equation has a finite number of solutions in each $[0, T]$.
- We can enforce this by requiring rate functions of ICTMC to be piecewise real-analytic functions.

Decidability

Decidability

- We need algorithms to solve ODEs with error guarantee (interval analysis).

Decidability

Decidability

- We need algorithms to solve ODEs with error guarantee (interval analysis).
- We need to find zeros of function $P(s, t) - p$ (root finding).

Decidability

Decidability

- We need algorithms to solve ODEs with error guarantee (interval analysis).
- We need to find zeros of function $P(s,t) - p$ (root finding).
- To answer the CSL query for highest until formulae, we need to know if $P(s,0) \bowtie p$ (zero test).

Decidability

Decidability

- We need algorithms to solve ODEs with error guarantee (interval analysis).
- We need to find zeros of function $P(s, t) - p$ (root finding).
- To answer the CSL query for highest until formulae, we need to know if $P(s, 0) \bowtie p$ (zero test).

Decidability

Decidability

- We need algorithms to solve ODEs with error guarantee (interval analysis).
- We need to find zeros of function $P(s, t) - p$ (root finding).
- To answer the CSL query for highest until formulae, we need to know if $P(s, 0) \bowtie p$ (zero test).

- It is not known if root finding and zero test are decidable.

Decidability

Theorem (Quasi-decidability)

Let $\varphi = \varphi(\mathbf{p})$ be a CSL formula, with constants $\mathbf{p} = (p_1, \ldots, p_k) \in [0,1]^k$ appearing in until formulae.

The CSL model checking for ICTMC problem is decidable for $\mathbf{p} \in E$, where E is an open subset of $[0,1]^k$, of measure 1.

Convergence of CSL truth

- Consider convergence of CSL properties: will properties that are true in the approximating ICTMC z_k eventually be true in the original full CTMC $Y_k^{(N)}$?

Asymptotic Correctness Theorem

Let $\varphi = \varphi(\mathbf{p})$ be a CSL formula, with constants $\mathbf{p} = (p_1, \ldots, p_k) \in [0,1]^k$ appearing in until formulae.

Then, for $\mathbf{p} \in E$, an open subset of $[0,1]^k$ of measure 1, there exists N_0 such that $\forall N \geq N_0$

$$s, 0 \models_{Y_k^{(N)}} \varphi \Leftrightarrow s, 0 \models_{z_k} \varphi.$$

Outline

Conclusions

- We have developed an application of mean field theory to model check properties of single agents in a large population.

Conclusions

- We have developed an application of mean field theory to model check properties of single agents in a large population.
- We focussed on CSL, providing a method to model check CSL formulae against time-inhomogeneous continuous time Markov chains.

Conclusions

- We have developed an application of mean field theory to model check properties of single agents in a large population.
- We focussed on CSL, providing a method to model check CSL formulae against time-inhomogeneous continuous time Markov chains.
- We have provided convergence results that guarantee almost sure consistence of the method.

Thank you!

Thank you!

quanticol

www.quanticol.eu