Fitting General Distributions within PEPA Terms

Nil Geisweiller
Office National d'Études et de Recherches Aérospatiales
Toulouse, France
Email: nil.geisweiller@cert.fr

***Abstract*— This paper presents an algorithm to fit distribution functions or data sets within PEPA terms. The advantages that motivate the author for fitting distributions within PEPA terms instead of Markov chains are :**

1) **It is easier to define prior knowledge on the structure of the model and therefore to reuse and handle it better.**
2) **PEPA is a well known and efficient algebra to design stochastic communication systems recognised by the probabilistic model checker PRISM and other performance evaluation tools.**

The given algorithm is based on the expectation-maximisation method (EM) and can be seen as an extension of the work of Soren Asmussen, Olle Nerman and Marita Olsson on the fitting procedure of phase-type distributions.

I. Introduction

Probabilistic model checking [3], [6], [7], [5] is a useful and versatile way to define performance constraints and check them automatically without any need of developing specific mathematical analysis. Apart from statistical model checking [16] the models that are checkable by the current model checkers must be Markovian[1] or semi-markovian[2] [12] but not yet generalised semi-markovian[3]. In spite of this limitation it is possible to represent a wide variety of complex behaviours using only markovian process, most of the time by adding a part of the process' history in its current state. Following the same idea it is also possible to represent non-exponential distributions describing the probability for taking a transition of a continuous-time Markov process by representing a single transition by a sub-Markov chain, which is called phase-type distribution [13]. So it is possible to insert quite realistic distributions in a markovian model. Furthermore there exist algorithms that attempt to find the best values of a phase-type distribution in order to fit data sets or distribution functions [4], [14]. However when the distribution to approach is the result of a set of components that the designer wants to model separately, these methods are not adapted because the obtained Markov chain is quite intricate and hard to analyse.

A. Overview of the Main Result

To address this difficulty we propose in this paper to extend the algorithm of Soren Asmussen, Olle Nerman and Marita Olsson [4] in a way so that it can work properly over a PEPA model, PEPA [11] for *Performance Evaluation Process Algebra* is a stochastic process algebra that permits to describe a CTMC in a compositional way. The goal of this method it to improve the realism of a model according to absorption durations gotten from the real system while keeping the model under a high level description, that is under a form of a PEPA program. Our algorithm takes in inputs 1) a PEPA model, 2) a set of absorption durations and returns the most (or almost the most) likely values of the unknown parameters of the model.

The difficulty is that when the PEPA model is translated to its underlying CTMC the unknown parameters, or say variables, are dispatched irregularly into the CTMC and blended with the other values and variables. To deal with these variables the algorithm has to consider the derivation graph of the PEPA model (see *Section* II-C for the definition of the derivation graph) and its underlying CTMC.

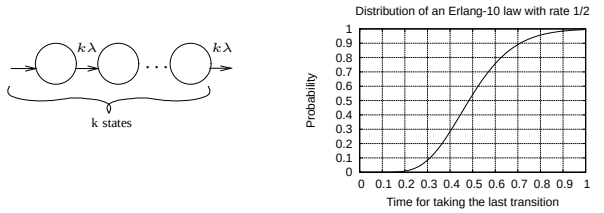


Fig. 1. *Erlang-k Distribution*

B. Outline

The paper is organised as follows : *Section* II recalls the definitions of phase-type distributions, PEPA and derivation graph. *Section* III gives an algorithm that addresses the presented issue. *Section* IV describes a possible use of this algorithm by extracting a model of the response time of a network without measuring it directly. Finally *Section* V concludes.

II. Basic Notions

A. Phase-Type Distributions

A *phase-type distribution*, PTD for short, is defined by a finite CTMC composed of several transient states (the phases) and one absorption state. It corresponds to the probability of being in the absorption state with respect to time. (see *Figures* 1, 2 and 3 for examples of special phase-type distributions called respectively Erlang, hyper-exponential and Coxian). *Figure* 4 is also a phase-type distribution, s_1 and s_2 are the phases and s_3 the absorption state.

We recall below the mathematical definition of a phase type distribution :

[1]memory-less property
[2]memory over a unique dimension of time
[3]memory over several dimensions of time

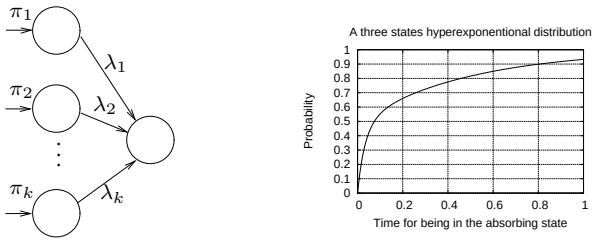


Fig. 2. *Hyper-exponential Distribution*

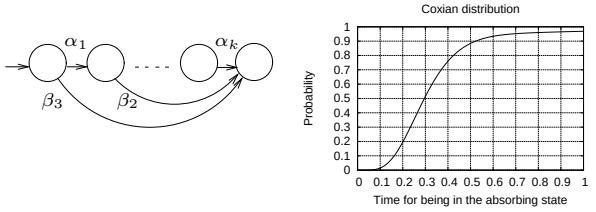


Fig. 3. *Coxian Distribution*

Definition 1: A phase-type distribution (PTD) is given by :

- S a finite set of states, including the absorption state, noted nil. S_{ph} is the set of phases, that is S without nil.
- $\mathbf{t} : S_{ph} \mapsto \mathbb{R}_+$ called the exit vector, is a column vector describing the density of going out of the set of phases.
- $\mathbf{T} : S_{ph} \times S_{ph} \mapsto \mathbb{R}$ the infinitesimal generator of phases, such that $\forall s \in S_{ph}$

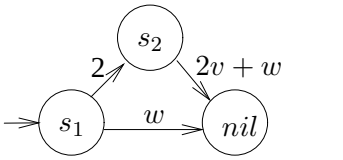
$$\mathbf{T}(s,s) = -\mathbf{t}(s) - \sum_{\substack{s' \in S_{ph}, \\ s' \neq s}} \mathbf{T}(s,s')$$

- $\pi : S_{ph}$ is the initial distribution such that :

$$\sum_{s \in S_{ph}} \pi(s) = 1$$

By default π is a row vector. If $\pi(s) = 1$ for some $s \in S_{ph}$ and $\pi(s') = 0$ for any $s' \neq s$, then we call s the initial state of the phase type distribution and we note π_s its initial distribution.

The infinitesimal generator, say $\mathbf{Q}$, of the CTMC embodying a phase type distribution with $\mathbf{T}$ as infinitesimal generator of phases and $\mathbf{t}$ as exit vector will be noted $\mathbf{Q} = \begin{pmatrix} \mathbf{T} & \mathbf{t} \\ 0,\ldots,0 & 0 \end{pmatrix}$.



$$\begin{pmatrix} -2+w & 2 & w \\ 0 & -2v-w & 2v+w \\ 0 & 0 & 0 \end{pmatrix}$$

Fig. 4. *On the up side is a CTMC with an absorption state, that is a phase type distribution, on the down side is its infinitesimal generator.*

B. PEPA with Variables

PEPA for *Performance Evaluation Process Algebra* is a stochastic process algebra defined by Jane Hillston [11] which allows to describe any finite[4] CTMC in terms of components executing actions and interacting together.

A PEPA model is composed by a set of declarations of PEPA terms and a root term defining the initial state of the process.

For the purpose of the paper a set of variables V is added in the definition of PEPA. V represents the set of unknown variables of the fitting problem. We also define $ev : V \mapsto \mathbb{R}_+$ an evaluation function of V. When it will be clear that we are speaking about the value of v instead of the variable itself the function ev will be omitted. The idle process nil has been also added to emphasise the definition of the PEPA model as a phase-type distribution.

Definition 2: Let $\mathcal{A}$ be a set of action names, or say actions, V a set of variables. τ represents the invisible action and $\top$ the unknown rate. Below is the definition of the rules defining the syntax of the PEPA terms and an informal description of its semantic :

$$P ::= (\alpha, r).P | P \underset{L}{\bowtie} P | P + P | P/L | A | nil$$

- $(\alpha, r).P$ is the *Prefix* operator, α is an action name, r the transition rate and P the description of the process after having taken the transition. In this paper a transition can be a positive real, the unknown rate or an element of V, that is $r \in V \cup \mathbb{R}^+ \cup \{\top\}$.
- $P \underset{L}{\bowtie} Q$, the *cooperation* operator, is the synchronised product between P and Q over the set of cooperation actions $L \subseteq \mathcal{A} \backslash \{\tau\}$.
- $P + Q$ is the *choice* operator.
- P/L is the hiding operator which makes unavailable the interactions belonging to L between P and the other processes.
- A is a *constant* declared by $A \overset{def}{=} P$.
- nil is the idle process. Note that nil is not a necessary key word to defined the absorption state in PEPA, instead it shall be seen as a macro to a term like $(\alpha, 1).P \underset{\{\alpha,\beta\}}{\bowtie} (\beta, 1).P$.

The underlying CTMC of a PEPA model is gotten by two steps :

1) Obtaining the *derivation graph* induced by the *Operational Semantic* of PEPA, a set of rules given in a Plotkin style [11].
2) Obtaining the CTMC from the *derivation graph* by deleting the action names from the transitions and lumping the equivalent ones.

Example 3: An example of a PEPA model is given below, the root term is $Model$:

[4]and also a subclass of infinite CTMCs

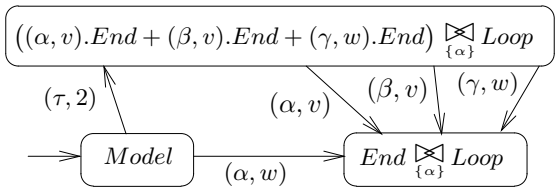


Fig. 5. *Example of a derivation graph. The underlying CTMC is the same as the one given in* Figure *4. Let's describe the correspondences between them : the state* $Model$ *becomes* s_1*, the state* $\big((\alpha,v).End + (\beta,v).End + (\gamma,w).End\big) \underset{\{\alpha\}}{\bowtie} Loop$ *becomes* s_2 *and the state* $End \underset{\{\alpha\}}{\bowtie} Loop$ *becomes the absorption one, that is* nil*. The transition* $(\tau,2)$ *becomes the one rated 2 and the transition* (α,w) *becomes the one rated* w*. The three transitions* (α,v)*,* (β,v) *and* (γ,w) *become the one rated* $2v+w$ *after aggregating them.*

$$
\begin{aligned}
Model &\overset{def}{=} Begin \underset{\{\alpha\}}{\bowtie} Loop \\
Begin &\overset{def}{=} (\tau,2).\big((\alpha,v).End + (\beta,v).End + (\gamma,w).End\big) \\
&+ \ (\alpha,w).End \\
Loop &\overset{def}{=} (\alpha,\top).Loop \\
End &\overset{def}{=} nil
\end{aligned}
$$

C. The Derivation Graph of a PEPA model

Let V be the set of variables of the fitting problem of a PEPA model.

Definition 4: A *derivation graph* is an oriented labelled multi-graph $(S, Act, s_\star, T)$:

- S is a set of states, every element of S is a PEPA term,
- Act is a set of action names, or say actions,
- $s_\star$ is the root term,
- T is a multi-set of transitions $\zeta : S \times S \mapsto Act \times (\mathbb{R}_+ \cup V \cup \{\top\})$

We have also to assume that the derivation graph we work on have an absorption state[5], this absorption state could be nil or $nil \underset{\emptyset}{\bowtie} nil$ or $(\alpha,1).P \underset{\{\alpha,\beta\}}{\bowtie} (\beta,1).P$ or any state which does not have any outgoing transition. We will now introduce some notations :

- $rate : T \mapsto \mathbb{R}_+ \cup V \cup \{\top\}$ denotes the rate of any transition of T.
- $\mathbb{T} : V \mapsto \mathcal{P}(T)$ defines for all $v \in V$ the multi-set $\mathbb{T}(v)$ that contains all transitions of rate v.

$$\forall v \in V \quad \mathbb{T}(v) = \{\zeta \in T \mid rate(\zeta) = v\}$$

The *Figure* 5 gives the derivation graph of the PEPA model given in the example 3. For a full explanation on how to derive a derivation graph from a PEPA model and the underlying CTMC from a derivation graph the reader is invited to consult [11].

We also define an infinite CTMC[6] $\mathcal{X} = (\mathcal{X}_t)_{t\in\mathbb{R}_+}$ associated with the derivation graph, the purpose of this CTMC is only theoretical and will be enlightened later. The state space of $\mathcal{X}$ is $S \times T \times \mathbb{N}$ and we define three additional stochastic processes in order to represent each component of the state space :

- $(X_t)_{t\in\mathbb{R}_+}$ denotes the state of the PEPA process at the time t, remark that $(S_t)_{t\in\mathbb{R}_+}$ is actually what is called the *underlying* CTMC of a PEPA model,
- $(T_t)_{t\in\mathbb{R}_+}$ denotes the last transition it has taken before or during t,
- $(C_t)_{t\in\mathbb{R}_+}$ denotes the number of transitions it has taken since its start.

III. The EM Algorithm within PEPA Terms

A. Overview of the EM Algorithm

The EM algorithm *(Expectation Maximisation)* [9], [15] is an iterative method to find parameters in a probabilistic model that maximise -or almost maximise- the likelihood of a data set. It is useful when the data is composed of partial observations and that it is not possible to directly maximise the likelihood. Here a complete observation, noted x, is the trajectory of the process defined by a PEPA model. However we have only a partial observation of x which is the time for the process to reach the absorption state nil, noted y. Let $\mathcal{X} = (\mathcal{X}_t)_{t\in\mathbb{R}_+}$ be an infinite CTMC associated with a PEPA model as defined in *Section* II-C and Y be a random variable denoting the absorption time of $\mathcal{X}$. Let χ be the product of n independent processes $\mathcal{X}$ and $\mathbf{Y}$ their absorption times. Let $\mathbf{x} = x_1, \ldots, x_n$ be the set of the unobserved trajectories and $\mathbf{y} = y_1, \ldots, y_n$ the associated observed durations. Since we do not know the trajectories associated to the observed durations it is not possible to directly calculate the parameters of the model (the variables of the PEPA model) which maximise the likelihood of $\mathbf{Y} = \mathbf{y}$. However it is possible to calculate the expectation of some relevant random variables (given an initial choice of the parameters of the model) and then maximise the likelihood of these expectations. Since they are not real observations but only expectations that depend of the current chosen model, it is possible to improve the current likelihood by repeating that process until a maximum (possibly local) is reached.

B. The Expectation Step

For any $v \in V$ let S_v be the set of states that have at least one outgoing transition of rate v :

$$\forall v \in V,\ S_V = \{s \in S_{ph} | \exists \zeta \in T,\ rate(\zeta) = v\}$$

Let $S_V \subseteq S_{ph}$ be the set of states that have at least one outgoing transition of some rate $v \in V$:

$$S_V = \bigcup_{v\in V} S_v$$

In this problem the relevant random variables associated with $\mathcal{X}$ are Z_v and N_v with $v \in V$. Z_v is the total time spent in all states $s \in S_v$ and N_v is the total number of times the transitions belonging to $\mathbb{T}(v)$ have been taken. Let $H =$

[5] if it has several we can just lump them together to get only one

[6] the reader has to be careful not to confuse this infinite CTMC with what is called the *underlying* CTMC of the derivation graph

$(Z_v, N_v)_{v\in V}$ be a family of random variables associated with $\mathcal{X}$, a value of H will be noted h.

The computation of the conditional expectations of Z_v and N_v knowing an observation y is given below (see *Appendix* A and B for their mathematical derivations) :

$$\mathbb{E}[Z_v|Y=y] = \frac{\sum_{s\in S_v} c_s^s(y)}{\pi_{s_\star}\mathbf{b}(y)}$$

$$\mathbb{E}[N_v|Y=y] = \frac{\sum_{\zeta\in\mathbb{T}(v)} ev(v)}{\pi_{s_\star}\mathbf{b}(y)} \times \begin{cases} c_{\zeta\bullet}^{\bullet\zeta}(y) & \text{if } \zeta\bullet \neq nil \\ a_{\zeta\bullet}(y) & \text{if } \zeta\bullet = nil \end{cases}$$

Where $\mathbf{a}$, $\mathbf{b}$ and $(\mathbf{c}^s)_{s\in S_{ph}}$ are $|S|+1$ vectors of functions defined from $\mathbb{R}_+$ to $\mathbb{R}_+^{S_{ph}}$:

$$\mathbf{a}(t) = \pi_{s_\star} e^{t\mathbf{T}}$$

$$\mathbf{b}(t) = e^{t\mathbf{T}}\mathbf{t}$$

$$\forall s \in S_{ph} \quad \mathbf{c}^s(t) = \int_0^t \pi_{s_\star} e^{u\mathbf{T}} \pi_s' e^{(t-u)\mathbf{T}}\mathbf{t}\,du$$

A component $s \in S_{ph}$ of a vector function, $\mathbf{a}(t)$ for instance, is noted $a_s(t)$. π_s' denotes the vector π_s transposed. In [4] it is shown that $\mathbf{a}$, $\mathbf{b}$ and $(\mathbf{c}^s)_{s\in S_{ph}}$ define three ordinary differential equation systems solvable using, in their case, the Runge-Kutta method in the fourth order.

We define now $\mathbf{Z}_v$ and $\mathbf{N}_v$ for all $v \in V$ which are respectively the total time spent in S_v of all processes of χ and the total number of times the transitions of rate v have been taken by all processes of χ. $\mathbf{Z}_v = n \times Z_v$ and $\mathbf{N}_v = n \times N_v$ since the n processes of χ are identical. We define the family of random variable $\mathbf{H} = (\mathbf{Z}_v, \mathbf{N}_v)_{v\in V}$, a value of $\mathbf{H}$ is noted $\mathbf{h}$. The expectations of $\mathbb{E}[\mathbf{Z}_v|\mathbf{Y}=\mathbf{y}]$ and $\mathbb{E}[\mathbf{N}_v|\mathbf{Y}=\mathbf{y}]$ are defined below :

$$\mathbb{E}[\mathbf{Z}_v|\mathbf{Y}=\mathbf{y}] = \sum_{k=1}^{n} \mathbb{E}[Z_v|Y=y_k]$$

$$\mathbb{E}[\mathbf{N}_v|\mathbf{Y}=\mathbf{y}] = \sum_{k=1}^{n} \mathbb{E}[N_v|Y=y_k]$$

C. The Maximisation Step

The goal of this step is to maximise the likelihood of $\mathbf{H}=\mathbf{h}$ when $\mathbf{h}$ contains the computed expectations of $\mathbf{Z}_v$ and $\mathbf{N}_v$, noted $\mathbf{z}_v$ and $\mathbf{n}_v$:

$$\mathbf{h} = (\mathbf{z}_v, \mathbf{n}_v)_{v\in V}$$

It is shown in [10] that maximising $L(\mathbf{H}=\mathbf{h})$ amounts to maximising the function below ($v_1, \ldots, v_m$ is the set of variables of V) :

$$L(v_1, \ldots, v_m) = \prod_{v\in V} e^{-v\times\mathbf{z}_v} \times v^{\mathbf{n}_v}$$

Then it is easy to verify (by deriving L and seeking its zeros) that $\forall v \in V$, $v = \frac{\mathbf{n}_v}{\mathbf{z}_v}$ maximises $L(v_1, \ldots, v_m)$. More intuitively the idea of the maximisation step consists of considering for any $v \in V$ a big transition, that is $\mathbb{T}(v)$, instead of a single transition (as in [4]) and a big state as well, S_v. The total number of times the big transition has been taken, that is $\mathbf{n}_v$ is divided by the total time spent in the big state, that is $\mathbf{z}_v$, the result of the division is the rate for taking v.

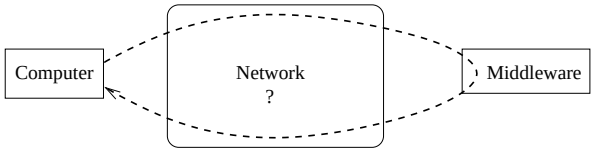


Fig. 6. *Scheme of the example*

D. Definition of the EM Algorithm for PTDV

The definition of the EM algorithm is now presented. Let L_{pre} and L_{new} be the logarithm of the likelihood $L(\mathbf{H}=\mathbf{h})$ for the previous and the current step of the algorithm. Let ϵ be the precision to be reached by the algorithm, that is $|L_{new} - L_{pre}| < \epsilon$. The algorithm is given below :

`Inputs : a data set and a derivation graph`
`Output : an evaluation function` ev

- `Choose an initialisation of` ev.
- `Initialise` L_{pre} `and` L_{new} `by` $-\infty$.
- `Until` $|L_{new} - L_{pre}| < \epsilon$ `do` :
  1) `E step` :
     - $L_{pre} := L_{new}$
     - $\forall\, v \in V$ `compute the expectations of` $\mathbf{Z}_v$ `and` $\mathbf{N}_v$, `noted` $\mathbf{z}_v$ `and` $\mathbf{n}_v$.
     - `Compute` L_{new}
  2) `M step` : $\forall v \in V\ ev(v) := \frac{\mathbf{n}_v}{\mathbf{z}_v}$.
- `Return` ev.

E. Fitting Continuous Distributions

According to [4], the EM algorithm can be used to approximate continuous distributions in the same way as data sets. See [4] and [10] for more details.

The next section gives an example of its use based on a real data set obtained from CERTI [8] an implementation of the real-time simulation architecture HLA [1], [2].

IV. Example : Extracting a Model of a Network

This section illustrates a possible use of the given algorithm through a simple example.

A. Description of the Problem

Let's take a system composed of three components, a computer, a network and another computer hosting a middleware, see *Figure* 6. The data set is obtained by sending messages of a fixed length from the computer to the network and measuring their echo response durations. Every message must :

1) cross the network to the middleware,

2) be processed by the middleware,
3) cross the network back to the computer.

The goal in this example is to obtain a good approximation of the response time of the network, knowing that :

1) we already have a prior model of the time response of the middleware,
2) we assume that crossing the network from the computer to the middleware or from the middleware to the computer gives equivalent behaviours.

We will first give a simplified description of the PEPA model that codes the behaviour of a message then present some results obtained with a more complex PEPA model and from a data set obtained from measurements of messages that has been used in the communication of a real-time distributed simulation.

Example 5: Let $System$ be the root term of the model. Let $V = v_1, \ldots, v_5$ be the set of variables of the fitting problem. $System$ is composed by the synchronised product of the components $NetworkToMid$, $Middleware$ and $NetworkFromMid$.

The terms $NetworkToMid$ and $NetworkFromMid$ denote a Coxian distribution with 3 states of phases. The terms $NTR1$ to $NTR3$ and $NFR1$ to $NFR3$ are used to express the phases of the Coxian distribution of the model of the network respectively from the computer to the middleware and from the middleware to the computer. The variables v_1 to v_5 are both used in $NetworkToMid$ and $NetworkFromMid$ in order to represent their identity of behaviour. The term $Middleware$ is an Erlang-4 distribution with rate 4λ where λ is a known value. There are two action names $inMid$ and $outMid$ defining respectively the interactions of the message going out of the network to the middleware and going out of the middleware to the network. When the message is in the wires of the network or in the silicium of the computer hosting the middleware the invisible action τ is used to model the internal changes.

$$
\begin{array}{rcl}
System & \stackrel{def}{=} & NetworkToMid \\
 & \bowtie_{\{inMid\}} & Middleware \\
 & \bowtie_{\{outMid\}} & NetworkMiddleware \\
 & & \\
NetworkToMid & \stackrel{def}{=} & NTR1 \\
NTR1 & \stackrel{def}{=} & (\tau, v_1).NTR2 \\
 & + & (outMid, v_2).nil \\
NTR2 & \stackrel{def}{=} & (\tau, v_3).NTR3 \\
 & + & (outMid, v_4).nil \\
NTR3 & \stackrel{def}{=} & (outMid, v_5).nil \\
 & & \\
Middleware & \stackrel{def}{=} & (inMid, \top).(\tau, \lambda).(\tau, \lambda) \\
 & & .(\tau, \lambda).(outMid, \lambda).nil \\
 & & \\
NetworkFromMid & \stackrel{def}{=} & (outMid, \top).NFR1 \\
NFR1 & \stackrel{def}{=} & (\tau, v_1).NFR2 \\
 & + & (\tau, v_2).nil \\
NFR2 & \stackrel{def}{=} & (\tau, v_3).NFR3 \\
 & + & (\tau, v_4).nil \\
NFR3 & \stackrel{def}{=} & (\tau, v_5).nil
\end{array}
$$

B. Results

The same model, but with 16 states for the network and 64 states for the middleware, has been tested with EMPEPA[7], a program developed by the author implementing the presented algorithm. The data set consists of 10000 absorption durations obtained by measuring the transmission time between two simulators during a distributed simulation execution.

1) Getting a Model of the Middleware: The algorithm has first been used to find an approximation of the model of the middleware, an Erlang-64 distribution. After 4 iterations[8] (a couple of hours) the algorithm converged toward the right value, see *Figure* 7.

2) Getting a Model of the Network: Then the complete model of the system has been given to the algorithm, the set of variables of the problem consists of 31 variables defining the Coxian distribution of $NetworkToMid$ and $NetworkFromMid$. The *Figure* 8 shows the result of the obtained model after 180 iterations (about 2 days of computation). The model of the network has been extracted very easily, by replacing the variables by the obtained values in the PEPA term $NetworkToMid$ or -which is the same- $NetworkFromMid$. *Figure* 9 shows the distribution of the one way response time of the network. *Figure* 10 shows the different obtained distributions together.

V. Conclusion and Possible Future Improvements

In this paper an algorithm based on the EM method has been presented to fit the parameters of a PEPA model according to

[7] http://sourceforge.net/projects/empepa

[8] The small number of iterations is due to the fact that the middleware is modelled by an Erlang distribution and with only one variable, λ. In theory only one iteration should suffi ce, perhaps this is the matter of some numerical imprecision in the fi rst iterations.

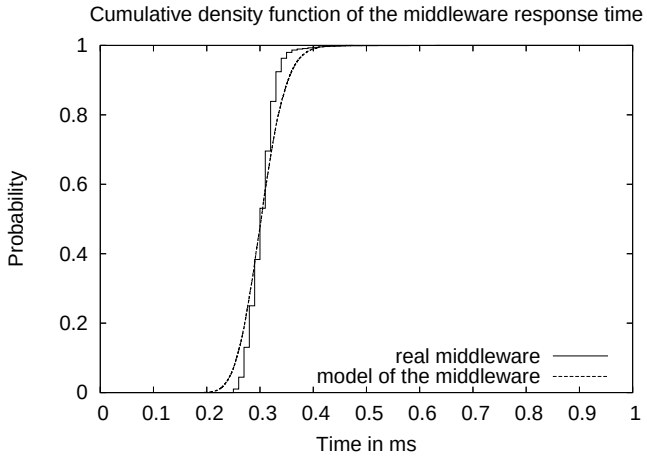


Fig. 7. *Cumulative density function of the middleware respond time*

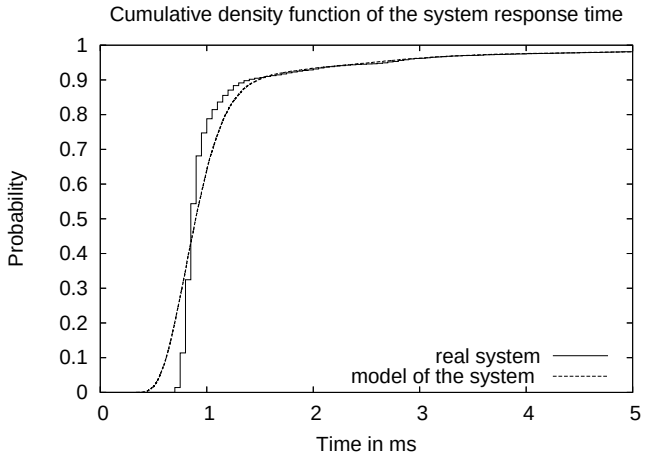


Fig. 8. *Cumulative density function of the whole system, that is the duration for a message to cross the network one way, being processed by the middleware and come back.*

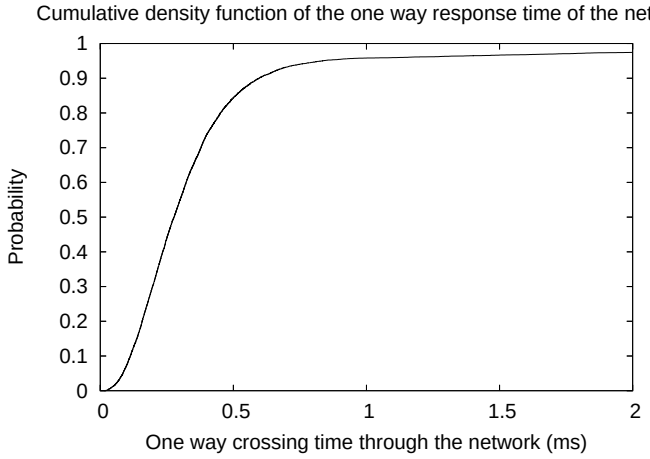


Fig. 9. *Distribution of one way crossing time of the network*

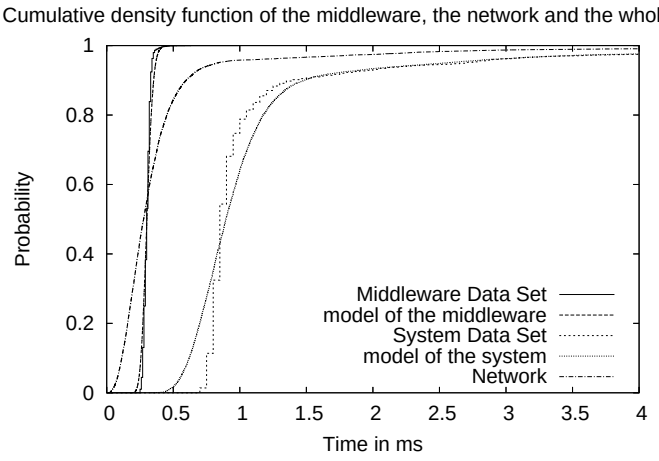


Fig. 10. *Recapitulating graph*

data sets or distribution functions. A simple example has been given to illustrate the utility of this algorithm by extracting the unknown values of a component modelling a one-way network. The present algorithm only works on the PEPA terms which have their interactions between passive and active transitions and with data set composed of absorption durations. Further improvements could be to make it work on the whole PEPA class and to consider partial observations of executions instead of simple absorption time observations. This study also leads to another more interesting and difficult issue which is the problem of finding the most likely PEPA model that explains a set of executions.

VI. Acknowledgement

Special thanks to Bruno d'Ausbourg and Joachim Reigle.

References

[1] Defense modeling and simulation office, high level architecture run-time infrastructure programmer's guide 1.3 version 5, rti 1.3 distribution. Technical report, Departement of Defense, 1998.

[2] Departement of defense (u.s), high level architecture interface specification, version 1.3. Technical report, Departement of Defense, 1998.

[3] A. Aziz, K. Sanwal, V. Singhal, and R. K. Brayton. Verifying continuous-time markov chains. In Rajeev Alur and Thomas A. Henzinger, editors, *Eighth International Conference on Computer Aided Verification CAV*, volume 1102, pages 269–276, New Brunswick, NJ, USA, / 1996. Springer Verlag.

[4] S. Asmussen, O. Nerman, and M. Olsson. Fitting phase-type distribution via the em algorithm. *Scandinavian Journal of Statistics*, 23:419–441, 1996.

[5] C. Baier, B. Haverkort, H. Hermanns, and J. Katoen. Model checking continuous-time markov chains by transient analysis. In *Proc. CAV*, 2000.

[6] C. Baier, J.-P. Katoen, and H. Hermanns. Approximate symbolic model checking of continuous-time markov chains. In *Proc. 10th International Conference on Concurrency Theory (CONCUR'99)*, volume 1664 of *Lecture Notes in Computer Science*, pages 146–161, 1999.

[7] C. Baier and M. Z. Kwiatkowska. Model checking for a probabilistic branching time logic with fairness. *Distributed Computing*, 11(3):125–155, 1998.

[8] B. d'Ausbourg and P. Siron. Certi : Evolutions of the onera rti prototype. In *Fall 2002 Simulation Interoperability Workshop*, 2002.

[9] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. Roy. Soc. Ser. B.*, 39:1–38, 1977.

[10] N. Geisweiller. Thèse sur la v´erifi cation et la mod´elisation probabiliste de systèmes complexes pour l'´evaluation de performances. onera, directeur : Bruno d'ausbourg, 2005.

[11] J. Hillston. *A Compositional Approach to Performance Modelling*. Cambridge University Press, 1996.

[12] G. G. I. Lopez, H. Hermanns, and J.-P. Katoen. Beyond memoryless distributions: Model checking semi-markov chains. In *PAPM-PROBMIV*, pages 57–70, 2001.

[13] M. Neuts. Matrix-geometric solution in stochastic models : An algorithmic approach. *The Johns Hopkins University Press*, 1981.

[14] A. Riska, V. Diev, and E. Smirni. An em-based technique for approximating long-tailed data sets with ph distributions. *Performance Evaluation*, 55:147–164, 2004.

[15] C. Wu. On the convergence properties of the em algorithm. *Ann. Statist.*, 11:95–103, 1983.

[16] H. L. S. Younes and R. G. Simmons. Probabilistic verifi cation of discrete event systems using acceptance sampling. In E. Brinksma and K. G. Larsen, editors, *Proceedings of the 14th International Conference on Computer Aided Verification*, volume 2404 of *Lecture Notes in Computer Science*, pages 223–235, Copenhagen, Denmark, July 2002. Springer.

APPENDIX

A. Conditional Expectation of the Sojourn Time in S_v

$$\mathbb{E}[Z_v|Y=y] = \int_0^\infty \mathbb{P}(X_t \in S_v | Y \in dy)$$

After y the process is in the absorption state nil, since $nil \notin S_V$ we can consider the integral between 0 and y.

$$\begin{aligned}\mathbb{E}[Z_s \mid Y=y] &= \int_0^y \mathbb{P}(X_t \in S_v \mid Y=y)dt \\ &= \int_0^y \sum_{s\in S_v} \mathbb{P}(X_t = s \mid Y=y)dt\end{aligned}$$

The next equation is obtained by using the Markov property :

$$\mathbb{P}(X_t = s \mid Y=y) = \frac{\mathbb{P}(X_t=s)\mathbb{P}(Y\in dy \mid X_t = s)}{\mathbb{P}(Y\in dy)}$$

Then we can replace the probabilities by their mathematical expressions using the definition of a Markov chain :

$$\mathbb{P}(X_t = s \mid Y=y) = \frac{\pi_{s_\star} e^{t\mathbf{T}} \pi'_s \pi_s e^{(y-t)\mathbf{T}}\mathbf{t}dt}{\pi_{s_\star} e^{y\mathbf{T}}\mathbf{t}}$$

Finally :

$$\begin{aligned}\mathbb{E}[Z_s \mid Y=y] &= \frac{\sum_{s\in S_v}\pi_s \int_0^y \pi_{s_\star} e^{t\mathbf{T}}\pi'_s e^{(y-t)\mathbf{T}}\mathbf{t}dt}{\pi_{s_\star}e^{y\mathbf{T}}\mathbf{t}} \\ &= \frac{\sum_{s\in S_v} c_s^s(y)}{\pi_{s_\star}\mathbf{b}(y)}\end{aligned}$$

B. Conditional Expectation of the Total Number of Times $\mathbb{T}(v)$ Has Been Taken

Let N_ζ for any $\zeta \in T$ be a random variable that defines the number of times the process $\mathcal{X}$ has taken the transition ζ. We assume first that $\zeta\bullet \neq nil$, and we define $N_\zeta^{\Delta t}$ a discreet version of N_ζ with the resolution Δt.

$$\begin{aligned}\mathbb{E}[N_\zeta^{\Delta t}|Y=y] &= \sum_{i=0}^{[y/\Delta t]-1} \mathbb{P}(C_{i\Delta t+\Delta t} \neq C_{i\Delta t}, T_{i\Delta t+\Delta t} = \zeta | Y\in dy) \\ &= \sum_{i=0}^{[y/\Delta t]-1} \frac{\mathbb{P}(C_{i\Delta t+\Delta t} \neq C_{i\Delta t}, T_{i\Delta t+\Delta t} = \zeta, Y\in dy)}{\mathbb{P}(Y\in dy)} \\ &= \sum_{i=0}^{[y/\Delta t]-1} \frac{\mathbb{P}(X_{i\Delta t+\Delta t} = \zeta\bullet, X_{i\Delta t} = \bullet\zeta, C_{i\Delta t+\Delta t} \neq C_{i\Delta t}, T_{i\Delta t+\Delta t} = \zeta, Y\in dy)}{\mathbb{P}(Y\in dy)}\end{aligned}$$

Then we use the Markov property of $\mathcal{X}$:

$$\begin{aligned}&= \sum_{i=0}^{[y/\Delta t]-1} \frac{\mathbb{P}(X_{i\Delta t} = \bullet\zeta)\mathbb{P}(X_{i\Delta t+\Delta t} = \zeta\bullet, C_{i\Delta t+\Delta t} \neq C_{i\Delta t}, T_{i\Delta t+\Delta t} = \zeta | X_{i\Delta t} = \bullet\zeta)\mathbb{P}(Y\in dy \mid X_{i\Delta t+\Delta t} = \zeta\bullet)}{\mathbb{P}(Y\in dy)} \\ &= \sum_{i=0}^{[y/\Delta t]-1} \frac{\sum_{\zeta\in\mathbb{T}(v)}(\pi_{s_\star}e^{i\Delta t\mathbf{T}}\pi'_{\bullet\zeta})(ev(rate(\zeta))\Delta t + o(\Delta t))(\pi_{\zeta\bullet}e^{(y-i\Delta t-\Delta t)\mathbf{T}}\mathbf{t})}{\mathbb{P}(Y\in dy)}\end{aligned}$$

Finally we consider the limit of Δt toward 0 :

$$\begin{aligned}\mathbb{E}[N_\zeta|Y=y] &= \frac{\int_0^y \pi_{s_\star}e^{t\mathbf{T}}\pi'_{\bullet\zeta}ev(rate(\zeta))\pi_{\zeta\bullet}e^{(y-t)\mathbf{T}}\mathbf{t}dt}{\mathbb{P}(Y\in dy)} \\ &= \frac{ev(rate(\zeta))\pi_{\zeta\bullet}\int_0^y \pi_{s_\star}e^{t\mathbf{T}}\pi'_{\bullet\zeta}e^{(y-t)\mathbf{T}}\mathbf{t}dt}{\mathbb{P}(Y\in dy)} \\ &= \frac{c_{\zeta\bullet}^{\bullet\zeta}(y)}{\pi_{s_\star}\mathbf{b}(y)}\end{aligned}$$

For the case $\zeta\bullet = nil$, the conditional expectation of the number of times ζ has been taken amounts to consider the conditional probability that ζ is the last incoming transition in the absorption state at time y :

$$\mathbb{E}[N_\zeta \mid Y=y] = \mathbb{P}(T_{y+\epsilon} = \zeta \mid Y=y) \tag{1}$$

The next equation is obtained by using the Markov property :

$$\begin{aligned}&= \frac{\mathbb{P}(X_y = \bullet\zeta)\mathbb{P}(Y\in dy \mid X_y = \bullet\zeta, T_{y+\epsilon}=\zeta)}{\mathbb{P}(Y\in dy)} \\ &= \frac{(\pi e^{y\mathbf{T}}\pi'_s)ev(rate(\zeta))}{\pi e^{y\mathbf{T}}\mathbf{t}} \\ &= \frac{ev(rate(\zeta))a_s(y)}{\pi\mathbf{b}(y)}\end{aligned}$$

It follows that :

$$\mathbb{E}[N_v|Y=y] = \frac{\sum_{\zeta\in\mathbb{T}(v)} ev(v)}{\pi_{s_\star}\mathbf{b}(y)} \times \begin{cases} c_{\zeta\bullet}^{\bullet\zeta}(y) & \text{if } \zeta\bullet \neq nil \\ a_{\zeta\bullet}(y) & \text{if } \zeta\bullet = nil \end{cases}$$