A numerical algorithm for the decomposition of cooperating structured Markov processes

Andrea Marin
DAIS
Università Ca' Foscari Venezia
via Torino, 155
Venezia - Italy
Email: marin@dais.unive.it

Samuel Rota Bulò
DAIS
Università Ca' Foscari Venezia
via Torino, 155
Venezia - Italy
Email: srotabul@dais.unive.it

Simonetta Balsamo
DAIS
Università Ca' Foscari Venezia
via Torino, 155
Venezia - Italy
Email: balsamo@dais.unive.it

***Abstract*—Modern computer systems consist of a large number of dynamic hardware and software components that interact according to some specific rules. Quantitative models of such systems are important for performance engineering because they allow for an earlier prediction of the quality of service. The application of stochastic modelling for this purpose is limited by the problem of the explosion of the state space of the model, *i.e.* the number of states that should be considered for an exact analysis increases exponentially and is thus huge even when few components are considered. In this paper we resort to product-form theory to deal with this problem. We define an iterative algorithm with the following characteristics: a) it deals with models with infinite state space and block regular structure (e.g. quasi-birth&death) without the need of truncation; b) in case of detections of product-form according to RCAT conditions, it computes the exact solution of the model; c) in case of non-product-form, it computes an approximate solution. The very loose assumptions allow us to provide examples of analysis of heterogeneous product-form models (e.g., consisting of queues with catastrophes and/or batch removals) as well as approximating non-product-form models with non-exponential service time distributions and negative customers.**

I. Introduction

Stochastic modelling of computer and communication systems has shown its importance in performance engineering by defining formalisms and analysis techniques capable of predicting the quality of service (QoS) and the reliability of hardware and software architectures in their earlier phase of design. Many formalisms, such as the Performance evaluation process algebra (PEPA) [1], stochastic Petri nets [2] or Markovian queueing networks [3], allow for the definition of models whose underlying stochastic processes are continuous time Markov chains (CTMC). In particular, process algebraic formalisms are appreciated for their formal semantics and compositionality properties that allow one to describe a complex system in a modular and hierarchical way. However, from the analysis prospective, all these approaches are limited by the problem of the state space explosion, i.e., in most of the cases, the state space of the model tends to grow exponentially with the number of its components. In order to overcome this problem, exact and approximated techniques have been proposed in literature. This paper focuses on the exact and approximate decomposition of stochastic models by resorting to product-form theory [3], [4], [5]. Consider a model consisting of M cooperating components $\mathcal{S} = \{1, \dots, M\}$ and let $\mathbf{m} = (m_1, \dots, m_M)$ be a vector describing its state, where m_i denotes the state of component i. Then the model is in product-form if its stationary state distribution $\pi(\mathbf{m})$ is proportional to the product of the stationary distributions of each component considered in isolation and opportunely parametrised $\pi^{(s)}(m_i)$, i.e.:

$$\pi(\mathbf{m}) \propto \prod_{s=1}^{M} \pi^{(s)}(m_s).$$

Assuming that each component has n states, and that the joint model state space is the Cartesian product of the state spaces of its components, a brute force approach, i.e., the solution of the global balance equations of the joint process, has a computational cost of $\mathcal{O}(n^{3M})$, while a product-form solution has a cost of $\mathcal{O}(Mn^3)$. Several research efforts have been devoted to characterising models with product-form solutions especially in the domain of queueing networks (e.g., [3], [4]). In [5] the compositional nature of process algebra is exploited to derive the Reversed Compound Agent Theorem (RCAT) which gives sufficient conditions for a cooperation of stochastic models to be in product-form. Similarly to [4] RCAT relies on the analysis of the reversed process of the components considered in isolation and this strongly limits its application from practitioners: reasoning on the reversed process and giving a physical interpretation to the product-form conditions may be a hard task. Moreover, from an algebraic point of view, in many relevant cases (e.g., [6], [7]) the computation of the stationary probabilities requires the solution of large systems of non-linear traffic equations.

a) Contributions: In this paper we present the Iterative Numerical Algorithm for Product-forms (INAP). The algorithm addresses the two problems concerning the analysis of product-form models, i.e., deciding if the product-form conditions are satisfied and computing the stationary distributions. As shown in [5], product-form conditions are in general not just structural but also depends on the transition rates in the reversed process of an isolated component. As a consequence, since deciding if a product-form exists requires the components to be parametrised and isolated, testing RCAT conditions is not a trivial task. The novel aspects of the algorithm are the following:

- It can study models with *infinite* state spaces with block regular structure as well as models with *finite* number of states as in [8][1]. This allows us to study a wide class of open models with different behaviours such as: G-queues with removal of batches of customers or flushing [7], G-queues with resets [9] and, more generally, queues with quasi birth&death (QBD) structure, queues with lower Hessenberg matrix structure. In dealing with unbounded models, we must also consider the problem of stability. This topic is considered in [10], however in this paper we resort to Neuts' drift condition [11] and Bini et. al work [12].
- One key point of our iterative algorithm consists in verifying the RCAT conditions in case of processes with Lower Hessenberg or QBD matrices as infinitesimal generator. We contribute to the state-of-the-art of product-form theory by deriving a closed-form matrix formula for this check, which avoids the computation of the reversed processes. Although these formulas are useful even outside the definition of INAP, they play an important role for the algorithm since they strongly increase its numerical stability and computational cost.
- When the algorithm reaches a fixed point, but the product-form conditions of RCAT cannot be verified, the computed solution can be interpreted as an approximation of the model's steady-state distribution. We will justify this claim in Section III, once the theoretical notions underlying the algorithm are introduced.

The main assumption required to apply INAP concerns the semantics of the cooperations among the components that must be pairwise. In other words, a transition of a component can trigger a transition in at most one other components (simultaneous state changes in more than two components are not considered).

b) Related works: Product-form theory has mainly been developed for queueing networks such as in [3], [6], [7] although some results have also been proved for process algebraic models [13], [5] and stochastic Petri nets [14], [15]. Ad-hoc algorithms for each of these formalisms have been proposed, however RCAT proved that working at a lower level of abstraction, i.e., by specifying the component cooperations as synchronising transitions of the Markov processes, allows for a heterogeneous model definition in product-form. Queueing Petri nets [16] is an example of such a heterogeneous modelling approach that may yield product-form stationary distribution. To the best of our knowledge, the most general algorithm for the detection and the computation of the solution of product-form models is that presented in [8]. The dynamics and convergence properties of the algorithm have experimentally shown to be robust, however its main limitation is that only models with a finite number of states can be studied, while many well-known classes of product-form models have an unbounded state-space (the topic is addressed also in [17], [18] for non-product-form models). In [19] a truncation mechanism is proposed, however this approach may reduce the solution accuracy and asks the modeller to decide how to truncate the components' state spaces. As concerns decomposition algorithms based on product-form approximations, some methods have been defined for the analysis of complex models in order to overcome the limitation due to the state space dimension, both for large product-form models and for non product-form models that can be approximated by a product-form solution. As in our case, most of the methods proposed in literature do not provide bounds on the approximation error. Some specific algorithms have been defined for the CTMCs underlying various models expressed in high-level formalisms, and they often take advantage of decompositions at the higher level of abstraction (e.g., decomposition and aggregation of queueing networks or stochastic Petri nets). In this field, Trivedi et al.'s fixed point iterations based on stochastic Petri nets have been widely explored (see [20] and the references therein). Fixed point iterations have shown their accuracy in practice in several works (see e.g. [21] and the references therein). With respect to these works, our approach is different because it is applied at a lower level of abstraction and hence can work with a wide class of possibly heterogeneous models. Moreover, its fixed point iterations are purely based on the analysis of the CTMC without being driven by modeller's intuition about the high-level behaviour. Concerning product-form decompositions for approximating stochastic models, a recent paper [22] introduces an algorithm that relies on the idea of minimising a non-convex function to derive the closest product-form distribution that approximates the correct distribution. The drawbacks of the approach are that they can be applied for models with finite state spaces and that the minimisation may fall into local minima resulting in very bad approximations.

The balance of the paper is as follows. Section II introduces the notation and briefly recalls the underlying theoretical results. Section III describes INAP while in Section IV we present the theoretical results about product-form form processes with block structured infinitesimal generators. Section V shows some numerical applications of INAP. Finally, Section VI concludes the paper with some final remarks.

II. Notation, definitions, background

In the paper we denote *scalar constants* using uppercase letters (*e.g.* N, T), (row) *vectors* using boldface lowercase symbols (*e.g.* $\boldsymbol{\pi}$, $\mathbf{e}$), *matrices* as uppercase teletype letters (*e.g.* $\mathtt{Q}$, $\mathtt{A}$) and *sets* using uppercase calligraphic (*e.g.* $\mathcal{S}$, $\mathcal{E}$). Any other entity different from the aforementioned ones will be denoted by lowercase symbols (*e.g.* $i, j, \alpha, \beta, \ell$). The sets of real and nonnegative real numbers are denoted by $\mathbb{R}$ and $\mathbb{R}_+$, respectively. The set of natural numbers is denoted by $\mathbb{N}$ and the set of natural numbers with the infinity symbol is denoted by $\mathbb{N}_\infty$.

[1] In the same paper, unbounded models are studied only if the process structures underlying the components are birth&death.

We denote by $\mathbf{e}$ a row vector of all 1s and by $\mathtt{I}$ the identity matrix. We define the operator $\mathrm{diag}(\mathbf{v})$, which produces a diagonal matrix with diagonal entries given by vector $\mathbf{v}$. Given a matrix $\mathtt{A}$, we denote by $\mathtt{A}^\top$ the transpose of $\mathtt{A}$ and, if $\mathtt{A}$ is square, we denote by $\overline{\mathtt{A}}$ the matrix defined as

$$\overline{\mathtt{A}} = \mathtt{A} - \mathrm{diag}(\mathbf{e}\,\mathtt{A}^\top)\,.$$

Let $\mathcal{S} = \{1, \cdots, M\}$ be the set of $M \in \mathbb{N}$ cooperating components that are defined as stochastic automata with possibly labelled transitions, with $\mathcal{L}$ a set of labels. A label $\ell \in \mathcal{L}$ is active in component $\alpha_\ell \in \mathcal{S}$ if all the rates of the transitions labelled ℓ are specified while it is passive in $\beta_\ell \in \mathcal{S}$ if there exists a probability assigned to each transition labelled ℓ. Intuitively, when automaton α_ℓ performs a transition labelled ℓ, then β_ℓ performs a given synchronising transition with the probability associated with it. We assume that the synchronising probability distributions for passive automata are well defined, and for each state they sum to 1, (i.e., a passive automaton cannot prevent an active to move). All the transitions are carried out in an exponentially distributed time (and this justifies the fact that the underlying process is a CTMC). Transitions labelled $\ell \in \mathcal{L}$ are carried out only jointly by α_ℓ and β_ℓ and this specifies how the synchronisation works. Observe that only pairwise synchronisations are allowed.

A *cooperation graph* is a multi-graph $g = (\mathcal{S}, \mathcal{E})$, where the set of components $\mathcal{S}$ corresponds to the set of vertices of the graph, and the set $\mathcal{E} = \{(\alpha_\ell, \beta_\ell)\}_{\ell \in \mathcal{L}}$ contains oriented edges indexed by labels in $\mathcal{L}$. Each label identifies a cooperation between two components in $\mathcal{S}$.

We denote by $\mathcal{A}_s^g$ ($\mathcal{P}_s^g$) the set of labels in $\ell \in \mathcal{L}$ identifying a cooperation in graph g where $s \in \mathcal{S}$ plays an active (passive) role:

$$\mathcal{A}_s^g = \{\ell \in \mathcal{L} \,|\, \alpha_\ell = s\} \;\text{ and }\; \mathcal{P}_s^g = \{\ell \in \mathcal{L} \,|\, \beta_\ell = s\}\,.$$

Hereafter, we write $\mathcal{A}_s$ ($\mathcal{P}_s$) for $\mathcal{A}_s^g$ ($\mathcal{P}_s^g$) when g can be inferred from the context without ambiguities.

Each component $s \in \mathcal{S}$ has a (possibly infinite) set $\mathcal{V}_s$ of states. Let $N_s = |\mathcal{V}_s| \in \mathbb{N}_\infty$ be the number of states of component s. Given a cooperation graph g, we have for each label $\ell \in \mathcal{L}$ a N_{α_ℓ}-dimensional matrix $\mathtt{T}^{(\ell)}$ specifying the rates of the active transitions with label ℓ of component α_ℓ, and a N_{β_ℓ}-dimensional stochastic matrix $\mathtt{J}^{(\ell)}$ indicating the corresponding passive transitions of component β_ℓ. Additionally, each component $s \in \mathcal{S}$ has an associated N_s-dimensional matrix $\mathtt{E}^{(s)}$ encoding rates of non cooperating transitions.

Observe that the matrices $\mathtt{T}^{(\ell)}$ and $\mathtt{E}^{(s)}$, for $\ell \in \mathcal{L}$ and $s \in \mathcal{S}$ have only non negative entries including the diagonal elements. This is due to the necessity of representing synchronising self-loop transition in states.

III. Algorithm definition

The algorithm we introduce in this section is based on a set of results pertaining to the study of stochastic models in product-form. Specifically, we consider the Reversed Compound Agent Theorem (RCAT) presented in [5], which states sufficient conditions for having a product-form solution in the joint state space of cooperating Markov processes.

Algorithm 1 INAP Algorithm: pseudo-code

```
Require: Cooperation graph g = (S, E)
  initialize the reverse rates {x_ℓ}_{ℓ∈L} (see §III-B)
  while not termination criterion met (see §III-E) do
    for all s ∈ S do
      compute Q^(s) according to (1)
      update the stationary distribution π^(s) (see §III-C)
    end for
    update the reverse rates {x_ℓ}_{ℓ∈L} (see §III-A)
  end while
  check convergence and product-form conditions (see §III-E)
  return {π^(s)}_{s∈S} and {x_ℓ}_{ℓ∈L}
```

Henceforth, we assume that the cooperating components forming a model have an underlying irreducible state space. Since we are considering also models with infinite state spaces, this condition is necessary but not sufficient for the ergodicity. Another important aspect to take into account is the structure of the joint process. Whenever the joint process' state space is the Cartesian product of the components' state spaces, then the probability obtained by INAP are normalised. In the other case (e.g., in case of closed queueing networks), this condition is not satisfied and the probability distributions computed by INAP can be interpreted as the un-normalised stationary probability for the ergodic states. According to RCAT, the steady-state distribution of the model is in product-form with respect to its component if there exists a set $\{x_\ell\}_{\ell \in \mathcal{L}}$ such that the infinitesimal generator $\overline{\mathcal{Q}}^{(s)}$ of each isolated component $s \in \mathcal{S}$ is determined by:

$$\mathtt{Q}^{(s)} = \mathtt{E}^{(s)} + \sum_{\ell \in \mathcal{A}_s} \mathtt{T}^{(\ell)} + \sum_{\ell \in \mathcal{P}_s} \mathtt{J}^{(\ell)} x_\ell \tag{1}$$

and the following relation holds:

$$x_\ell\,\mathbf{e} = \boldsymbol{\pi}^{(\alpha_\ell)} \mathtt{T}^{(\ell)} \mathrm{diag}(\boldsymbol{\pi}^{(\alpha_\ell)})^{-1}\,, \tag{2}$$

where $\boldsymbol{\pi}^{(s)}$ are obtained for all $s \in \mathcal{S}$ as the unique solution of:

$$\boldsymbol{\pi}_s \overline{\mathtt{Q}}^{(s)} = \mathbf{0} \qquad \text{and} \qquad \mathbf{e}\,\boldsymbol{\pi}_s^\top = 1\,. \tag{3}$$

We refer to x_ℓ as the *reverse rate* associated with the cooperation label $\ell \in \mathcal{L}$ [4], [5]. The basic idea of our INAP algorithm is to start with an initial set of reversed rates $\{x_\ell\}_{\ell \in \mathcal{L}}$, which are required to set up the system (3) for all $s \in \mathcal{S}$ and find the steady-state probabilities $\{\boldsymbol{\pi}^{(s)}\}_{s \in \mathcal{S}}$. These probabilities, in turn, are then used to update the reverse rates $\{x_\ell\}_{\ell \in \mathcal{L}}$. This is more formally depicted by Algorithm 1. We keep updating the steady-state probabilities and the reverse rates until convergence (see [8] for more details). In what follows, we discuss in detail each step of the algorithm and the optimisations that have been implemented.

A. Computation of the reversed rates x_ℓ

The first problem we address is how to compute the reversed rates in the isolated components because most of the algorithm definition relies on this aspect. The effective computation of the reversed rates according to Equation (2) is a non-trivial task and usually causes numerical stability problems due to some state stationary probabilities that may tend to 0. Indeed, Equation (2) involves a non-linear relation between the steady-state probabilities (which are, in general, unknown). Moreover, Equation (2) does not hold in case of models which are not in product-form. For these reasons, we compute x_ℓ as a weighted mean of the right-hand-side of (2), where the weights are given by the steady-state probability $\boldsymbol{\pi}^{(\alpha_\ell)}$. This is achieved by right-multiplying both sides of (2) by ${\boldsymbol{\pi}^{(\alpha_\ell)}}^\top$, which yields the following simplified formula

$$x_\ell = \boldsymbol{\pi}^{(\alpha_\ell)} \mathtt{T}^{(\ell)} \mathbf{e}^\top . \tag{4}$$

The rationale of this choice is that states with higher stationary probabilities affect the behaviour of the cooperations more than those with a lower one. Moreover, in case of product-form the expected reversed rate obviously corresponds to the constant reversed rate incoming to each state.

B. Initialising the reversed rates x_ℓ

The definition of the infinitesimal generators underlying the components relies on the knowledge of the synchronising transition reversed rates as shown by Equation (1). Therefore, the first step of Algorithm 1 is the initialisation of the reversed rates. The initialisation is done randomly within the set of *feasible reversed rates* whose definition follows.

Definition 1 (Feasible reverse rates)**.** *A set $\{x_\ell\}_{\ell \in \mathcal{L}}$ of reversed rates is feasible if the following conditions are satisfied:*

1) *For each $\ell \in \mathcal{L}$,*

$$\min\left(\mathtt{T}^{(\ell)}\mathbf{e}^\top\right) \le x_\ell \le \max\left(\mathtt{T}^{(\ell)}\mathbf{e}^\top\right), \tag{5}$$

where $\min$ and $\max$ assume the value of the minimum and maximum components of the argument vectors.

2) *For each model $s \in \mathcal{S}$, the infinitesimal generator $\overline{\mathtt{Q}}^{(s)}$ corresponds to an ergodic chain.*

Note that for an ergodic model s the expected reversed rate x_ℓ, $\ell \in \mathcal{A}_s$, as defined by Equation (4) must satisfy Inequality (5). Indeed, it suffices to observe that Equation (4) defines a convex linear combination of the entries of $\mathtt{T}^{(\ell)}\mathbf{e}^\top$. The same inequality trivially holds if the model satisfies RCAT product-form conditions since $\mathtt{T}^{(\ell)}\mathbf{e}^\top$ would be a constant vector.

Although it is easy to sample the initial set of reversed rates according to the interval given by Condition (5), satisfying Condition 2 of Definition 1 is harder. We solve this issue in Section III-D in terms of a constrained minimisation problem.

C. Solution of the components in isolation

Once the infinitesimal generator $\overline{\mathtt{Q}}_s$ of component $s \in \mathcal{S}$ is determined at a given iteration using Equation (1), its steady state solution can be computed. Recall that since the set of reversed rates is feasible, then the model is surely ergodic. The solution of the model can be carried out in different ways, for instance exploiting theoretical results about symbolic solutions [23]. Otherwise, the system of global balance equations can be directly solved. Finally, if the infinitesimal generator has a block regular structure, a matrix geometric solution can be computed as proposed in Section IV.

D. Stability issues

So far, we ignored the possibility that some of the systems (3) might have no solution even if the underlying process is irreducible. This can happen in case of components with infinitely many states, which are unstable due to bad choices of the reverse rates. If this situation arises while the algorithm is iterating we cannot conclude that the model is unstable, but only that the temporary choice of the reversed rates causes one or more component to be unstable. However, in these cases Equation (4) cannot be applied to compute the reversed rates for the following iterations. To overcome this issue, we introduce a procedure aimed at adjusting the reversed rates $\mathcal{X} = \{x_\ell\}_{\ell \in \mathcal{L}}$ to $\mathcal{X}' = \{x'_\ell\}_{\ell \in \mathcal{L}}$ in such a way as to be feasible according to Definition 1 and be as close as possible to $\mathcal{X}$. More formally, we should solve the minimisation problem

$$\mathcal{X}' \in \arg\min_{\mathcal{Y}} \sum_{\ell \in \mathcal{L}} (x_\ell - y_\ell)^2 \tag{6}$$

with the constraint that $\mathcal{Y} = \{y_\ell\}_{\ell \in \mathcal{L}}$ must be feasible according to Definition 1. In components with infinite state spaces and underlying block regular structure of Lower Hessenberg Matrix type, or QBD, the stability check can be done by means of the so called *drift condition*. Roughly speaking, the drift condition states that the (positive) drift to higher numbered levels must be less than the (negative) drift to lower numbered ones. More details are given in Section IV. For many practical instances of infinite state models, a passive label ℓ either contributes to the positive drift or to the negative drift. We call these models *monotonic with respect to the passive labels*. For this class of models, the solution of optimisation problem (6) is simple, and can always be found if the feasible space for the reversed rates is non-empty.

E. Convergence and stopping criteria

The algorithm terminates because one of the following three conditions is satisfied:

(S1) A set of feasible rates cannot be found.

(S2) For each component $s \in \mathcal{S}$, given a tolerance $\varepsilon > 0$:

$$\left\| \left(\boldsymbol{\pi}^{(s)}\right)^{[k]} - \left(\boldsymbol{\pi}^{(s)}\right)^{[k-1]} \right\| < \varepsilon ,$$

where $\left(\boldsymbol{\pi}^{(s)}\right)^{[k]}$ is the stationary distribution at iteration $k > 1$. In Section IV we discuss how to implement this test for models with infinite states.

(S3) The maximum number of iterations has been reached.

If the algorithm terminates due to (S1), then:

- if the components are monotonic with respect to the passive labels, then the model does not admit an exact product-form solution. If the model is stable but not in product-form, then the INAP approximation is unstable and no solution can be found. Intuitively, this could happen if the model is really close to instability.
- if the components are not monotonic with respect to the passive labels, then the algorithm used to solve (6) could have failed to find a feasible solution. However, observe that often intuition can drive the modeller to define a non empty set of feasible reversed rates in order to prevent the occurrence of such a behaviour.

When the algorithm terminates due to (S2), then one of the following cases arises:

- In the last interaction some reversed rates have been changed according to problem (6). In this case, the proposed result is not meaningful since some of the components in isolation are unstable.
- Otherwise if RCAT condition (2) holds, then the algorithm computed the correct solution of the model. If it does not hold, then the solution should be considered as an approximation. The check of RCAT condition (2) for model with infinite states is addressed in Section IV.

Condition (S3) is needed since INAP shares with most of the other fixed point algorithms for approximate analysis of stochastic models the lack of a proof of convergence. Therefore, although we could never observe in practice cyclic behaviours, we have to consider this possibility.

Observe that the space of feasible reversed rates is compact. Hence, by Brouwer's fixed point theorem, if the space of feasible reversed rates is non-empty, there exists at least one fixed point. However, this is not sufficient to guarantee that, given the initialization, a fixed point will be reached.

IV. Matrix Geometrics methods and product-forms

This section reviews the fundamental notions about matrix analytic methods. Matrix Geometric methods [11], [24] exploit the regular block-structure of the CTMC underlying a class of Markovian queues to efficiently compute their steady-state distribution.

A. Block Lower Hessenberg Markov Chains

In this section we consider Markov chains whose infinitesimal generator $\overline{\mathtt{Q}}$ exhibits a block lower Hessenberg structure:

$$\overline{\mathtt{Q}} = \begin{pmatrix} \mathtt{B}_{00} & \mathtt{B}_{01} & \mathtt{0} & \mathtt{0} & \mathtt{0} & \mathtt{0} & \mathtt{0} & \cdots \\ \mathtt{B}_{10} & \mathtt{B}_{11} & \mathtt{A}_0 & \mathtt{0} & \mathtt{0} & \mathtt{0} & \mathtt{0} & \cdots \\ \mathtt{B}_{20} & \mathtt{B}_{21} & \mathtt{A}_1 & \mathtt{A}_0 & \mathtt{0} & \mathtt{0} & \mathtt{0} & \cdots \\ \mathtt{B}_{30} & \mathtt{B}_{31} & \mathtt{A}_2 & \mathtt{A}_1 & \mathtt{A}_0 & \mathtt{0} & \mathtt{0} & \cdots \\ & & & \ddots & \ddots & \ddots & \ddots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}, \qquad (7)$$

where

- $\mathtt{A}_0, \mathtt{A}_1, \mathtt{A}_2$ are square matrices with the same dimension K,
- $\mathtt{B}_{00}$ is a square matrix with dimension H,
- $\mathtt{B}_{i0}$, $i > 0$ are $K \times H$ matrices,
- $\mathtt{B}_{01}$ is a $H \times K$ matrix,
- $\mathtt{B}_{i1}$, $i > 0$ are square matrices with dimension K.

As a special case, we derive later the results for quasi birth&death processes. We aim at finding a stationary distribution $\boldsymbol{\pi}$ from the system $\boldsymbol{\pi}\overline{\mathtt{Q}} = \mathbf{0}$. Consider a decomposition of the stationary distribution as $\boldsymbol{\pi} = (\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \dots)$, Neuts has shown that a matrix geometric solution exists to this problem, *i.e.* there exists a positive matrix $\mathtt{R}$ such that:

$$\boldsymbol{\pi}_n = \boldsymbol{\pi}_{n-1}\mathtt{R} \qquad (8)$$

for all $n > 1$, whereas $\boldsymbol{\pi}_0$ and $\boldsymbol{\pi}_1$ are determined as the solution of the following linear system of equalities:

$$(\boldsymbol{\pi}_0, \boldsymbol{\pi}_1) \begin{pmatrix} \mathtt{B}_{00} & \mathtt{B}_{01} \\ \sum_{i=1}^{\infty} \mathtt{R}^{i-1}\mathtt{B}_{i0} & \sum_{i=1}^{\infty} \mathtt{R}^{i-1}\mathtt{B}_{i1} \end{pmatrix} = (\mathbf{0}, \mathbf{0}), \qquad (9)$$

under the normalizing constraint:

$$\boldsymbol{\pi}_0 \mathbf{e}^\top + \boldsymbol{\pi}_1 \left(\sum_{i=1}^{\infty} \mathtt{R}^{i-1} \right) \mathbf{e}^\top = 1\,. \qquad (10)$$

In [11] the author proves that matrix $\mathtt{R}$ must satisfy the following relation:

$$\mathtt{A}_0 + \mathtt{R}\mathtt{A}_1 + \sum_{i=2}^{\infty} \mathtt{R}^i \mathtt{A}_i = \mathtt{0}\,.$$

Starting from this relation, the following iterative scheme for the computation of $\mathtt{R}$ can be derived:

$$\mathtt{R}^{[k+1]} = -\mathtt{A}_0\mathtt{A}_1^{-1} - \sum_{i=2}^{\infty} \left(\mathtt{R}^{[k]}\right)^i \mathtt{A}_i\mathtt{A}_1^{-1}\,,$$

for $k \geq 0$ and $\mathtt{R}^{[0]} = \mathtt{0}$. Neuts has shown that the iterative scheme is nondecreasing and converges to matrix $\mathtt{R}$.

In our context, this kind of process underlies one or more components in a cooperation. Therefore, we are actually assuming that all the matrices characterizing a component $s \in \mathcal{S}$ have a block regular structure of the type (7). Moreover, recall that the matrices $\mathtt{J}^{(\ell)}$, $\mathtt{T}^{(\ell)}$, $\mathtt{E}^{(s)}$ defining the components do not have the diagonal elements defined as the opposite of the sum of the row elements. Indeed, if $\mathtt{Q}$ defined by Equation (1) has a lower Hessenber structure, then trivially also $\overline{\mathtt{Q}}$ does. We use $\mathtt{B}_{00}^{(\ell)}$ to denote the upper left block of matrix $\mathtt{T}^{(\ell)}$ and similarly for the other blocks. As usual, we assume that for each component s with lower Hessenberg structure there exist constants $V^{(s)}$ and $W^{(s)}$ such that for all $\ell \in \mathcal{A}_s$ we have:

- $i \geq V^{(s)} \implies \mathtt{B}_{ih}^{(\ell)} = \mathtt{B}_{V^{(s)}h}^{(\ell)}$ with $h \in \{0, 1\}$
- $i > W^{(s)} \implies \mathtt{A}_i^{(\ell)} = \mathtt{0}$

In the following proposition we assume, without loss of generality, $V^{(s)} > W^{(s)}$ and write simply V and W omitting the component label.

Theorem 1 (Computation of expected reversed rate for Lower Hessenberg matrices)**.** *The expected reversed rate defined by Equation* (4) *can be written as follows in case of components exhibiting an underlying process with block-regular structure of the type shown in (7):*

$$
\begin{aligned}
x_\ell = \boldsymbol{\pi}_0 &\left(\mathtt{B}_{00}^{(\ell)}\mathbf{e}^\top + \mathtt{B}_{01}^{(\ell)}\mathbf{e}^\top\right) \\
&+ \sum_{i=1}^{V-1} \boldsymbol{\pi}_1 \mathtt{R}^{i-1}\left[\left(\mathtt{B}_{i0}^{(\ell)} - \mathtt{B}_{V0}^{(\ell)}\right)\mathbf{e}^\top + \left(\mathtt{B}_{i1}^{(\ell)} - \mathtt{B}_{V1}^{(\ell)}\right)\mathbf{e}^\top\right] \\
&\qquad - \sum_{i=1}^{W} \boldsymbol{\pi}_1 \mathtt{R}^{i-1}\left(\sum_{z=i}^{W}\mathtt{A}_z^{(\ell)}\right)\mathbf{e}^\top \\
&+ \boldsymbol{\pi}_1(\mathtt{I}-\mathtt{R})^{-1}\left(\mathtt{B}_{V0}^{(\ell)}\mathbf{e}^\top + \mathtt{B}_{V1}^{(\ell)}\mathbf{e}^\top + \mathtt{A}^{(\ell)}\mathbf{e}^\top\right) \qquad (11)
\end{aligned}
$$

where $\mathtt{A}^{(\ell)} = \sum_{i=0}^{W} \mathtt{A}_i^{(\ell)}$.

Proof: Equation (4) for lower Hessenberg matrices becomes:

$$
\begin{aligned}
\boldsymbol{\pi}_0 &\left(\mathtt{B}_{00}^{(\ell)}\mathbf{e}^\top + \mathtt{B}_{01}^{(\ell)}\mathbf{e}^\top\right) \\
&+ \boldsymbol{\pi}_1 \sum_{i=1}^{W} \mathtt{R}^{i-1}\left(\mathtt{B}_{i0}^{(\ell)}\mathbf{e}^\top + \mathtt{B}_{i1}^{(\ell)}\mathbf{e}^\top + \sum_{z=0}^{i-1}\mathtt{A}_z^{(\ell)}\mathbf{e}^\top\right) \\
&+ \boldsymbol{\pi}_1 \sum_{i=W+1}^{V-1} \mathtt{R}^{i-1}\left(\mathtt{B}_{i0}^{(\ell)}\mathbf{e}^\top + \mathtt{B}_{i1}^{(\ell)}\mathbf{e}^\top + \mathtt{A}^{(\ell)}\mathbf{e}^\top\right) \\
&+ \boldsymbol{\pi}_1 \sum_{i=V}^{\infty} \mathtt{R}^{i-1}\left(\mathtt{B}_{V0}^{(\ell)}\mathbf{e}^\top + \mathtt{B}_{V1}^{(\ell)}\mathbf{e}^\top + \mathtt{A}^{(\ell)}\mathbf{e}^\top\right)
\end{aligned}
$$

Since the spectral radius of $\mathtt{R}$ is lower than 1, the infinite sum of the expression above can be conveniently rewritten as:

$$
\begin{aligned}
\boldsymbol{\pi}_1(\mathtt{I}-\mathtt{R})^{-1}&\left(\mathtt{B}_{V0}^{(\ell)}\mathbf{e}^\top + \mathtt{B}_{V1}^{(\ell)}\mathbf{e}^\top + \mathtt{A}^{(\ell)}\mathbf{e}^\top\right) \\
&- \sum_{i=1}^{V-1} \boldsymbol{\pi}_1 \mathtt{R}^{i-1}\left(\mathtt{B}_{V0}^{(\ell)}\mathbf{e}^\top + \mathtt{B}_{V1}^{(\ell)}\mathbf{e}^\top + \mathtt{A}^{(\ell)}\mathbf{e}^\top\right)
\end{aligned}
$$

After rearranging the terms and some algebraic simplifications, Equation (11) can be derived. ∎

Corollary 1. *If a model $s \in \mathcal{S}$ satisfies RCAT conditions, then for each $\ell \in \mathcal{A}_s$, x_ℓ as defined in (11) is the constant reversed rate associated with label ℓ.*

Proof: The proof follows straightforwardly from the definition of x_ℓ given by Equation (4). ∎

Remark 1 (Stopping criterion)**.** *The verification of the stopping condition*

$$
\left\|\left(\boldsymbol{\pi}^{(s)}\right)^{[k]} - \left(\boldsymbol{\pi}^{(s)}\right)^{[k-1]}\right\| < \varepsilon\,,
$$

in case of Block Lower Hessenberg Markov Chains becomes

$$
\begin{aligned}
&\left\|\boldsymbol{\pi}_0^{[k]} - \boldsymbol{\pi}_0^{[k-1]}\right\|^2 \\
&\quad + \left\|\boldsymbol{\pi}_1^{[k]}\left(\mathtt{I}-\mathtt{R}^{[k]}\right)^{-1} - \boldsymbol{\pi}_1^{[k-1]}\left(\mathtt{I}-\mathtt{R}^{[k-1]}\right)^{-1}\right\|^2 < \varepsilon\,,
\end{aligned}
$$

where $\boldsymbol{\pi}_0^{[k]}$, $\boldsymbol{\pi}_1^{[k]}$ are the first two blocks of the decomposition of $\boldsymbol{\pi}^{(s)}$ and $\mathtt{R}^{[k]}$ is the rate matrix, all at iteration k.

Remark 2 (Checking RCAT condition)**.** *In order to verify that a product-form solution has been found, we have to check that the RCAT condition* (2) *is fullfilled for all $\ell \in \mathcal{L}$. To this end, we compute for each $\ell \in \mathcal{L}$ the residual of equation* (2) *after post-multiplication of both sides by diag($\boldsymbol{\pi}_{\alpha_\ell}$) and we impose that the maximum of those residuals should not exceed a tolerance parameter ε:*

$$
\max_{\ell\in\mathcal{L}} \left\|\boldsymbol{\pi}^{(\alpha_\ell)}\left(x_\ell \mathtt{I} - \mathtt{T}^{(\ell)}\right)\right\| < \varepsilon\,.
$$

The computation of the norm for all $\ell \in \mathcal{L}$, needed to check the RCAT condition, can be achieved in closed-form according to Theorem 2 assuming $\mathtt{Z} = x_\ell \mathtt{I} - \mathtt{T}^{(\ell)}$.

Theorem 2. *Let $\mathtt{Z}$ be a matrix with block structure and block names defined as in* (7) *and let $\boldsymbol{\pi} = (\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \dots)$ be a normalized probability distribution satisfying $\boldsymbol{\pi}_i = \boldsymbol{\pi}_{i-1}\mathtt{R}$, for all $i > 1$. Then*

$$
\begin{aligned}
&\|\boldsymbol{\pi}\mathtt{Z}\|^2 \\
&= \sum_{j\in\{0,1\}} \left\|\boldsymbol{\pi}_0 \mathtt{B}_{0j} + \boldsymbol{\pi}_1 \sum_{i=1}^{V-1} \mathtt{R}^{i-1}\left(\mathtt{B}_{ij} - \mathtt{B}_{Vj}\right) + \boldsymbol{\pi}_1(\mathtt{I}-\mathtt{R})^{-1}\mathtt{B}_{Vj}\right\|^2 \\
&\qquad + \boldsymbol{\pi}_1 \mathtt{U}\left[(\mathtt{V}^\top \mathtt{Y}\mathtt{Y}^\top \mathtt{V}) \circ \mathtt{W}\right]\mathtt{U}^\top \boldsymbol{\pi}_1^\top
\end{aligned}
$$

where $\circ$ is the entrywise matrix product operator (Hadamard product),

$$
\mathtt{R} = \mathtt{U}\,\mathit{diag}(\mathrm{s})\,\mathtt{V}^\top
$$

is the singular value decomposition of matrix $\mathtt{R}$, and $\mathtt{Y}, \mathtt{W} = (w_{ij})$ are defined as follows

$$
\mathtt{Y} = \mathtt{A}_0 + \mathtt{R}\mathtt{A}_1 + \mathtt{R}^2\mathtt{A}_2 \quad \mathit{and} \quad w_{ij} = (1 - s_i s_j)^{-1}\,.
$$

Proof: By exploiting the structure of $\mathtt{Z}$ we expand the norm on the left-hand-side of the equality as follows

$$
\begin{aligned}
&\|\boldsymbol{\pi}\mathtt{Z}\|^2 \\
&= \left(\sum_{j\in\{0,1\}} \left\|\boldsymbol{\pi}_0 \mathtt{B}_{0j} + \boldsymbol{\pi}_1 \sum_{i=1}^{\infty} \mathtt{R}^{i-1}\mathtt{B}_{ij}\right\|^2\right) + \sum_{i=0}^{\infty} \left\|\boldsymbol{\pi}_1 \mathtt{R}^i \mathtt{Y}\right\|^2
\end{aligned}
$$

Here, we can get rid of the infinite summation in the first terms on the right-hand-side by rewriting it as follows

$$
\sum_{j\in\{0,1\}} \left\|\boldsymbol{\pi}_0 \mathtt{B}_{0j} + \boldsymbol{\pi}_1 \sum_{i=1}^{V-1} \mathtt{R}^{i-1}\left(\mathtt{B}_{ij} - \mathtt{B}_{Vj}\right) + \boldsymbol{\pi}_1(\mathtt{I}-\mathtt{R})^{-1}\mathtt{B}_{Vj}\right\|^2,
$$

whereas we can get rid of the infinite summation of the second term on the right-hand-side as follows:

$$\sum_{i=0}^{\infty} \left\| \boldsymbol{\pi}_1 \mathtt{R}^i \mathtt{Y} \right\|^2 = \boldsymbol{\pi}_1 \left[\sum_{i=0}^{\infty} \mathtt{R}^i \mathtt{Y}\mathtt{Y}^\top (\mathtt{R}^i)^\top \right] \boldsymbol{\pi}_1$$

$$= \boldsymbol{\pi}_1 \mathtt{U} \left[\sum_{i=0}^{\infty} \operatorname{diag}(\mathbf{s})^i \mathtt{V}^\top \mathtt{Y}\mathtt{Y}^\top \mathtt{V} \operatorname{diag}(\mathbf{s})^i \right] \mathtt{U}^\top \boldsymbol{\pi}_1$$

$$\boldsymbol{\pi}_1 \mathtt{U} \left[(\mathtt{V}^\top \mathtt{Y}\mathtt{Y}^\top \mathtt{V}) \circ \sum_{i=0}^{\infty} (\mathbf{s}^i)^\top \mathbf{s}^i \right] \mathtt{U}^\top \boldsymbol{\pi}_1$$

where $\mathbf{s}^i$ denotes vector $\mathbf{s}$ with entries raised to the ith power. Finally, the result derives from the equality $\sum_{i=0}^{\infty} (\mathbf{s}^i)^\top \mathbf{s}^i = \mathtt{W}$. ■

Note that Theorems 1 and 2 give a new procedure to check RCAT conditions for all the models with Lower Hessenberg block structured infinitesimal generator. First, the expected reversed rate must be computed according to the closed matrix formula of Theorem 1 and then using Theorem 2 one can check if the reversed rates are constant without the need of a truncation.

Remark 3 (Stability of the queue)**.** *Stability of queues with Lower Hessenberg structures has been widely studied in literature and for the sake of brevity we recall here few notions. For our purposes, the key-point is that according to the drift conditions presented in [24], [12], deciding if rate matrix* $\mathtt{R}$ *has spectral radius lower than* 1 *requires only algebraic operations on the block-matrices of the infinitesimal generator. Therefore, this can be done efficiently while solving the optimization problem* (6) *when required.*

B. QBD Processes

We consider QBD processes as special cases of Lower Hessenberg matrices where: $\mathtt{B}_{11} = \mathtt{A}_1$, $\mathtt{B}_{21} = \mathtt{A}_2$, $\mathtt{B}_{i0} = \mathtt{0}$ for $i \geq 2$, $\mathtt{B}_{i1} = \mathtt{0}$ for $i > 2$ and $\mathtt{A}_i = \mathtt{0}$ for $i > 2$. The iterative computation of matrix $\mathtt{R}$ simplifies to:

$$\mathtt{R}^{[k+1]} = -\mathtt{A}_0 \mathtt{A}_1^{-1} - \left(\mathtt{R}^{[k]} \right)^2 \mathtt{A}_2 \mathtt{A}_1^{-1}, \quad k \geq 0, \qquad (12)$$

where $\mathtt{R}^{[k]}$ is the matrix at the kth iteration step, with $\mathtt{R}^{[0]} = \mathtt{0}$. Faster iterative approaches for the computation of $\mathtt{R}$ in case of QBD processes have been defined (see, e.g., [25], [12]).

The following Corollary is the formulation of Theorem 1 for QBD processes. The proof is purely algebraic and is omitted for the sake of brevity.

Corollary 2 (Computation of expected reversed rate for QBD)**.** *The expected reversed rate defined by Equation* (4) *can be written as follows in case of components exhibiting an underlying QBD process with block-regular structure:*

$$x_\ell = \boldsymbol{\pi}_0 \left(\mathtt{B}_{00} \mathbf{e}^\top + \mathtt{B}_{01} \mathbf{e}^\top \right) + \boldsymbol{\pi}_1 \left(\mathtt{B}_{10} \mathbf{e}^\top - \mathtt{A}_2 \mathbf{e}^\top \right) + \boldsymbol{\pi}_1 (\mathtt{I} - \mathtt{R})^{-1} \left(\mathtt{A}_0 + \mathtt{A}_1 + \mathtt{A}_2 \right) \mathbf{e}^\top . \qquad (13)$$

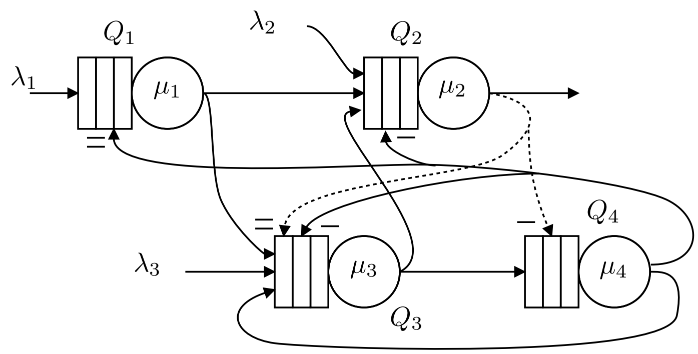

Fig. 1. G-network studied in Section V-A. Different line styles are used just for readability. Symbol $-$ denotes a batch removal signal, while $=$ a catastrophe signal.

V. Numerical validation

A. Network of heterogeneous queues in product-form

We test the INAP algorithm for models with infinite state spaces in a network of heterogeneous queues that yields a product-form steady-state distribution.

a) Model description.: The first example deals with the class of queueing networks introduced by Gelenbe in [6] called G-networks. We show that, differently from what previously proposed in [8], the algorithm developed in this paper is able to solve models consisting of more general nodes as those presented in [7], [26]. The G-network consists of 4 nodes $(Q_1, \ldots, Q_4)$ with independent exponentially distributed service time. The arrival of external customers at $Q_1, \ldots, Q_3$ are modelled by independent homogeneous Poisson processes with rates $\lambda_1, \ldots, \lambda_3$. As usual, we assume independence between the arrival processes and the service times. Once a customer departs from a node, it can: a) leave the network; b) enter another node as a standard customer. In this case, if the destination node is empty then its service starts immediately, otherwise it has to wait in the queue according to a *First Come First Served* discipline; c) enter another queue as a batch removal signal. In this case, at the customer arrival epoch t the queue length decreases by $\min\{n_i^{[t]}, B_i\}$ customers, where $n_i^{[t]}$ denotes the number of customers at the destination node i at epoch t and B_i is a random variable representing the size of the batch to remove at node i; d) enter another queue as a catastrophe signal and in this case, the node is flushed. The routing of customers and signals in the network is probabilistic and state-independent. In Figure 1 we show the model that we study.

b) Modelling the nodes.: Table I associates a label with each synchronisation of the network. The 4 nodes of the network can be easily modelled according to a QBD structure or as a block Lower Hessenberg Matrix. As an instance, let us consider Q_3. It can receive both catastrophe signals (from Q_2) and batch removal signals (from Q_3), whereas positive customers arrive from the outside with rate λ_3, from Q_1 and Q_4. Departures are possible to nodes Q_2 and Q_4 as positive customers. In Table II we show the matrices that describe this node assuming that the batch size B_3 is deterministically

Label	Active	Passive	Type	Prob.
0	Q_4	Q_1	C	P_{41}^C
1	Q_1	Q_2	P	P_{12}
2	Q_3	Q_2	P	P_{32}
3	Q_4	Q_2	B	P_{42}^B
4	Q_1	Q_3	P	P_{13}
5	Q_4	Q_3	P	P_{43}
6	Q_2	Q_3	C	P_{23}^C
7	Q_4	Q_3	B	P_{43}^B
8	Q_3	Q_4	P	P_{34}
9	Q_2	Q_4	B	P_{24}^B

TABLE I
SYNCHRONISATIONS IN THE NETWORK OF FIGURE 1. LABELS P, B, C DENOTE A POSITIVE CUSTOMER ARRIVAL, A BATCH REMOVAL SIGNAL AND A CATASTROPHE SIGNAL, RESPECTIVELY.

equal to 3. Therefore, nodes Q_1 and Q_3 are modelled as lower

$$\mathrm{E}^{(3)} = \begin{pmatrix} 0 & \lambda_3 & 0 & 0 & 0 & \dots \\ 0 & 0 & \lambda_3 & 0 & 0 & \dots \\ 0 & 0 & 0 & \lambda_3 & 0 & \dots \\ 0 & 0 & 0 & 0 & \lambda_3 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \end{pmatrix}$$

$$\mathrm{J}^{(4)} = \mathrm{J}^{(5)} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & \dots \\ 0 & 0 & 1 & 0 & 0 & \dots \\ 0 & 0 & 0 & 1 & 0 & \dots \\ 0 & 0 & 0 & 0 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \end{pmatrix}$$

$$\mathrm{J}^{(6)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \dots \\ 1 & 0 & 0 & 0 & 0 & \dots \\ 1 & 0 & 0 & 0 & 0 & \dots \\ 1 & 0 & 0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \end{pmatrix}$$

$$\mathrm{J}^{(7)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \dots \\ 1 & 0 & 0 & 0 & 0 & \dots \\ 1 & 0 & 0 & 0 & 0 & \dots \\ 1 & 0 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & 0 & \dots \\ 0 & 0 & 1 & 0 & 0 & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \end{pmatrix}$$

TABLE II
MATRIX DESCRIPTION OF Q_3 CONSIDERED IN SECTION V-A.

Hessenberg block matrices, whereas Q_2 and Q_4 (which do not accept catastrophe signals) are modelled as QBD processes.

c) Numerical validation.: Table III gives the set of network's parameters that we used. INAP reached a fixed point with a tolerance of 10^{-9} in 12 iterations. The numerical procedure verifies RCAT's Condition 2, therefore we can conclude that the computed result is the product-form steady-state distribution. In order to evaluate the accuracy of the numerical approach, we derive the non-linear system of rate equations characterising the G-network and evaluate the norm of the vector of residuals once the unknown are replaced by the values obtained by INAP. In [7], [26] the authors prove that each node of this type of network must yield a geometric expression for the stationary probability distribution, $\pi_i(n_i) = (1-\rho_i)\rho_i^{n_i}$, where ρ_i is the load factor of node Q_i and n_i its number of customers. Therefore, we can reduce the number of unknowns of the models to the load factors $\rho_1, \dots, \rho_4$. These are the solutions of the following non-linear system of equations:

Arrival rates	
λ_1	3.0
λ_2	4.2
λ_3	3.8
Service rates	
μ_1	2.3
μ_2	6.2
μ_3	3.1
μ_4	6.0
Batch sizes	
B_2	3
B_4	3

Routing probabilities	
P_{12}	0.3
P_{13}	0.7
P_{23}^C	0.2
P_{24}^B	0.3
P_{20}	0.5
P_{32}	0.4
P_{34}	0.6
P_{41}^C	0.1
P_{42}^B	0.1
P_{43}	0.6
P_{43}^B	0.2

TABLE III
PARAMETRISATION OF THE G-NETWORK OF FIGURE 1.

Load factor	Computed value
ρ_1	0.8994655
ρ_2	0.8866789
ρ_3	0.7447235
ρ_4	0.1734745

TABLE IV
LOAD FACTORS OF THE QUEUES OF FIGURE 1 COMPUTED BY INAP WITH THE PARAMETRISATION GIVEN IN TABLE III.

$$\begin{cases} \rho_1(\lambda_1 + \rho_4\mu_4 P_{41}^C + \mu_1) = \lambda_1 + \rho_1^2\mu_1 \\ \rho_2(\lambda_2 + \rho_1\mu_1 P_{12} + \rho_3\mu_3 P_{32} + \rho_4\mu_4 P_{42}^B + \mu_2) \\ \quad = \lambda_2 + \rho_1\mu_1 P_{12} + \rho_3\mu_3 P_{32} + \rho_2^2\mu_2 + \rho_2^4\rho_4\mu_4 P_{42}^B \\ \rho_3(\lambda_3 + \rho_1\mu_1 P_{13} + \rho_4\mu_4(P_{43} + P_{43}^B) + \rho_2\mu_2 P_{23}^C + \mu_3) \\ \quad = \lambda_3 + \rho_1\mu_1 P_{13} + \rho_4\mu_4 P_{43} + \rho_3^2\mu_3 + \rho_3^4\rho_4\mu_4 P_{43}^B \\ \rho_4(\rho_3\mu_3 P_{34} + \rho_2\mu_2 P_{24}^B + \mu_4) \\ \quad = \rho_3\mu_3 P_{34} + \rho_4^2\mu_4 + \rho_4^4\rho_2\mu_2 P_{24}^B \end{cases}$$

The values computed by INAP are shown in Table IV. The vector of residuals has a norm of $6.366 \cdot 10^{-10}$.

B. Comparison with algorithms based on the truncation of the stochastic process

In [19] the authors revise the algorithm proposed in [8] in order to consider model with infinite state spaces. The main idea consists in the definition of a special operator that truncates the state space of a model so that the states with stationary probability lower than a given threshold are not considered. The algorithm is then applied to study a G-network with catastrophe nodes and single customer deletion nodes (as well as standard exponential nodes). The model is in product-form and consists of 10 nodes whose stationary distributions has a geometric form: $\pi^{(i)}(n_i) = (1-\rho_i)\rho_i^{n_i}$, where n_i denotes the number of customers at node i, $\pi^{(i)}(n_i)$ its stationary probability and ρ_i is the load factor of node i. In [19] the authors state the non-linear system of equations whose solution give ρ_i, $i = 1, \dots, 10$. Although the number of iterations of the algorithm proposed here are the same as that proposed in [19], the former is more accurate since the vector of residuals has a norm of $1.8683 \cdot 10^{-9}$, while for the latter it has a norm of $9.4891 \cdot 10^{-5}$. Moreover, the truncation mechanism proposed is not efficient for models in which some components have load factors close to 1, because the number of states that must be considered can be very high.

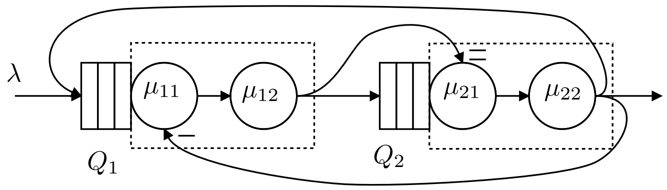

Fig. 2. Network studied in Section V-C.

Label	Active	Passive	Type	Prob.
0	Q_1	Q_2	P	P_{12}
1	Q_1	Q_2	C	P_{12}^C
2	Q_2	Q_1	N	P_{21}^N
3	Q_2	Q_1	P	P_{21}

TABLE V
SYNCHRONISATIONS IN THE NETWORK OF FIGURE 2. LABELS P, N, C DENOTE A POSITIVE CUSTOMER ARRIVAL, A NEGATIVE CUSTOMER ARRIVAL AND A CATASTROPHE SIGNAL, RESPECTIVELY.

In the case-study of [19] the number of states of one queue reaches 161 and the matrix corresponding to its infinitesimal generator must be inverted at each algorithm step. Using the results proposed here, the complexity of the matrix operations depends on the size of the blocks characterising their structure and hence are in general much lower. Moreover, truncation may cause numerical instability in the process of computation of the steady-state distribution of each node even in isolation. With respect to these problems for the Lower Hessenberg case, the algorithm we propose in this paper is more accurate, more efficient and does not require to specify how the state space should be truncated. Finally, the algorithm proposed here is able to provide approximate solutions in case of non-product-form model specifications.

C. Product-form approximation of ./Hypo/1 queues with negative customers and catastrophes

a) Model description: In this section we consider a network with feedback consisting of two nodes (Q_1, Q_2) with independent hypoexponential service time distributions. We assume for both the queues two stages of service with rates μ_{i1} and μ_{i2} for node $i = 1, 2$. Customers arrive at Q_1 according to an independent, homogeneous Poisson process with rate λ. If a negative customer arrives at Q_1 when the first stage of service is busy, then the queue length is decreased by 1, whereas if it arrives when Q_1 is empty or with the second stage of service busy, then the negative customer vanishes. Q_2 has a similar behaviour, but at a signal arrival epoch, if the first stage of service is busy, then the queue is flushed. Figure 2 illustrates the network topology.

b) Modelling the nodes: Table V gives the synchronising labels with the associated probabilities. Nodes Q_1 has a QBD underlying structure where the size of A_i matrices is 2, while node Q_2 accepts catastrophe signals and hence exhibits an underlying block regular structure that resembles a Lower Hessenberg matrix.

c) Numerical analysis: We apply INAP for the analysis of the model of Figure 2 parametrised as specified in Table VI. We study the model under different arrival rates λ. INAP

Service rates	
μ_{11}	13.0
μ_{12}	20.0
μ_{21}	7.0
μ_{22}	18.0

Routing prob.	
P_{12}	0.8
P_{12}^C	0.2
P_{21}	0.4
P_{21}^N	0.2
P_{20}	0.4

TABLE VI
PARAMETERS USED FOR THE ANALYSIS OF THE NETWORK OF FIGURE 2. P_{20} DENOTES THE PROBABILITY THAT A CUSTOMER EXITS THE NETWORK AFTER LEAVING NODE Q_2.

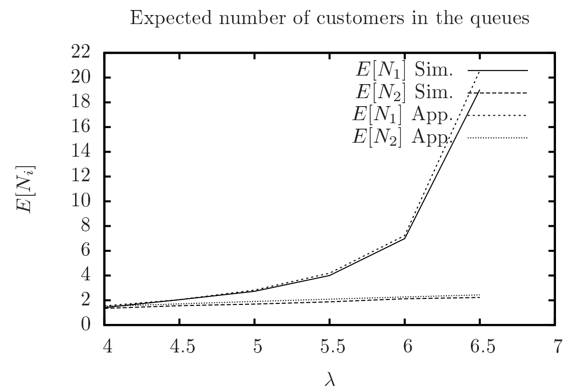

Fig. 3. Expected number of customers in Q_1 and Q_2 obtained for different values of the arrival rate λ.

detects that the model is not in product-form, as well-known in queueing theory. However, it still provides an approximated result based on a decomposition of the network. In Figure 3 we compare the expected number of customers in Q_1 and Q_2 under different arrival rates obtained by INAP and by a simulation tool. Simulations are run in Timenet [27] on a Generalised Stochastic Petri net model of the queueing network of Figure 2. The simulation parameters are: confidence interval of 95% with a maximum width of 5% of the estimated value. As we can note, INAP approximations and Timenet estimates differ in the worst case for less than 10% of the estimated mean.

VI. CONCLUSION

Decomposition of general stochastic models into product-form approximating models has been an important topic of research for the community of performance evaluation. Specifically, the applicability of fixed point iterations has been investigated in many works (see, e.g., [20], [21] and the references therein). With respect to the state-of-the-art, we contribute to this line of research by exploiting RCAT product-forms to provide exact model decompositions, when a set of conditions is satisfied, or approximate ones, otherwise. In practice, testing RCAT conditions for general and heterogeneous models is a difficult task that requires a theoretical analysis of the model's components. Specifically, the computation of the transition rates in the reversed processes of the components is computationally expensive and numerically unstable for models with infinite state spaces. The first contribution of the paper is the definition of a novel strategy to derive the reversed rates of the components' synchronising transitions that is very efficient both for models with finite and infinite block structured state spaces. Basically, the method develops in two phases: in the former we compute an "expected reversed rate" and in the latter we test RCAT condition by checking if all the reversed rates associated to the same synchronising active label are equal to the expected value. We showed how to apply this result for models with infinite state spaces but with block regular structure of the type Lower Hessenber or QBD and proved that both the reversed rates and RCAT test can be computed through a closed matrix formula. These approaches make INAP computationally efficient and numerically stable. The class of components with this types of block-regular structures includes a large variety of well-known queueing models such as those studied in [24], [6], [7] and many other that could be or not in product-form. From the applicability point of view, the paper gives a contribution to the exact or approximate analysis of heterogeneous models. Indeed, since models are specified at a very low level, different types of components can be integrated. Moreover, differently from what proposed in [20], INAP uses very loose assumption on the model synchronisations (the pairwise cooperation is the strictest one). We believe that INAP should be applied for studying product-form models with pairwise synchronisations thanks to its capability of automatic detection (within the numerical precision) of product-forms - therefore automatising the proof -, to its ability of performing an automatic analysis of the model stability and finally to the computational efficiency required to solve the possibly non-liner systems of rate equations that are needed for deriving the stationary distributions.

INAP convergence is very fast and in all our experiments the number of iterations have been lower than 20 with a tolerance of $\varepsilon = 10^{-6}$. INAP is currently implemented in Java, and the tool exploits the polymorphism and class abstractions to allow the extension to new model classes possibly defined by the users. With this implementation we showed some case-studies in which we consider heterogeneous queueing network models. The product-form solution is automatically derived when it exists. Otherwise, we showed that INAP derives a good approximation of non-product-form models. However, the quality of such an approximation strongly depends on how "far" is the model to be in in product-form as spotted also in [22]. We think that an important improvement could be the implementation of INAP within existing tools for stochastic modelling such as the PEPA Eclipse Plug-in [28].

Future efforts will be devoted to relax the hypothesis on the pairwise synchronisations in order to encompass trigger-based component cooperations [29] and to improve the accuracy of the approximation when the matrices describing the passive transitions are not stochastic.

REFERENCES

[1] J. Hillston, "A Compositional Approach to Performance Modelling," Ph.D. dissertation, Department of Computer Science, University of Edinburgh, 1994.

[2] M. A. Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis, *Modelling with generalized stochastic Petri nets*. Wiley, 1995.

[3] F. Baskett, K. M. Chandy, R. R. Muntz, and F. G. Palacios, "Open, closed, and mixed networks of queues with different classes of customers," *J. ACM*, vol. 22, no. 2, pp. 248–260, 1975.

[4] F. Kelly, *Reversibility and stochastic networks*. New York: Wiley, 1979.

[5] P. G. Harrison, "Turning back time in Markovian process algebra," *Theoretical Computer Science*, vol. 290, no. 3, pp. 1947–1986, 2003.

[6] E. Gelenbe, "Product form networks with negative and positive customers," *J. of Appl. Prob.*, vol. 28, no. 3, pp. 656–663, 1991.

[7] ——, "G-Networks with signals and batch removal," *Prob. in the Eng. and Informational Sciences*, vol. 7, pp. 335–342, 1993.

[8] A. Marin and S. Rota Bulò, "A general algorithm to compute the steady-state solution of product-form cooperating Markov chains," in *Proc. of MASCOTS 2009*, London, UK, September 2009, pp. 515–524.

[9] E. Gelenbe and J. M. Fourneau, "G-networks with resets," *Perf. Eval.*, vol. 49, no. 1-4, pp. 179–191, 2002.

[10] E. Gelenbe and M. Schassberger, "Stability of product form G-Networks," *Prob. in the Eng. and Informational Sciences*, vol. 6, pp. 271–276, 1992.

[11] M. F. Neuts, *Matrix Geometric Solutions in Stochastic Models*. Baltimore, Md: John Hopkins, 1981.

[12] D. Bini, G. Latouche, and B. Meini, *Numerical Methods for Structured Markov Chains*. Oxford University Press, 2005.

[13] M. Sereno, "Towards a product form solution for stochastic process algebras," *The Computer Journal*, vol. 38, no. 7, pp. 622–632, December 1995.

[14] J. L. Coleman, W. Henderson, and P. G. Taylor, "Product form equilibrium distributions and a convolution algorithm for Stochastic Petri nets," *Perf. Eval.*, vol. 26, no. 3, pp. 159–180, 1996.

[15] S. Balsamo, P. Harrison, and A. Marin, "Methodological construction of product-form stochastic petri-nets for performance evaluation," *J. of System and Software*, p. To appear, 2012.

[16] F. Bause, "Queueing Petri nets: A formalism for the combined qualitative and quantitative analysis of systems," in *Proc. of 5th Int. Workshop on Petri Nets and Performance Models*, Toulouse (France), 1993, pp. 14–23.

[17] I. Mitrani, A. Ost, and M. Rettelbach, *Quantitative Methods in Parallel Systems*. Springer, 1995, ch. TIPP and the Spectral Expansion Method.

[18] N. Thomas and S. Gilmore, "Applying quasi-separability to markovian process algebra," in *6th Process Algebra and Performance Modelling Workshop*, 1998.

[19] S. Balsamo, G. Dei Rossi, and A. Marin, "A numerical algorithm for the solution of product-form models with infinite state spaces," in *Proc. of EPEW 2010: Comp. Perf. Eng.* Bertinoro, Italy: LNCS 6342/2010, 2010, pp. 191–206.

[20] V. Mainkar and K. Trivedi, "Sufficient conditions for existence of a fixed point in stochastic reward net-based iterative models," *IEEE Trans. on Soft. Eng.*, vol. 22, no. 9, pp. 640–653, 1996.

[21] R. Gaeta, M. Gribaudo, D. Manini, and M. Sereno, "Fixed Point Approximations for TCP Behaviour in Networks of Routers Iimplementing Heterogeneous Queue Management Policies," in *Proc. of. 13th Conf. on Measurement, Modeling, and Eval. of Comp. and Comm. Syst. (MMB)*, Nurnberg, Germany, 2006.

[22] P. Buchholz, "Product form approximations for communicating Markov processes," *Perf. Eval.*, vol. 67, no. 9, pp. 797 – 815, 2010, special Issue: QEST 2008.

[23] A. Marin and S. Rota Bulò, "Explicit solutions for queues with hypo-exponential service time distribution and application to product-form approximations," in *Proc. of Valuetools 2012 Conf.* Paris, FR: ICST, 2012, pp. 1–10.

[24] M. F. Neuts, *Structured stochastic matrices of M/G/1 type and their application*. New York: Marcel Dekker, 1989.

[25] G. Latouche and Y. Ramaswami, "A logaritmic reduction algorithm for Quasi Birth and Death processes," *J. of Appl. Prob.*, vol. 30, pp. 650–674, 1994.

[26] P. G. Harrison, "Compositional reversed Markov processes, with applications to G-networks," *Perf. Eval., Elsevier*, vol. 57, no. 3, pp. 379–408, 2004.

[27] A. Zimmermann, J. Freiheit, R. German, and G. Hommel, "Petri net modelling and performability evaluation with timenet 3.0," in *TOOLS '00: Proc. of the 11th Int. Conf. on Computer Performance Evaluation: Modelling Techniques and Tools*. London, UK: Springer-Verlag, 2000, pp. 188–202.

[28] M. Tribastone, A. Duguid, and S. Gilmore, "The PEPA Eclipse Plug-in," *Perf. Eval. Review*, vol. 36, no. 4, pp. 28–33, 2009.

[29] E. Gelenbe, "G-networks with triggered customer movement," *J. of Appl. Prob.*, vol. 30, pp. 742–748, 1993.