Performance Engineering with Product-form Models: Efficient Solutions and Applications

Simonetta Balsamo
Dipartimento di Informatica
Università Ca' Foscari Venezia
via Torino, 155
Venice, Italy
balsamo@dsi.unive.it

Andrea Marin
Dipartimento di Informatica
Università Ca' Foscari Venezia
via Torino, 155
Venice, Italy
marin@dsi.unive.it

ABSTRACT

Performance engineering plays a pivotal role in the successful design of software system and the software development process. Stochastic modelling has been widely applied to predict and evaluate or estimate system performance. We consider the specification of models in terms of compositions of simpler components and their efficient solution. Various formalisms or classes of stochastic models have been applied for system performance engineering and evaluation. These formalisms includes queueing networks, Stochastic Petri Nets, and Stochastic Process Algebras. Their dynamic behaviour can be usually represented by an underlying stochastic (Markov) process. For each formalism some classes of product-form models have been identified, starting from the first remarkable results for BCMP queueing networks. For some product-form models various efficient algorithms have been defined. We discuss the problem of identifying and characterize classes of product-form models. We compare the properties of the various modeling formalisms, their solution and the combination of product-form (sub)models into a heterogeneous model. We illustrate the application of product-form stochastic models for system performance engineering with some examples of tools for the solution of heterogeneous models formed by synchronized sub-models, and some practical applications.

Categories and Subject Descriptors

C.4 [**PERFORMANCE OF SYSTEMS**]: Modeling techniques

General Terms

Performance

Keywords

Queueing theory, Product-form solutions.

1. INTRODUCTION

Performance engineering plays a pivotal role in the software development process and the successful design of software architectures [42], communication protocols, or hardware architectures all along the development process. Modelling is an important and useful approach for performance evaluation and system validation and it can provide prediction and comparison of design alternatives. System performance engineering deals with the representation and analysis of the system dynamic based on models to provide feedback in the system development process. In this context, stochastic modelling has been widely applied to evaluate or estimate the performance of both software [8] and hardware architectures of computer systems. These often consist of a possibly large number of interacting components that have probabilistic behaviours. A successful quantitative analysis of such systems depends on several factors: the ability to derive an adequate model from the system specifications, the characterisation of the workload, and the availability of methods for deriving the desired performance indices and their interpretation at the system level. We deal with the specification of models in terms of compositions of simpler components and their efficient solution. Specifically, we focus on the class of stochastic models with product-form solutions. These are characterised by separable equilibrium state distributions that can be efficiently computed by considering each model component in isolation. System performance engineering based on stochastic models requires the choice of an appropriate formalism in order to model develop, analyze and evaluate significant system models. Various formalisms or classes of stochastic models have been applied for system performance engineering and evaluation. Their dynamic behaviour can be usually represented by an underlying stochastic process. These formalisms include queueing networks, Stochastic Petri Nets, Stochastic Process Algebras, stochastic (Markov) processes. For each formalism some classes of product-form models have been identified, starting from the first remarkable results for BCMP queueing networks [12]. Some product-form models can be solved by various efficient algorithms that have been proposed in literature. However, for some classes of models, additional conditions and constraints have to be satisfied in order to apply such algorithms. Specific features and properties have been studied and identified for various classes of stochastic models, depending on the specific formalism. More recent research focused on model properties that can be expressed in terms of the underlying Markov process.

Product-form has been widely investigated in queueing network domain. Queueing network models (QN) have been extensively applied to represent and analyze various types of resource sharing systems, such as production, communication and computer systems, and they have proved to be a powerful and versatile tool for system performance evaluation and prediction. QNs provide an abstract/black-box notation, thus allowing easier feedback and model comprehension, especially in a component-based software development process. A QN is a collection of service centers representing the system components that provide service to a collection of customers that represent the users. The customers' competition for the resource service corresponds to queueing into the service centers. The analysis of the QN consists of evaluating a set of performance measures, such as resource utilization and throughput and customer response time. This class of performance models provide a good balance between a relative high accuracy in the performance results and the efficiency in model analysis and evaluation, mainly thanks to product-form solutions. Product-form QNs have a simple closed form expression of the stationary state distribution that allowed the performance community to define efficient algorithms to evaluate average performance measures [34, 16]. Other performance modelling formalisms such as the Stochastic Petri Nets (SPN) and Stochastic Process Algebra (SPA) have been proposed allowing the definition of more expressive models and more complex cooperation and synchronization among system components. Some classes of product-form models of SPN [38], Markovian SPA (MPA) [30] and Stochastic Automata Network (SAN) [40] have been proposed in literature. These formalisms also allow one to specify cooperation of sub-models at the level of stochastic Markov processes. In this tutorial we discuss the characterisation of classes of product-form models for various formalisms, their solution and the combination of product-form (sub)models into a heterogeneous model. Identifying product-form models is not an easy task. In Section 2 we discuss the relevant problems in the characterisation of classes of product-form models. In Section 2.1 we deal with various types of product-form QN models, and we discuss their properties, such as local and station balance, $M \Rightarrow M$ property, reversibility and quasi-reversibility. Section 2.2 presents the characterization of product-form models expressed in other formalisms such as SPN and SPA and the cooperation at Markov process level. We discuss their properties by reviewing the important result of the Reversed Compound Agent Theorem (RCAT) [25], and one of its extensions (GRCAT). Section 3 introduces heterogeneous modelling where various types of sub-models expressed by different formalisms can be combined. Section 4 addresses the solution algorithms for product-form stochastic models. Section 5 presents an example of a tool for the solution of heterogeneous models that consist of synchronised sub-models, and Section 6 some practical applications of product-form performance models. Conclusions are presents in Section 7.

2. CHARACTERISATION OF A CLASS OF PRODUCT-FORM MODELS

Characterising the stochastic processes underlying product-form models has been one of the main topics addressed in product-form theory. For queueing networks, several authors tried to achieve this goal from different abstraction levels. Since product-form appeared also for other formalisms (MPA, (G)SPN, SA), it appeared clear that more general characterisations were needed in ordered to have a unified approach to product-form modelling. In this section, we briefly describe both the results that have been formulated for queueing networks and those that appeared later in literature, and then, when possible, we compare them.

2.1 Characterisation of product-form QNs

Before proceeding with the definition of product-form properties in QNs we need to introduce some notation and nomenclature.

2.1.1 Nomenclature and notation about QNs

In this paper we refer to a *queueing station* (or simply to *station*) to identify the *service room* and the *queue*. Customers are characterised by a *class* and a *chain*. While a class is a temporary characterisation, i.e., probabilistic class switching may occur, a chain is permanent. Chains form a partition of the classes, and each chain may be either open or closed. In the former case Poisson distributed arrivals from the outside and departures are allowed, while in the latter one the chain population is constant. Customers are distinguishable only through their classes. We assume independence among the service times and the arrival processes. A QN consists of N stations whose state is denoted by $\mathbf{m}_i$, $1 \leq i \leq N$, and the joint state is denoted by $\mathbf{m} = (\mathbf{n}_1, \ldots, \mathbf{m}_N)$. For some QNs, the states of the stations are fully described by the numbers of customers that are present for each class (take for instance the processor sharing queueing discipline with exponential service time [12]), i.e., the arrival order is not relevant. In these cases, we identify the global state with the same population of $\mathbf{m}$ for all the stations and classes with the exception of class r of station i which has one more customer than $\mathbf{m}$ with $\mathbf{m} + \mathbf{e}_{ir}$. A station is called *work-conserving* if there is no artificial creation or loss of work in the system [32]. In what follows we refer to QNs assuming state-independent probabilistic routing and the usual independence hypothesis among the service and arrival times.

2.1.2 Properties related to product-form in QNs

Product-form in QNs has been characterised at different levels of abstraction:

- CTMC level: this is the case for local balance, reversibility, quasi-reversibility, Markov implies Markov property. In this context we consider the migration of a customer from a queue to another as the only possible interactions among the queues. However, it is possible for a queue to have an *internal* transition, i.e., a state transition which does not correspond to a job completion or a customer arrival.

- Queueing discipline level: this is the case for the station balance property. In this context the conditions for the product-form are formulated in terms of properties of the queuing discipline of each station in the network. From the modeller point of view, these conditions are probably the most interesting because they give easily-understandable restrictions on the type of queues one may use in order to obtain a product-form model. On the other hand, as we will discuss later on, station balance is a stricter condition than others formulated at the CTMC level.

The following is a review about the characterisations of product-form QNs:

Local balance. This is one of the first properties that has been investigated in product-form theory [14]. Let $\mathbf{m}$ be an ergodic state of a QN, and let $\mathbf{m}' = \mathbf{m} + \mathbf{e}_{ir}$ if r belongs to an open chain, or $\mathbf{m}' = \mathbf{m} + \mathbf{e}_{ir} - \mathbf{e}_{js}$ if r and s belong to a closed chain. Let $\pi(\mathbf{m})$ by the state probability distribution of $\mathbf{m}$. Then the local balance property holds if for all ergodic states $\mathbf{m}$[1]:

$$\pi(\mathbf{m})\mu_{ir}(\mathbf{m}) = a_{ir}(\mathbf{m}')\pi(\mathbf{m}'),$$

where a_{ir} denotes the rate with which the model goes from state $\mathbf{m}'$ to $\mathbf{m}$ due to a class r customer arrival at station i and $\mu_{ir}(\mathbf{m})$ is the service rate for Class r cuastomers are station i. It is worthwhile noting that the local balance is a property defined on the global state and hence, in general, neither easy to check nor modular. When a QN satisfies this property then it has a product-form solution.

The $M \Rightarrow M$ property. The Markov implies Markov property ($M \Rightarrow M$) has been introduced by Muntz very soon in the developing of product-form theory [39]. A work-conserving station satisfies this property if, when considered in isolation, under class-independent Poisson arrival processes it exhibits Poisson independent departure processes. Formally, let $\Gamma^{+r}(\mathbf{m})$ be the set of states in which that station has one more customer of class r with respect to $\mathbf{m}$, let λ_r be the rate of the r-th Poisson arrival process, and $q(\mathbf{m}^{+r} \to \mathbf{m})$ be the transition rate from state $\mathbf{m}^{+r}$ to $\mathbf{m}$. Then the $M \Rightarrow M$ property holds if and only if for all the ergodic states $\mathbf{m}$ and classes r [39]:

$$\lambda_r = \sum_{\mathbf{m}^{+r} \in \Gamma^{+r}} \frac{\pi(\mathbf{m}^{+r})q(\mathbf{m}^{+r} \to \mathbf{m})}{\pi(\mathbf{m})} \quad (1)$$

A network of stations that yield the $M \Rightarrow M$ property has product-form solution.

Reversibility and quasi-reversibility. Reversibility and quasi-reversibility applied to stochastic networks are deeply studied in [33]. We can see QNs as special cases of stochastic networks. A station is reversible if its underlying CTMC is reversible (e.g. $M/M/1$ systems are reversible). A network of reversible stations satisfies the *detailed balance* property and has product-form solution. Let $\mathbf{m}(t)$ denote the state of a station at time t. A station is quasi-reversible if at any epoch t_0, $\mathbf{m}(t_0)$ is independent of: a) the arrival times of class r customers subsequent to t_0 and b) the departure time of class r prior to t_0, with r a generic class of the network. A QN whose stations are quasi-reversible has product-form solution.

It is worthwhile noting that the $M \Rightarrow M$ property, the reversibility and quasi-reversibility require conditions that can be tested for each station of the QN in isolation. This means that the joint process of the QN is not needed and hence these properties are modular with respect to the local balance.

Station balance. This property has been introduced in [15] and is important because it states the condition for the product-form of a station in terms of high level model property. A station yields the station balance if the rate with which a customer at any queue position receives service is proportional to the probability that a customer of the same class will arrive at that position.

[1]Note that local balance may be formulated even for stations with Coxian service time distribution, and scheduling disciplines in which the arrival order is important.

2.1.3 Comparison among the product-form properties

We now present a comparison among the aforementioned product-form properties. In this section we frequently refer to the BCMP theorem that is briefly reviewed in Appendix A. It is important to point out that this comparison holds in the domain of QNs for work-conserving stations. First of all, it is easy to see that QN reversibility implies quasi-reversibility but the opposite is not true, hence the latter property is more general. Indeed, if Jackson's QN consists of reversible exponential queues (M/M/c queue when isolated), BCMP stations are not, in general, reversible. Quasi-reversibility and the $M \Rightarrow M$ property are equivalent as proved in [39, 33]. For scheduling disciplines that do not distinguish customers on the basis of their classes we have that product-form and local balance are equivalent [32]. The local balance implies the $M \Rightarrow M$ property for each station (it suffices to consider a QN consisting of a single station and apply the definition) whereas the opposite is not true when the queueing discipline considers the customer classes. The station balance property is stricter than quasi-reversibility (and hence $M \Rightarrow M$). For instance FCFS BCMP station is quasi-reversible but does not yield station balance. However, the remaining BCMP queueing disciplines satisfy the station balance property. Indeed, the stations that satisfy this property are the only ones that give product-form solution under non-exponential service time distribution [15]. We recall that this comparison that is classical in literature is valid under the set of assumptions given at the beginning of the paragraph.

2.1.4 Other QNs with product-form

Other classes of QNs than BCMP have been identified in literature. Some extensions to the set of the BCMP queueing stations have been proved with the $M \Rightarrow M$ property (see, e.g., [1, 35, 20]). More interesting, from a theoretical point of view, is the class of models known as G-networks introduced in [21]. In these models the conservation law does not hold, i.e., the incoming flow of customers is not preserved as outgoing flow. Indeed, customers may be either standard customers or signals. When a signal arrives at a queue it may originate a non-standard behaviour, and the most common are:

Trigger: at the arrival at a non-empty queue, this signal removes a customer and put it in another queue chosen probabilistically (even none is a valid choice). If the queue is empty the signal vanishes. The product-form is proved in [22].

Catastrophe: at the arrival epoch it empties the destination queue instantaneously. The product-form is proved in [17, 31].

Resets: at the arrival epoch the population of the queue is reset to a given value. The product-form is proved in [23].

The quasi-reversibility property may be extended in order to include the G-networks as proposed in [18]. Differently from work-conserving QNs, G-networks do not yield the one-step behaviour property and, in general, require the solution of a non-linear system of traffic equations.

Product-forms in QNs may arise even for models with finite capacity and blocking as summarised in [5] although the aforementioned properties do not necessary hold. As an instance one may consider a saturated station (state $\mathbf{m}_F$) and easily observe that $M \Rightarrow M$ property does not hold because $\Gamma^{+r}(\mathbf{m}_F) = \emptyset$ for each class r. However, we observe that this class of product-form models does not enjoy the high compositionality properties that quasi-reversible models do. In fact, the proof of the separable solution relies on special properties of the joint state space.

2.2 (G)RCAT characterisation

Product-form has been widely investigated in queueing network domain. However, more expressive models appeared in literature such as the Stochastic Petri Nets (SPNs) [38], several Markovian Process Algebra (MPA) (see, e.g., [30] for the definition of the Performance Evaluation Process Algebra (PEPA)) and Stochastic Automata Network (SAN) [40]. These formalisms allow one to specify models in terms of cooperation of sub-models that may be in general more complicated than the cooperation of queueing networks. In this section we review an important result, the Reversed Compound Agent Theorem (RCAT) [25], and one of its extensions (GRCAT) [37] and show how the class of product-form models it is able to study is very general and, specifically, includes the results based on the quasi-reversibility (and hence $M \Rightarrow M$) on queueing networks. In this section, we first briefly explain the type of cooperation among models -as we will show it should be quite similar to a class of cooperations defined in PEPA and in SAN network theory-. Then we state the main theoretical result, i.e., (G)RCAT.

2.2.1 *Cooperation at CTMC level*

People who are familiar with PEPA synchronisation will surely note the analogies with what we are going to describe in this paragraph. Note that we just deal with pairwise interactions, i.e., a transition in a model may cause a transition just for another model. Consider sub-models S_i and S_j and suppose that we desire to express the fact that a transition labelled with a in S_i can occur only if S_j performs a transition labelled with b, and vice-versa. Specifically, if S_i and S_j are in states s_i, s_j such that they are able to perform a transition labelled with a and b, respectively, that take the sub-models to state s'_i and s'_j, then they can move simultaneously to state s'_i and s'_j. The rate at which this joint transition occurs is decided by the active sub-model that can be S_i or S_j. We express such a cooperation between S_i and S_j, with S_i active, as follows:

$$S_i \underset{(a^+,b^-)}{\overset{y}{\times}} S_j,$$

which means that transitions labelled by a is S_i are active with respect to the cooperation with transitions labelled by b of S_j and originate a models where the joint transitions are labelled with y. Figure 1 shows a graphical representation of such a cooperation. The fact that the resulting model is still Markovian should be obvious because the synchronisation inherits the properties derived for that of PEPA. Note that the major difference is that we can synchronise different labels and assign a different name to the resulting transitions. This happens because we would like a modeller to be able to use a library of models whose labels have a local scope. In this way, the library items can be created independently and instantiated several times in the same model. Note that if a in S_i is active in a cooperation, then all the transitions labelled by a must have a rate in $\mathbb{R}^+$, while if the label is passive the transitions have an unspecified rate with will be denoted by the PEPA symbol $\top$. We say that a model S_i is *closed* if all the transitions have a real rate or is *open* otherwise. Closed models have a well-defined underlying CTMC and may be studied in isolation.

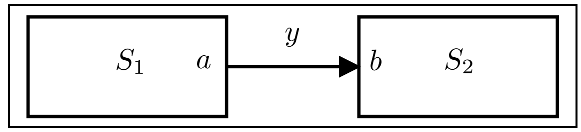

Figure 1: Cooperation between S_1 label a, and S_2 label b. The arrow goes from the active to the passive transitions.

In what follows, $S_1\{a \leftarrow \lambda\}$ is the sub-model S_1 in which all the transitions labelled by a take rate λ.

2.2.2 *(G)RCAT*

In this section we state a very general characterisation of product-form models. With respect to the original version published in [25] we use the formulation that allows multiple pairwise cooperations [27]. Given a model S_i we denote by $\mathcal{A}_i$ ($\mathcal{P}_i$) the set of its labels which participate as active (passive) in some cooperations. However before giving the theorem, we want to help the intuition about what we are doing. Let S be the model defined in terms of the pairwise cooperations of $S_1, \ldots, S_N$. Models S_i are in general open, since some passive labels may be present in their definitions. Finding a product-form solution for S may be seen as the problem of finding a proper closure to $S_1, \ldots, S_N$, i.e., a correct specification for each of their passive labels, such that for each ergodic state of S, $s = (s_1, \ldots, s_N)$, its steady-state probability $\pi(s)$ is proportional to product of the steady-state probabilities of the closed models considered in isolation $\pi_i(s_i)$:

$$\pi(s) \propto \prod_{i=1}^{N} \pi_i(s_i)$$

THEOREM 1 (RCAT). *Let S be the closed model defined as the cooperation of $S_1, \ldots, S_N$ on active labels $\mathcal{A} = \cup_{i=1}^{N} \mathcal{A}_i$ and passive $\mathcal{P} = \cup_{i=1}^{N} \mathcal{P}_i$. Assume that the following structural conditions are satisfied for all S_i:*

1. *if $a \in \mathcal{A}_i$ then each state of S_i has exactly one incoming transition labelled by a*
2. *if $a \in \mathcal{P}_i$ then each state of S_i has exactly one outgoing transition labelled by a*

Furthermore, assume that there exists a set of positive rates $\mathcal{R} = \{K_j, a_j \in \mathcal{A}\}$ such that we can close the models as follows:

$$\forall b_t \in \mathcal{P}_i,\ S_i \underset{(b_t^-,a_j^+)}{\overset{y}{\times}} S_k,\ S_i^c = \{b_t \leftarrow K_j\}$$

and K_j is the reversed rate of all the transitions labelled by a_j in S_k^c. Then for each ergodic state $s = (s_1, \ldots, s_N)$ of S

we have:

$$\pi(s) \propto \prod_{i=1}^{N} \pi_i(s_i),$$

with π_i the steady-state probability distribution of S_i^c.

The computations of the rates of $\mathcal{R}$ is the difficult part when applying this theorem. However, since these values must be the reversed rates of all the active synchronising transitions of each closed sub-model, then a system of possibly non-linear equations can be formulated. This system of equations, called *rate equation system*, has been shown to be equivalent to the traffic equation system when modelling Jackson's QNs. Structural Condition 1 of Theorem 1 can be relaxed in order to include the case of more active transitions with the same label incoming into the same state of a model. This extension, called GRCAT, is studied in [37] where it is proved that value K_j must be the constant sum of the reversed rates incoming into every state of S_k^c. More formally, let s_k be a state of S_k^c, and let $\gamma_k^{a_j}(s_k)$ the set of states s'_k from which s_k is reachable though a transition labelled by a_j with rate $q(s'_k \xrightarrow{a_j} s_k)$. Then, we have that:

$$K_j = \frac{\sum_{s'_k \in \Gamma_j^{a_j}(s_k)} \pi_k(s'_k) q(s'_k \xrightarrow{a_j} s_k)}{\pi_k(s_k)}, \tag{2}$$

by applying the results on revered processes proved in [33].

2.2.3 Comparison with the other product-form characterisations

In this section, we address the problem of relating (G)RCAT with the characterisations of product-form QNs presented in Section 2.1. If we consider the underlying processes of a QN in which we use active transitions to denote job-completion events, and passive transitions to denote customer arrival events, then we can see that Equation (2) becomes Equation (1) by setting $K_j = \lambda_j$, i.e., the arrival rate of class j customers. However, Equation (2) is more general, not only because it allows the analysis of non-working-conserving queues (see [26] for RCAT applied to several G-networks models), but also because we may analyse models with finite states spaces as shown in [36]. Another important aspect to point out concerns the fact that the $M \Rightarrow M$ property implicitly assumes that when studying the queue in isolation under independent Poisson arrival processes, Equation (1) must hold for any set of arrival rates, whereas Equation (2) requires the check only for the rates assumed by the passive transition according to Theorem 1 specification. This may lead to rate-dependent product-form conditions that, in general, QNs do not have.

The direct consequence of GRCAT result is that all quasi-reversible queueing models can be composed, under very general assumptions, with models satisfying RCAT conditions mainintaing the product-form solution.

3. HETEROGENEOUS MODELLING

For heterogeneous modelling we refer to the possibility of specifying a model by means of sub-models expressed by different formalisms. The idea is not new as witnessed for instance by [3, 13] for what concerns the composition of QNs and SPNs. The problem of defining a semantics for the cooperation of stochastic models is not easy, however we aim to give a solution that relies on the cooperation mechanism defined in Section 2.2.1. This simplifies the task since some kinds of cooperation that are allowed within the sub-model specification (e.g., the cooperation between two active transitions as possible in PEPA, or the cooperation among more than two sub-models as allowed by SPNs) are now allowed when specifying their composition. This is motivated by the fact that we aim to obtain a product-form model, and hence we base the semantics of sub-models composition on the pairwise active-passive cooperation which is considered in (G)RCAT formulation. A formalism may be used to specify a sub-model if it allows the modeller to assign labels to the state-transitions in the underlying stochastic process. This is obvious for some formalisms (in general those that allow compositionality and have a strong semantics, such as SPNs, PEPA models, and SAN) but may result difficult for others such as QNs. The problem may be overcome by defining SPN, PEPA or SAN models equivalent to queueing stations. This problem has been addressed in literature, showing that the solution is not obvious especially when multi-class stations are considered (see, e.g., [4, 24, 9, 11]).

4. SOLUTION ALGORITHMS

Stochastic models with product-form solutions are known to be computationally tractable. Several algorithms have been defined in QN domain and some are available for SPNs [19, 41]. These algorithms compute the normalising constant and/or a set of mean performance indices of closed product-form models. However, less attention has been traditionally devoted to the definition of efficient algorithms for the parametrisation of the components that form the whole joint model. Indeed, for QNs this part corresponds to the solution of the system of traffic equations which is known to be linear and usually with a relative small number of equations (the exact dimension depends on the QN structures, i.e., on the number of classes and stations). Even for product-form SPNs of the type studied in [29, 19] the parametrisation of each sub-model requires the solution of a linear set of traffic equations. However, the introduction of G-networks [21] and, more recently, the applications of RCAT [25] have shown that the parametrisation of the sub-models may require the solution of a non-linear set of traffic equations. For this reason we distinguish two types of algorithms: one for the solution of the product-form model (hence they may compute un-normalised steady-state probabilities) and one for the computation of the normalising constant or directly the mean performance indices.

4.1 Solution of the rate equation system

In the RCAT terminology, the equations leading to the determination of the elements of set $\mathcal{R}$ defined in Theorem 1 are called *rate equations*. If we apply RCAT to derive the product-form solution of Jackson's QNs then we observe that the rate equations correspond to the QN traffic equations [25], and a similar result holds for BCMP QNs [37], and G-networks [25, 26]. However, in general, RCAT allows hybrid modelling, and hence the rate equations have not a form which is known a priori.

To the best of our knowledge only two algorithms have been proposed for the solution of RCAT rate equations. The first [2] is based on the symbolic generation of the rate equations and leaves to other software the problem of the solution. Unfortunately, even small models may lead to rate equation systems with high degree whose symbolic solution is computationally unfeasible. However, given the system, numerical techniques may be applied. In this paper we focus on the second algorithm proposed in [36, 6] which relies on purely numerical techniques. The advantages of this approach will be illustrated later, but its very general field of application joint with the efficient computational complexity makes it a good choice for the analysis of heterogeneous product-form models. This is based on an iterative schema which is able both to decide is an RCAT based product-form solution exists and the parametrisation of the sub-models. When the joint-process state space is the Cartesian product of the state spaces of the sub-models then the algorithm also returns the normalised steady-state distribution. The convergence to the correct solution of the iterative schema has been proved only for some classes of models, and hence false-negative answers may occur. However, several practical applications have never exhibited such a undesired behaviour.

4.1.1 INAP

In this part we present the Iterative Numerical Algorithm for Product-form (INAP) which has been defined in [36] and then extended (INAP+) in order to improve the convergence rate and to deal with models with infinite state spaces in [6]. Algorithm 1 illustrates the simplest formulation INAP, i.e., without the optimizations and without the dynamic truncation mechanism that allows the solution of product-form models whose sub-models consist of infinite state spaces. Note also that although the algorithm has been introduced for RCAT models, it may be easily extended in order to include GRCAT ones. We assume to have a set of N cooperating sub-models $S_1, \ldots, S_N$ whose state spaces are denoted by $\mathcal{S}_1, \ldots, \mathcal{S}_N$, $N > 0$. Cooperations are pairwise as specified in Section 2.2.1 and we assume that RCAT structural conditions are satisfied. The algorithm aims to find the correct parametrisation of each sub-model, i.e., the rates to be assigned to the transitions that are passive in the cooperations. Once these rates are found, it checks if the reversed rates of the active transitions in the closed sub-models are constant. INAP starts with a random initialisation of the steady-state probabilities of the closed sub-models and from these it computes the reversed rates of all the active transitions for each sub-model. Suppose that we know the steady-state distribution π_k of closed sub-model S_k^c and that there exists a transition $\alpha \xrightarrow{a} \beta$, where $a \in \mathcal{A}_k$ occurs as an active label in some cooperation. The reversed rate of this transition may be computed as [33]: $q_k(\alpha \xrightarrow{a} \beta)\pi_k(\alpha)/\pi_k(\beta)$. Since the algorithm is iterative, in general, the reversed rates of the active transitions in an inner step of the computation may be not constant. This does not preclude the sub-model to be in product-form when the algorithm converges but we need to choose a single value for the closure of the sub-model with the passive label which synchronises with a. Although we have shown that different strategies may be adopted, we observed experimentally that the fastest convergence rate is obtained by taking the weighted mean of the reversed rates. The weight of each transition reversed rate is the steady-state probability of the destination state. This leads to a very simple formula for the computation of the rate to be used for the closure of the cooperating sub-model:

$$K_a = \sum q_k(\alpha \xrightarrow{a} \beta)\pi_k(\alpha).$$

INAP terminates the iteration in two cases:

- the maximum number of iterations M is reached;
- the distance between two consecutive steady-state distributions is less than a user-defined precision ϵ, for all the sub-models.

At the end, the solution is considered valid if the reversed rates of all the active action types for each process are constant within the precision ϵ or, in other words, the difference between the maximum and the minimum reversed rates corresponding to every active action type must be less than ϵ. The proof of convergence has been given only for special cases.

Although the algorithm may be extended in order to deal with models with infinite state spaces, we point out that for simple cases the standard version is sufficient. This is the case for models whose underlying stochastic process (once they are closed) is a birth-and-death process, e.g., exponential queues and G-queues.

In some cases, e.g., for closed QNs of exponential queues, RCAT rate equations may form a under-determined system and hence infinite solutions are valid. INAP converges to one of the solutions, but unfortunately this may be the trivial one. For instance, if we consider a closed QN of exponential queues (Gordon and Newell QN) INAP may converge to the solution of the traffic equations in which the relative visit ratio to each queue is 0. In this case the modeller should choose a reference station and remove it from the QN. The customers exiting from the reference station will be replaced with an external arrival with arbitrary rate $\lambda > 0$. Other strategies may be needed according to the considered model.

4.2 Computation of the normalising constant and open problems

When (G)RCAT rate equations have a unique solution and the state-space of the joint-model is the Cartesian product of the state spaces of each sub-model, then INAP computes the normalised steady-state distribution of each sub-model and of the joint-model. Although this case is frequent for open systems, many other present more problems in the normalisation of the stationary probabilities of the joint-process. In these cases the computation of the normalising constant may result a difficult task. For BCMP QNs several algorithms have been defined, and the most common are those based on the convolution or on the so-called Mean Value Analysis (although different ones have been developed). We refer to [16, 34] for a review of some of these algorithms. Here, we would like to point out that the all the algorithms on product-form QNs take advantage of the fact that deciding if a joint-state belongs to the ergodic part of the underlying CTMC can be done in constant time. For instance, in a Gordon and Newell QN with K initial customers and N stations any distribution of the customers in the stations is an ergodic state of the underlying CTMC. However, this property is not always satisfied. Probably, the most important case is that of product-form SPNs, in which a Convolution [19] and a Mean Value Analysis [41] algorithms are defined but a property on the reachability of the states must be checked, and hence the exponential complexity of generating the whole state space must be always faced but is some cases in which that property is structurally satisfied.

Obviously, this drawback highly limits the application of these algorithms because when using product-forms one would

Algorithm 1: INAP simplified without optimizations.

Input: agents $S_1, \ldots, S_N$; precision ϵ; maximum number of iterations M

Output: unnormalized stationary distribution π of the joint-process

Randomly initialize π_k for all $k = 1, \ldots, N$

$n = 0$

repeat

  for $k = 1, \ldots, N$ **do**

    $\pi_k^{prev} \leftarrow \pi_k$

  for $k = 1, \ldots, N$ **do**

    foreach $a \in \mathcal{A}_k$ **do**

      Let $j : S_j \times_{(a^+, b^-)} S_k$

      `/*` Λ `is the set of the weighted reversed rates of` a `*/`

      $\Lambda \leftarrow \left\{ q_k(\alpha \xrightarrow{a} \beta)\pi_k(\alpha) \,:\, \alpha, \beta \in \mathcal{S}_k \right\}$

      `/* set the rates of the passive actions */`

      $S_j^c \leftarrow S_j\{b \leftarrow \mathrm{sum}(\Lambda)\}$

  Compute π_k of S_k^c for all $k = 1, \ldots, N$

  $n \leftarrow n + 1$

until $n > M$ ***or*** $\forall k = 1, \ldots, N. \|\pi_k - \pi_k^{prev}\| < \epsilon$;

`/* Check if the reversed rates are constant */`

for $k = 1, \ldots, N$ **do**

  foreach $a \in \mathcal{A}_k$ **do**

    `/*` Λ' `is the set of the reversed rates of` a `*/`

    $\Lambda \leftarrow \{ q_k(\alpha \xrightarrow{a} \beta)\pi_k(\alpha)/\pi_k(\beta) \,:\, \alpha, \beta \in \mathcal{S}_k \}$

    if $\Lambda \neq \emptyset$ ***and*** $\max(\Lambda) - \min(\Lambda) > \epsilon$ **then**

      fail: MARCAT product-form not identified

return $\{\pi_k\}_{k=1,\ldots,N}$

like to take advantage of the separable solution and hence avoid the generation of the joint-state-space. However, observe that with respect to the solution of the CTMC underlying the joint-process, the efficiency of the algorithms presented in [19, 41] is higher since the solution of the linear system of GBEs is not required.

Therefore, we can say that, in general, the computation of the normalising constant is not a trivial task, and we think that future research efforts should be devoted to this topic.

5. A TOOL FOR HETEROGENEOUS MODELLING

The product-form analysis for heterogeneous models can be performed thought a tool that simplifies the model specifications and efficiently decides if the product-form exists and, in this case, computes the steady-state distributions of the model component in isolation. This is possible thanks to the combination of the idea of synchronisation for product-form models that underlies RCAT formulation with INAP. In this brief section we show how it is possible for a tool to work with a library of stochastic models whose transitions are labelled -independently of the formalism which is used to specify them-. Basically, the idea is that the labels used in the model definitions work as input/output interfaces. Hence, the modeller specifies a synchronisation by connecting the label of one model with that of another by a directed arc from the active to the passive. Although the practice is simple, we should point out an aspect. As a trivial illustrative example, consider Jackson's QN of Figure 2-(A) whose underlying processes are depicted in Figure 2-(B). The co-

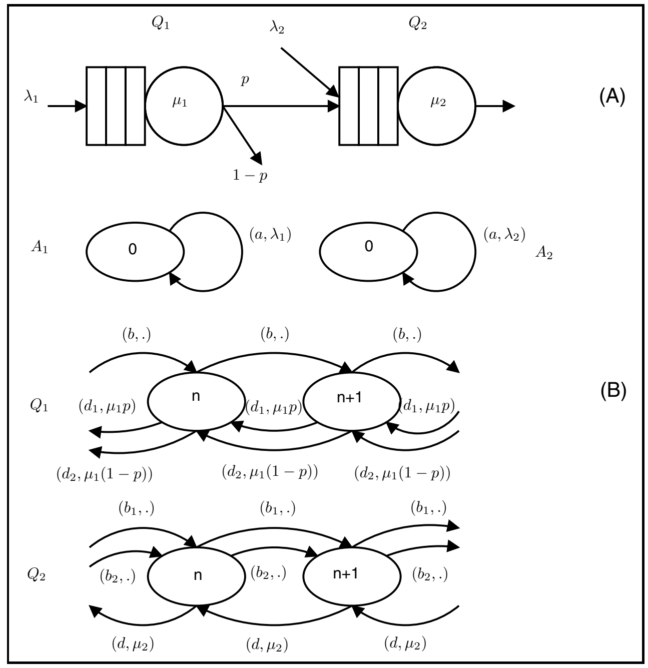

Figure 2: Example of Jackson's QN. (A) The model. (B) The underlying processes.

operations are:

$$A_1 \underset{(a^+, b^-)}{\overset{y_1}{\times}} Q_1 \quad A_2 \underset{(a^+, b_1^-)}{\overset{y_2}{\times}} Q_2 \quad Q_1 \underset{(d_1^+, b_2^-)}{\overset{y_3}{\times}} Q_2$$

Observe that although the model consists of two independent arrival processes and two exponential queues, the modeller should use three library models, since the processes underlying Q_1 and Q_2 are different (note that we assume that once a model is imported one can specify the real number corresponding to symbolic transition rates). This obviously looks unnatural because what the modeller should be able to do is depicted in Figure 3, where Q_1 and Q_2 are instances of the same module. Nevertheless, it is possible to show

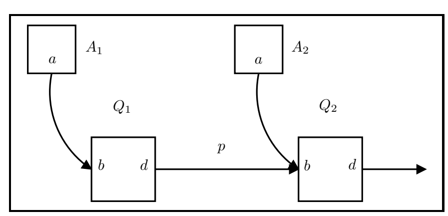

Figure 3: Modular modelling of Jackson's QN of Figure 2.

that the passage from the modular specification of Figure 3 (which includes probabilistic synchronisations and multiple active labels synchronising with the same passive) to that of Figure 2-(B) may be performed automatically [7].

6. PRACTICAL APPLICATION

In this section we apply the theoretical results and methodologies that we have previously illustrated to an illustrative example. We first describe the system which will be analysed, then we provide a product-form model and discuss the underlying assumptions. Finally, we show how the stationary distribution, and hence the performance measures, may be derived by applying INAP algorithm joint with the results about product-form models that have been studied in literature.

6.1 System description

We consider a software architecture in which two classes of customers compete for two distinct services, see Figure 4 for an informal diagram describing the system. Customers (of both classes) arrive at the communication line from the outside. Once they are sent to the processing phase, they are served and the results are sent back through the same communication line. Then, the results are analysed and if the answer is considered acceptable then the computation terminates, otherwise the customer is sent back to the system. The processing of Class 1 and 2 customers are not independent and some (even catastrophic) interferences may occur. We now describe each component of the system in details.

- *Communication lines:* The communication infrastructure consists of a K independent lines which can work in parallel. As a policy for the flux-control the sending processes must have a maximum window size of $n < K$. In other words, the processes generating the requests can have at most n packets being sent. The other ones are queued according to a First Come First Served (FCFS) scheduling discipline. Observe that there are four groups of customers that use the communication lines: Classes 1 and 2 from the outside to the processing phases and in the opposite direction.

- *3-phases processing:* Both the services for the customers of Class 1 and 2 consist of 3 phases. The first phase is computed according to a standard round-robin scheduling while the second and third according to a FCFS one. With a fixed probability, a job completion at the first phase of Class 2 invalidates all the computations at the phase 1 of Class 1, and hence they are discarded. Vice versa, a job completion at phase 2 of Class 1 may destroy the current computation at phase 2 of Class 2. Also phases 3 of the computation may conflict. However, in this case a recovery procedure can be performed and the job processing at phase 3 are done again without customer loss.

- *Validation of the result:* The results are validated according with a round-robin scheduling discipline and this requires a fixed amount of computation for each request.

We aim to compute the throughput of the system for Class 1 and Class 2 customers, and, as an instance, the mean response time of the third phase of service.

6.2 Model description

We propose a model of the system consisting of 8 interconnected blocks, as depicted by Figure 5. The description of the blocks follows.

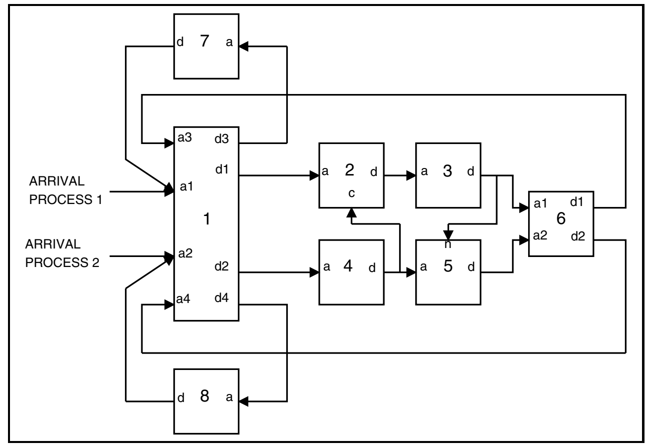

Figure 5: Model of the system of Figure 4: block structure.

Arrival processes. As usual, in order to obtain a product-form solution, we assume the arrival processes are independent Poisson processes with rate λ_1 and λ_2 for Classes 1 and 2, respectively.

Communication lines (BLOCK 1). The communication lines are modelled by a Multiple Server with Concurrent Classes of Customers (MSCCC) station [35] in the generalised version proposed in [20]. Customers belong to one of four groups labelled from 1 to 4: group 1 (2) customers are those of Class 1 (2) directed to computation phases, while group 3 (4) customers are those of Class 1 (2) coming from the computation phases. At any time, at most K customer can be processed and, among these, at most n can belong to the same group. Service time is exponentially distributed with mean $1/\mu_1$. Note that if $n \geq K$, then the station is a simple multi-class, multi-server queueing station and if $n = 1$ then we have the MSCCC station proposed in [35]. This type of service station is known to be quasi-reversible. According to the model depicted in Figure 5 the transitions corresponding to customer arrivals are labelled by $a1, \ldots, a4$ and are passive in the cooperations (incoming arrows), whereas those corresponding to customer departures are labelled by $d1, \ldots, d4$ and are active in the cooperations (outgoing arrows). Under this assumptions on the roles of the transitions in the cooperations, we can say that quasi-reversibility implies that GRCAT conditions are satisfied [37]. We refer to the original papers for a detailed and formal description of the model and its analysis. It is worthwhile pointing out that we may see this complicated block as a black-box, meaning we do not actually need to know if its definition is given in terms of process algebra equations or GSPN structure.

Class 1, phase 1 (BLOCK 2). This part of the model consists of a queueing station with Processor Sharing (PS) queueing discipline and exponential distributed service time with mean $1/\mu_2$. Block 4 can cause some events that destroy all the customers being processed at the station. Therefore, we must consider some catastrophe transitions, i.e., transitions that take the model from any state to the state representing the empty station. The stochastic process underlying this block is depicted by Figure 6. Observe that this station is not quasi-reversible (the arrival flow is not preserved) even if it is known [17] that a network of such stations has

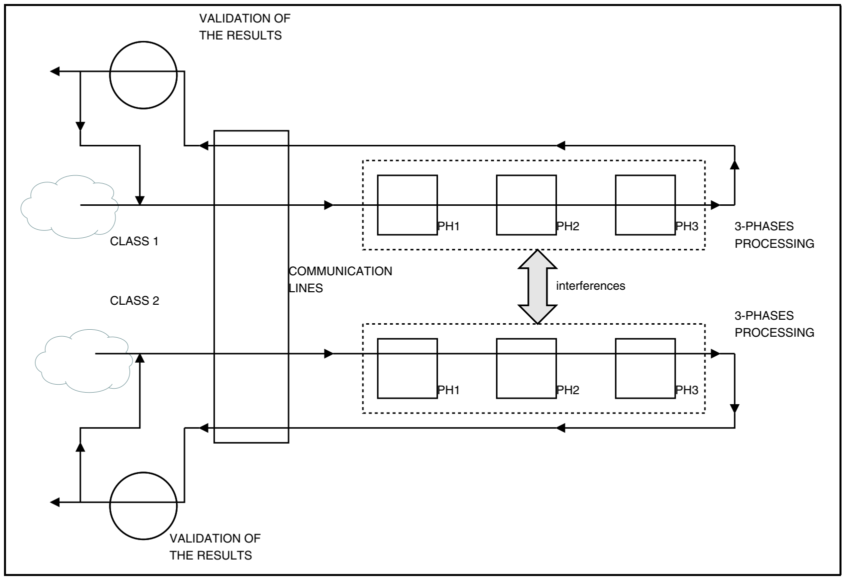

Figure 4: System analysed in Section 6.

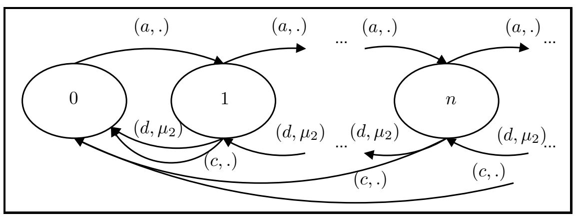

Figure 6: Underlying process of a queue with catastrophe.

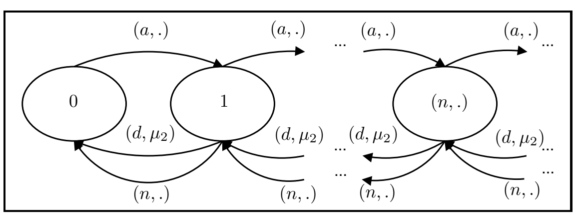

Figure 7: Underlying process of a queue with negative customers.

.

product-form solution. In [26] the author proved that the process depicted in Figure 6 satisfies RCAT conditions.

Class 2, phase 2 (BLOCK 5). This block is very similar to BLOCK 2. However, the interference with BLOCK 3 causes the destruction of just one customer, the one being served. We have another example of a non-quasi-reversible process (see Figure 7) whose composition with others with the same structure yields a product-form solution [21]. This model satisfies RCAT conditions as proved in [25].

Classes 1 and 2, phase 3 (BLOCK 6). We represent this block by means of the SPN depicted by Figure 8. In this case, SPN transitions labelled by T_{a1} and T_{a2} are associated with the process transitions labelled by $a1$ and $a2$ that are passive in the cooperations (according to Figure 5), whereas SPN transitions T_{d1} and T_{d2} are associated with process

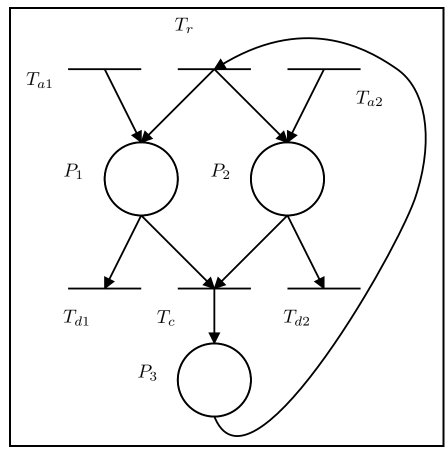

Figure 8: SPN describing BLOCK 6 of the model of Figure 5.

transitions labelled by $d1$ and $d2$ that are active in the cooperations. All the transitions has exponentially distributed service time. The rates of T_{d1} and T_{d2} are the service rates of phase 3 for class 1 and 2, μ_{61} and μ_{62}, respectively. Transition T_c models the conflicts that may occur during phase 3 processing and its rate must be determined according to the probability of conflict. Let us consider that both places P_1 and P_2 contains a customer. The probability of conflict p_c is equal to the probability that T_C removes both these customers before that the fastest between T_{D1} and T_{D2} completes its work. This latter time is an exponential random variable with rate $\mu_{61} + \mu_{62}$ and hence we have $p_c = \mu_c/(\mu_{61} + \mu_{62})$. Transition T_r models the repair of the requests after a conflict. This SPN belongs to the class of product-form models studied in [28] which is strictly related with that studied in [29, 19].

BLOCKS 3,4,7,8. The remaining blocks are considered as simple exponential queues whose arrivals transitions are passive in the cooperations whereas the active ones are active. Note that this choice does not take into account the differences among the scheduling disciplines of the stations. However, the average performance indices are not affected by this simplification by the well-know properties of the quasi-reversible queues. The service rates of stations associated with blocks 3, 4, 7, 8 are μ_3, μ_4, μ_7 and μ_8, respectively.

Modelling the cooperations. In order to complete the model description we discuss the synchronisations. With the purpose of giving a shorter and more readable list of cooperations we use

$$S_7 \underset{(d^+,a1^-)}{\overset{y_{71},p_{71}}{\times}} S_1$$

to denote that BLOCK 7 synchronises with BLOCK 1 with probability p_{71}. The other cooperations are:

$$S_1 \underset{(d1^+,a^-)}{\overset{y_{12}}{\times}} S_2 \quad S_2 \underset{(d^+,a^-)}{\overset{y_{23}}{\times}} S_3 \quad S_3 \underset{(d^+,a1^-)}{\overset{y_{36},p_{36}}{\times}} S_6$$

$$S_3 \underset{(d^+,n^-)}{\overset{y_{35}^{-},p_{35}^{-}}{\times}} S_5 \quad S_1 \underset{(d2^+,a^-)}{\overset{y_{14}}{\times}} S_4 \quad S_4 \underset{(d^+,a^-)}{\overset{y_{45},p_{45}}{\times}} S_5$$

$$S_4 \underset{(d^+,c^-)}{\overset{y_{42}^{-},p_{42}^{-}}{\times}} S_2 \quad S_5 \underset{(d^+,a2^-)}{\overset{y_{56}}{\times}} S_6 \quad S_6 \underset{(d2^+,a4^-)}{\overset{y_{612}}{\times}} S_1$$

$$S_6 \underset{(d_1^+,a_3^-)}{\overset{y_{611}}{\times}} S_1 \quad S_1 \underset{(d3^+,a^-)}{\overset{y_{17}}{\times}} S_7 \quad S_1 \underset{(d4^+,a^-)}{\overset{y_{18}}{\times}} S_8$$

$$S_8 \underset{(d^+,a2^-)}{\overset{y_{81},p_{81}}{\times}} S_1 \quad A_1 \underset{(a^+,a1^-)}{\overset{y_{01}}{\times}} S_1 \quad A_2 \underset{(a^+,a2^-)}{\overset{y_{02}}{\times}} S_1$$

The last two synchronisations model the arrival of the customers from the outside where the arrival processes A_1 and A_2 are identical to those showed in Figure 2. The obvious restriction on the probabilities are assumed: $0 \le p_i \le 1$ for all i, $p_{42}^{-} + p_{45} = 1$, $p_{35^-} + p_{36} = 1$, $p_{71} < 1$ and $p_{81} < 1$. Let use denote by x_i the rate that (G)RCAT associates with the passive transitions in the cooperation labelled by y_i. Knowing these rates allow us to consider the BLOCKs in isolation.

6.3 Analysis

The previous part of the section has shown that each of the blocks which form the model of Figure 5 satisfies (G)RCAT and hence they may be composed to obtain a product-form model. The following steps of the model analysis are: 1) the parametrisation of each BLOCK and 2) its analysis in isolation.

Parametrisation of each BLOCK.

We propose two solutions of this problem. The first one is based on setting up RCAT rate equations and leaves their solution to dedicated software. The second one uses INAP as illustrated in Algorithm 1. In order to derive (G)RCAT rate equations we must consult the list of cooperations and apply the results proved in literature. Since BLOCK 1 is a quasi-reversible station, then we straightforwardly have $x_{12} = x_{71} + \lambda_1$, where the right hand-side is the total rate assigned to the arrival transitions labelled by $a1$. Similarly, we derive: $x_{17} = x_{611}$, $x_{18} = x_{612}$, $x_{14} = \lambda_2 + x_{81}$. The other quasi-reversible stations are BLOCKs 3, 4, 7, 8 whose analysis gives the equations: $x_{36} = x_{23}p_{36}$, $x_{35}^{-} = x_{23}p_{35}^{-}$, $x_{45} = x_{14}p_{45}$, $x_{42}^{-} = x_{14}p_{42}^{-}$, $x_{71} = x_{17}p_{71}$, $x_{81} = x_{18}p_{81}$. For what concerns BLOCK 6, in [28] we prove that the reversed rates of the transitions labelled by $d1$ ($d2$) is equal to the forward rate of those labelled by $a1$ ($a2$) independently of the rates of μ_r and μ_c. Therefore, we have: $x_{611} = x_{36}$ and $x_{612} = x_{56}$.

BLOCK	Parameters
BLOCK 1	$\mu = 0.5$, $K = 20$, $n = 15$
BLOCK 2	$\mu_2 = 6$
BLOCK 3	$\mu_3 = 3.2$, $p_{36} = 0.85$, $p_{35}^{-} = 0.15$
BLOCK 4	$\mu_4 = 7$, $p_{42}^{-} = 0.10$, $p_{45} = 0.90$
BLOCK 5	$\mu_5 = 6$
BLOCK 6	$\mu_{61} = 4.1$, $\mu_{62} = 7$, $\mu_C = 1.5$, $\mu_R = 5$
BLOCK 7	$\mu_7 = 12$, $p_{71} = 0.20$
BLOCK 8	$\mu_8 = 12$, $p_{81} = 0.25$

Table 1: Example of parametrisation of the model depicted in Figure 5.

More attention should be devoted to BLOCKs 3 and 5. In the former, the reversed rate of the transitions labelled by d is [26]:

$$x_{23} = \frac{(x_{12} + \mu_2 + x_{42}^{-}) - \sqrt{(x_{12} - \mu_2 - x_{42}^{-})^2 + 4x_{12}x_{42}^{-}}}{2}.$$

Transitions labelled by d in the latter block have the following reversed rate [25]: $x_{56} = \mu_5 x_{45}/(x_{35}^{-} + \mu_5)$. Solving the system for all x_i gives the correct parametrisation of each sub-model that can eventually be studied in isolation. The difficult points of this approach are basically two. The first one concerns the ability of the modeller of knowing the results from the literature that allows the definition of RCAT rate equations. The second one is that the system of equations is non-linear and, although in this case a symbolic solution may be obtained (although its expression results quite difficult to manipulate), often this is not computationally efficient.

The alternative approach relies on the application of the numerical algorithm INAP. In this case we do not need to set up the rate equation systems and just need to draw a model very similar to that depicted by Figure 5 and enter the numerical parametrisation. From a practical point of view, the modeller defines the system by choosing the appropriate sub-models from a library and connects them by arcs. (G)RCAT conditions may be automatically checked and the value for the passive transitions are computed. It is worthwhile noting that, for this computation, the quasi-reversible stations may be replaced by simple Jackson's stations thus simplifying the computation. Let us assume the numerical parametrisation given in Table 1. For this case, the numerical solution is:

$$\begin{array}{lll} x_{17} = 2.46575, & x_{18} = 5.3096, & x_{36} = 2.46575, \\ x_{45} = 5.69466, & x_{71} = 0.49315, & x_{81} = 1.3274, \\ x_{611} = 2.46575, & x_{612} = 5.3096, & x_{14} = 6.3274, \\ x_{23} = 2.90088, & x_{56} = 5.3096, & x_{35}^{-} = 0.435132, \\ x_{12} = 3.49315, & x_{42}^{-} = 0.63274 & \end{array} \tag{3}$$

Analysis of the blocks in isolation.

Since the blocks are now parametrised and the system is open, the analysis may be carried on quite easily. In particular, we refer to [20] for the effective computation of the distribution of the number of customers and its expected value of BLOCK 1. For what concerns BLOCK 6 the derivation of the stationary distribution and the mean performance indices is done in [10]. Let m_1, m_2 and m_3 be the number of customers in P_1, P_2, and P_3. Then the steady-state distribution π_6 is given by:

$$\pi_6(m_1, m_2, m_3) = \left(1 - \frac{x_{36}}{\mu_{61}}\right)\left(1 - \frac{x_{56}}{\mu_{62}}\right)\left(1 - \frac{\mu_c x_{36} x_{61}}{\mu_{61}\mu_{62}}\right) \cdot \left(\frac{x_{36}}{\mu_{61}}\right)^{m_1}\left(\frac{x_{56}}{\mu_{62}}\right)^{m_2}\left(\frac{\mu_c x_{36} x_{61}}{\mu_{61}\mu_{62}}\right)^{m_3}.$$

From this, after some trivial algebra we may derive the mean number of customers of Class 1 at BLOCK 6 ($\overline{N}_{61}$), in steady-state as the sum of the mean number of customers in P_1 and P_3:

$$\overline{N}_{61} = x_{36}\left(\frac{1}{\mu_{61} - x_{36}} + \frac{x_{56}\mu_c}{\mu_r \mu_{61}\mu_{62} - x_{36}x_{56}\mu_c}\right).$$

Note that since there is not loss of customers, then we may apply Little's result in order to derive the mean response time at this station. For instance, for Class 1 customers this is given by: $\overline{R}_{61} = N_{61}/x_{36}$. In our numerical example we obtain $\overline{N}_{61} = 1.6674$ and $\overline{R}_{61} = 0.67620$.

Finally, an interesting performance measure is the throughput to the outside of BLOCK 7 (BLOCK 8) which gives us the reduction on the incoming flow of Class 1 (Class 2) customers due to the interference along the processing phases. The throughput of the system for Class 1 customers is $X_1 = x_{17}(1 - p_{71})$, that in the numerical example is $X_7 = 1.9726$.

7. CONCLUSION

In this paper we reviewed some recent results about product-form models and showed how they may be applied in order to obtain a heterogeneous modelling technique. Specifically, we focused our attention on RCAT [25] and its formulation given in [37]. These results have been compared with previous ones defined for queueing networks. Product-form models play an important role for the performance evaluation community thanks to their efficiently tractable solutions. Nevertheless, we pointed out some practical problems: the first concerns the computation of the parameters that allow the analysis of the sub-models in isolation and the second concerns the computation of the normalising constant for closed or mixed models. We have reviewed INAP, i.e., a general numerical algorithm that aims to solve the former problem, whereas the definition of a general algorithm for the computation of the normalising constant, as far as we know, is still an open problem. Finally, we presented an example of a system with a corresponding open model (hence avoiding the problem of normalising the probabilities) that consists of several interacting parts defined by means of different formalisms. Some average performance indices have been derived exploiting the separable steady-state distribution. In our opinion, a pivotal importance for future research efforts assume the problem of the normalisation of the probabilities in case of closed and mixed models, and the definition of general techniques to approximate non-product-form models by means of product-form ones.

8. REFERENCES

[1] Afshari, P. V., Bruell, S. C., and Kain, R. Y. Modeling a new technique for accessing shared buses. In *Proc. of the Computer Network Performance Symp.* (New York, NY, USA, 1982), ACM Press, pp. 4–13.

[2] Argent-Katwala, A. Automated product-forms with meercat. In *SMCtools 06: workshop on Tools for solving structured Markov chains* (Pisa, Italy, 2006), vol. 201, p. 10.

[3] Balbo, G., Bruell, S. C., and Ghanta, S. Combining queueing network and generalized stochastic Petri nets for the solution of complex models of system behavior. *IEEE Trans. on Computers 37* (1998), 1251–1268.

[4] Balbo, G., Bruell, S. C., and Sereno, M. On the relations between BCMP Queueing Networks and Product Form Solution Stochastic Petri Nets. *Proc. of 10th Int. Workshop on Petri Nets and Performance Models, 2003.* (2003), 103–112.

[5] Balsamo, S., De Nitto Persone', V., and Onvural, R. *Analysis of Queueing Networks with Blocking.* Kluwer Academic Publishers, 2001.

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

[7] Balsamo, S., Dei Rossi, G., and Marin, A. A tool for the numerical solution of cooperating Markov chains in product-form. In *Proc. Of Int. Conf. HET-NETs* (Zakopane, PL, January, 2010), pp. 311–324.

[8] Balsamo, S., Marco, A. D., Inverardi, P., and Simeoni, M. Model-based performance prediction in software development: a survey. *IEEE Transactions on Software Engineering 30*, 5 (May 2004), 295–310.

[9] Balsamo, S., and Marin, A. On representing multiclass M/M/k queues by generalized stochastic Petri nets. In *Proc. of ECMS/ASMTA-2007 Conf.* (Prague, Czech Republic, 4-6 June 2007), pp. 121–128.

[10] Balsamo, S., and Marin, A. Separable equilibrium distributions for systems with unreliable wireless links. In *Wireless Days (WD), 2010 IFIP* (Venice, Italy, Oct. 2010), pp. 1 –5.

[11] Balsamo, S., and Marin, A. Representing LCFSPR BCMP service centers with Coxian service time distribution. In *Proc. of Valuetools '07 Conf.* (Nantes, France, October 23-25, 2007).

[12] Baskett, F., Chandy, K. M., Muntz, R. R., and Palacios, F. G. Open, closed, and mixed networks of queues with different classes of customers. *J. ACM 22*, 2 (1975), 248–260.

[13] Bause, F. 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.

[14] Chandy, K. M. The analysis and solutions for general queueing networks. In *Proc. of 6th Annual Princeton Conf. on Inform. Sci. and Systems* (Princeton, N.J., March, 1972), pp. 219–223.

[15] Chandy, K. M., J. H. Howard, J., and Towsley, D. F. Product form and local balance in queueing networks. *J. ACM 24*, 2 (1977), 250–263.

[16] Chandy, K. M., and Sauer, C. H. Computational algorithms for product form queueing networks. *Commun. ACM 23*, 10 (1980), 573–583.

[17] Chao, X. A queueing network model with catastrophes and product form solution. *Op. Res. Letters 18*, 2 (1995), 75–79.
[18] Chao, X., Miyazawa, M., and Pinedo, M. *Queueing networks: customers, signals and product form solutions*. Wiley, 1999.
[19] Coleman, J. L., Henderson, W., and Taylor, P. G. Product form equilibrium distributions and a convolution algorithm for Stochastic Petri nets. *Perform. Eval., Elsevier 26* (1996), 159–180.
[20] Crosby, S., and A.E., K. Product for solutions for multiserver centres with concurrent classes of customers. *Perf. Eval. 11* (1990), 265–281.
[21] Gelenbe, E. Product form networks with negative and positive customers. *Journal of Applied Prob. 28*, 3 (1991), 656–663.
[22] Gelenbe, E. G-networks with triggered customer movement. *J. of Appl. Prob.* (1993), 742–748.
[23] Gelenbe, E., and Fourneau, J. M. G-networks with resets. *Perform. Eval., Elsevier 49*, 1-4 (2002), 179–191.
[24] Harrison, P. G. Reversed processes, product forms, non-product forms and a new proof of the BCMP theorem. In *Int. Conf. on the Numerical Solution of Markov Chains (NSMC 2003), Urbana IL, USA, September 2-5 2003* (September 2003), pp. 289–304.
[25] Harrison, P. G. Turning back time in Markovian process algebra. *Theoretical Computer Science 290*, 3 (January 2003), 1947–1986.
[26] Harrison, P. G. Compositional reversed Markov processes, with applications to G-networks. *Perform. Eval., Elsevier 57*, 3 (2004), 379–408.
[27] Harrison, P. G., and Lee, T. T. Separable equilibrium state probabilities via time reversal in Markovian process algebra. *Theoretical Computer Science 346*, 1 (2005), 161–182.
[28] Harrison, P. G., and Marin, A. A compositional construction of product-form Petri-nets using Ercat. Tech. rep., Department of Computing, Imperial College, 2009.
[29] Henderson, W., Lucic, D., and Taylor, P. G. A net level performance analysis of Stochastic Petri Nets. *J. Austral. Math. Soc. Ser. B 31* (1989), 176–187.
[30] Hillston, J. *A Compositional Approach to Performance Modelling*. PhD thesis, Department of Computer Science, University of Edinburgh, 1994.
[31] J.M. Fournau, L. Kloul, F. Q. G-networks with jumps back to zero. In *Proc. of MASCOTS '95* (Duram, NC, USA), pp. 28–32.
[32] Kant, K. *Introduction to Computer System Performance Evaluation*. McGraw-Hill, 1992.
[33] Kelly, F. *Reversibility and stochastic networks*. Wiley, New York, 1979.
[34] Lavenberg, S. S. *Computer Performance Modeling Handbook*. Academic Press, New York, 1983.
[35] Le Boudec, J. Y. A BCMP extension to multiserver stations with concurrent classes of customers. In *SIGMETRICS '86/PERFORMANCE '86: Proc. of the 1986 ACM SIGMETRICS Int. Conf. on Computer performance modelling, measurement and evaluation* (New York, NY, 1986), ACM Press, pp. 78–91.
[36] Marin, A., and Bulò, S. R. 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.
[37] Marin, A., and Vigliotti, M. G. A general result for deriving product-form solutions of markovian models. In *Proc. of First Joint WOSP/SIPEW Int. Conf. on Perf. Eng.* (San Josè, CA, USA, 2010), ACM, pp. 165–176.
[38] Molloy, M. K. Performance analysis using stochastic petri nets. *IEEE Trans. Comput. 31*, 9 (1982), 913–917.
[39] Muntz, R. R. Poisson departure processes and queueing networks. Tech. Rep. IBM Research Report RC4145, Yorktown Heights, New York, 1972.
[40] Plateau, B. On the stochastic structure of parallelism and synchronization models for distributed algorithms. *SIGMETRICS Perform. Eval. Rev. 13*, 2 (1985), 147–154.
[41] Sereno, M., and Balbo, G. Mean Value Analysis of stochastic Petri nets. *Perform. Eval. Elsevier 29* (1997), 35–62.
[42] Smith, C. U., and Williams, L. G. Five steps to establish software performance engineering. In *CMG Conf.* (Reno, Nevada, USA, 2006), pp. 507–516.

APPENDIX

A. THE BCMP THEOREM

BCMP QNs consist of stations with 4 types of disciplines: First Come First Served (FCFS), Last Come First Served with Preemptive Resume (LCFSPR), Processor Sharing (PS) and Infinite Server (IS) (see [12] for a formal description). The QN may have open and closed chains, and the routing is state-independent. Class switching is allowed, and $p_{ir,js}$ denotes the probability that a customer enters station j with class s after a job completion at station i class r. $p_{ir,0}$ denotes the probability that a customer leaves the system after a job completion at station i, class r belonging to an open chain. Note that while FCFS stations must have exponential service time distribution whose rate does not dependent on the class of the customer in service, the remaining 3 types can have Coxian distributed service time whose parameters depend on the customer classes. Since the QN is in product-form, we can study each of the N station in isolation under independent Poisson arrival processes whose rate is e_{ir} for station i and class r. e_{ir} is the solution of a linear system of equations called traffic equation system which is defined for each QN chain. Let $\mathcal{R}^{(c)}$ the set of classes belonging to chain c, then the corresponding traffic equation system is:

$$e_{ir} = \lambda_{1r} + \sum_{j=1}^{N} \sum_{s \in \mathcal{R}^{(c)}} e_{js} p_{js,ir}, \qquad i = 1, \ldots, N \quad r \in \mathcal{R}^{(c)} \tag{4}$$

where λ_{ir} denotes the external arrival rate at queue i with class r belonging to an open chain. Note that for closed chains System (4) is under-determined, and all the non-trivial solutions differ for a multiplying factor. In this cases, any non-trivial solution can be chosen but then a normalizing constant must be computed.