Deriving the rate equations characterising product-form models and application to propagating synchronisations

Peter G. Harrison
Dept. of Computing, Imperial College London
South Kensington Campus
London, SW7 2AZ, U.K.
Email:pgh@doc.ic.ac.uk

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

***Abstract*—Performance engineering often uses stochastic modelling as a powerful approach to the quantitative analysis of real systems. Product-form Markovian models have the property that the steady-state analysis can be carried out efficiently and without the need for solving the system of global balance equations. The Reversed Compound Agent Theorem (RCAT) gives sufficient conditions for the model to have a product-form solution. In this paper we show its application in the case of instantaneous synchronisations of more than two components at the same time. Although examples of this class of product-form models are already known, the results shown here are novel. We introduce the idea of Propagation of Instantaneous Transitions (PITs) to model multi-way synchronisations as successive pairwise ones in the case of product-form. An algorithm that derives the system of equations that must be solved to obtain the steady-state distribution is presented. Two examples of new product-form models are then derived as a consequence of these contributions. The first is a queueing network with finite capacity nodes, a skipping policy, and partial flushing as a congestion handling mechanism. The second is a queueing network with nodes that may have negative queue lengths, where an unbounded customer deletion mechanism is introduced.**

***Index Terms*—Queueing theory, Product-form solutions.**

I. Introduction

Stochastic models play an important role in the performance evaluation of computer systems. In particular, models with underlying continuous time Markov chains (CTMCs), such as queueing networks [18], Markovian process algebra [15], stochastic Petri nets [19], have been widely used to study the performance and reliability of both hardware and software architectures. However, when systems consist of several interacting components, stochastic analysis faces the problem of the state space explosion. Informally, the number of states of the model representing the system tends to grow exponentially or combinatorially with the number of components. Product-forms can tackle models with this behaviour in an efficient way by means of a structural decomposition that facilitates the computation of the model's equilibrium probabilities as the (normalised) product of the equilibrium probabilities of its components considered in isolation. Known models with product-form solution have been defined in terms of various formalisms, e.g. [3], [6], [2]. In [11], the problem of characterising a large class of product-form models is addressed and the Reversed Compound Agent Theorem (RCAT) is proved. Based on this result, several well-known product-forms have been re-proved in a modular and compact way and new ones have been derived. One of the challenges of carrying out a product-form analysis is the decomposition of the model into separate components. Roughly speaking, each of these components must be parametrised in order to take into account the effect of the others. In queueing networks of the type studied in [3], it is required to solve the *linear* system of traffic equations. However, in the introduction of G-networks [6], Gelenbe showed that more complex and expressive networks encompassing both positive and negative customers have product-forms with non-linear traffic equations. This topic has been further explored in [11], where it is shown how to derive the system of equations that characterises the product-form model, starting from its definition at a low level of abstraction. More specifically, the cooperating components are defined in terms of cooperating CTMCs. Since we can therefore deal with more general models than queueing networks, we refer to this system of equations as the *rate equations* of the model. Differently from what happens with queueing networks, when applying RCAT, more than one equation may be associated with the same component. These constraints may cause rate-dependent product-form conditions. This paper addresses the problem of deriving algorithmically the rate equations associated with a given a model and its components. The existence of the rate equations' solution ensures the product-form of the model. The idea of the algorithm is novel and is based on a recursion on the number of interactions among the components. Although, for the sake of readability, we mainly study queueing networks with some special behaviours, the algorithm we propose can be applied to any type of model, even one consisting of heterogeneous components. The second contribution of the paper is to show the algorithm at work in the case of models with non-pairwise synchronisations, such as those studied in [20], [8], [5]. All these models share the characteristic that at a given epoch, more than two components can instantaneously

VALUETOOLS 2012, October 09-12, Cargèse, France

DOI 10.4108/valuetools.2012.250312

change their state, i.e., the type of synchronisation is more sophisticated than the one studied in [3]. In [12], [1], it is shown that iterated applications of RCAT can deal with special cases of such models. In the present work, we formalise such synchronisations with product-form by introducing the idea of propagation of instantaneous transitions (PITs) and we show how to derive algorithmically the system of rate equations. This is particularly challenging in the case of chains or even cycles of PITs. Finally, we study two new product-form models based on PITs. The former is a queueing network with finite capacity, where the skipping policy is adopted to handle arrivals at saturated queues [20], [1] and a congestion-handling algorithm is modelled by means of a partial flushing of the queues. The latter is a network of queues with positive and negative customers, also having positive or negative queue lengths, as well as partial flushing (chains of PITs modelling customer deletions). With respect to other models proposed in the literature, e.g. [5], this network has the property that, given the state, the number of deleted customers is finite but unbounded.

The remainder of the paper is the following. Section II introduces the notions of PITs and chains and Section III gives a process algebraic notation to describe such a class of product-form models. Section IV presents the algorithm to derive the system of rate equations and shows an application to a toy-example. In Section V, the new product-forms for the networks of finite capacity queues with flushing and for queues with positive and negative length are derived. Finally, Section VI gives some concluding remarks.

II. Chains and PITs

Product-form models with synchronisations defined in terms of propagating instantaneous signals are not new. The first remarkable result was obtained by Gelenbe in [8], where G-networks with triggers were defined. G-networks [6] are a class of product-form queueing networks in which positive and negative customers are allowed; positive customers behave exactly as the customers of Jackson's queueing networks [16], increasing the queue length at their arrival epochs. Negative customers may arrive from the outside at a queueing station or a positive customer may switch to a negative one after a service completion at another node, according to a state-independent routing probability matrix. At its arrival epoch at a node, a negative customer deletes a positive one if any is present, or it vanishes otherwise. In [8], it is shown that negative customers may act as triggers, i.e., move a customer from a non-empty queue to another queue chosen probabilistically. From a theoretical point of view, the proof that this model is still in product-form is important because, differently from previous models [16], [3], [6], timed synchronisations are not pairwise, meaning that more than two model components (i.e. constituent processes) can change their states simultaneously. A general characterisation of this class of models based on an extended formulation of Kelly's *quasi-reversibility* is given in [4, Ch. 5]. The idea of multi-wise instantaneous synchronisation has been further extended in the class of G-networks investigated by Fourneau et al. [5] to encompass *chains* of instantaneous state changes. A more general theoretical analysis of this property is carried out in [12], [1] where it is shown that this type of synchronisation can be modelled as the propagation of instantaneous transitions. The advantage of this approach is twofold: it allows for a specification of the models in terms of pairwise synchronisations and, as a consequence, the product-form can be derived as an iterative application of the Reversed Compound Agent Theorem (RCAT). In order to clarify how RCAT can be applied to derive the product-forms of networks of cooperating Markov processes with Propagation of Instantaneous Transitions (PITs), we utilise the example based on G-networks depicted in Figure 1. Notice that this introductory example belongs to the class of G-networks studied in [8] and also characterised in [4, Ch. 5]. Customers

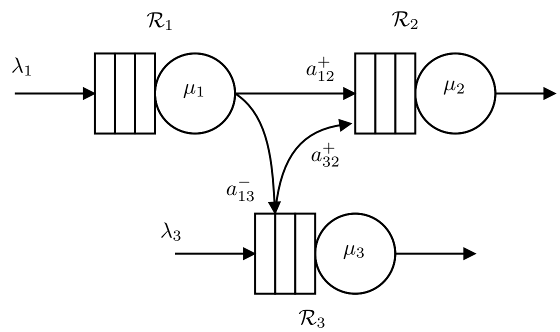

Fig. 1. G-network with trigger generated after job completion in $\mathcal{R}_1$ that moves a customer from $\mathcal{R}_3$ to $\mathcal{R}_2$.

arrive from the outside at stations $\mathcal{R}_1$ and $\mathcal{R}_3$ according to independent, homogeneous Poisson processes with rates λ_1 and λ_3, respectively. Service times at the stations are exponential random variables, which are independent, with rates μ_1, μ_2, μ_3. After a job completion at $\mathcal{R}_1$, a customer may enter $\mathcal{R}_2$ as a positive customer (a_{12}^+) or move to $\mathcal{R}_3$ as a trigger (a_{13}^-). In the latter case, if $\mathcal{R}_3$ is not empty, then its queue length is decreased by one and a customer is added to $\mathcal{R}_2$ (a_{32}^+). Observe that if $\mathcal{R}_1$, $\mathcal{R}_3$ are non-empty, a trigger causes a change in the state of the three nodes simultaneously. Using RCAT we can deal with these cases as follows. Consider the processes underlying each node in isolation, as depicted by Figure 2. The synchronisation between the departure of a positive customer from $\mathcal{R}_1$ and its arrival at $\mathcal{R}_2$ is modelled with a standard active/passive synchronisation as proposed in [15], i.e., a transition with a specified rate (active) forces a transition with unspecified rate $\top$ (passive) to move jointly. Active synchronising transitions cannot occur without a corresponding passive one. Type a_{12}^+ is therefore active in $\mathcal{R}_1$ and passive in $\mathcal{R}_2$. The behaviour of triggers is rather different. Indeed, when a customer leaves $\mathcal{R}_1$ as a trigger for $\mathcal{R}_3$, it removes a customer from the nonempty queue (a_{13}^-) and instantaneously adds a customer to $\mathcal{R}_2$ (a_{32}^+). In Figure 2, this behaviour is depicted by assigning two types to the same transition: one passive and one active. The rate of the latter is unknown and denoted by x_{13}^- (the

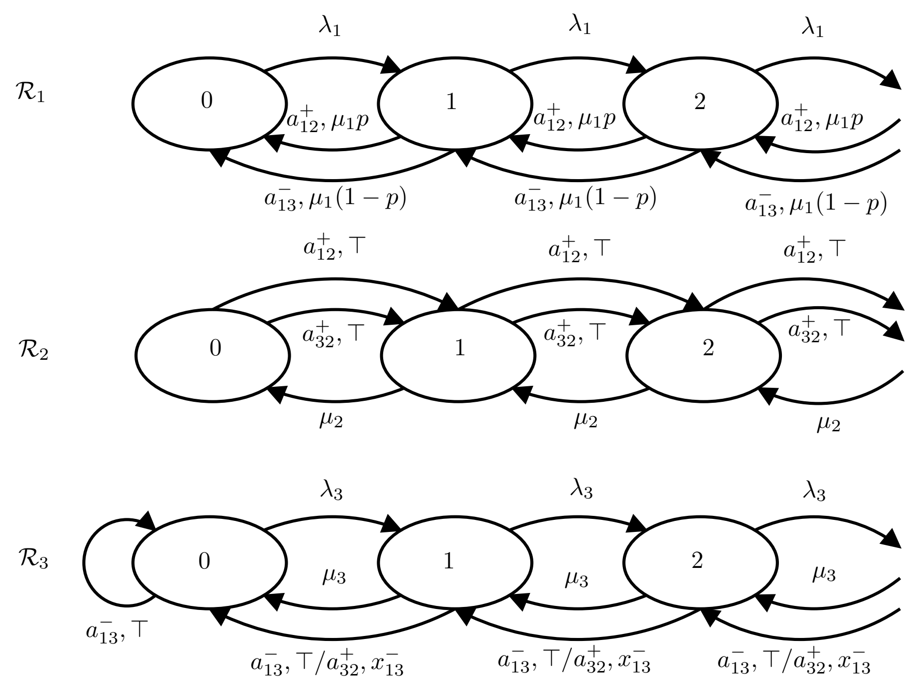

Fig. 2. Processes underlying the G-network of Figure 1.

reason for this name will be clear soon enough). Observe that a trigger arriving at $\mathcal{R}_3$, when this is empty, does not move any customer to $\mathcal{R}_2$ so that the invisible passive action in state 0 of $\mathcal{R}_3$ does not have the double type.

A product-form can be derived via application of RCAT. As a matter of terminology, we consider an *isolated component* as referring to a component of the original model that does not cooperate with any other and whose (originally) passive transitions' rates are specified as real numbers; the same value for all passive transitions with the same type. Informally, in its simplest formulation RCAT conditions require that:

- If a synchronising type ℓ is passive in a component, then all states of that component must have one outgoing transition with type ℓ
- If a synchronising type ℓ is active in a component, then all states of that component must have one incoming transition with type ℓ
- In the isolated components, all the transitions with the same active synchronising types must have the same reversed rate.

Although relaxations of these conditions are possible, for the sake of brevity we limit our analysis to cases encompassed by RCAT as stated above. We refer to the first two as the *structural conditions* of RCAT, while the latter is the *rate condition*. Let x_ℓ be the reversed rate associated with synchronising type ℓ. Then the component that is passive with respect to ℓ is isolated by setting all the rates of its transitions with that type to x_ℓ. For this reason, in Figure 2, active type a_{32}^+ appears with rate x_{13}^- (which abbreviates $x_{a_{13}^-}$).

Applying RCAT to the network of Figure 1, we first note that RCAT's structural conditions are satisfied, by inspection of Figure 2. In general, the reversed rate $\overline{q}_\ell(Q,P)$ of a transition with type ℓ from state P to Q and with rate $q_\ell(P,Q)$, can be computed as [17], [11]:

$$\overline{q}_\ell(Q,P) = \frac{\pi(P)}{\pi(Q)} q_\ell(P,Q) \tag{1}$$

in a Markov process with stationary probability function π. However, in this case, since the CTMCs underlying the components are reversible, we can write down the reversed rates by inspection of the corresponding forward rates. Indeed, all the transitions a_{12}^+ (a_{13}^-) have the same reversed rate x_{12}^+ (x_{13}^-), which can be computed as:

$$x_{12}^+ = \lambda_1 p\,, \quad x_{13}^- = \lambda_1(1-p)\,.$$

We devote more attention to transitions with type a_{32}^+ in $\mathcal{R}_3$. These are active with rate (in the isolated process) x_{13}^-, which is set by the passive synchronisation according to RCAT. Therefore, their reversed rate is constant with value:

$$x_{32}^+ = \lambda_3 x_{13}^- / (x_{13}^- + \mu_3)\,.$$

Observe that once a customer is added to $\mathcal{R}_2$ due to type a_{32}^+, there can be no further instantaneous propagation since RCAT's structural conditions are not satisfied, because state 0 of $\mathcal{R}_2$ does not have an incoming transition with type a_{32}^+. However, if we modify $\mathcal{R}_2$ to $\mathcal{R}_2'$ by adding an invisible transition on state 0, as shown in Figure 3, type ℓ has constant reversed rate provided we choose rate $\gamma_\ell = \mu_2 x_{32}^+ / (x_{32}^+ + x_{12}^+)$ for the invisible action.

Propagation of instantaneous transitions in the way shown in Figure 3 is interesting because it allows for the construction of *chains* of instantaneous transitions of finite but unbounded

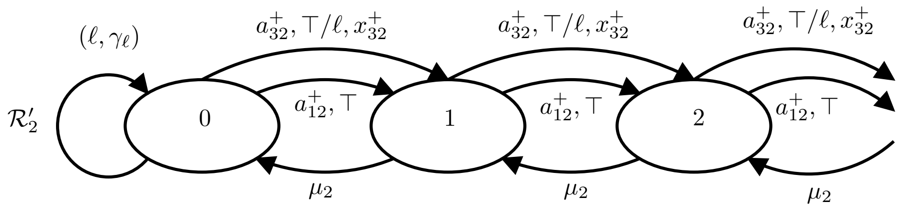

Fig. 3. Propagation of Instantaneous Transitions.

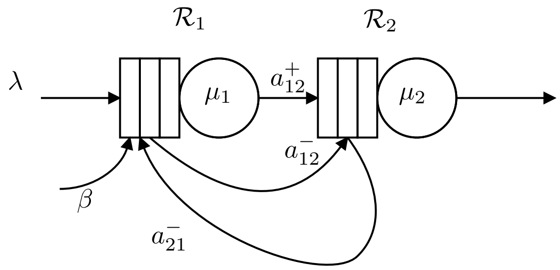

Fig. 4. G-network with iterative customer removals.

$\mathcal{R}_1$	$\mathcal{R}_2$
1. $P_n = (\tau,\ \lambda).P_{n+1}$	1. $Q_n = (a_{12}^+, \top).Q_{n+1}$
2. $P_{n+1} = (a_{12}^+, \mu_1).P_n$	2. $Q_{n+1} = (\tau, \mu_2).Q_n$
3. $P_{n+1} = (a_{12}^-, \beta).P_n$	3. $Q_{n+1} = (a_{12}^- \to a_{21}^-, \top).Q_n$
4. $P_{n+1} = (a_{21}^- \to a_{12}^-, \top).P_n$	4. $Q_0 = (a_{12}^-, \top).Q_0$
5. $P_0 = (a_{21}^-, \top)$	

TABLE I
MPA DEFINITION OF THE G-NETWORK DEPICTED BY FIGURE 4.

length. These chains may also form cycles in a similar fashion to what is studied in [5]. Consider the model depicted by Figure 4. Positive customers arrive from the outside according to a homogeneous Poisson process with rate λ and are served first at $\mathcal{R}_1$ and then at $\mathcal{R}_2$, queues with independent, exponential service times with rates μ_1 and μ_2, respectively. PITs are started in $\mathcal{R}_1$ with rate β and cause a positive customer deletion in $\mathcal{R}_1$ if the queue is not empty. This instantaneously propagates to $\mathcal{R}_2$, back to $\mathcal{R}_1$, then to $\mathcal{R}_2$ again and so on, the chain terminating when one of the two queues is empty on arrival of the PIT.

Using RCAT to study product-form models with chains of instantaneous transitions is convenient because it enhances the modularity of the analysis and facilitates the derivation of the steady-state probability distribution in heterogeneous models including, but not limited to, those considered in [8], [7], [9], [5]. One of the open problems when considering such models is the derivation of the system of rate equations. This depends both on the way the components interact with each other and on the internal structure of each component, as shown by the example of Figure 1. In Section IV we propose an algorithm that derives the rate equations for a given model that consists of a set of components that may interact by means of PITs, including when cycles arise in the chains of propagation.

III. A PROCESS-ALGEBRAIC NOTATION

In this section we define a process algebraic syntax to specify models with PITs. Since it closely resembles that of well-known process algerbras such as the Performance Evaluation Process Algebra (PEPA) [15], we just introduce the novel equation type that we use in this paper. We write:

$$P = (a \to b, \top).Q \tag{2}$$

to denote a passive action with type a that takes process P to Q and instantaneously synchronises as active on type b. Note that, since we are interested only in product-form applications of this type of synchronisations, the rate at which the transition synchronises on type b is well-defined and equal to the reversed rate of the passive action with type a.

Example 1: According to the process algebraic notation proposed here, $\mathcal{R}_1$ and $\mathcal{R}_2$, depicted in Figure 4, can be formally specified as in Table I, where τ denotes a non-synchronising internal type.

IV. RATE EQUATIONS

The algorithm we present in this section derives the set of rate equations for a model given by a set of interacting components. The algorithm works for PITs with arbitrary finite length or topology; hence cycles are admitted. We first briefly recap our notation.

Definition 1: Let $\mathcal{M}$ be the model consisting of components $\mathcal{C} = \{\mathcal{R}_1, \dots \mathcal{R}_N\}$. Each component $\mathcal{R}_i$ is a set of Markovian Process Algebra (MPA) equations $s \in \mathcal{R}_i$. The set of synchronising types is denoted by $\mathcal{L}$, with generic type $\ell \in \mathcal{L}$. $\mathcal{A}(\mathcal{R}_i)$ denotes the set of synchronising types that are active in $\mathcal{R}_i$, and $\mathcal{P}(\mathcal{R}_i)$ is the analogous set of passive. Finally, the *internal action type* τ denotes the special type that never synchronises.

Algorithm 1 is a recursion that considers a synchronising type ℓ at each step, removes its synchronisation from the model by replacing the $\top$ in the passive component with the corresponding reversed rate. At the same time, it generates the set of equations corresponding to x_ℓ. When it terminates, it returns the set of rate equations for the model. If they admit a solution, then the model has a product-form given by the resulting set of values of the x_ℓ.

Algorithm 1 Algorithm

```
function DeriveRateEquations(𝒞, ℒ)
                                   ▷ 𝒞 = {ℛ_1, ... ℛ_N} is the set of MPA agents
                       ▷ ℛ_i is the set of statements forming the i-th simple agent
                                       ▷ ℒ is the set of synchronising types
    ℰ ← ∅
    if ℒ ≠ ∅ then
        choose ℓ ∈ ℒ
        Let ℛ_A ∈ 𝒞 s.t. ℓ ∈ 𝒜(ℛ_A)
        Let ℛ_P ∈ 𝒞 s.t. ℓ ∈ 𝒫(ℛ_P)
        for all s ∈ ℛ_A do
            s ← s{ℓ := τ}
            Set ℰ as the set of rate eq. associated with ℓ
        end for
        for all s ∈ ℛ_P do
            if s = (ℓ, ⊤).Q for some Q ∈ ℛ_P then
                s ← (τ, x_ℓ).Q
            else if s = (ℓ → a, ⊤).Q for some Q ∈ ℛ_P, a ∈ 𝒜(ℛ_P) then
                s ← (a, x_ℓ).Q
            else if s = (ℓ → τ, ⊤).Q for some Q ∈ ℛ_P then
                s ← (τ, x_ℓ).Q
            end if
        end for
    end if
    return ℰ ∪ DeriveRateEquations(𝒞, ℒ \ {ℓ})
end function
```

A. Derivation of reversed rates

In general, deriving the system of rate equations associated with a synchronising type may be difficult, and the procedure defined in [11] has to be applied. In practice, the equations for the reversed rates can be found by one of the following approaches:

a) Using the steady state probabilities: This method straightforwardly applies Equation (1) to derive the reversed rates of each synchronising active transition. If the steady-state probability functions of each component in the model are unknown, for a model with C components, N_c states in component c and $|\mathcal{L}|$ synchronising active types, the number of equations is $\mathcal{O}(N+|\mathcal{L}|\cdot N/C)$ and the number of unknowns is $\mathcal{O}(N+|\mathcal{L}|)$, where $N=\sum_c N_c$.

b) Analysis of the cycles: This method relies on the application of Kolmogorov's criteria. It does not require the introduction of support variables such as the stationary state probabilities. The first of Kolmogorv's criteria states that for each state of the process, the sum of the rates of the outgoing transitions in the forward and in the reversed processes are the same. The second states that for each cycle of the process, the product of the rates in the forward and in the reversed processes are the same. In [11], it is proved that only the minimal cycles need be considered. The main advantage of this approach is that the number of unknowns does not depend on the number of states since it is exactly the number of synchronising types $|\mathcal{L}|$. However, the number of equations depends heavily on the structure of the process and on its cycles in particular. In the worst case, the number of cycles may grow exponentially with the number of states; however, in most practical applications, the number of cycles depends linearly on the number of process algebra equations that specify each component, which reduces the complexity to $\mathcal{O}(|\mathcal{L}|\cdot T)$, assuming that each component consists of T equations.

c) Exploiting properties of the underlying processes: In many relevant cases, the processes underlying the model components have a special structure that can drive and greatly simplify the formulation of the rate equations. A remarkable example is the class of processes whose stationary distribution is geometric, such as those presented in [16], [6], [8], [7], [9], [12]. In all these cases, each transition type introduces only one unknown and one equation in the system of rate equations, as we show in Section IV-B.

B. Step by step example

In this section we show Algorithm 1 at work on the model defined by Example 1 and shown in Figure 4. The set of components $\mathcal{C}$ is $\{\mathcal{R}_1,\mathcal{R}_2\}$, where the $\mathcal{R}_i$ are defined in Table I and $\mathcal{L}=\{a_{12}^+,a_{12}^-,a_{21}^-\}$. We start with $\mathcal{E}=0$. In the literature, it is well known that components $\mathcal{R}_1$ and $\mathcal{R}_2$, at equilibrium, have geometric stationary distributions and we therefore exploit this knowledge to derive the rate equations. The order in which we consider the types is completely arbitrary.

Step 1: Type $a^+_{12} \in \mathcal{A}(\mathcal{R}_1) \cap \mathcal{P}(\mathcal{R}_2)$. Rule 2 in $\mathcal{R}_1$ becomes

$$P_{n+1} = (\tau, \mu_1).P_n \quad (n \geq 0)$$

Rule 1 in $\mathcal{R}_2$ becomes

$$Q_n = (\tau, x^+_{12}).Q_{n+1}$$

and the set of rate equations $\mathcal{E}$ is updated with $x^+_{12} = \lambda\mu_1/(\mu_1 + \beta + x^-_{21})$.

Step 2: Type $a^-_{12} \in \mathcal{A}(\mathcal{R}_1) \cap \mathcal{P}(\mathcal{R}_2)$. In $\mathcal{R}_1$, Rules 3 and 4 instantiate respectively to:

$$P_{n+1} = (\tau, \beta).P_n \quad \text{and} \quad P_{n+1} = (a^-_{21} \to \tau, \top).P_n\,,$$

while Rules 3 and 4 in $\mathcal{R}_2$ become:

$$Q_{n+1} = (a^-_{21}, x^-_{12}).Q_n \quad \text{and} \quad Q_0 = (\tau, x^-_{12}).Q_0\,.$$

Notice that the effective rate of the transitions labelled a^-_{12} in $\mathcal{R}_1$ is $\beta + x^-_{21}$ (see Rules 3 and 4 of $\mathcal{R}_1$); therefore the set $\mathcal{E}$ is updated with $x^-_{12} = \lambda(x^-_{21} + \beta)/(x^-_{21} + \beta + \mu_1)$.

Step 3: Type $a^-_{21} \in \mathcal{P}(R_1) \cap \mathcal{A}(R_2)$. In $\mathcal{R}_1$ Rules 4 and 5 become, respectively:

$$P_{n+1} = (\tau, x^-_{21}).P_n \quad \text{and} \quad P_0 = (\tau, x^-_{21}).P_0\,.$$

and in $\mathcal{R}_2$, Rule 3 becomes:

$$Q_{n+1} = (\tau, x^-_{12}).Q_n\,.$$

$\mathcal{E}$ is updated with the rate equation $x^-_{21} = x^+_{12}x^-_{12}/(x^-_{12} + \mu_2)$.

After *Step 3*, Algorithm 1 terminates, giving a set of 3 rate equations in the unknowns $x^+_{12}, x^-_{12}, x^-_{21}$. Note that for G-networks, the existence of a solution to the rate equations is proved in [10], [14]. The isolated components are now simple MPA agents without synchronisations – i.e. essentially independent. Therefore, the computation of the equilibrium joint state probabilities, when they exist, is trivial, namely the product of the (local) stationary probability functions of the isolated components.

C. Correctness and complexity

The correctness of the algorithm hinges on the observation that the solution of the system of rate equations $\mathcal{E}$ is a sufficient condition for the set of components $\mathcal{C}$ cooperating on types $\mathcal{L}$ to be in product-form.

Proposition 1: In the model $\mathcal{M}$ of Definition 1, suppose $|\mathcal{L}| > 0$ and let the passive actions with type $b \in \mathcal{L}$ be assigned rate-variable x_b in an application of RCAT [11]. Let v_a be a solution for x_a that satisfies the rate equations for one type $a \in \mathcal{L}$. Then, if RCAT's conditions hold, the product-form of the model $\mathcal{M}$ with types $\mathcal{L}$ is the same as the product-form of the model $\mathcal{M}_a$ with types $\mathcal{L}\backslash\{a\}$, defined by assigning all passive actions of type a the value v_a in their component-process in $\mathcal{M}$.

Proof: We refer to the statement of RCAT given in [14], and less formally in section II, where all states have an outgoing instance of each passive action and an incoming instance of each active action in the appropriate components – in other words the constraint equations are satisfied vacuously in [14]. First, in the model $\mathcal{M}_a$, all passive actions in any component with a type $b \in \mathcal{L}\backslash\{a\}$ are outgoing from all states of that component since the same can be said of b in model $\mathcal{M}$. Similarly, all active actions in any component with a type $b \in \mathcal{L}\backslash\{a\}$ are incoming to all states of that component in $\mathcal{M}_a$. Now, in $\mathcal{M}_a$, all instances of the passive action of type a are replaced by non-synchronising actions with rate v_a, exactly as prescribed by RCAT in $\mathcal{M}$. It is therefore required to prove that the rate equations for $\{x_b \mid b \in L\backslash\{a\}\}$ in $\mathcal{M}_a$ have the same solution as they do in $\mathcal{M}$. But the rate equations of $\mathcal{M}_a$ are, by construction, precisely the rate equations of $\mathcal{M}$ with the substitution $x_a := v_a$, i.e., back-substituting for x_a. $\diamondsuit$

Concerning the computational complexity, we observe that the number of recursive calls is exactly equal to the number of synchronising types. We assume that, using opportune indexing structures, the analysis of the components' rules s can be done in constant time. If the model consists of components with a birth & death underlying structure, then the algorithm has time complexity $\mathcal{O}(|\mathcal{L}|)$. Otherwise, with the first method described in Section IV-A, we have a time complexity of $\mathcal{O}(|\mathcal{L}| \cdot N/C)$, with $N = \sum_{i=1}^{C} N_c$ and N_c is the number of states of component c.

V. Examples

This section shows some illustrative examples of the methodology proposed.

A. Finite capacity queues with congestion control

Consider a tandem of nodes $\mathcal{R}_1, \ldots, \mathcal{R}_N$ with exponential service times, rates $\mu_1, \ldots, \mu_N$ and a buffer with finite positive capacity $B_1, \ldots, B_N$, as illustrated by Figure 5. When node i is full, it generates a flushing signal that instantaneously propagates to node $(i \bmod N) + 1$ until an empty node is found. We also adopt the skipping policy presented in [20], [1] to deal with saturated queues. Informally, when a customer arrives at a full queue $\mathcal{R}_i$, it is discarded with a certain probability p_i or it immediately tries to enter the following station with probability $(1 - p_i)$, and iterates these trials until it either finds a non-saturated queue or leaves the system. Customers arrive from the outside at each node $\mathcal{R}_i$ according to independent Poisson processes with rates λ_i. As far as we know, this model has not been studied in the literature and the product-form is derived in a purely automatic way by applying Algorithm 1. Figure 6 illustrates the process underlying component $\mathcal{R}_i$, with $1 < i < N$. $\mathcal{R}_1$ differs from the others because it lacks the transition with synchronising type $a^+_{i-1,i}$ and type $a^-_{i-1,i}$ should be replaced by $a^-_{N,1}$. $\mathcal{R}_N$ differs because the synchronising type $a^+_{i,i+1}$ is replaced by the internal type τ and type $a^-_{i,i+1}$ is replaced by type $a^-_{N,1}$.

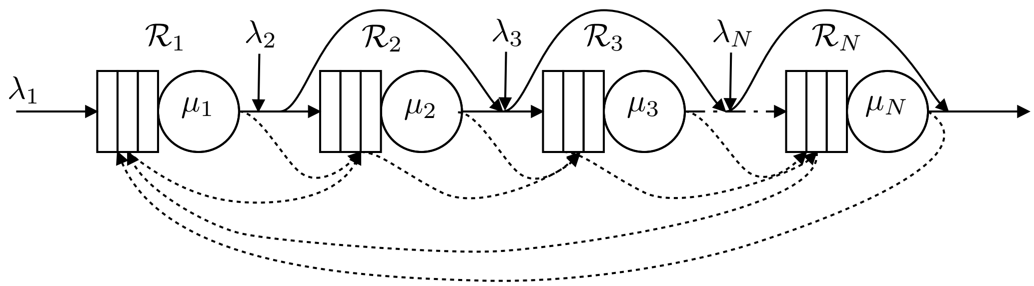

Fig. 5. Tandem of nodes with finite capacity with skipping and congestion control policy.

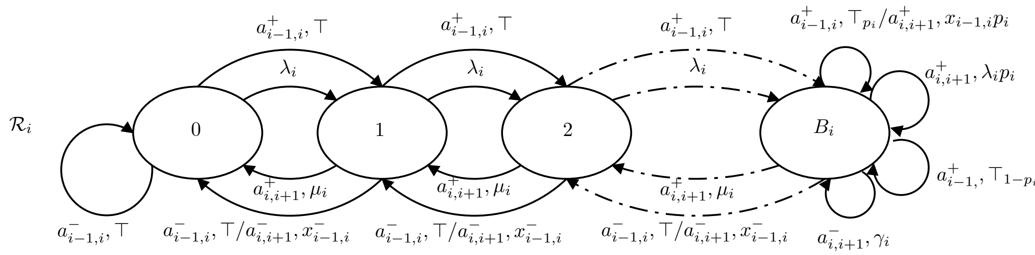

Fig. 6. Processes underlying the finite capacity nodes of Figure 5.

For $1 < i < N$, calculating the appropriate reversed rates yields the following equations for active type $a^+_{i,i+1}$:

$$x^+_{i,i+1} = (\lambda_i + x_{i-1,i}) \frac{\mu_i}{\mu_i + x^-_{i-1,i}}, \tag{3}$$

$$x^+_{i,i+1} = (\lambda_i + x_{i-1,i}) p_i . \tag{4}$$

For active type $a^-_{i,i+1}$, the equations are:

$$x^-_{i,i+1} = (\lambda_i + x^+_{i-1,i}) \frac{x^-_{i-1,i}}{\mu_i + x^-_{i-1,i}}, \tag{5}$$

$$x^-_{i,i+1} = \gamma_i . \tag{6}$$

Consulting Equations (3) and (4) generated by Algorithm 1, we immediately derive a product-form rate condition, *viz*, $p_i = \mu_i/(\mu_i + x^-_{i-1,i})$. Similarly, from Equations (5) and (6), we obtain $\gamma_i = (\lambda_i + x^+_{i-1,i})x^-_{i-1,i}/(\mu_i + x^-_{i-1,i})$. Therefore, in practice, we need only to solve Equations (3) and (5). For model $\mathcal{R}_1$, we set $x^+_{i-1,i} = 0$ and substitute $x^-_{i-1,i}$ by $x^-_{N,1}$. For model $\mathcal{R}_N$, we ignore the equations for $x^+_{N,N+1}$ and substitute $x^-_{N,N+1}$ by $x^-_{N,1}$.

Given the solution of the system of rate equations, the steady-state probability distribution of the model of Figure 5 is:

$$\pi(n_1, \ldots, n_N) = G \prod_{i=1}^{N} \rho_i^{n_i} ,$$

for $(n_1, \ldots, n_N) \in [0, B_1] \times [0, B_2] \times \ldots \times [0, B_N]$, the set of ergodic states, where $\rho_i = (\lambda_i + x^+_{i-1,i})/(\mu_i + x^-_{i-1,i})$ for $1 < i \le N$, $\rho_1 = \lambda_1/(\mu_1 + x^-_{N,1})$ and G is the normalising constant,

$$G = \prod_{i=1}^{N} \frac{1 - \rho_i}{1 - \rho_i^{B_i+1}} .$$

B. Queues with negative populations

In this section we introduce a queueing network model with instantaneous transition propagations, in which nodes may have negative queue length. We prove the product-form for this class of queueing networks and show that Algorithm 1 derives automatically the system of rate equations. We first describe the behaviour of an isolated queue and then derive the product-form for the network. The product-form solution requires some rate-dependent conditions, which follow in a natural way on applying RCAT.

1) The single queue: We consider a queue $\mathcal{R}_i$ with positive or negative population, exponential service times and independent, homogeneous Poisson arrivals. The population of the queue grows (in unit increments) at rate λ_i, but decreases due to service completions at rate μ_{i1} when the population is strictly positive, and with rate μ_{i2} otherwise. The queue's population can decrease also because of a customer destruction. This event occurs with rate γ_i but only has effect (i.e., actually deletes a customer) with probability p_{i1}, when the queue length is positive, and p_{i2}, otherwise. Figure 7 shows the process underlying this type of queue. The equilibrium probabilities of the isolated queue are then given by:

$$\pi_i(n) = \begin{cases} \pi_{i0}\rho_{i1}^{n} & \text{if } n \ge 0 \\ \pi_{i0}\rho_{i2}^{-n} & \text{if } n < 0 \end{cases} , \tag{7}$$

where

$$\rho_{i1} = \frac{\lambda_i}{\mu_{i1}+\gamma_i p_{i1}},$$
$$\rho_{i2} = \frac{\mu_{i2}+\gamma_i p_{i2}}{\lambda_i} \text{ and}$$
$$\pi_{i0} = \frac{1-\rho_{i1}\rho_{i2}}{(1-\rho_{i1})(1-\rho_{i2})}.$$

Table II gives the process algebraic equations for this queue when embedded in a queueing network. Arrivals are modelled by passive transitions and we use the assignment of a double type to some transitions to allow for the PITs. For the sake of simplicity, we assume unique arrival and departure streams for each synchronising type. The stability conditions are $\rho_{i1} < 1$ and $\rho_{i2} < 1$. Now, we study the conditions for this queue to yield a product-form stationary distribution. In order for the output of the queue to be an input of another (either as positive or negative customers), we require the transitions corresponding to job completions and those corresponding to customer destructions to have constant reversed rates.

Process $\mathcal{P}_i$	Description
1. $P^i_{n+1} = (a, \top).P^i_n\,, n \in \mathbb{Z}$	Arrival
2. $P^i_n = (b, \mu_{i1}).P^i_{n-1}\,, n > 0$	Departure
3. $P^i_n = (b, \mu_{i2}).P^i_{n-1}\,, n \leq 0$	Departure
4. $P^i_n = (c \to d, \top_{p_{i1}}).P^i_{n-1}\,, n > 0$	Succ. del.
5. $P^i_n = (c, \top_{1-p_{i1}}).P^i_n\,, n > 0$	Unsucc. del.
6. $P^i_n = (c \to d, \top_{p_{i2}}).P^i_{n-1}\,, n \leq 0$	Succ. del.
7. $P^i_n = (c, \top_{1-p_{i2}}).P^i_n\,, n \leq 0$	Unsucc. del.

TABLE II
DEFINITION OF THE COMPONENTS OF THE QUEUEING NETWORK OF SECTION V-B. TRANSITIONS LABELLED c ARE THE PITS OF THE COMPONENTS.

In order for the transitions labelled b and c to have constant reversed rates, x_b and x_c, respectively, we must have:

$$x_a\frac{\mu_{i2}}{x_c p_{i2}+\mu_{i2}} = x_a\frac{\mu_{i1}}{x_c p_{i1}+\mu_{i1}},$$
$$x_a\frac{x_c p_{i2}}{x_c p_{i2}+\mu_{i2}} = x_a\frac{x_c p_{i1}}{x_c p_{i1}+\mu_{i1}},$$

which are satisfied if and only if:

$$p_{i1}\mu_{i2} = p_{i2}\mu_{i1}. \tag{8}$$

Note that the rate condition does not depend on the reversed rates x_a and x_c; hence it can be checked for each queue in isolation.

The process is ergodic if and only if $\rho_{i1} < 1$ and $\rho_{i2} < 1$, i.e.:

$$\mu_{i2} + x_c p_{i2} < x_a < \mu_{i1} + x_c p_{i1}. \tag{9}$$

Hence, it is necessary that $\mu_{i1} + x_c p_{i1} > \mu_{i2} + x_c p_{i2}$. We solve the inequality for p_1 using rate constraint (8):

$$\mu_{i2} + x_c\frac{p_{i1}\mu_{i2}}{\mu_{i1}} < \mu_{i1} + x_c p_{i1},$$
$$p_{i1}x_c(\mu_{i1}-\mu_{i2}) > \mu_{i1}(\mu_{i2}-\mu_{i1}).$$

Observe that, if $\mu_{i1} > \mu_{i2}$, then the inequality is satisfied for all $p_{i1} \in [0,1]$; otherwise it is impossible. Therefore, a necessary stability condition is $\mu_{i1} > \mu_{i2}$. Similarly for p_2,

$$\mu_{i2} + x_c p_{i2} < \mu_{i1} + x_c\frac{p_{i2}\mu_{i1}}{\mu_{i2}},$$
$$p_{i2}x_c(\mu_{i1}-\mu_{i2}) > \mu_{i2}(\mu_{i2}-\mu_{i1}).$$

Again, for $\mu_{i1} > \mu_{i2}$ the inequality is always satisfied for all $p_{i2} \in [0,1]$. Summing up, stability condition (9) can be satisfied only if $\mu_{i1} > \mu_{i2}$, given that Equation (8) holds. Observe that this is also necessary for the stability of the queue when there is no customer deletion, i.e., when $x_c = 0$.

2) The queueing network: Consider now the queueing network depicted by Figure 8. Henceforth we assume $p_{i1} = 1$ and choose p_{i2} such that the rate constraint (8) holds and the queues are stable ($i = 1, 2$). Arcs are labelled with the types of the synchronising transitions a^+_{12}, a^-_{12} and a^-_{21}. λ is the arrival rate of positive customers and β is the arrival rate of customers that start the PITs. The total rate γ at which a destruction signal is received by the first queue is $\beta + x^-_{21}$. The variables x^+_{12}, x^-_{12} and x^-_{21} are the solutions of the following equations:

$$\begin{cases}\gamma = \beta + x^-_{21}\\ x^-_{12} = \lambda\gamma/(\gamma+\mu_{11})\\ x^+_{12} = \lambda\mu_{11}/(\gamma+\mu_{11})\\ x^-_{21} = x^+_{12}x^-_{12}/(x^-_{12}+\mu_{21})\end{cases}. \tag{10}$$

Remark 1: Since the queue length can be negative, one may wonder if the deletion-cycle always terminates. A necessary condition for having infinite cycles of deletion is that $p_{i2} = 1$, for $i = 1, 2$. We now prove that the product-form condition (8) and the necessary stability condition are sufficient to prevent infinite deletion-cycles. Suppose that $p_{12} = 1$. Then the rate condition (8) becomes $\mu_{11} = \mu_{12}p_{11}$. However, this contradicts the stability condition $\mu_{i1} > \mu_{i2}$, and hence it follows that $p_{i2} < 1$ is a necessary condition for the queue to be stable and in product-form. This prevents infinite cycles of customer deletions in a stable queue, although their length is unbounded.

Once the system of rate equations is solved, the product-form solution is given by Equation (7), where λ_i should be replaced by the sum of the rates corresponding to positive customer arrivals at node i. In the example of Figure 8, these arrival rates are equal to λ and x^+_{12} for $\mathcal{R}_1$ and $\mathcal{R}_2$, respectively. Similarly, γ_i is replaced by the sum of the rates that cause customer destructions, i.e., in the example, $\beta + x^-_{21}$ and x^-_{12}), respectively. As an instance of numerical solution, Figures 9 and 10 show the plots of the average number of customers in the queues as functions of λ for the following set of parameters: $\mu_{11} = 3.0$, $\mu_{21} = 2.0$, $\beta = 0.5$, $\mu_{12} = 1.0$, $\mu_{22} = 1.0$, $p_{12} = 1/3$, $p_{22} = 1/2$.

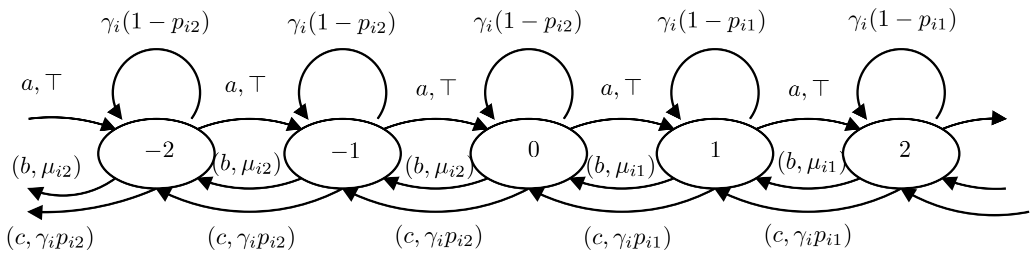

Fig. 7. Queue with positive and negative population and customer destruction.

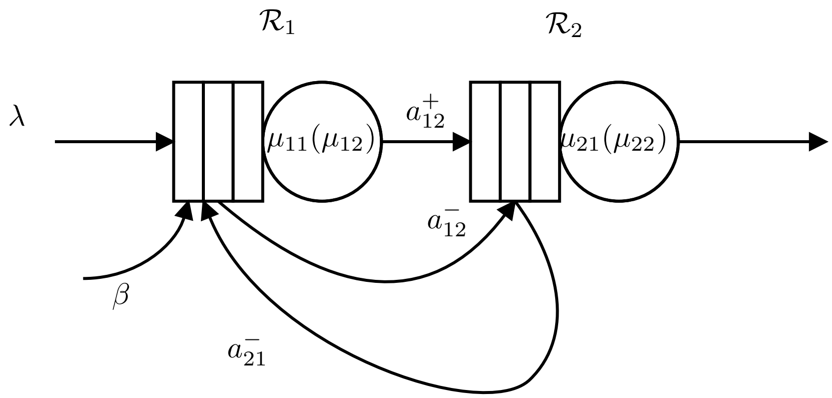

Fig. 8. Case study: a queueing network with positive or negative population and PITs.

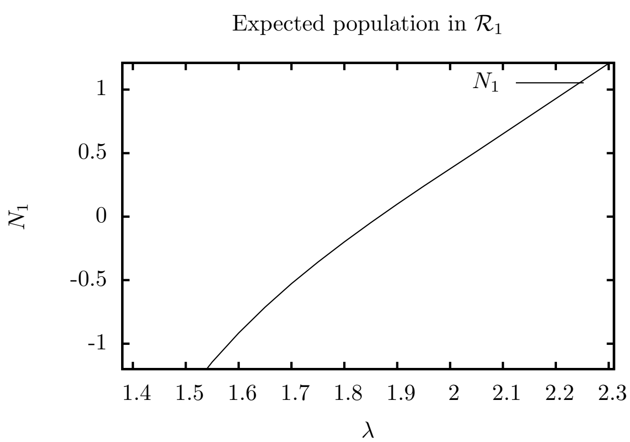

Fig. 9. Expected queue length N_1 in $\mathcal{R}_1$ as a function of λ in the example of Figure 8.

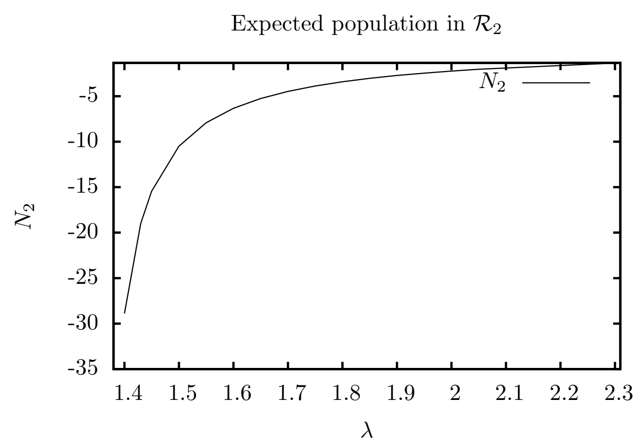

Fig. 10. Expected queue length N_2 in $\mathcal{R}_2$ as a function of λ in the example of Figure 8.

VI. Conclusion

In this paper we have proposed an algorithm to compute the system of rate equations associated with a class of product-form models that can have multi-way synchronisations and be studied using iterated applications of the Reversed Compound Agent Theorem (RCAT) [11]. The main idea consists of a recursive analysis of the pairwise synchronisations between the model's components. At each step, one synchronisation is removed and the corresponding rate equations are added to the system. We have shown the resulting algorithm to be very efficient when the components have a special underlying structure, such as a birth & death CTMC. The main contribution is the algorithmic analysis of product-form models whose components' synchronisations are not pairwise. For instance, this is the case for G-networks with triggers [8], in which triggers move a customer from one queue to another therefore changing the state of three components at the same epoch. Following the approach of [12], we proposed a formalism to describe such synchronisations in product-form models in terms of successive pairwise synchronisations by introducing the notion of Propagating Instantaneous Transitions (PITs). In particular, Algorithm 1 can be applied to automatically derive the set of rate equations. We also showed how PITs can be applied to study queueing networks with finite capacity and a skipping policy as defined in [20], [1]. Finally, it is worthy of note that Algorithm 1 is a step toward the automatic proof of the product-form for heterogeneous models, i.e., consisting of components taken from queueing theory, stochastic Petri nets, process algebra, or other domain.

Future research efforts will be devoted to the extension of Algorithm 1 to encompass the product-forms that can be studied using the Extended formulation of RCAT (ERCAT) [13].

REFERENCES

[1] BALSAMO, S., HARRISON, P. G., AND MARIN, A. A unifying approach to product-forms in networks with finite capacity constraints. In *Proc. of the 2010 ACM SIGMETRICS Int. Conf. on Measurement and Modeling of Computer Systems* (New York, NY, USA, 14-18 June 2010), V. Misra, P. Barford, and M. S. Squillante, Eds., pp. 25–36.

[2] BALSAMO, S., HARRISON, P. G., AND MARIN, A. Methodological Construction of Product-form Stochastic Petri-Nets for Performance Evaluation. *J. of System and Software 85*, 7 (2012), 1520–1539.

[3] 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.

[4] BOUCHERIE, R. J., AND VAN DIJK, N. M. *Queueing Networks: A Fundamental Approach*. International Series in Operations Research & Management Science. Springer, 2011.

[5] FOURNEAU, J. M., AND QUESSETTE, F. Computing the steady-state distribution of G-networks with synchronized partial flushing. In *ISCIS, 21th International Symposium* (Istanbul, Turkey, 2006), pp. 887–896.

[6] GELENBE, E. Product form networks with negative and positive customers. *J. of Appl. Prob. 28*, 3 (1991), 656–663.

[7] GELENBE, E. G-Networks with signals and batch removal. *Prob. in the Eng. and Informational Sciences 7* (1993), 335–342.

[8] GELENBE, E. G-networks with triggered customer movement. *J. of Appl. Prob. 30* (1993), 742–748.

[9] GELENBE, E., AND FOURNEAU, J. M. G-networks with resets. *Perf. Eval. 49*, 1-4 (2002), 179–191.

[10] GELENBE, E., AND SCHASSBERGER, M. Stability of product form G-Networks. *Prob. in the Eng. and Informational Sciences 6* (1992), 271–276.

[11] HARRISON, P. G. Turning back time in Markovian process algebra. *Theoretical Computer Science 290*, 3 (2003), 1947–1986.

[12] HARRISON, P. G. Compositional reversed Markov processes, with applications to G-networks. *Perf. Eval., Elsevier 57*, 3 (2004), 379–408.

[13] HARRISON, P. G. Reversed processes, product forms and a non-product form. *Elsevier Linear Algebra and Its Applications 386* (July 2004), 359–381.

[14] 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.

[15] HILLSTON, J. *A Compositional Approach to Performance Modelling*. PhD thesis, Department of Computer Science, University of Edinburgh, 1994.

[16] JACKSON, J. R. Jobshop-like queueing systems. *Management Science 10* (1963), 131–142.

[17] KELLY, F. *Reversibility and stochastic networks*. Wiley, New York, 1979.

[18] LAVENBERG, S. S. *Computer Performance Modeling Handbook*. Academic Press, New York, 1983.

[19] MARSAN, M. A., CONTE, G., AND BALBO, G. A class of generalized stochastic petri nets for the performance evaluation of multiprocessor systems. *ACM Trans. Comput. Syst. 2*, 2 (1984), 93–122.

[20] PITTEL, B. Closed exponential networks of queues with saturation: The Jackson-type stationary distribution and its asymptotic analysis. *Math. of Op. Res. 4*, 4 (1979), 357–378.