

PRODUCT-FORM IN G-NETWORKS

ANDREA MARIN

Department of Environmental Sciences, Informatics and Statistics Università Ca' Foscari Venezia, Italy
E-mail: marin@dais.unive.it

The introduction of the class of queueing networks called G-networks by Gelenbe has been a breakthrough in the field of stochastic modeling since it has largely expanded the class of models which are analytically or numerically tractable. From a theoretical point of view, the introduction of the G-networks has led to very important considerations: first, a product-form queueing network may have non-linear traffic equations; secondly, we can have a product-form equilibrium distribution even if the customer routing is defined in such a way that more than two queues can change their states at the same time epoch. In this work, we review some of the classes of product-forms introduced for the analysis of the G-networks with special attention to these two aspects. We propose a methodology that, coherently with the product-form result, allows for a modular analysis of the G-queues to derive the equilibrium distribution of the network.

1. INTRODUCTION

Queueing network models have been extensively applied to represent and analyze resource sharing systems, such as production, communication and computer systems [5,60]. One of the main problems in the analysis of queueing networks regards the analytical tractability of the model. Indeed, although in many cases the stochastic process underlying the queueing network is a continuous time Markov chain (CTMC), the standard approaches for the computation of the equilibrium performance indices based on the direct solution of the system of global balance equations (GBEs) are not applicable since the size of the state space makes the algorithms numerically unstable or very time and space expensive.

A class of queueing networks enjoys a property which is known as *product-form*. Informally, we say that a queueing network is in product-form if for each positive recurrent state of the underlying CTMC its stationary probability can be expressed in terms of a product of functions depending only on the state of a single queue. More formally, if a network of Markovian queues consists of N stations, then let $\mathbf{n} = (n_1, \dots, n_N)$ be a positive recurrent state where n_i denotes the state of queue i . The network is in product-form if for each positive recurrent \mathbf{n} , we have:

$$\pi(\mathbf{n}) = \frac{1}{G} \prod_{i=1}^N g_i(n_i), \quad (1)$$

where π is the stationary probability function and G is the normalizing constant. The computation of G is usually simple if the set of positive recurrent states \mathcal{S} of the underlying

CTMC is the Cartesian product of the positive recurrent states reachable by each queue. In these cases, G can be computed by normalizing the stationary probabilities of each isolated queue, that is, for each i we compute G_i such that:

$$\sum_{n_i: (n_1, \dots, n_i, \dots, n_N) \in \mathcal{S}} g_i(n_i) = G_i$$

and we derive $G = \prod_{i=1}^N G_i$. Open queueing networks usually satisfy this property [6]; see, e.g., [8,51]. Conversely, in closed queueing networks the set of positive recurrent states \mathcal{S} is a proper subset of the Cartesian product of the positive recurrent states reachable by each queue. Hence, the computation of the normalizing constant G becomes more challenging and ad hoc algorithms must be applied; see [5] and the references therein for a survey. Examples of networks with closed topology are [8,45]. The definition of functions g_i depends on the network topology in particular on the probabilistic routing matrix. In many cases [8,45,51], product-form queueing networks assume a state-independent probabilistic routing of the customers among the stations. Then, function g_i s depend on the queueing discipline and service time distribution of station i and on the visit ratio of the customers at station i . The computation of the visit ratio requires the solution of the linear system of *traffic* equations.

In 1989 [19], Gelenbe proposes a Markovian queueing network model that introduces important novelties with respect to the existing literature: the G-networks. This class of models maintains the desirable property of being analytically tractable since they are in product-form but are more general than those previously known. Indeed, in their original formulation, G-networks included two classes of customers: Positive and negative. Positive customers behave exactly as positive customers in Jackson networks [51], while when a negative customer arrives at station, it immediately deletes a positive customer if any is present or vanishes otherwise. Positive customers can change their class to negative customers according to a state-independent probabilistic law. One of the consequences of this change of paradigm, is that the network's traffic equations do not form any more a linear system of equations. Therefore, the analysis of G-networks usually relies on the numerical solution of the non-linear system of traffic equations. The existence and the uniqueness of the solution for ergodic models is proved in [40]. After 1989, several extensions of G-networks have been proposed in the literature by Gelenbe, Fourneau, Harrison, and other authors including the introduction of multi-class positive customers in the style of the BCMP networks [10,36]. We think that one of the most remarkable extensions of G-networks is the introduction of signals that perform a triggered customer movement. A trigger [20] is generated immediately after a service completion at a certain queue Q_1 and moves to another queue Q_2 according to a state-independent probabilistic routing. If Q_2 is empty, then the trigger vanishes, otherwise it removes one customer from Q_2 and adds it to a third queue Q_3 chosen probabilistically. From a theoretical point of view the introduction of the triggers is extremely interesting because we have more than two network stations that change their state at the same time epoch. Previous characterizations of product-form queueing networks (see, e.g., [52,59]) used to assume pairwise synchronizations between the queueing stations. The introduction of new extensions of G-networks is still an important area of research (see e.g., [3,4,18,37,55]).

The application area of G-networks is wide and includes neural network learning [22–24,42], image processing and recognition [30,41], signal processing [34], performance evaluation [38,39], bioinformatics [25,53,54,61], optimization [43], energy packet networks [27–29,44], cognitive networks [26,32,33,35] just to mention some examples. Nevertheless, the

flexibility of these models is promising to address other challenges in terms of performance evaluation of algorithms or learning such as those proposed in [1,63].

In this survey we focus on the analysis of the product-form in G-networks, starting from the original model proposed by Gelenbe in 1989 [19] and then showing how (and why) more recent behaviors maintain the product-form property. We explore the connections between the product-form characterizations such as the quasi reversibility [52] and its extended formulation [9], the Reversed Compound Agent Theorem (RCAT) [46,49].

This paper is structured as follows. In Section 2, we introduce the G-networks with pairwise synchronizations, that is, those in which at any time epoch at most two queues can change their state. In Section 3, we consider more sophisticated types of synchronizations such as the triggers. Section 4 aims at reviewing very important signal behaviors like the resets, the catastrophes and the batch removals and giving a uniform approach to their analysis. In Section 5, we present a final example. Finally, Section 6 gives some final remarks.

2. G-NETWORKS WITH PAIRWISE SYNCHRONIZATIONS

In this section, we study the class of G-networks whose interactions are pairwise, that is, at each time epoch at most two queueing stations change their states. Differently from the original paper, instead of specifying the transitions of the CTMC underlying the whole queueing network, we give a formal language to specify the behavior of each station in isolation and the way it cooperates with the rest of the network. In order to do this, we resort to a formalism which is similar to the Performance Evaluation Process Algebra (PEPA) introduced by Hillston in [50]. Informally, we describe a stochastic networks in terms of synchronizing Markov chains (in a similar fashion of what has been proposed in [52]). Each queueing station has an underlying description in which some transitions are labeled. The occurrence of labeled transitions models the synchronization. A labeled transition may be either active or passive: Active ones have a rate that represents the parameter of an exponentially distributed random variable, while passive ones do not have a rate (we use the symbol \top). A labeled transition can be carried out only jointly between the active and the passive components. Unlabeled transitions are carried out individually by each queueing station with a delay that is exponentially distributed with a certain rate.

Example 1 Modelling of a simple G-network: Let us consider the G-network depicted by Figure 1. The probabilistic routing is such that $\sum_{j=0}^N p_{ij}^+ + p_{ij}^- = 1$ for all $i = 1, \dots, 3$. In Figure 2, we show the model underlying the G-network and its PEPA description is shown in Table 1. We should notice the passive self-loops on states 0 of the models. Consider for instance the label a_{13}^- that models the departure of a negative customer from stations

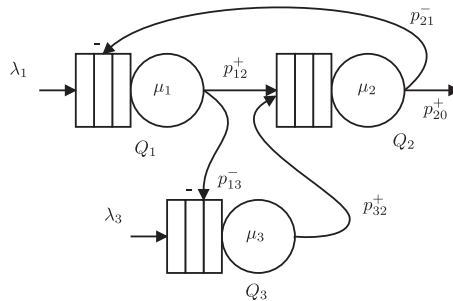


FIGURE 1. Simple G-network with positive and negative customers.

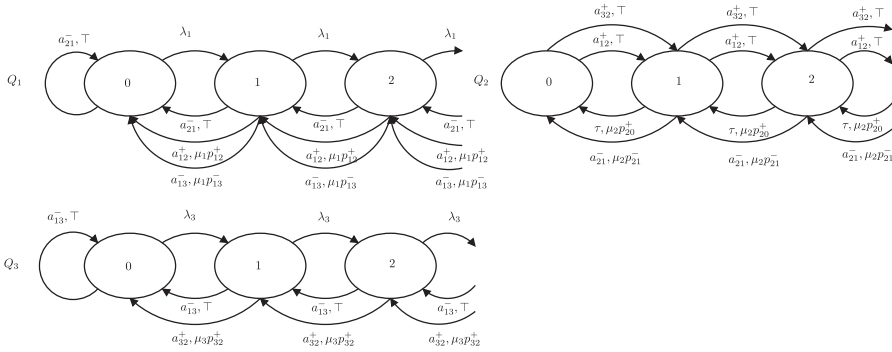


FIGURE 2. Model underlying the G-network of Figure 1.

TABLE 1. PEPA Specification of the Model of Figure 1.

$Q_1^0 = (\tau, \lambda_1).Q_1^1 + (a_{21}^-, \top).Q_1^0$ $Q_1^n = (\tau, \lambda_1).Q_1^{n+1} + (a_{21}^-, \top).Q_1^{n-1}$ $+ (a_{12}^+, \mu_1 p_{12}^-).Q_1^{n-1}$ $+ (a_{13}^-, \mu_1 p_{12}^-).Q_1^{n-1} \quad n > 0$	$Q_2^0 = (a_{32}^+, \top).Q_2^1 + (a_{12}^+, \top).Q_2^1$ $Q_2^n = (a_{32}^+, \top).Q_2^{n+1} + (a_{12}^+, \top).Q_2^{n+1}$ $+ (a_{20}^+, \mu_2 p_{20}^+).Q_2^{n-1}$ $+ (a_{21}^-, \mu_2 p_{21}^-).Q_2^{n-1} \quad n > 0$
$Q_3^0 = (\tau, \lambda_3).Q_3^1 + (a_{13}^-, \top).Q_3^0$ $Q_3^n = (\tau, \lambda_3).Q_3^{n+1} + (a_{13}^-, \top).Q_3^{n-1}$ $+ (a_{32}^+, \mu_3 p_{32}^+).Q_3^{n-1}$	$G = Q_1^0 \boxtimes_L Q_2^0 \boxtimes_L Q_3^0$

1 to 3. Notice that according to PEPA semantics the synchronizing transitions can occur only jointly. The self-loop labeled a_{13}^- in the specification of Q_3 models the event of a departure of a negative customer from Q_1 directed to Q_3 when the latter queue is empty. In this case, the negative customer has no effect on Q_3 (i.e., the self-loop is executed) and the number of customers in Q_1 is decreased by 1.

Now, we discuss the advantages of having introduced a process algebraic notation. This allows for a modular description of the G-network. However, the most interesting aspect of this approach is that the product-form of G-networks can be studied modularly, that is, by considering the isolated components instead of the whole queueing network. In this paper, we apply RCAT introduced by Harrison in [46] and then extended in [49,58] which gives a process algebraic characterization of a class of product-form models that includes the quasi-reversible queues [9,52]. In this way, the process algebraic modular description of the components is projected to the modular product-form analysis of the G-networks. Some of the models that we survey in this paper have been previously studied by RCAT in [47].

THEOREM 1: *Let Q_1, Q_2, \dots, Q_N be a network of queues. Then assume the following conditions hold:*

- *For each state s_i of Q_i if the synchronizing label a is active in Q_i then each state must have one incoming transition labeled a . Moreover, the reversed rates of all the transitions labeled a is the same $x_a > 0$;*
- *For each state s_i of Q_i if the synchronizing label a is passive in Q_i then each state must have one outgoing transition labeled a ;*

Then, the network is in product-form and each positive recurrent state of the joint process has the equilibrium distribution given by Eq. (1), where functions g_i are the equilibrium distribution of queue Q_i in which all the passive transitions labeled a assume the same value x_a , that is, the constant reversed rate in the synchronizing components.

Example 2: We now apply Theorem 1 to study the network of Example 1. First of all observe that all the passive (active) labels are enabled in (enter into) all the states of the models underlying the stations. It is also straightforward to prove that the reversed rate associated with the active transitions are constant. Let us consider label a_{12}^+ which is active in Q_1 . The CTMC underlying Q_1 is reversible [52] so the reversed rate associated with the sum of the three death transitions is birth rate λ_1 . In order to derive the reversed rate associated just with the transition labeled a_{12}^+ we use the result proved in [48], that is, the reversed rate is split among the three *parallel* transitions in proportion with their forward rates, that is:

$$x_{12}^+ = \lambda_1 \frac{\mu_1 p_{12}^+}{x_{21}^- + \mu_1 p_{12}^+ + \mu_1 p_{13}^-} = \lambda_1 \frac{\mu_1 p_{12}^+}{x_{21}^- + \mu_1}.$$

Similarly, we derive x_{13}^- as:

$$x_{13}^- = \lambda_1 \frac{\mu_1 p_{13}^-}{x_{21}^- + \mu_1}.$$

Analogously, we derive the reversed rates of the transitions in Q_2 and Q_3 :

$$x_{21}^- = (x_{32}^+ + x_{12}^+) p_{21}^-, \quad x_{32}^+ = \lambda_3 \frac{\mu_3}{\mu_3 + x_{13}^-}.$$

The solution of these rate equations gives the values to replace the \top in the components and hence find their equilibrium distribution, that is, functions g_i of Eq. (1). For G-networks the rate equations are equivalent to the traffic equations and hence under stability they admit a unique feasible solution as proved in [40,48].

3. G-NETWORKS WITH MULTI-WAY SYNCHRONIZATIONS

In this section we study G-networks in which the initial notion of negative customer is generalized to that of signal. From the point of view of characterizing the product-form in G-networks this becomes a challenging problem because signals may cause a state transition in more than two stations simultaneously. An example of such a behavior is the trigger. A trigger may be generated after a job completion at a station Q_i and moves a customer from a station Q_j to Q_k . If Q_j is empty, then the trigger vanishes. It is interesting to note that for this kind of behavior a PEPA description in which each component is described by a simple process cannot be used. Let us inspect Figure 3(B) showing the transitions associated with the trigger depicted in Figure 3(A). We notice that the synchronization between Q_1 and Q_2 must happen both in the case in which Q_2 is in state 0 and when it is in a state $n > 0$. However, in the former case the trigger does not affect the state of Q_3 , whereas in the latter it adds a customer. The issue here is that Q_3 should be able to distinguish if the synchronization between Q_1 and Q_2 happened when Q_2 was empty or not, but there is no way to achieve this with a single label. A similar observation can be done also for modular specifications based on Kronecker’s algebra as those studied in [11,12]. In [49] we have introduced a new way for specifying this class of synchronizations in G-networks based on the idea of *propagation of instantaneous transitions* (PITs). Informally,

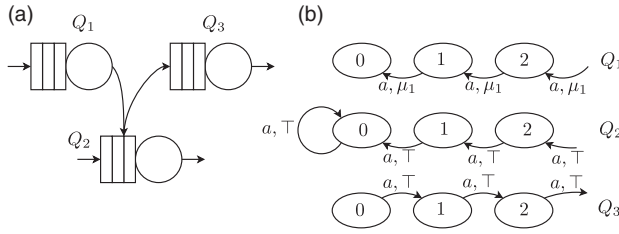


FIGURE 3. Wrong PEPA modelling of a G-network trigger.

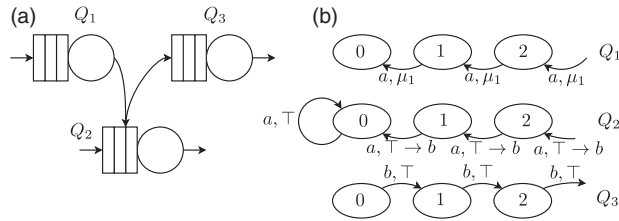


FIGURE 4. Representation of a trigger by using PITs.

in this synchronization specification when an active/passive synchronization occurs we have that it may propagate instantaneously with a different label. Let us revisit the example of Figure 3 which is correctly represented with the new formalism in Figure 4. Assume that Q_2 is in state $n_2 > 0$ and that Q_1 moves from $n_1 > 0$ to $n_1 - 1$. Then a synchronization occurs on label a that instantaneously propagates to Q_3 with label b . We write $a, \top \rightarrow b$ to describe this effect. Now Q_3 synchronizes on b a moves from state n_3 to $n_3 + 1$. Notice that if Q_2 is empty, Q_1 still synchronizes with Q_2 on label a but the self-loop transition does not propagate and hence the state of Q_3 is not changed.

The advantage of this modular specification of G-networks is manifold. First, we can easily specify heterogeneous networks with different kind of synchronizations, for example, triggers but also resets or catastrophes as we will see later on. But the most important advantage is the easiness of proving the product-form of the equilibrium distribution for models specified in this way, that is, we do not need to resort to the GBEs. The main idea here is that to prove that a component satisfies RCAT conditions we proceed in two steps:

1. We verify the first condition of Theorem 1 for all the passive labels.
2. Let a be a passive label for a component. We assume to know the rate x_a associated with this label and verify that the active transitions satisfy the second condition of Theorem 1. If the transition is a PIT, that is, of the form $a, \top \rightarrow b$ we consider b as an active transition with *forward* rate x_a .

Example 3: We consider once more a G-network with triggers as depicted in Figure 5, where $p + q = 1$. We start by studying the model of Q_1 in isolation. The only passive label is c which is outgoing from every state. Let us assume that all the transitions labeled c have a rate x_c . Labels a and d are active and their reversed rates are clearly constant since the CTMC underlying the queue is a Birth&Death process and hence reversible. We immediately obtain the reversed rates x_a and x_d as:

$$x_a = (\lambda_1 + x_c)q, \quad x_d = (\lambda_1 + x_c)p.$$

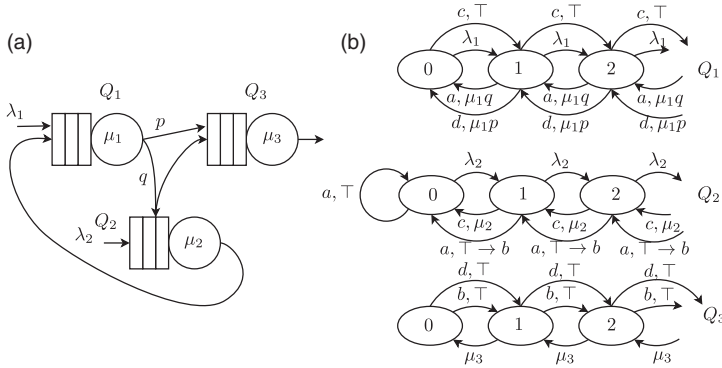


FIGURE 5. G-network with trigger.

Let us consider Q_2 . Here, a is passive and is outgoing from every state. Let us consider active label a which is associated with the propagation of a -synchronizations on states $n > 0$. We treat b as a normal active transition whose forward rate is x_a and hence we obtain:

$$x_b = \lambda_2 \frac{x_a}{x_a + \mu_2}, \quad x_c = \lambda_2 \frac{\mu_2}{x_a + \mu_2}.$$

Finally, in Q_3 we easily verify the conditions on passive labels. The solution of the above non-linear rate equation system is:

$$\begin{aligned} x_a &= \frac{1}{2} \left(\lambda_1 q - \mu_2 + \sqrt{\Delta} \right), & x_b &= \frac{\lambda_1 q + 2\lambda_2 q + \mu_2 - \sqrt{\Delta}}{2q}, \\ x_c &= \frac{-\lambda_1 q - \mu_2 + \sqrt{\Delta}}{2q}, & x_d &= \frac{p \left(\lambda_1 q - \mu_2 + \sqrt{\Delta} \right)}{2q}, \end{aligned}$$

where $\Delta = 4\lambda_2\mu_2q + (\lambda_1q + \mu_2)^2$. Notice that the reversed rates x_a, x_b, x_c, x_d are strictly positive for any value of the forward rates. Now we can obtain the load factor of each station:

$$\rho_1 = \frac{x_c + \lambda_1}{\mu_1}, \quad \rho_2 = \frac{\lambda_2}{\mu_2 + x_a}, \quad \rho_3 = \frac{x_d + x_b}{\mu_3}.$$

The stationary distribution, if $\rho_i < 1$ for all $i = 1, 2, 3$ is given by Eq. (1), where $g_i(n_i) = (1 - \rho_i)\rho_i^{n_i}$ and $G = 1$.

The PITS can have a cyclic topology and hence it may reproduce sophisticated behaviors such as the partial flushing [13,16,17] previously proposed in the G-network theory.

Example 4: We consider a G-network of the class studied in [13] as depicted in Figure 6. In this example a positive customer leaving Q_1 can join Q_2 with probability q as a standard positive customer, while with probability $1 - q$ it affects Q_2 as a signal. If Q_2 is empty the signal has no effect, whereas if $n_2 > 0$ a customer is removed and there is an instantaneous propagation of the signal to Q_3 . The signal here behaves exactly as in Q_2 but it propagates to Q_2 . Therefore, when a customer leaves Q_1 as a signal for Q_2 the effect is an instantaneous iterative deletion of customers from Q_2 and Q_3 which ends when the first empty queue is encountered. For instance, consider the state $(3, 2, 1)$ where each component represents the state of queue Q_1, Q_2 and Q_3 , respectively. In the CTMC underlying the G-network we have

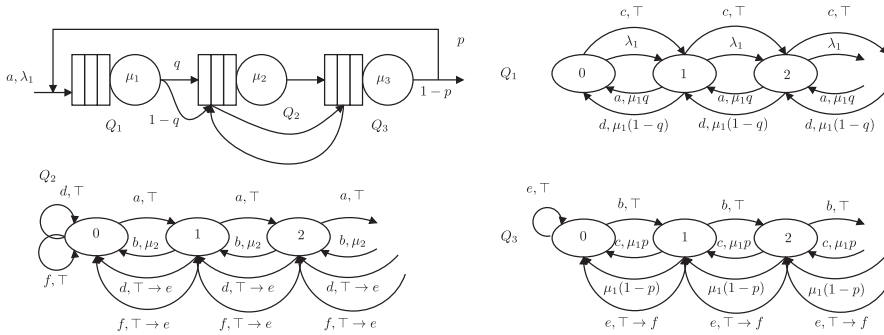


FIGURE 6. G-network with partial flushing.

a transition to state $(2, 0, 0)$ with rate $\mu_1(1 - q)$. Let us derive the rate equations according to the approach proposed in [49] by considering Q_1 in isolation. Assuming that the rate of the transitions labeled c is x_c we immediately obtain the equations associated with x_a and x_d (which are clearly the same for each state):

$$x_a = (x_c + \lambda_1)q, \quad x_d = (x_c + \lambda_1)(1 - q).$$

In Q_2 we assume that all the transitions labeled a, d and f have rates x_a, x_d and x_f , respectively. Hence, we have $x_b = x_a\mu_2/(\mu_2 + x_d + x_f)$. In the analysis of label e , we must note that for each state there are two incoming arcs labeled e : one deriving from the propagation of a synchronization on d and the other on f . Therefore, we must sum the reversed rates of these transitions as shown in [58]:

$$x_e = x_a \frac{x_d + x_f}{\mu_2 + x_d + x_f}.$$

Finally, the analysis of Q_3 gives the following rate equations:

$$x_c = x_b \frac{\mu_1 p}{\mu_1 + x_e}, \quad x_e = x_b \frac{x_e}{\mu_1 + x_e}.$$

It is possible to show that these rate equations have a unique valid solution and hence the G-network is in product-form according to Eq. (1), with $g_i(n_i) = (1 - \rho_i)\rho_i^{n_i}$, $G = 1$, and:

$$\rho_1 = \frac{x_c + \lambda_1}{\mu_1}, \quad \rho_2 = \frac{x_a}{\mu_2 + x_d + x_f}, \quad \rho_3 = \frac{x_b}{\mu_1 + x_e}.$$

4. ON THE RELATIONS AMONG SOME G-NETWORK PRODUCT-FORMS

So far, we have studied G-networks in which the model underlying each station is simple in the sense that it can be studied as a reversible component. As a consequence the derivation of the rate equations is simple. Nevertheless, we note that the network studied in Example 4 is peculiar because the “jumps” in the process underlying a single station are not necessarily among adjacent states, that is, from n to $n + 1$ or $n - 1$. The iterative instantaneous deletion of customers can indeed move one station from state n to a state $n' \leq n$ which depends on the population of the network. In this section, we consider a set of G-networks whose process underlying a single station cannot be trivially seen as a Birth&Death process and

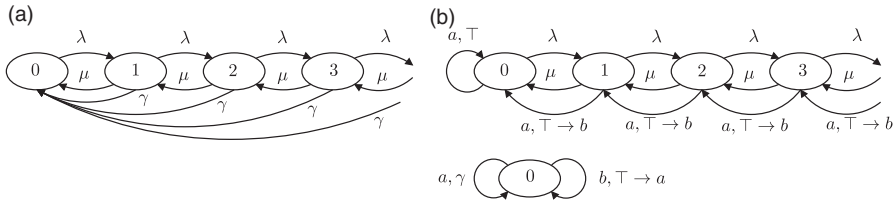


FIGURE 7. G-queue with catastrophes. (A) Underlying CTMC. (B) Modeling with PITs.

hence is more complicated to study with respect to the G-queues seen so far. Nevertheless, we show that an appropriate modelling with the PITs allows us to derive the equation for the equilibrium distribution *as if* we are dealing with a reversible process.

4.1. G-networks with Catastrophes

In [9,15] the authors consider a G-network in which signals can completely flush a queue. According to [9] we call this type of signal *catastrophe*. Figure 7(A) shows the CTMC underlying a single G-queue with catastrophes while Figure 7(B) shows its modeling with PITs and an underlying Birth&Death process. By inspecting the latter figure we can immediately see that if the rate equations are satisfied then the equilibrium probability is geometrically distributed and that the throughput of the queue is the (constant) reversed rate of the transitions with rate μ , that is, $\lambda\mu/(\mu + x_a)$. Let us consider the model with PITs. From Figure 7(B) we derive the following system of rate equations:

$$\begin{cases} x_b = \lambda x_a / (\mu + x_a) \\ x_a = \gamma + x_b \end{cases}$$

whose unique solution is

$$x_a = \frac{1}{2} \left(\gamma + \lambda - \mu + \sqrt{(\gamma + \lambda - \mu)^2 + 4\gamma\mu} \right),$$

$$x_b = \frac{1}{2} \left(-\gamma + \lambda - \mu + \sqrt{(\gamma + \lambda - \mu)^2 + 4\gamma\mu} \right)$$

from which we derive:

$$\rho = \frac{\gamma + \lambda + \mu - \sqrt{(\gamma + \lambda - \mu)^2 + 4\gamma\mu}}{2\mu}, \tag{2}$$

which reproduces the result presented in [15]. However, notice that differently from this work, we do not use the GBEs of the whole network to prove the product-form, but we study the isolated components reduced to a reversible process and hence the proof is simplified.

4.2. Gelenbe&Fourneau Resets (G-resets)

What we call Gelenbe&Fourneau resets (G-resets) are a class of signals introduced in [31] that behave as follows. When a G-reset arrives at a non-empty queue, then it removes one customer as a negative signal. When the G-reset arrives at an empty queue, then its population jumps to $n > 0$ selected with probability $\pi(n)/(1 - \pi(0))$, where $\pi(n)$ is the marginal equilibrium distribution of that queue. In order to model such a G-queue with a

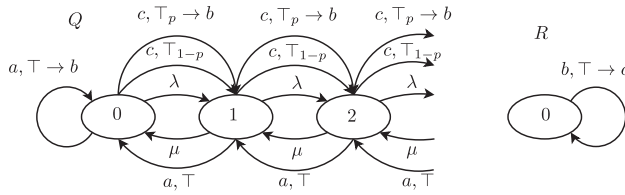


FIGURE 8. G-queue with Gelenbe&Fournau resets.

simplified model with an underlying Birth&Death process we resort again to the propagation of instantaneous transitions. First, we introduce the following notation: In the case that for a state s there are more than one outgoing passive transition with the same label, we can assign a probability of synchronization in an analogue way of what happens for SAN [62], PEPA [49,50]. Here, with an abuse of notation, we write as subscript of the symbol \top the probability that the synchronization occurs with that transition, e.g., from state s we may have two outgoing transitions with label a written as (a, \top_p) and (a, \top_{1-p}) . In this case, when the active component performs a transition labeled a the passive performs the first transition with probability p and the latter with probability $1 - p$. The sum of the probabilities associated with the same label must always be 1.

Let us consider the model with PITs depicted in Figure 8. We assume that the events generating a signal labeled a occur with rate γ . We observe that the structural conditions of RCAT are satisfied for the passive transitions since the sum of the probability of synchronization outgoing each state is 1. As concerns the active transitions, we observe that every state of the models has one incoming active transition. The reset takes the state of the queue to $n > 0$, where n is chosen according to a geometric distribution with parameter $1 - p$. Thus, we have the following rate equations:

$$\begin{cases} x_b = \gamma & \text{State 0 in } Q \\ x_b = (\mu + \gamma)x_cp/(\lambda + x_c) & \text{State } n > 0 \text{ in } Q \\ x_c = x_b & \text{State 0 in } R, \end{cases}$$

which can be satisfied if and only if $p = (\gamma + \lambda)/(\gamma + \mu)$ and gives as a solution $x_a = x_b = x_c = \gamma$. Thus $p = \rho$, where ρ is such that $\pi(n) = (1 - \rho)\rho^n$. This reproduces the result presented in [31].

4.3. Fournau Resets

Fournau’s resets have been proposed in [14] and behaves as the G-resets with the difference that when a reset signal arrives at a non-empty queue it vanishes. Proving the product-form equilibrium distribution of a G-network with F-resets by resorting to the system of GBEs does not allow us to fully exploit the similarity of this signal behavior with the one called G-reset. Conversely, the high-level approach that we are adopting allows us to derive the equilibrium distribution very quickly. The model associated with the F-reset based on PITs is shown in Figure 9. The rate equations are:

$$\begin{cases} x_b = \gamma & \text{State 0 in } Q \\ x_b = \mu x_cp/(\lambda + x_c) & \text{State } n > 0 \text{ in } Q \\ x_c = x_b & \text{State 0 in } R, \end{cases}$$

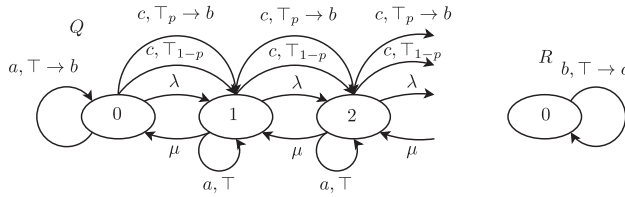


FIGURE 9. G-queue with Fourneau resets.

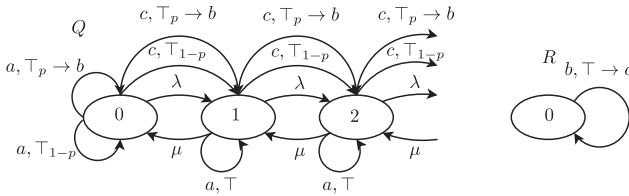


FIGURE 10. G-queue with Harrison resets.

which can be satisfied if and only if $p = (\gamma + \lambda)/\mu$ and gives as a solution $x_a = x_b = x_c = \gamma$. Thus $p = \rho$, where ρ is such that $\pi(n) = (1 - \rho)\rho^n$. This reproduces the result presented in [14].

4.4. Harrison Resets

The resets proposed by Harrison in [47] are similar to those proposed by Fourneau in [14] but when the signal arrives at the empty queue the destination state is chosen according to the equilibrium distribution for $n \geq 0$ (for the F-resets and G-resets we have $n > 0$). We will see that this modification on the reset behavior has important consequences. The model is shown in Figure 10. We can easily check that RCAT structural conditions are satisfied. The rate equations are:

$$\begin{cases} x_b = \gamma p & \text{State 0 in } Q \\ x_b = \mu x_c p / (\lambda + x_c) & \text{State } n > 0 \text{ in } Q \\ x_c = x_b & \text{State 0 in } R. \end{cases}$$

The system admits a solution if $p = \lambda / (\mu - \gamma)$ and gives as a solution $x_a = x_b = x_c = \gamma \lambda / (\mu - \gamma)$. Thus $p = \rho$, where ρ is such that $\pi(n) = (1 - \rho)\rho^n$. This reproduces the result presented in [47]. It is interesting to notice that H-resets can propagate themselves as PITs differently from F-resets [49]. The motivation is that the final state of a F-reset can never be the one representing the empty queue and hence RCAT structural conditions are trivially not satisfied.

4.5. G-Networks with Batch Removal

In [21] Gelenbe proves that if the effect of a signal arriving at a G-queue is that of removing a batch of customers, then the G-network maintains a product-form equilibrium distribution. The batch size can be modeled by an arbitrary discrete random variable. If the queue has the same or less customers than the batch size, then the effect of the signal is flushing the queue. Also the product-form of this class of G-queues can be reduced to the analysis of a Birth&Death process by resorting to the PITs [49]. Let D be the random variable that models the batch size at a given G-queue and let $\vec{p} = (p_1, p_2, \dots)$ be a vector of conditional

probabilities such that p_1 is the probability of deleting at least one customer and p_i is the probability of deleting at least i customers given that $D \geq i - 1$. The model with PITs uses labels a_i and b_i , $i > 0$, to count the number of deletions. The model Q for the G-queue is defined as follows:

- From each state $n \geq 0$ there is a transition with rate λ to state $n + 1$;
- From each state $n > 0$ there is a transition with rate μ to state $n - 1$ and for each $i = 1, 2, \dots$ there is a transition to state $n - 1$ labeled $a_i, \top \rightarrow b_i$;
- At state 0, for each $i = 1, 2, \dots$ there is a self-loop labeled a_i, \top .

The model R for the propagation of the PITs consists of a single state with the following transitions:

- A self-loop labeled $a_1, \gamma p_1$, where γ is the arrival rate of the signal at the queue
- For each $i = 2, 3, \dots$ there is a self-loop labeled $b_i, \top_{p_{i+1}} \rightarrow a_{i+1}$.

It is easy to see that RCAT structural conditions are satisfied. Hence, in order to derive the equilibrium distribution we have to find a solution to the following system of rate equations:

$$\begin{cases} x_{a_1} = \gamma p_1 \\ x_{a_i} = x_{b_{i-1}} p_i \\ x_{b_i} = \lambda x_{a_i} / \left(\mu + \sum_{i=1}^{\infty} x_{a_i} \right) \end{cases} \quad \text{for } i > 1.$$

Then, the equilibrium distribution of the queue is geometrically distributed with $\rho = \lambda / (\mu + \sum_{i=1}^{\infty} x_{a_i})$.

5. CASE STUDY

In this section, we apply the results presented so far to study an heterogeneous G-network. The model is shown in Figure 11. Customers arrive from the outside at stations Q_1 and Q_2 according to independent Poisson processes with rates λ_1 and λ_2 , respectively. After being served by Q_1 , a customer can enter Q_3 as a positive customer or a catastrophe signal with probability p_a and p_b , respectively ($p_a + p_b = 1$). After being served by Q_2 the customer moves to Q_4 . After being served by Q_3 a customer can move to Q_2 as a positive customer with probability p_c or can join Q_1 as a signal. The effect of this signal is to iteratively delete the customers of Q_1 and Q_2 starting from Q_1 until the first empty queue is found. Clearly, we have $p_c + p_d = 1$. After being served by Q_4 , a customer can either

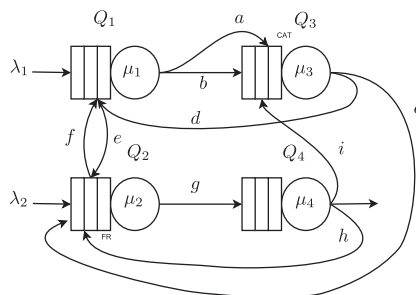


FIGURE 11. Example of G-network in product-form.

leave the network with probability $1 - p_h - p_i$ or move to Q_2 as a signal behaving as a Fourneau reset with probability p_h or move to Q_3 as a negative customer with probability p_i , $p_i + p_h \leq 1$. Henceforth x_t , with $t = a, \dots, i$ denotes the reversed rate associated with the synchronization t depicted in Figure 11. We start with the analysis of Q_1 . According to the models of Section 3 we have the following rate equations:

$$\begin{cases} x_a = \lambda_1 \mu_1 p_a / (\mu_1 + x_d + x_f), \\ x_b = \lambda_1 \mu_1 p_b / (\mu_1 + x_d + x_f), \\ x_e = \lambda_1 (x_d + x_f) / (\mu_1 + x_d + x_f) \end{cases}$$

and $\rho_1 = \lambda_1 / (\mu_1 + x_d + x_f)$. Now we study Q_2 . We studied in the previous section that the load factor of a G-queue with Fourneau resets is:

$$\rho_2 = \frac{\underbrace{(\lambda_2 + x_c + x_h)}_{\text{arrivals}}}{\underbrace{(\mu_2 + x_e)}_{\text{death rate}}},$$

Therefore, we can easily derive the rate equations as:

$$\begin{cases} x_g = \rho_2 \mu_2 \\ x_f = \rho_2 x_e \end{cases}$$

Let us consider Q_3 . The load factor of a G-queue with catastrophes is given by Eq. (2). In our case, the total death rate of the process underlying the queue is $\mu_3 + x_i$; hence we obtain:

$$\rho_3 = \frac{x_b + (\mu_3 + x_i) + x_a - \sqrt{(x_b + x_a - (\mu_3 + x_i))^2 + 4x_a(\mu_3 + x_i)}}{2(\mu_3 + x_i)},$$

and hence:

$$\begin{cases} x_c = \rho_3 \mu_3 p_c \\ x_d = \rho_3 \mu_3 p_d. \end{cases}$$

Finally Q_4 originates the following rate equations:

$$\begin{cases} x_h = x_g p_h \\ x_i = x_g p_i \end{cases}$$

and $\rho_4 = x_g / \mu_4$. The solution of the rate equation system gives the load factor ρ_i for the product form solution:

$$\pi(n_1, n_2, n_3, n_4) = \prod_{i=1}^4 (1 - \rho_i) \rho_i^{n_i}.$$

6. CONCLUSION

In this paper, we have addressed the problem of characterizing the product-form of G-networks. The product-form of G-networks has been a breakthrough in the analysis of Markovian models since it enlarges the class of models which are analytically or numerically tractable. In order to characterize the G-networks' product-form, we observe that most of the single class G-networks have a product-form which is based on the product of

the marginal geometric distributions of the single queues. From this observation we have shown that it is possible to use the PITs presented in [49] in order to reduce the generally non-reversible CTMC underlying a G-queue to an equivalent (in terms of steady-state distribution) reversible CTMC. The analysis of reversible CTMCs simplifies the derivation of the equilibrium distribution of the G-network. There is clearly a trade-off in choosing the technique to use in order to prove the product-form of a G-network. In fact, on the one hand the standard approach consisting in guessing the expression of the equilibrium distribution and verifying that it satisfies the system of GBEs is very general and does not require one to introduce formalisms to specify the synchronizations of the stations which form the queueing network. However, this approach requires a strong intuition about the candidate expression for the product-form if the G-network has complicated behavior, and also requires long proofs which must take into account all the possible transitions in the CTMC underlying the networks. On the other hand, we have the RCAT-based proofs which rely on the verification of some sufficient conditions for the product-form. Therefore, we may have product-forms which are not “detected” by RCAT (see, e.g., [7] for a counterexample to the necessity of RCAT and ERCAT conditions for the product-form). Another drawback is that since the proofs are modular, that is, they verify some conditions for each isolated station, there must be a formalism that specifies how these stations cooperate once they are connected in a network. This issue should not be underestimated. For instance we showed that the standard PEPA-based synchronization semantics or the one based on the Kronecker’s product operator are not suitable for modelling the G-networks triggers’ behaviors. However, the main advantage of RCAT is that, when applicable, it allows for very compact proofs even for heterogeneous networks such as those studied in [2,49]. Finally, it is important to notice that the rate equations associated with the models with PITs can be automatically derived with the algorithm proposed in [49] and solved numerically with one of the general algorithms available in the literature such as those proposed in [2,9,56,57].

References

1. Albarelli, A., Rodolà, E., Bergamasco, F., & Torsello, A. (2011). A non-cooperative game for 3D object recognition in cluttered scenes. In *Proceedings of Int. Conf. on 3D Imaging, Modeling, Processing, Visualization and Transmission, 3DIMPVT*, Hangzhou, China, pp. 252–259.
2. Balsamo, S., Dei Rossi, G., & Marin, A. (2010). A numerical algorithm for the solution of product-form models with infinite state spaces. In *Proceedings of EPEW* (A. Aldini, M. Bernardo, L. Bononi, & L.S. Cortellessa, V. Eds.), Berlin Heidelberg: Springer, vol. 6342, pp. 191–206.
3. Balsamo, S., Harrison, P.G., & Marin, A. (2010). A unifying approach to product-forms in networks with finite capacity constraints. In *Proceedings of SIGMETRICS*, New York, NY, USA, pp. 25–36.
4. Balsamo, S., Harrison, P.G., & Marin, A. (2012). Methodological Construction of Product-form Stochastic Petri-Nets for Performance Evaluation. *Journal of System and Software* 85(7): 1520–1539.
5. Balsamo, S. & Marin, A. (2007). Queueing networks in formal methods for performance evaluation. In *Formal Methods for Performance Evaluation* (M. Bernardo & S. J. Hillston, Eds.), LNCS, vol. 4468, Berlin, Heidelberg: Springer, pp. 34–82.
6. Balsamo, S. & Marin, A. (2011). Performance engineering with product-form models: efficient solutions and applications. In *Proceedings of ICPE’11—Second Joint WOSP/SIPEW International Conference on Performance Engineering*, Karlsruhe, Germany, pp. 437–448.
7. Balsamo, S. & Marin, A. (2013). Separable solutions for Markov processes in random environments. *European Journal of Operational Research* 229(2): 391–403.
8. Baskett, F., Chandy, K.M., Muntz, R.R., & Palacios, F.G. (1975). Open, closed, and mixed networks of queues with different classes of customers. *Journal of ACM* 22(2): 248–260.
9. Chao, X., Miyazawa, M., & Pinedo, M. (1999). *Queueing networks. Customers, Signals and Product Form Solutions*. West Sussex, England: Wiley.
10. Fourneau, J.M., Gelenbe, E., & Suros, R. (1996). G-Networks with multiple classes of positive and negative customers. *Theoretical Computer Science* 155: 141–156.

11. Fourneau, J.M., Plateau, B., & Stewart, W.J. (2007). Product form for stochastic automata networks. In *Proceedings of Valuetools*. ICST, Brussels, Belgium: ICST, pp. 1–10.
12. Fourneau, J.M., Plateau, B., & Stewart, W.J. (2008). An algebraic condition for product form in stochastic automata networks without synchronizations. *Performance Evaluation* 65(11–12): 854–868.
13. Fourneau, J.M. & Quesette, F. (2006). Computing the steady-state distribution of G-networks with synchronized partial flushing. In *Proceedings of ISCIS, 21th International Symposium*, Istanbul, Turkey, pp. 887–896.
14. Fourneau, J.-M. (2007). Closed G-networks with resets: product form solution. In *Proceedings of QEST*, Edinburgh, Scotland, UK, pp. 287–296.
15. Fournau, J.-M., Kloul, L., & Quesette, F. (1995). G-networks with jumps back to zero. In *Proceedings of ACM/IEEE MASCOTS*, Duram, NC, USA, pp. 28–32.
16. Fourneau, J.-M., Kloul, L., & Quesette, F. (2000). Multiple class G-networks with iterated deletions. *Performance Evaluation* 42(1): 1–20.
17. Fourneau, J.-M., Kloul, L., & Verchère, D. (2000). Multiple class G-networks with list-oriented deletions. *European Journal of Operational Research* 126(2): 250–272.
18. Fourneau, J.-M. & Wolter, K. (2015). Mixed networks with multiple classes of customers and restart. In *Proceedings of ASMTA*, Albena, Bulgaria, pp. 73–86.
19. Gelenbe, E. (1989). Random neural networks with negative and positive signals and product form solution. *Neural Computation* 1(4): 502–510.
20. Gelenbe, E. (1993). G-networks with triggered customer movement. *Journal of Applied Probability* 30: 742–748.
21. Gelenbe, E. (1993). G-networks with signals and batch removal. *Probability in the Engineering and Informational Sciences* 7: 335–342.
22. Gelenbe, E. (1993). G-networks: A unifying model for neural nets and queueing networks. In *Proceedings of MASCOTS*, San Diego, CA, USA, pp. 3–8.
23. Gelenbe, E. (1993). Learning in the recurrent random neural network. *Neural Computation* 5: 154–164.
24. Gelenbe, E. (1994). A unifying model for neural and queueing networks. *Annals of Operations Research* 48: 433–461.
25. Gelenbe, E. (2007). Steady-state solution of probabilistic gene regulatory networks. *Physics Review E* 76(3): 031903.
26. Gelenbe, E. (2009). Steps toward self-aware networks. *Communications of the ACM* 52(7): 66–75.
27. Gelenbe, E. (2011). Energy packet networks: ICT based energy allocation and storage. In *Proceedings of GreeNets*, Colmar, France, pp. 186–195.
28. Gelenbe, E. (2012). Energy packet networks: smart electricity storage to meet surges in demand. In *Proceedings of SIMUTOOLS*, Sirmione-Desenzano, Italy, pp. 1–7.
29. Gelenbe, E. (2014). Adaptive management of energy packets. In *Proceedings of IEEE COMPSAC*, Vasteras, Sweden, pp. 1–6.
30. Gelenbe, E., Feng, Y., & Krishnan, K. (1996). Neural network methods for volumetric magnetic resonance imaging of the human brain. *Proceedings of IEEE* 84(10): 1488–1496.
31. Gelenbe, E. & Fourneau, J.-M. (2002). G-networks with resets. *Performance Evaluation* 49: 179–191.
32. Gelenbe, E., Lent R., & Xu, Z. (2001). Design and performance of cognitive packet networks. *Performance Evaluation* 46(2–3): 155–176.
33. Gelenbe, E., Lent, R., & Xu, Z. (2001). Measurement and performance of a cognitive packet network. *Computer Networks* 37(6): 691–701.
34. Gelenbe, E., Harman, K., & Krolik, J. (1998). Learning neural networks for detection and classification of synchronous recurrent transient signals. *Signal Processing* 64(3): 233–247.
35. Gelenbe, E. & Kazhmaganbetova, Z. (2014). Cognitive packet network for bilateral asymmetric connections. *IEEE Transactions Industrial Informatics* 10(3): 1717–1725.
36. Gelenbe, E. & Labeled, A. (1998). G-networks with multiple classes of signals and positive customers. *European Journal of Operational Research* 108(2): 293–305.
37. Gelenbe, E. & Marin, A. (2015). Interconnected wireless sensors with energy harvesting. In *Proceedings of ASMTA*, Albena, Bulgaria, pp. 87–99.
38. Gelenbe, E. & Morfopoulou, C. (2011). A framework for energy-aware routing in packet networks. *Computer Journal* 54(6): 850–859.
39. Gelenbe, E. & Morfopoulou, C. (2012). Power savings in packet networks via optimised routing. *MONET* 17(1): 152–159.
40. Gelenbe, E. & Schassberger, M. (1992). Stability of product form G-networks. *Probability in the Engineering and Informational Sciences* 6: 271–276.

41. Gelenbe, E., Sungur, M., Cramer, C., & Gelenbe, P. (1996). Traffic and video quality with adaptive neural compression. *Multimedia Systems* 4(6): 357–369.
42. Gelenbe, E. & Timotheou, S. (2008). Random neural networks with synchronized interactions. *Neural Computation* 20(9): 2308–2324.
43. Gelenbe, E., Timotheou, S., & Nicholson, D. (2010). Fast distributed near-optimum assignment of assets to tasks. *Computer Journal* 53(9): 1360–1369.
44. Gelenbe, E. & Tugce Ceran, E. (2015). Central or distributed energy storage for processors with energy harvesting. In *Proceedings of Sustain IT*, Madrid, Spain, pp. 1–3.
45. Gordon, W.J. & Newell, G.F. (1967). Cyclic queueing networks with exponential servers. *Operations Research* 15(2): 254–265.
46. Harrison, P.G. (2003). Turning back time in Markovian process algebra. *Theoretical Computer Science* 290(3): 1947–1986.
47. Harrison, P.G. (2004). Compositional reversed Markov processes, with applications to G-networks. *Performance Evaluation* 57(3): 379–408.
48. Harrison, P.G. & Lee, T.T. (2005). Separable equilibrium state probabilities via time reversal in Markovian process algebra. *Theoretical Computer Science* 346(1): 161–182.
49. Harrison, P.G. & Marin, A. (2014). Product-forms in multi-way synchronizations. *Comp. J.* 57(11): 1693–1710.
50. Hillston, J. (1996). *A compositional approach to performance modelling*. Cambridge, UK: Cambridge University Press.
51. Jackson, J.R. (1963). Jobshop-like queueing systems. *Management Science* 10: 131–142.
52. Kelly, F. (1979). *Reversibility and stochastic networks*. New York: Wiley.
53. Kim, H. & Gelenbe, H. (2012). Stochastic gene expression modeling with hill function for switch-like gene responses. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 9(4): 973–979.
54. Kim, H., Park, T., & Gelenbe, E. (2014). Identifying disease candidate genes via large-scale gene network analysis. *IJDDB* 10(2): 175–188.
55. Marin, A., Balsamo, S., & Harrison, P.G. (2012). Analysis of stochastic Petri nets with signals. *Performance Evaluation* 69(11): 551–572.
56. Marin, A. & Bulò, S.R. (2009). A general algorithm to compute the steady-state solution of product-form cooperating Markov chains. In *Proceedings of MASCOTS*, London, UK, September 2009, pp. 515–524.
57. Marin, A., Rota Bulò, S., & Balsamo, S. (2012). A numerical algorithm for the decomposition of cooperating structured Markov processes. In *Proceedings of MASCOTS*, pp. 401–410.
58. Marin, A. & Vigliotti, M.G. (2010). A general result for deriving product-form solutions of Markovian models. In *Proceedings of First Joint WOSP/SIPEW International Conference on Performance Engineering*, San Josè, CA, USA: ACM, pp. 165–176.
59. Muntz, R.R. (1972). Poisson departure processes and queueing networks. Yorktown Heights, New York, Technical Report. IBM Research Report RC4145.
60. Lazowska, E.D., Zahorjan, J.L., Graham, G.S., & Sevcick, K.C. (1984). *Quantitative system performance: computer system analysis using queueing network models*. Englewood Cliffs, NJ: Prentice-Hall.
61. Phan, H.T.T., Sternberg, M.J.E., & Gelenbe, E. (2012). Aligning protein-protein interaction networks using random neural networks. In *Proceedings of IEEE International Conference on Bioinformatics and Biomedicine, BIBM*, Philadelphia, PA, USA, pp. 1–6.
62. Plateau, B. (1985). On the stochastic structure of parallelism and synchronization models for distributed algorithms. *SIGMETRICS Perform. Eval. Rev.* 13(2): 147–154.
63. Torsello, A., Rodolà, E., & Albarelli, A. (2011). Sampling relevant points for surface registration. In *Proceedings of International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission, 3DIMPVT*, Hangzhou, China, pp. 290–295.