Product-forms in multi-wise synchronisations

Peter G. Harrison* and Andrea Marin**

*Department of Computing, Imperial College, London SW7 2BT, UK
**DAIS, Università Ca' Foscari Venezia, Italy
Email: pgh@doc.ic.ac.uk marin@dais.unive.it

A new algorithm is given to find product-form solutions for the joint equilibrium probabilities in a class of synchronised Markov processes. This is based on, and proved by, multiple applications of the Reversed Compound Agent Theorem (RCAT) and can describe chains of pairwise synchronisations that occur in a prescribed order. The length of the sequence is unbounded but finite with probability one. Several applications are given to illustrate the methodology, which include various modes of resets in queueing networks with negative customers. In particular, it is shown that there is a type of reset that can propagate further transitions in a chain actively. Furthermore, a number of completely new product-form models, for example where the transitions in a chain are non-homogeneous, are given.

1. 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 [1], Markovian process algebra [2] and stochastic Petri nets [3], 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 state space explosion, where the number of states of the model representing the system may 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 various formalisms, e.g. [4, 5, 6]. In [7], 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 [4], it is required to solve the *linear* system of traffic equations. However, in the case of the more expressive G-networks [5, 8, 9, 10], a product-form requires non-linear traffic equations to be satisfied. G-networks are a class of product-form queueing networks in which both positive and negative customers are allowed. Positive customers behave exactly as the customers of Jackson's queueing networks [11], increasing the queue length at their arrival epochs. Negative customers may arrive at a queueing station either from outside the network or as a result of a positive customer switching 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 vanishes otherwise. This topic has been further explored in [7], where it is shown how to derive the system of equations characterising a product-form model by defining the cooperating components in terms of synchronising CTMCs. Since the methodology can in this way deal with more general models than just queueing networks, we refer to this system of equations as the *rate equations* of the model. In contrast to the analysis of queueing networks, when applying RCAT, more than one rate equation may be associated with the same component, an important feature that may cause rate-dependent product-form conditions.

1.1. Contribution

From the computational point of view, this paper addresses the problem of deriving algorithmically the rate equations associated with a given model of cooperating components. The existence of the rate equations' solution ensures the product-form of the model. The algorithm is based on a novel 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, e.g. queues and Petri nets. The second contribution of the paper shows the algorithm at work in the case of models with synchronisations amongst several components, i.e. not pairwise, such as those studied in [12, 9, 13]. All these models share the characteristic that at a given epoch, more than two components can instantaneously change their state, i.e., the type of synchronisation is more sophisticated than that of pairwise departure-arrival coupling as in [4], for example. In [14, 15], it is shown that iterated applications of RCAT can deal with certain cases of non-pairwise synchronisation. In the present work, we formalise this notion by introducing the idea of *propagation of instantaneous transitions* (PITs) and we show how to generate algorithmically the corresponding system of rate equations. This is particularly challenging in the case of chains or even cycles of PITs. Finally, we study three product-form models based on PITs. The first is based on the so called resets introduced for G-networks in [10]. We compare the different variants of resets that have been proposed in the literature [10, 14, 16], and then consider the ability of each to participate in a chain of PITs. We show that only those proposed in [14] can be part of such a chain of PITs. The second model we consider is a queueing network with finite capacity, where the skipping policy is adopted to handle arrivals at saturated queues [12, 15] and a congestion-handling algorithm is modelled by means of partial flushing of the queues. The latter is a network of queues with positive and negative customers, also having the novel characteristic of positive or negative queue lengths, as well as partial flushing by means of chains of PITs representing customer deletions. With respect to other models proposed in the literature, e.g. [13], this network has the property that, given the state, the number of deleted customers is finite but unbounded.

1.2. Related work

The analysis of product-form models with multi-wise synchronisations were first studied by Gelenbe in [9, 5] for G-networks. The same author showed that the non-linear solution of traffic equations associated with these models admit a unique solution [17]. These equations can be numerically hard to solve, as showed in [13], where it is shown that the simple fixed point iterations fail to find the solution for some networks' configurations. In [15] a multi-wise synchronisation among queues with finite capacities and exponential service time distributions is used to establish a product-form for networks that adopt a skipping policy to handle arrivals at saturated queues. Propagations of signals are viewed as multi-wise synchronisations in [14] but no algorithm is presented nor is a formalism specified to describe precisely the components involved. Multi-wise synchronisations have not always been studied as propagations of signals, as shown by the stochastic Petri net product-form analysis given in [6]. However, when possible, the propagation approach adopted in this paper enhances the modularity of the analysis and reduces its time-complexity. In [18, Ch. 5] previous results on G-networks are reviewed, showing that they can be treated by an extended formulation of the quasi-reversibility property introduced therein. Concerning G-networks with resets, the seminal work was published by Gelenbe and Fourneau in [10], to which they added the sequels [19, 16]. In [14] the notion of resets was enriched by proposing another formulation with propagation properties. This paper is a revised and extended version of [20], presented at the Valuetools 2012 conference. Specifically, we have added:

- the propagation of resets and chains of heterogeneous signals;
- a full proof of correctness of the proposed algorithm;
- some algorithmic model transformations that allow for the application of the Reversed Compound Agent Theorem (RCAT) to models that do not satisfy its conditions. As a consequence we show the equivalence of RCAT and the result given in [21];
- more worked examples.

1.3. Structure of the paper

The remainder of the paper is organised as follows. Section 2.2 introduces the notions of PITs and chains and Section 4.1 describe such a class of product-form models using a process-algebraic notation. Section 4.2 presents the algorithm that derives the system of rate equations and illustrates its application on a toy-example. In Section 5, new product-forms for networks of finite capacity queues with flushing and for queues with positive and negative length are derived, as well as for networks with propagating resets. The paper concludes in Section 6.

2. BACKGROUND

2.1. Product-forms in pairwise cooperations

Product-forms in Jackson [11], Gordon and Newell [22] and BCMP [4] networks only deal with pairwise synchronisations between queues, meaning that at most two stations of the queueing network can change their states at the same instant. Product-forms in models with pairwise synchronisations have been studied extensively and several characterisations have been formulated in terms of both high-level descriptions (e.g. queueing disciplines [23]) and the underlying stochastic processes – see [24, 7, 21] and more recently [18, Ch. 5]. In this section we first give the semantics of pairwise synchronisations and summarise the product-form theorem of [7]. We consider stochastic models that consist of several components $\mathcal{R}_1, \ldots, \mathcal{R}_M$, each with a countable set of states. A transition (or *action*) between two states in any component is assigned a label (or *type*) from the set of symbols $\mathcal{A} \cup \{\tau\}$, where $\mathcal{A}$ is the Roman alphabet. The labels in $\mathcal{A}$ are used for any action that participates in a synchronisation between components, while the type τ is restricted to internal actions that do not synchronise. An action $a \in \mathcal{A}$ is active in a component $\mathcal{R}_i$ if all its occurrences in $\mathcal{R}_i$ have a rate that is a positive real number, while it is passive if all its occurrences have an unspecified rate $\top$. Each synchronising type is active in one component and passive in one other – giving a pairwise synchronisation. A component can perform an action with a synchronising type only jointly with its counterpart and the joint action has the rate specified by the active component. A formal semantics for this type of synchronisation and a derivation, assuming transition times are exponentially distributed, of an underlying CTMC can be found, for instance, in [25, 2]. Observe that a set of components whose transitions are all labelled τ are trivially in product-form because they are independent. The algorithm presented in Section 4 aims at reducing a non-trivial cooperation of components to a model consisting of such independent components.

As a matter of terminology, we consider an *isolated component* as referring to a component of the original model that does not synchronise with any other and whose (originally) passive transitions' rates are re-specified as real numbers; the same value for all passive transitions with the same type. We denote the rate of a transition with type a by $\mathbf{r}(a)$. 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 transitions sharing the same active synchronising type must have the same reversed rate.

Although it is possible to relax some of these conditions, for the sakes of brevity and clarity we limit our analysis to cases covered by RCAT as stated above. We refer to the first two as the *structural conditions* and the last as the *rate condition*. Let x_ℓ be the reversed rate associated with active synchronising type ℓ. Then the component that is passive with respect to ℓ is isolated by setting the rates of all of its transitions with type ℓ to x_ℓ.

2.2. Chains and PITs

Product-form models with synchronisations defined in terms of propagating instantaneous signals are not new. In [9], 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 [11, 4, 5], the 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 [18, Ch. 5]. The idea of multi-wise instantaneous synchronisation has been further extended in the class of G-networks investigated by Fourneau et al. [13] to encompass *chains* of instantaneous state changes. A more general theoretical analysis of this property is carried out in [14, 15] 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 RCAT.

2.3. Positive triggers

A *positive trigger*, or just trigger, is a negative customer that arrives at a node, j say, and sends an ordinary customer from there, if any is present, to another queue, k say [9]. That is, a trigger reduces the queue length of the node j at which it arrives by one and increases the queue length by one at the node k to which it sends the customer it removes. If the trigger is emanated from another node, i say, as a result of a service completion there, three nodes are involved in a composite transition: the queues at i and j are decremented by one, and the queue at k is incremented by one.

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 positive triggers depicted in Figure 1. Notice that this introductory example belongs to the class of G-networks studied in [9] and also characterised in [18, Ch. 5]. Customers 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 $\mathcal{R}_1$,

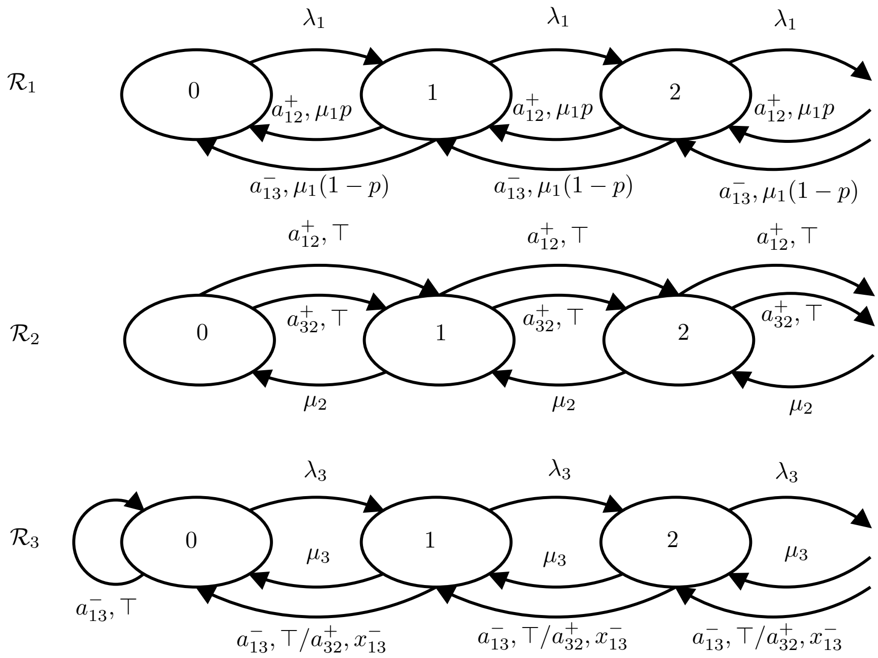

FIGURE 2. Processes underlying the G-network of Figure 1.

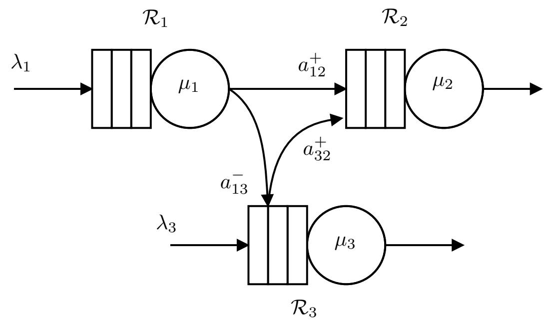

FIGURE 1. G-network with trigger generated after job completion in $\mathcal{R}_1$ that moves a customer from $\mathcal{R}_3$ to $\mathcal{R}_2$.

$\mathcal{R}_3$ 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 shown in 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 [2], i.e., a transition with a specified rate (active) forces a transition with unspecified rate $\top$ (passive) to occur 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 reason for this choice of label will become 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.

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 [24, 7]:

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

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:

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

Now consider the transitions with type a_{32}^+ in $\mathcal{R}_3$. These are active with rate (in the isolated process) x_{13}^-, which was set by their passive synchronisation with $\mathcal{R}_1$. 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

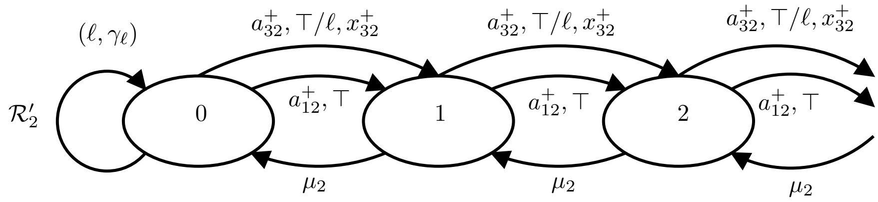

FIGURE 3. Propagation of Instantaneous Transitions.

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.

2.4. Negative triggers and their propagation

A *negative trigger* is similar to a positive one except that the customer removed from node j is changed into a trigger, which may be either positive or negative. If positive, four nodes are involved since on going to node k, the positive trigger will send a customer, if one is present, to another node ℓ. If negative, however, the arrival at node ℓ will also be a trigger which could lead to a chain of negative triggers in a network that will come to a halt only when one arrives at an empty queue or converts into a positive trigger or sends a customer outside the network (like an ordinary negative customer). If cycles are allowed, a whole subset of a network's queues could be emptied. Notice that triggers cause transitions in a sequence of nodes in a strictly specified order. For example, suppose a negative trigger emanates from node 1, passes to node 2 and then to node 3 if node 2 is not empty, so that the node-sequence is $[1, 2, 3]$. This is not the same as a negative trigger with node-sequence $[1, 3, 2]$, for suppose the state of the nodes just before the trigger leaves node 1 is $(3, 2, 0)$. For node-sequence $[1, 2, 3]$, the resulting state will be $(2, 1, 0)$, the negative trigger deleting one positive customer at nodes 1 and 2 but having no effect at node 3. For node-sequence $[1, 3, 2]$, however, the resulting state would be $(2, 2, 0)$, since the chain of triggers would end at node 3 and not continue to node 2.

As an illustrative example we consider the network shown in Figure 4. This example shows that PITs allow for the construction of *chains* of instantaneous transitions of finite but unbounded length. These chains may also form cycles in a similar fashion to what is studied in [13]. 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 parameters μ_1 and μ_2, respectively. PITs are started in $\mathcal{R}_1$

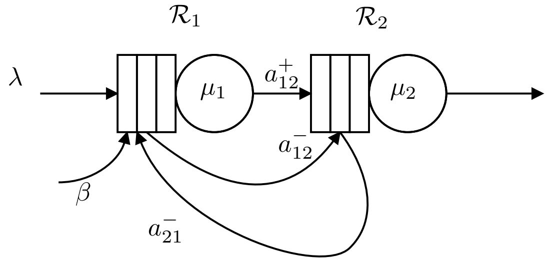

FIGURE 4. G-network with iterative customer removals.

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 [9, 26, 10, 13]. 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 4.2 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.

2.5. Resets and their propagation

After their introduction in 1989, G-networks have been extended in many ways. In particular, resets allow jumps from a queue's empty state to an 'equilibrium state' in a certain sense [10].

More formally, a reset is a kind of negative customer, or *signal*, that may cause a change of state on arrival at an empty queue. Resets may be modelled as passive transitions that synchronise with (active) departures at another queueing node.

In the literature, three types of reset have been proposed:

1. G-resets, introduced by Gelenbe and Foureau

in [10], which cause a decrement of one in the queue length on arrival of a signal at a non-empty queue and, on arrival of a signal to an empty queue, a transition to queue length $n > 0$ with probability $\pi(n)/(1 - \pi(0))$;

2. H-resets, introduced by Harrison in [14], which are very similar to G-resets but cause a transition, on arrival of a signal at an empty queue, to queue length $n \geq 0$ with probability $\pi(n)$ – notice that it is therefore possible for a reset to have *no effect*, with probability $\pi(0)$;
3. F-resets, introduced by Fourneau in [16], which behave as G-resets on arrival at an empty queue but *have no effect on non-empty queues*.

In fact all of these variants yield geometric equilibrium probability distributions. For H-resets this is shown in [14]. For the other variants, let us define a generic reset-queue by the following generators:

- Arrival-transitions with rate λ^+ from all states n to $n+1$ $(n \geq 0)$;
- Service-completion-transitions with rate μ from all states $n+1$ to n $(n \geq 0)$;
- Signal-transitions with rate λ^- from all states $n+1$ to n $(n \geq 0)$;
- Signal-transitions with rate $\nu\pi(n)/(1-\pi(0))$ from state 0 to $n > 0$, for some parameter $\nu > 0$.

Direct solution of the balance equations then yields geometric equilibrium state probabilities of the form $\pi(n) = (1 - \rho)\rho^n$, where $\rho = \frac{\lambda^+ + \nu}{\lambda^- + \mu}$. For G-resets, $\nu = \lambda^-$, giving $\rho_G = \frac{\lambda^+ + \lambda^-}{\mu + \lambda^-}$; for F-resets, $\lambda^- = 0$ and $\rho_F = \frac{\lambda^+ + \nu}{\mu}$. For H-resets, a separate calculation is necessary because of the 'invisible transition' from state 0 to itself, and we can show that $\rho_H = \frac{\lambda^+}{\lambda^- + \mu - \nu}$. In all three variants, all active actions correspond to service completions, which go into every state, and all passive actions, representing either positive or negative (reset) arrivals, are enabled in every state. RCAT is therefore satisfied since we must include an invisible passive action from any state to itself when signals have no effect on that state, so that the node sending the signal is not blocked. A product-form solution is therefore guaranteed in all three cases and all we have to do is solve the rate equations.

3. SPLITTING PASSIVE AND ACTIVE ACTIONS

In some cases, the idea of pairwise synchronisation as the joint transition between an active and a passive component can be difficult to handle for practical purposes. For instance, a queueing discipline satisfies the station balance property if the customer service rate at a particular queue position is proportional to the probability that the customer arrives at that position [23]. Station balance is known to be sufficient for product-form, and its definition implies a probabilistic choice of queue position at a customer arrival epoch; the state reached immediately after the arrival is chosen probabilistically. In our terminology, we refer to this problem as *split passive actions*, since the customer arrival is naturally modelled by means of multiple passive actions.

Conversely, the are also product-forms in which a component's state can be reached with more than one active action with the same label. As an instance, we consider the Last Come First Served with Preemptive Resume (LCFSPR) queueing discipline in BCMP queueing networks with Coxian service time distribution [4]. For simplicity, consider the state representing the empty queue at a single class station. The empty state can be reached by any of the states encoding the queue with 1 customer in any of the Coxian stages of service. Since all these transitions have the same effect on the queues receiving the served customer, it is natural to use the same label for all of them and consequently the empty state in the active queue can be reached by *multiple incoming active actions*.

In the next two subsections, we show that split passive actions and multiple incoming active actions are dual to each other but extend neither the modelling power of the basic active/passive synchronisation nor the resulting product-forms when they exist. However, they allow a much more compact notation for practical applications, they enhance the modularity of model specifications and they facilitate easier automation.

3.1. Split passive actions

When a state s has two or more outgoing instances of a passive action a we have to define a semantics for the synchronisation. It is natural to assign a synchronisation probability to each of the outgoing passive transitions in the style proposed, for instance, in [2, 25, 27]. Suppose that state s has n (we allow $n = \infty$) outgoing transitions labelled a and let $w_1, \dots, w_n$ be the probability of synchronising with each of them, $w_i > 0$ and $\sum_{i=1}^{n} w_i = 1$. Then the transition associated with probability w_i synchronises with the active transition labelled a and rate λ with the joint rate $w_i\lambda$.

Associating the probabilities to the passive transitions can be very useful for practical modelling. For example it allows one to define probabilistic routing in a queueing network in a modular way. This type of synchronisation can be described by RCAT after a model transformation, as follows. We assign to each instance of the outgoing passive transition a different name, say $\{a_1, \dots, a_n\}$ (where the set may be finite or infinite). Then, to achieve the required probabilistic synchronisation with an active action a, we replace each active instance of a, with rate λ say, by parallel instances (i.e., going out from and coming in to the same states) of active actions $\{a_1, \dots, a_n\}$ with rates $\{w_1\lambda, \dots, w_n\lambda\}$. This transformation must be iterated

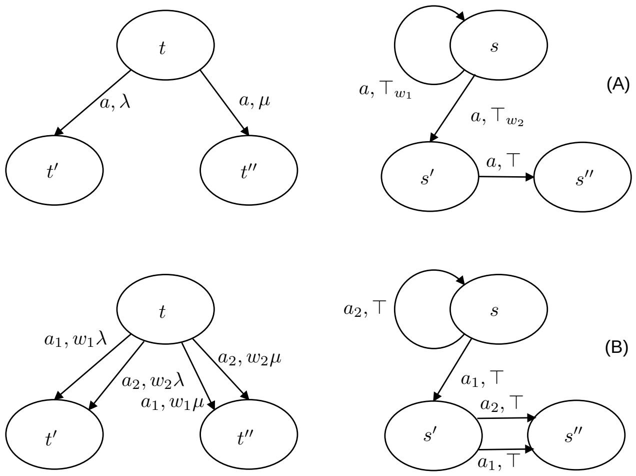

FIGURE 5. Example of split passive actions in abbreviated probabilistic form and its equivalent pure form.

over all probabilistic synchronisations. In this way we can describe (G- and F-) resets at state $s = 0$ exactly when $w_i = \pi(i)/(1 - \pi(0))$ for $i > 0$. However, it is rather cumbersome to modify all the active transitions, which are in components other than the one representing the reset queue itself. An example of an abbreviated, or "shorthand", specification with its transformed, non-probabilistic version is shown in Figure 5 where $(a, \top_{w_i})$ denotes that the passive transition labelled a synchronises with probability w_i.

From the point of view of product-form analysis, if RCAT is applied to the shorthand version directly, the theorem holds (giving the same result) if and only if it holds for the former, unabbreviated specification. This is firstly because the rate constraints still hold, since the synchronising probabilities sum to one. Moreover, in the former version, the rate equation for action a_i is $x_{a_i} = \overline{\mathbf{r}}(a_i)$, where $\overline{\mathbf{r}}(a_i)$ is the reversed rate of the active action a_i with forward rate $w_i\lambda$. But since the active actions $\{a_1, a_2, \ldots\}$ are parallel with total rate λ, the reversed rate of active action a_i is $w_i\overline{\mathbf{r}}(a)$, as in the latter, shorthand specification.

3.2. Multiple incoming active actions

An interesting dual to split passive actions is the case of a state into which there are more than one *active* actions of the same type. In the reversed process, these become split passive actions, with associated probabilities that are proportional to the reversed rates of the corresponding active actions. This must be the case in a product-form network for the following reason. Suppose a synchronising action a is active in component i and passive in j. Consider the passive transition $s'_j \xrightarrow{a,x_a}_j s_j$ and two active transitions $s'_i \xrightarrow{a,\lambda'}_i s_i$, $s''_i \xrightarrow{a,\lambda''}_i s_i$. Then, by Equation 1, the reversed rates of the *synchronised* transitions are $\lambda'(\pi_i(s'_i)/\pi_i(s_i))K$ and $\lambda''(\pi_i(s''_i)/\pi_i(s_i))K$, respectively, where $K = \pi_j(s'_j)/\pi_j(s_j)$ and $\pi_i(\cdot), \pi_j(\cdot)$ denote marginal equilibrium probability functions in components i, j respectively. In the local component i, the reversed rates of the corresponding two instances of active action a are $\lambda'(\pi_i(s'_i)/\pi_i(s_i))$ and $\lambda''(\pi_i(s''_i)/\pi_i(s_i))$, which are in the same ratio.

In fact, this idea relates to an extension of RCAT recently proposed by Marin and Vigliotti [21]. This replaces RCAT's requiring the reversed rates of all individual transitions with the same synchronising active action type to be equal, by the condition that the sum of the reversed rates of all the incoming transitions with the same active type be the same in all states. Clearly, when there is at most one incoming transition for each active action type in each state, both conditions are satisfied. In fact, the "extension" may be proved by the original RCAT itself, so there is no need to provide a new theorem. However, the proof is not obvious, so that the extension is important from the point of view of clarity and automation. An outline of the proof is the following.

Suppose that a network satisfies the proposed extension of RCAT and that in component i, the transitions $s_{i1} \xrightarrow{a,\lambda_1}_i s_i, \ldots, s_{in} \xrightarrow{a,\lambda_n}_i s_i$ are active ($n \geq 1$) and in component j, transition $s'_j \xrightarrow{a,x_a}_j s_j$ is passive. Now define actions $a_1, \ldots, a_n$ and replace the above active transitions in component i by $s_{i1} \xrightarrow{a_1,\lambda_1}_i s_i, \ldots, s_{in} \xrightarrow{a_n,\lambda_n}_i s_i$ respectively, and *all* passive transitions of the form $s' \xrightarrow{a,x_a}_j s$ in component j by the n *parallel* passive transitions $s' \xrightarrow{a_1,x_{a1}}_j s, \ldots, s' \xrightarrow{a_n,x_{an}}_j s$ (including the above case $s' = s'_j, s = s_j$). This circumvents the problem posed by the multiple transitions into the particular state s_i in component i corresponding to active actions of type a. All other instances of active action type a in component i are replaced by the parallel active actions $a_1, \ldots, a_n$ and the procedure repeated to cover all further cases of multiple incoming actions of the same type a_m for $1 \leq m \leq n$. There will be one such case for each state, other than s_i, that also had multiple incoming active actions of type a initially and each will generate a number of new independent action types, $a_{mm'}$ say, for $1 \leq m' \leq n'$, where n' is the number of actions of type a incoming to that state initially. This guarantees, by the hypothesis of the proposed extension, that all outgoing passive and incoming active transitions in the transformed model occur in all states, as required by (original) RCAT. The remaining question is, how to apportion the rates of each parallel active transition in each step so as to ensure that the reversed rate is the same for every instance of each a_m, $1 \leq m \leq n$ (in the first step, without loss of generality). This is straightforward: for every transition $s' \xrightarrow{a,\mu}_i s$ that is replaced by $\{s' \xrightarrow{a_m,\mu_m}_i s \mid 1 \leq m \leq n\}$ in component i, we set $\mu_m = \mu(\overline{\lambda_m}/\sum_{\ell=1}^{n}\overline{\lambda_\ell})$, where $\overline{\lambda_\ell}$ is the reversed rate in isolated component i of the instance of action type a_ℓ referred to above. RCAT may now be applied

$\mathcal{R}_1$	$\mathcal{R}_2$
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 1. MPA definition of the G-network depicted by Figure 4.

in its original form, giving the same product-form as Marin and Vigliotti's extension.

4. AUTOMATIC DERIVATION OF PRODUCT-FORM

In this section we provide an algorithm for the derivation of the rate equations associated with a stochastic model of synchronising components. In order to specify synchronisations with PITs formally, we use a process algebraic formalism.

4.1. Extended process algebra notation

We adapt the syntax of the Performance Evaluation Process Algebra (PEPA) [2] to specify models with PITs. It is assumed that the reader has some familiarity with PEPA and we just introduce the novel equation type that we use. 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 synchronisation, the rate at which the transition synchronises on type b is well-defined and equal to the reversed rate of the synchronising active action with type a, by application of RCAT to the first synchronisation.

Example 1. Consider the model depicted by Figure 4 and described in Section 2. Table 1 shows the equations describing the model using the notation for the PITs that has just been introduced.

4.2. 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 4.1. *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 $\mathcal{P}(\mathcal{R}_i)$ the corresponding set of synchronising passive types. 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 rate equations corresponding to x_ℓ. When it terminates, it returns the complete set of rate equations for the model. If they admit a solution, then the model has a product-form given by the resulting values of the x_ℓ, which fully parameterise the isolated components.

4.3. 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 [7] has to be applied. In practice, the equations for the reversed rates can be found by one of the following approaches:

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 not known, for a finite 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 $N = \sum_c N_c$.

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 sums 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 [7], 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

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
```

component consists of T equations.

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 [11, 5, 9, 26, 10, 14]. 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 4.4.

4.4. 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 1 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 $\mathcal{R}_2$ give:

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

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 [17, 28]. The isolated components are now simple MPA agents without synchronisations, i.e. are 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.

4.5. Correctness and complexity

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

In mathematical terms, any CTMC C can be represented by a graph in which a transition from state s to state t, with instantaneous rate α, is named by some symbol, a say. Treating all transitions thus leads to arcs labelled (a, α). Such a labelled transition system is equivalent to our process algebraic formalism, except that the symbol $\top$ must be replaced by a variable uniquely associated with its label-counterpart, e.g. x_a. Of course, in a model with empty cooperation set $\mathcal{L}$, this situation does not arise. Given this isomorphism, we may talk in terms of properties pertaining to CTMCs when dealing with stochastic process algebra.

Lemma 4.1. *Suppose that synchronising components $\mathcal{R}_i$, representing CTMCs C_i, satisfy the conditions of RCAT. Then, for all $a \in \mathcal{L}$,*

$$\underset{L}{\bowtie} \mathcal{R}_i \quad \textit{and} \quad \underset{L\backslash\{a\}}{\bowtie} \mathcal{R}_i\big((a,\top) \leftarrow (a, \overline{\mathbf{r}}(a)) \mid a \in \mathcal{P}(\mathcal{R}_i)\big)$$

have RCAT rate equations with the same solutions for $\{x_b \mid b \neq a \in \mathcal{L}\}$ and hence the same product-form, where $\overline{\mathbf{r}}(\mathbf{a})$ denotes the reversed rate of active transitions labelled $a \in \mathcal{L}$ and $\leftarrow$ denotes the substitution operator (which can only be applicable in one component $\mathcal{R}_i$ on the right hand side, by hypothesis).

Proof. First, the rate condition of RCAT ensures that $\overline{\mathbf{r}}(a)$ is well defined; in fact $\overline{\mathbf{r}}(a)$ may contain instances of $\{x_b \mid b \neq a \in \mathcal{L}\}$. Moreover, $x_a = \mathbf{r}(\overline{a})$ is one of the rate equations produced by RCAT for the cooperation on the lhs. The rate equations on the rhs are the same as the remaining ones on the lhs, in which the substitution $x_a = \overline{\mathbf{r}}(a)$ is made. The rate equations of the rhs are therefore precisely those of the lhs in which the substitution $x_a \leftarrow \mathbf{r}(\overline{a})$ is also made, i.e. in which the variable x_a is eliminated.

That both cooperations have the same product-form is now immediate from RCAT. □

This result immediately gives the following proposition that we require:

Proposition 4.1. *In the model $\mathcal{M}$ of Definition 4.1, suppose $|\mathcal{L}| > 0$ and that each passive action with type $b \in \mathcal{L}$ is assigned rate-variable x_b in an application of RCAT [7]. Let $\{v_b \mid b \in \mathcal{L}\}$ be a solution of the rate equations for $\{x_b\}$. 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 $\mathcal{L}\backslash\{a\}$, for any one of the types $a \in \mathcal{L}$, defined by assigning, to all the unspecified rates of actions of type a, the value v_a in the component in which they are passive in $\mathcal{M}$.*

Proof. 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 $\mathcal{M}$. The rate equations for $\{x_b \mid b \in L\backslash\{a\}\}$ in $\mathcal{M}_a$ have the same solution as in $\mathcal{M}$ by Lemma 4.1. □

Observe that, as a consequence of Proposition 4.1, the correctness of Algorithm 1 does not depend on the order in which the types in $\mathcal{L}$ are considered. Concerning the computational complexity of Algorithm 1, we observe that the number of recursive calls is exactly equal to the number of synchronising types. We assume that, using suitable indexing structures, the analysis of each 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 4.3, we have a time complexity of $\mathcal{O}(|\mathcal{L}| \cdot N/C)$, with $N = \sum_{i=1}^{C} N_c$ and N_c the number of states of component c.

5. STOCHASTIC NETWORKS WITH PITS

In this section we study some stochastic networks with multi-wise synchronisations using the methodology proposed in Section 4.

5.1. Propagation of H-resets

Here we show that H-resets, differently from the other types of resets (G- and F-), can propagate in PITs. Consider the model depicted in Figure 6. Notice that we use the splitting of passive actions discussed in Section 3 for the sake of a simple notation. Therefore, when we write $\top_p$, we mean that the synchronisation on the associated passive transition occurs with probability p. Observe that the probabilistic synchronisation for label h occurs only in state 0 and that it is well defined under the stability assumption since then $\sum_{i=0}^{\infty} \pi(i) = 1$, where $\pi(i)$ denotes the steady state probability of state i.

Now suppose we wish to allow a reset to synchronise as the *active* action with some passive action in another node (i.e., label ℓ in Figure 6). The reversed rates of these active actions are constant since the reversed rate of a reset from state 0 to state i is $\pi(0)/\pi(i) \cdot \pi(i)x_h = \pi(0)x_h$, a constant. It is worthy of note that this

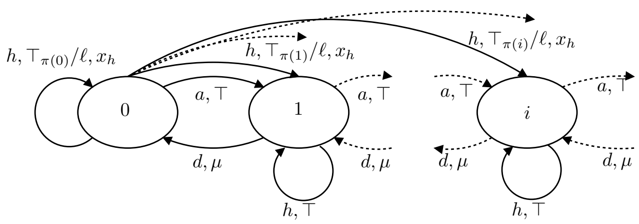

FIGURE 6. Process underlying a queue with H-resets. Positive customer and reset arrivals are modelled by passive actions (a and h, respectively) while customer departure is modelled by active action d.

holds even for non-geometric queues, i.e., H-resets can propagate maintaining a product-form even in the case of state-dependent arrival or service rates.

Since G-resets and F-resets do not have active reset-actions coming in to state 0 and so violate the second condition of RCAT, they cannot propagate according to our notion of PITs. H-resets do satisfy these conditions and can therefore participate actively, possibly contributing to chains of signals involving resets. H-resets are therefore a much richer concept.

5.1.1. Example

We now apply the automatic analysis of PITs to the queueing network of Figure 7. This network has two peculiarities: the first is that the third station, denoted by $\mathcal{R}_3$, has state-dependent arrival rate and therefore is known to be, in general, not in product-form. However, as discussed before, since this station interacts with the other ones only by propagating the H-resets, then the network is still in product-form. The second peculiarity is that the propagation of H-resets forms a cycle. In practice, a reset can be originated by $\mathcal{R}_1$ immediately after a job completion and this goes to $\mathcal{R}_3$. If $\mathcal{R}_3$ is empty, then its state is possibly changed according to $\mathcal{R}_3$'s stationary distribution and the reset propagates to $\mathcal{R}_2$ and back to $\mathcal{R}_3$ until one of the two stations is *not* empty, so that the cycle terminates. The effect, therefore, is to ensure that either $\mathcal{R}_2$ or $\mathcal{R}_3$ is non-empty immediately after the generation of a H-reset by $\mathcal{R}_1$.

The queuing network is expressed by using the process algebraic notation in Table 2, where $p^+_{12}+p^H_{12} = 1$ and $p^+_{21} + p^+_{20} = 1$. By applying Algorithm 1, and observing that a^H_{32} has multiple instances coming in to the same state (see Section 3), we obtain the following system of rate equations:

$$\begin{cases} x^+_{12} = (\lambda_1 + x^+_{21})p^+_{12} \\ x^H_{13} = (\lambda_1 + x^+_{21})p^+_{13} \\ x^H_{32} = \pi_3(0)\left(x^H_{13} + x^H_{23}\right) \\ x^H_{23} = \left(1 - \frac{x^+_{12}}{\mu_2 - x^H_{32}}\right) x^H_{32} \\ x^+_{21} = \frac{x^+_{12}}{\mu_2 - x^R_{32}} \mu_2 p^+_{21} \end{cases} . \qquad (3)$$

Assuming stability, the solution of this system exists and is unique (see [28]). Notice that $\mathcal{R}_3$ may have a non-geometric distribution so we left $\pi_3(0)$ in the system of equations. If $\lambda_3(n) = \lambda_3$ and $\mu_3(n) = \mu_3$, then $\pi_3(0) = 1 - \lambda_3/(\mu_3 - x^H_{13} - x^H_{23})$. In this case the System (3) can be solved numerically. In Figure 8 we show the perceived arrival rate of H-reset signals at $\mathcal{R}_3$ coming from $\mathcal{R}_2$ for the following set of parameters: $\lambda_3 = 0.1$, $\mu_1 = \mu_2 = \mu_3 = 1.0$, $p^+_{12} = p^H_{13} = p^+_{21} = p^+_{20} = 0.5$. We note that while λ_1 grows, the load factor in $\mathcal{R}_2$ grows due to the arrival of positive customers and that of $\mathcal{R}_3$ grows due to the arrivals of H-reset signals. The H-reset signals propagate to $\mathcal{R}_2$ contributing to the augmenting of its load-factor. The plot of Figure 8 shows the propagation rate from $\mathcal{R}_2$ back to $\mathcal{R}_3$ of the reset. We observe an initial growing phase that is caused by the more frequent reset signals generated by $\mathcal{R}_1$ but then, after reaching a maximum, we observe a decreasing phase due to the fact that $\pi_2(0)$ decreases as the load factor of $\mathcal{R}_2$ increases. The product-form expression, in

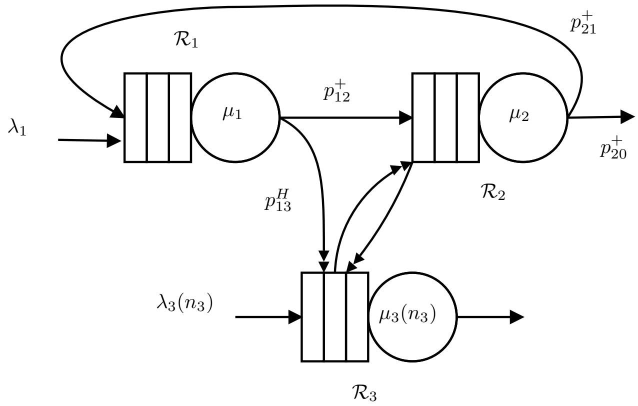

FIGURE 7. Network with propagating H-resets. The double arrows denote the H-reset propagation.

$\mathcal{R}_1$	
1. $P_n = (\tau, \lambda).P_{n+1}$	$n \geq 0$
2. $P_n = (a^+_{21}, \top).P_{n+1}$	$n \geq 0$
3. $P_n = (a^+_{12}, \mu_1 p^+_{12}).P_{n-}$	$n > 0$
4. $P_n = (a^H_{13}, \mu_1 p^H_{13}).P_{n-1}$	$n > 0$
$\mathcal{R}_2$	
1. $Q_n = (a^+_{12}, \top).Q_{n+1}$	$n \geq 0$
2. $Q_n = (a_{21}, \mu_2 p^+_{21}).Q_{n-1}$	$n > 0$
3. $Q_0 = (a^H_{32} \to a^H_{23}, \top_{\pi_2(i)}).Q_i$	$i \geq 0$
4. $Q_n = (a^H_{32}, \top).Q_n$	$n > 0$
5. $Q_n = (\tau, \mu_2 p^+_{20}).Q_{n-1}$	$n > 0$
$\mathcal{R}_3$	
1. $R_n = (\tau, \lambda_3(n)).R_{n+1}$	$n \geq 0$
2. $R_n = (\tau, \mu_3(n)).R_{n-1}$	$n > 0$
3. $R_0 = (a^H_{13} \to a^H_{32}, \top_{\pi_3(i)}).R_i$	$i \geq 0$
4. $R_n = (a^H_{13}, \top).R_n$	$n > 0$
5. $R_0 = (a^H_{23} \to a^H_{32}, \top_{\pi_3(i)}).R_i$	$i \geq 0$
6. $R_n = (a^H_{23}, \top).R_n$	$n > 0$

TABLE 2. MPA definition of the network of Figure 7.

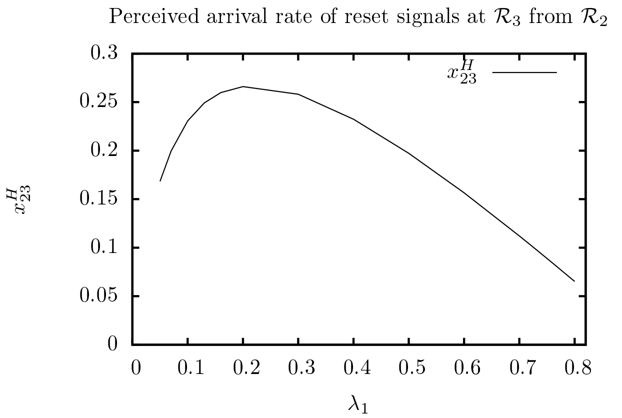

FIGURE 8. Perceived arrival rate of H-reset signals at $\mathcal{R}_3$ coming from $\mathcal{R}_2$.

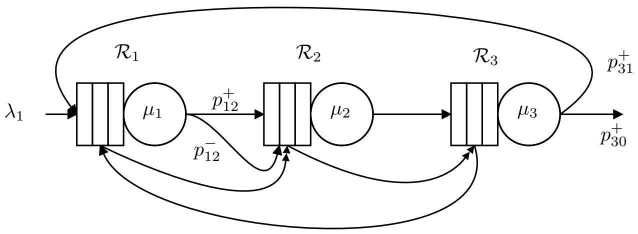

FIGURE 9. Heterogeneous propagation of H-resets and negative triggers.

the case that $\mathcal{R}_3$ has a geometric distribution, is:

$$\pi(n_1, n_2, n_3) = \prod_{i=1}^{3} (1 - \rho_i)\rho_i^{n_i} ,$$

where $\rho_1 = (\lambda_1 + x_{21}^+)/\mu_1$, $\rho_2 = x_{12}^+/(\mu_2 - x_{32}^H)$ and $\rho_3 = \lambda_3/(\mu_3 - x_{13}^H - x_{23}^H)$.

5.1.2. *Inhomogeneous PITs*

In this example we study the queueing network depicted by Figure 9. The peculiarity of this example is that the type of signal propagated as PITs is not homogeneous. First, component $\mathcal{R}_1$ starts the PIT chain after a job completion with probability p_{12}^+; if $\mathcal{R}_2$ is non-empty, the trigger generated by $\mathcal{R}_1$ deletes a customer in $\mathcal{R}_2$ and propagates to $\mathcal{R}_3$ as an H-reset. If $\mathcal{R}_3$ is empty, the propagation continues to $\mathcal{R}_1$ as a negative customer. The cycle is now closed, but notice that the propagation from $\mathcal{R}_1$ to $\mathcal{R}_2$ is no longer as a negative trigger but as a H-reset. This implies that the computation of the reversed rate associated with the propagation of the PIT from $\mathcal{R}_2$ to $\mathcal{R}_3$ requires a non-trivial analysis of the multiple active incoming actions discussed in Section 3. The application of Algorithm 1 gives the following system of rate equations:

$$\begin{cases} x_{31}^+ = \rho_3\mu_3 p_{31}^+ \\ x_{12}^+ = \rho_1\mu_1 p_{12}^+ \\ x_{12}^- = \rho_1\mu_1 p_{12}^- \\ x_{12}^H = (1-\rho_1)x_{31}^H \\ x_{23}^+ = \rho_2\mu_2 \\ x_{23}^H = (1-\rho_2)x(\lambda_1 + x_{31}^+)/(\mu_1 + x_{31}^-) \end{cases} ,$$

where $\rho_1 = (\lambda_1 + x_{31}^+)/(\mu_1 + x_{31}^-)$, $\rho_2 = x_{12}^+/(\mu_2 + x_{12}^- - x_{13}^H)$ and $\rho_3 = x_{23}/(\mu_3 - x_{23}^H)$.

5.2. Finite capacity queues with congestion control

Consider a tandem of nodes $\mathcal{R}_1, \dots, \mathcal{R}_N$ with exponential service times, rates $\mu_1, \dots, \mu_N$ and buffers with finite positive capacities $B_1, \dots, B_N$, as illustrated by Figure 10. 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 [12, 15] 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 but the product-form is derived in a purely automatic way by applying Algorithm 1. Figure 11 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 $a_{N,1}^-$. For $1 < i < N$, calculating the appropriate reversed rates yields the following equations for active type $a_{i,i+1}^+$:

$$x_{i,i+1}^+ = (\lambda_i + x_{i-1,i})\frac{\mu_i}{\mu_i + x_{i-1,i}^-} , \quad (4)$$

$$x_{i,i+1}^+ = (\lambda_i + x_{i-1,i})p_i . \quad (5)$$

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

$$x_{i,i+1}^- = (\lambda_i + x_{i-1,i}^+)\frac{x_{i-1,i}^-}{\mu_i + x_{i-1,i}^-} , \quad (6)$$

$$x_{i,i+1}^- = \gamma_i . \quad (7)$$

Consulting Equations (4) and (5) 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 (6) and (7), 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 solve Equations (4) and (6). For model $\mathcal{R}_1$,

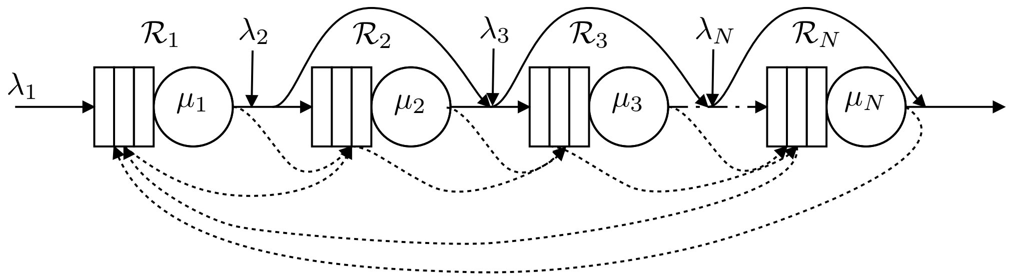

FIGURE 10. Tandem of nodes with finite capacity, skipping and congestion control policy.

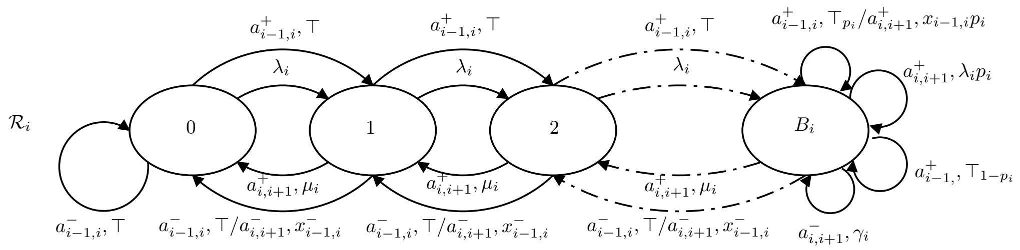

FIGURE 11. Process underlying a finite capacity node i of Figure 10.

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 rate equations, the steady-state probability distribution of the model of Figure 10 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 $1 < i \leq N$, $\rho_1 = \lambda_1/(\mu_1 + x^-_{N,1})$ and G is the normalising constant,

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

5.3. 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 find the product-form for this class of queueing networks and show how Algorithm 1 derives the rate equations automatically. 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.

5.3.1. 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 12 shows the process underlying this type of queue. The equilibrium probability function of the isolated queue is then given by:

$$\pi_i(n) = \begin{cases} \pi_{i0}\rho_{i1}^{n} & \text{if } n \geq 0 \\ \pi_{i0}\rho_{i2}^{-n} & \text{if } n < 0 \end{cases} , \tag{8}$$

where

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

Table 3 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 $\rho_{i1} < 1$

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. $P^i_n = (b, \mu_{i2}).P^i_{n-1}\,, n \le 0$	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. $P^i_n = (c \to d, \top_{p_{i2}}).P^i_{n-1}\,, n \le 0$	Succ. del.
7. $P^i_n = (c, \top_{1-p_{i2}}).P^i_n\,, n \le 0$	Unsucc. del.

TABLE 3. Definition of the components of the queueing network of Section 5.3. Transitions labelled c are the PITs of the components.

and $\rho_{i2} < 1$. We now study the conditions for this queue to yield a product-form stationary distribution. In order that the output of a queue be an input of another (of either positive or negative customers), we require the transitions corresponding to job completions and to customer destructions to have constant reversed rates.

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:

$$p_{i1}\mu_{i2} = p_{i2}\mu_{i1}. \qquad (9)$$

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.:

$$\mu_{i2} + x_c p_{i2} < x_a < \mu_{i1} + x_c p_{i1}. \qquad (10)$$

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 (9):

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

Observe that, if $\mu_{i1} > \mu_{i2}$, then the inequality is satisfied for all $p_{i1} \in [0, 1]$; otherwise it is false. 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 (10) can be satisfied only if $\mu_{i1} > \mu_{i2}$, given that Equation (9) holds. Observe that this is also necessary for the stability of the queue when there is no customer deletion, i.e., when $x_c = 0$.

5.3.2. The queueing network

Consider now the queueing network of Figure 13. Henceforth we assume $p_{i1} = 1$ and choose p_{i2} such that the rate constraint (9) 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:

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

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 condition (9) and the necessary stability condition are sufficient to prevent infinite deletion-cycles. Suppose that $p_{12} = 1$. Then condition (9) 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 precludes infinite cycles of customer deletions in a stable queue, although their length is unbounded.

Once the rate equations are solved, the product-form solution is given by Equation (8), 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 13, 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 14 and 15 show the plots of the average number of customers in the queues as functions of λ for the following set of parameters: $\mu_{11} = 3.0$, $\mu_{21} = 2.0$, $\beta = 0.5$, $\mu_{12} = 1.0$, $\mu_{22} = 1.0$, $p_{12} = 1/3$, $p_{22} = 1/2$.

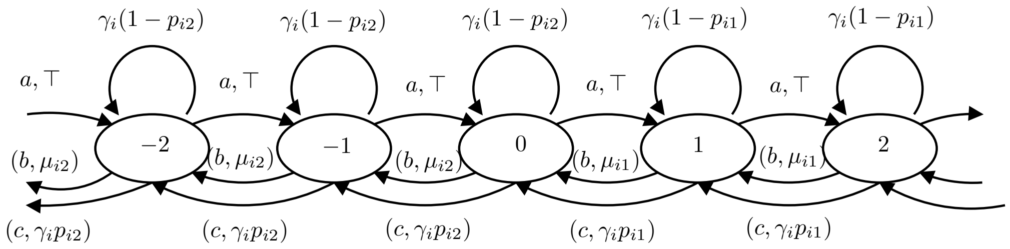

FIGURE 12. Queue with positive and negative population and customer destruction.

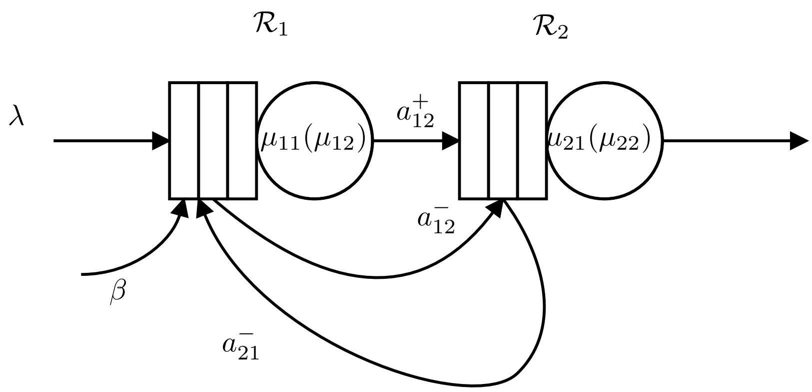

FIGURE 13. Case study: a queueing network with positive or negative population and PITs.

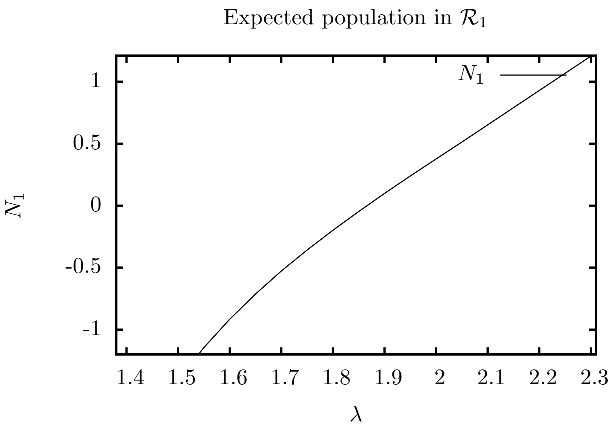

FIGURE 14. Expected queue length N_1 in $\mathcal{R}_1$ as a function of λ in the example of Figure 13.

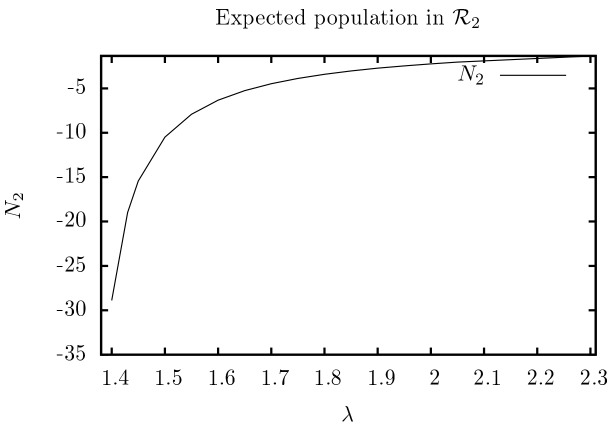

FIGURE 15. Expected queue length N_2 in $\mathcal{R}_2$ as a function of λ in the example of Figure 13.

6. CONCLUSION

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) [7]. 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 [9], 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 [14], 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 [12, 15]. 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 product-forms that can be derived using the more general, extended formulation of RCAT (ERCAT) [29].

REFERENCES

[1] Lavenberg, S. S. (1983) *Computer Performance Modeling Handbook*. Academic Press, New York.

[2] Hillston, J. (1994) A Compositional Approach to Performance Modelling. PhD thesis Department of Computer Science, University of Edinburgh.

[3] Marsan, M. A., Conte, G., and Balbo, G. (1984) A class of generalized stochastic petri nets for the performance evaluation of multiprocessor systems. *ACM Trans. Comput. Syst.*, **2**, 93–122.

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

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

[6] Balsamo, S., Harrison, P. G., and Marin, A. (2012) Methodological Construction of Product-form Stochastic Petri-Nets for Performance Evaluation. *J. of System and Software*, **85**, 1520–1539.

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

[8] Gelenbe, E. and Rosenberg, C. (1990) Queues with slowly varying arrival and service processes. *Management Science*, **36**, 928–937.

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

[10] Gelenbe, E. and Fourneau, J. M. (2002) G-networks with resets. *Perf. Eval.*, **49**, 179–191.

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

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

[13] Fourneau, J. M. and Quessette, F. (2006) Computing the steady-state distribution of G-networks with synchronized partial flushing. *ISCIS, 21th International Symposium*, Istanbul, Turkey, pp. 887–896.

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

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

[16] Fourneau, J. M. (2007) Closed G-networks with Resets: product form solution. *Int. Conf. on the Quantitative Eval. of Systems (QEST)*, pp. 287–296.

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

[18] Boucherie, R. J. and van Dijk, N. M. (2011) *Queueing Networks: A Fundamental Approach* International Series in Operations Research & Management Science. Springer.

[19] Gelenbe, E. and Fourneau, J. M. (2004) Flow equivalence and stochastic equivalence in G-Networks. *Comp. Management Sci.*, **1**, 179–192.

[20] Harrison, P. G. and Marin, A. (2012) Deriving the rate equations characterising product-form models and application to propagating synchronisations. *Proc. of Valuetools 2012*, Cargèse, France, pp. 107–116.

[21] Marin, A. and Vigliotti, M. G. (2010) A general result for deriving product-form solutions of Markovian models. *Proc. of First Joint WOSP/SIPEW Int. Conf. on Perf. Eng.*, San Josè, CA, USA, pp. 165–176. ACM.

[22] Gordon, W. J. and Newell, G. F. (1967) Cyclic queueing networks with exponential servers. *Oper. Res.*, **15**, 254–265.

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

[24] Kelly, F. (1979) *Reversibility and stochastic networks.* Wiley, New York.

[25] Plateau, B. (1985) On the stochastic structure of parallelism and synchronization models for distributed algorithms. *SIGMETRICS Perform. Eval. Rev.*, **13**, 147–154.

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

[27] Buchholz, P. (2010) Product form approximations for communicating Markov processes. *Perf. Eval.*, **67**, 797 – 815. Special Issue: QEST 2008.

[28] Harrison, P. G. and Lee, T. T. (2005) Separable equilibrium state probabilities via time reversal in Markovian process algebra. *Theoretical Computer Science*, **346**, 161–182.

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