# Semi-Product-Form Solution for Models with State-Dependent Rates


Nigel Thomas
School of Computing Science,
Newcastle University, UK.
nigel.thomas@ncl.ac.uk

Andrea Marin
Dipartimento di Informatica,
Universita Ca' Foscari di Venezia, Italy.
marin@dsi.unive.it

Peter Harrison
Department of Computing,
Imperial College London, UK.
pgh@doc.ic.ac.uk



In this paper we consider the problem of finding a decomposed solution to a Markov model where the action rates may depend on the global state space. To do this we consider regular cycles in the underlying state space and show that a semi-product form solution exists when the functions describing the action rates have specific forms. The approach is illustrated with a simple queueing example although it clearly extends to more general cases. The results for semi-product form solutions are not entirely new, however the method by which they are derived is both novel and intuitive.


## 1 Introduction

One approach to tackling the state space explosion problem common to all compositional modelling techniques is through the exploitation of, so called, *product-form solutions*. Essentially, a product-form is a decomposed solution where the steady state distribution of a whole system can be found by multiplying the marginal distributions of its components. The quest for product-form solutions in stochastic networks has been a major research area in performance modelling for over 30 years. Most attention has been given to queueing networks and their variants, but there have also been other significant examples, e.g. [1, 3, 12].

Recent work by Harrison on the *Reversed Compound Agent Theorem* (RCAT) has exploited properties of the reversed process to derive product form solutions to models expressed in stochastic process algebra [6, 7, 8, 9]. This has resulted in a body of work defining the identification of product forms at the syntactic level, based on corresponding active and passive actions in synchronising components. The method outlined in this paper can be used to consider classes of model not amenable to solution by the standard RCAT method. Furthermore, the resultant decomposition is not strictly a product form, as it includes terms relating to the global state space. Nevertheless, we can exploit the same properties of the reversed process to derive expressions based on the cycles which arise in the underlying CTMC and hence an efficient scalable decomposition.

Most product form results in queueing networks rely on regular structure within the underlying state space. Arrival rates at a given node are clearly dependent on the number of jobs at preceding nodes, however service rates are generally constant or depend only on the local state of a component (in a limited fashion). There are a number of existing results where particular transition rates may depend on the state of other components, e.g. [4, 5, 13]. In the work presented here we consider the case where any transition rate may depend on some function of the current global state. This generalises our previous results in this area [7, 11]. Clearly this creates an apparently strong dependency between components and it is perhaps counter-intuitive that a product form solution could exist in such situations. In fact, as we show, the resulting decomposition will include terms relating to the rate functions and hence the


Submitted to:
PASTA 2010

global state. As such this is not strictly a product form solution, and so we use the term *semi-product form*.

Bonald and Proutiere [2] consider a general queueing network model and show that it is insensitive to the service distribution (and hence has a semi-product form solution) subject to a general balance condition. The example we include here is clearly subject to the same condition. However, our approach differs from that taken in [2] in that we use the reversed process and Kolmogorov's generalised criteria to find conditions (equivalent to those of Bonald and Proutiere [2]) on the rate functions. This gives a simpler and more intuitive means of deriving the semi-product form solution.

## 2 Outline of the method

For every stationary Markov process, there is a reversed process with the same state space and the same steady state probability distribution, i.e. $\pi_{\mathbf{x}} = \pi'_{\mathbf{x}}$, where $\pi_{\mathbf{x}}$ and $\pi'_{\mathbf{x}}$ are the steady state probabilities of being in state $\mathbf{x}$ in the forward and reversed process respectively. Furthermore, the forward and reversed processes are related by the transitions between states; there will be a non-zero transition rate between states $\mathbf{x}$ and $\mathbf{x}'$ in the reversed process, $q'_{\mathbf{x},\mathbf{x}'}$ iff there is a non-zero transition rate between states $\mathbf{x}'$ and $\mathbf{x}$ in the forward process, $q_{\mathbf{x}',\mathbf{x}}$. A special case is the *reversible* process, where the reversed process is stochastically identical to the forward process, so that $q'_{\mathbf{x},\mathbf{x}'} = q_{\mathbf{x}',\mathbf{x}}$; an example is the M/M/1 queue. The reversed process is easily found if we already know the steady state probability distribution (see Kelly [14] for example).

The forward and reversed probability fluxes balance at equilibrium, i.e.

$$\pi'_{\mathbf{x}} q'_{\mathbf{x},\mathbf{x}'} = \pi_{\mathbf{x}'} q_{\mathbf{x}',\mathbf{x}} \tag{1}$$

and so, since $\pi_{\mathbf{x}} = \pi'_{\mathbf{x}}$, we find:

$$q'_{\mathbf{x},\mathbf{x}'} = \frac{\pi_{\mathbf{x}'} q_{\mathbf{x}',\mathbf{x}}}{\pi_{\mathbf{x}}}$$

Kolmogorov's criteria utilise these balance equations to relate the forward and reversed transitions directly, as follows

$$\pi_{\mathbf{x}} q_{\mathbf{x},\mathbf{x}'} \pi_{\mathbf{x}'} q_{\mathbf{x}',\mathbf{x}} = \pi'_{\mathbf{x}'} q'_{\mathbf{x}',\mathbf{x}} \pi'_{\mathbf{x}} q'_{\mathbf{x},\mathbf{x}'}$$

Since $\pi_{\mathbf{x}} = \pi'_{\mathbf{x}}$, we find:

$$q_{\mathbf{x},\mathbf{x}'} q_{\mathbf{x}',\mathbf{x}} = q'_{\mathbf{x}',\mathbf{x}} q'_{\mathbf{x},\mathbf{x}'}$$

More generally, we observe that

$$q_{\mathbf{x_1},\mathbf{x_2}} q_{\mathbf{x_2},\mathbf{x_3}} \dots q_{\mathbf{x_n},\mathbf{x_1}} = q'_{\mathbf{x_1},\mathbf{x_n}} \dots q'_{\mathbf{x_3},\mathbf{x_2}} q'_{\mathbf{x_2},\mathbf{x_1}}$$

That is, for any cycle length $n$, the product of the transition rates in the forward process must equal the product of the transition rates in the reversed process of the same cycle taken in reverse order. This is referred to as Kolmogorov's general criteria.

In order to find a possible (semi) product form solution, we first identify a notional reversed model of the system. That is, an equivalent model where time is reversed. For any given formalism there are a number of rules that can be followed to *structurally reverse* any given model (beyond the scope of this short paper). That is, to reverse each component of a model in to derive a *notional* reversed model. If this notional reversed model is the actual reversed CTMC, then Kolmogorov's criteria will be satisfied and a (semi) product form solution will exist. Hence we can use Kolmogorov's criteria as a test for our notional reversed model, to see if it is the actual reversed CTMC.

## 3 Example: 3 node Towsley model with state dependent routing

In order to illustrate the approach we consider a very simple example based on the classic adaptive routing model proposed by Towsley [15].

There is one main node and two secondary nodes ($k = 1,2$) in a closed queueing network. Following service at the main node a job will proceed to either secondary node according to some (state dependent) probability. Following service at a secondary node a job will return to the main node with certainty. In Towsley's model the rates are fixed and only the routing probabilities are state dependent. Here we make all rates and probabilities state dependent. The service rates at the secondary nodes are $\mu_k(i_1,i_2)$, $k = 1,2$; $\lambda_k(i_1,i_2)$ denotes the service rate at the main node of jobs proceeding to node $k$, $k = 1,2$. Where $i_1$ and $i_2$ are the number of jobs at secondary nodes 1 and 2 respectively; obviously there will be $n-i_1-i_2$ jobs at the main node, where $n$ is the total population.

The Kolmogorov criteria for the *minimal cycles* are as follows:

$$\lambda_1(i_1,i_2)\mu_1(i_1+1,i_2) = \overline{\mu}_1(i_1,i_2)\overline{\lambda}_1(i_1+1,i_2) \tag{2}$$

$$\lambda_2(i_1,i_2)\mu_2(i_1,i_2+1) = \overline{\mu}_2(i_1,i_2)\overline{\lambda}_2(i_1,i_2+1) \tag{3}$$

$$\begin{aligned}&\lambda_1(i_1,i_2)\lambda_2(i_1+1,i_2)\mu_1(i_1+1,i_2+1)\mu_2(i_1,i_2+1) = \\ &\overline{\mu}_2(i_1,i_2)\overline{\mu}_1(i_1,i_2+1)\overline{\lambda}_2(i_1+1,i_2+1)\overline{\lambda}_1(i_1+1,i_2)\end{aligned} \tag{4}$$

$$\begin{aligned}&\lambda_2(i_1,i_2)\lambda_1(i_1,i_2+1)\mu_2(i_1+1,i_2+1)\mu_1(i_1+1,i_2) = \\ &\overline{\mu}_1(i_1,i_2)\overline{\mu}_2(i_1+1,i_2)\overline{\lambda}_1(i_1+1,i_2+1)\overline{\lambda}_2(i_1,i_2+1)\end{aligned} \tag{5}$$

Hence, it follows that,

$$\begin{aligned}\overline{\mu}_1(i_1,i_2) &= x_1\lambda_1(i_1,i_2)\\ \overline{\mu}_2(i_1,i_2) &= x_2\lambda_2(i_1,i_2)\\ \overline{\lambda}_1(i_1,i_2) &= \mu_1(i_1,i_2)/x_1\\ \overline{\lambda}_2(i_1,i_2) &= \mu_2(i_1,i_2)/x_2\end{aligned}$$

where $x_1$ and $x_2$ are constants.

Thus we can find the necessary condition for satisfing the Kolmogorov criteria,

$$\begin{aligned}&\frac{\lambda_1(i_1,i_2)}{\lambda_1(i_1,i_2+1)}\frac{\mu_1(i_1+1,i_2+1)}{\mu_1(i_1+1,i_2)} = \\ &\frac{\lambda_2(i_1,i_2)}{\lambda_2(i_1+1,i_2)}\frac{\mu_2(i_1+1,i_2+1)}{\mu_2(i_1,i_2+1)}\end{aligned} \tag{6}$$

Since the Kolmogorov criteria are satisfied, a product form solution exists when (6) is satisfied. This solution will have the form,

$$\pi_{ij} = \pi_{00}\prod_{m=0}^{i-1}\frac{\lambda_1(m,0)}{\mu_1(m+1,0)}\prod_{n=0}^{j-1}\frac{\lambda_2(i,n)}{\mu_2(i,n+1)} \quad ,i+j \leq N$$

where $\pi_{00}$ is the steady state probability that all the jobs are located at the main node.[1]

[1] Calculating this probability, *the normalising constant*, is potentially costly in many product form networks, however in this case the normalising constant may be computed very effiently using mean value analysis.

### 3.1 Fixed rates

In Towsley's paper [15] the rates are fixed, but the routing probabilities are state dependent: $\mu_k(i_1,i_2) = \mu_k$, $\lambda_k(i_1,i_2) = p_k(i_1,i_2)\lambda$, where $p_2(i_1,i_2) = 1 - p_1(i_1,i_2)$. Hence,

$$\begin{aligned}(1-p_1(i_1,i_2))p_1(i_1,i_2+1) &= \\ (1-p_1(i_1+1,i_2)p_1(i_1,i_2) & \end{aligned} \tag{7}$$

This is clearly satisfied if

$$p_1(i_1,i_2) = \frac{i_1+a}{b(i_1+i_2)+c}$$

Recall that $p_2(i_1,i_2) = 1 - p_1(i_1,i_2)$, hence

$$p_k(i_1,i_2) = \frac{i_k+a_k}{i_1+i_2+c} \tag{8}$$

Where $c = a_1 + a_2$. (8) is equivalent to Towsley's result, although there are clearly also other $p_k$'s satisfying (7), i.e. Towsley's result was not complete. This result trivially extends to more secondary nodes.

### 3.2 State dependent rates

Now recall (6).

$$\begin{aligned}\frac{\lambda_1(i_1,i_2)}{\lambda_1(i_1,i_2+1)}\frac{\mu_1(i_1+1,i_2+1)}{\mu_1(i_1+1,i_2)} &= \\ \frac{\lambda_2(i_1,i_2)}{\lambda_2(i_1+1,i_2)}\frac{\mu_2(i_1+1,i_2+1)}{\mu_2(i_1,i_2+1)} &\end{aligned}$$

This is a generalisation of Towsley's result where the service rates are state dependent. (6) is clearly satisfied if

- Rates are either dependent on the total jobs at all secondary nodes, or only on the jobs at the target secondary node:
  $\lambda_k(i_1,i_2) = \lambda_k f_\lambda(\sum_{\forall j} i_j)$ or $\lambda_k(i_1,i_2) = \lambda_k(i_k)$
  and
  $\mu_k(i_1,i_2) = \mu_k f_k(\sum_{\forall j} i_j)$ or $\mu_k(i_1,i_2) = \mu_k(i_k)$
- Functional dependence is the same at all secondary nodes:

  $$\lambda_k(i_1,i_2) = \lambda_k f(i_1,i_2)$$

  and

  $$\mu_k(i_1,i_2) = \mu_k f(i_1,i_2).$$

- Arrivals and departure rates at each secondary node are dependent on the total jobs at all other secondary nodes:

  $$\lambda_k(i_1,i_2) = \lambda_k f_k(\sum_{j\neq k} i_j)$$

  and

  $$\mu_k(i_1,i_2) = \mu_k f_k(\sum_{j\neq k} i_j).$$

Where $f(.)$, $f_k(.)$ and $f_\lambda(.)$ are arbitrary functions.

## 4 Minimal cycles and multiple dimensions

In the derivation above we have employed the notion of *minimal cycles* without explanation. The reader will have observed that the longest minimal cycles used are simple squares, consisting of services at the main node of one type, then the other type, followed by services at the secondary nodes of the first type and then the second type. Why, one might ask, do we not consider mulitple services of each type? The reason is simply that we do not need to, as such cycles would produce multiple redundant equations, even in the case of state dependent rates. The proof of this condition is not complex for the example included here. A more precise definition and an outline of the general proof is given in [10]. One of the consequences of the notion of minimal cycles applied to this model is that the approach described applies to an arbirtary number of queues, since each minimal cycle will involve only the main node and at most one secondary node.

## 5 Additional Remarks

The illustrative example presented here is very simple; however the method clearly applies to more general cases, subject to the following provisos.

1. It should be possible to derive a notional reversed model to be tested using Kolmogorov's criteria. As well as the queueing example here, some examples of how to achieve this in stochastic process algebra were given in [11]. Other examples involving stochastic Petri nets have been undertaken but are as yet unpublished.
2. The state space should be regular, allowing a minimal set of regular cycles to be identified.
3. The number of dimensions (governed by the number of components) should be manageable. In the example here it is a simple matter to add more nodes, resulting in more, but not longer, minimal cycles.

## References


[1] G. Balbo, S. Bruell, and M. Sereno, Embedded processes in generalized stochastic Petri nets, in: *Proc. 9th Intl. Workshop on Petri Nets and Performance Models*, pp. 71-80, 2001.

[2] T. Bonald, A. Proutiere, Insensitivity in processor-sharing networks, *Performance Evaluation*, **49**, pp. 193-209, 2002.

[3] R.J. Boucherie. A Characterisation of Independence for Competing Markov Chains with Applications to Stochastic Petri Nets, *IEEE Trans. on Software Eng.*, **20**(7), pp. 536544, 1994.

[4] R.J. Boucherie and N.M. van Dijk, Product forms for queueing networks with state-dependent multiple job transitions, *Advances in Applied Probability*, **23**, pp. 152-187, 1991.

[5] X. Chao, M. Miyazawa and M. Pinedo, *Queueing Networks: Customers, Signals and Product Form Solutions*, Wiley, 1999.

[6] P.G. Harrison, Turning back time in Markovian process algebra, *Theoretical Computer Science*, January, 2003.

[7] P.G. Harrison, Reversed processes, product-forms and a non-product-form, *Linear Algebra and Its Applications*, July, 2004.

[8] P.G. Harrison, Compositional reversed Markov processes, with applications to G-networks, *Performance Evaluation*, 2004.

[9] P.G. Harrison, Product-forms and functional rates, *Performance Evaluation*, **66**, pp. 660-663, 2009.

[10] P.G. Harrison, Process algebraic non-product-forms, *Electronic Notes in Theoretical Computer Science*, **151**(3), 2006.

[11] P.G. Harrison and N. Thomas, Product form solution in PEPA via the reversed process, in: *Next Generation Internet: Performance Evaluation and Applications*, LNCS 5233, Springer-Verlag, 2010.

[12] W. Henderson and P.G. Taylor, Embedded Processes in Stochastic Petri Nets, *IEEE Trans. on Software Eng.*, **17**(2), pp. 108-116, 1991.

[13] W. Henderson and P.G. Taylor, State-dependent Coupling of Quasireversible Nodes, *Queueing Systems: Theory and Applications*, **37**(1/3), pp. 163-197, 2001.

[14] F.P. Kelly, *Reversibility and stochastic networks*, Wiley, 1979.

[15] D. Towsley, Queuing Network Models with State-Dependent Routing, *Journal of the ACM*, **27**(2), pp. 323-337, 1980.