

Sum of product forms solutions to MSCCC queues with job type dependent processing times

Citation for published version (APA):

Adan, I. J. B. F., Visschers, J. W. C. H., & Wessels, J. (1998). *Sum of product forms solutions to MSCCC queues with job type dependent processing times*. (Memorandum COSOR; Vol. 9819). Technische Universiteit Eindhoven.

Document status and date:

Published: 01/01/1998

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Sum of product forms solutions to MSCCC queues with job type dependent processing times

Ivo Adan, Jeremy Vissschers and Jaap Wessels
Eindhoven University of Technology, Eindhoven, the Netherlands

September 4, 1998

Abstract

Queueing models with simultaneous resource possession can be used to model production systems at which the production process occupies two or more resources (machines, operators, product carriers etc.) at the same time. A special class of these queueing models is the class of MSCCC queues, for which the stationary distribution has a product form. This was shown by Berezner et al. whose result depends on one special characteristic of MSCCC queues, being the processing times are *job type independent* exponentially distributed. However in many production situations processing times are *job type dependent*. Therefore we examined MSCCC queues with *job type dependent* exponentially distributed processing times. We determined the equilibrium probabilities of two special models using a *detailed* state description, for which a solution using an aggregated state description is known. Comparing these two solutions we gained more insight in the structure of the solution to more general models for which such an aggregated state description no longer has the Markov property.

1 Introduction

Queueing models with simultaneous resource possession can be used to model production systems at which different resources are needed at the same time to process a product or customer. In many production systems these resources could be machines, operators, product carriers and/or tools used to make a product while in a large computer network it could be buses, memory modules and processors that are needed to satisfy a request from a user.

When there are two resources, a set of job type dependent machines and a common pool of operators, one of two assignment rules is commonly used. The first rule, First Assigned First Served (FAFS), assigns an arriving job to a machine of its type as soon as the machine becomes available. The operators then process the jobs in order of assignment. By contrast, the second rule, First Come First Served (FCFS), leaves the jobs in the queue. The operators then take the first job in the queue for which there is a machine available. When using this last rule the model becomes an MSCCC queue if it is assumed that the processing times are job type independent exponentially distributed and that jobs arrive according to a Poisson process.

This class of MSCCC queues is contained in the class of Order Independent queues, for which Berezner et al.[1] proved that the equilibrium probabilities have a product form (section 2). For the MSCCC queues this result depends on one main characteristic, the processing times should be *job type independent* exponentially distributed, which is not always valid in practice. For some special models with job type dependent processing times, solutions are known; for example the $M|H_m|1$ queue (m types of jobs and only one operator), for which the stationary distribution is a sum of m product forms (section 3.1), and the MSCCC queue with job type

dependent processing times and as many operators as machines (section 3.2). The solution to these two models however is only known using an aggregated state description. To analyse these models in general the use of such a state description is not always possible, because the Markov property is lost in the aggregation. The use of a detailed state description, on the other hand, yields a large state space, which is hard to analyse without some insight in the structure of the solution.

In our research we are trying to gain some insight into the structure of the solution to the equilibrium equations of MSCCC queues with job type *dependent* exponentially distributed processing times. We are no longer looking for a product form, but for a linear combination of one or more product forms. We analysed the two special, for which in an aggregated state description the solution is known, using a detailed state description. The structure of the solution to these special models can help us find the structure of the solution in general.

2 MSCCC queues

The class of MSCCC queues is a special class of queueing systems with simultaneous resource possession with two different resources. At an MSCCC queue jobs of type $c \in \mathcal{C}$ (with \mathcal{C} the set of job types) arrive according to a Poisson process with intensity λ_c and are processed at the machines of type c in a FCFS order. There are B_c machines of type c , which are operated by a common pool of operators of size K . The processing times are exponentially distributed with mean $1/\mu$ and are job type independent. An example of an MSCCC queue is shown in figure 1.

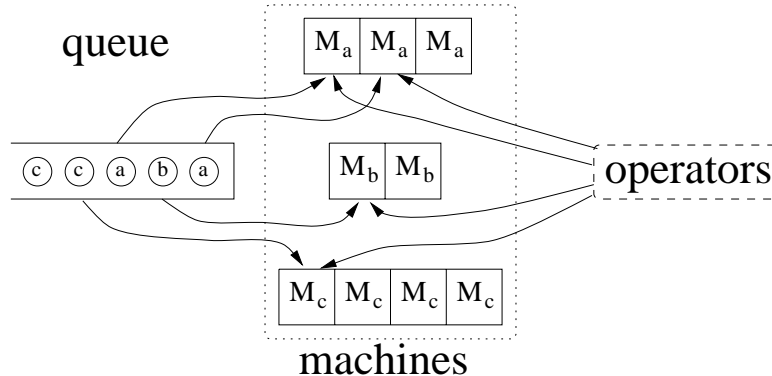


Figure 1: An example of an MSCCC queue

Berezner et al[1] showed that the solution to the equilibrium probabilities for MSCCC queues have a product form. Their result holds for a larger class of queueing systems, the Order Independent queues (O.I. queues). In their analysis they denote states as a sequence of jobs $c_n \cdots c_1$, with $c_i \in \mathcal{C}$. To position i in the queue a relative departure rate $s_i(c_i \cdots c_1)$ is assigned that only depends on the queue $c_i \cdots c_1$. The total relative departure rate of this queue is defined as

$$k(c_i \cdots c_1) = \sum_{j=1}^i s_j(c_j \cdots c_1)$$

This function should be independent of the order of the jobs in $c_n \cdots c_1$, hence the name Order Independent queues. They also defined the function $\mu(n)$, as the total absolute departure rate if the queue length equals n and derived the solution to the equilibrium probabilities $\pi(c_n \cdots c_1)$:

$$\pi(c_n \cdots c_1) = \prod_{i=1}^n \frac{\lambda_{c_i}}{\mu(n)k(c_i \cdots c_1)} \pi(0)$$

For MSCCC queues the function $k(c_i \cdots c_1)$ denotes the number of operators that is processing a job times the intensity μ . If we define $n_c(c_i \cdots c_1)$ as the number of type c customers in $c_i \cdots c_1$, then for $k(c_i \cdots c_1)$ the following holds:

$$k(c_i \cdots c_1) = \mu \cdot \min \left\{ K, \sum_{c \in \mathcal{C}} \min \{B_c, n_c(c_i \cdots c_1)\} \right\} \quad (1)$$

With $\mu(n) = 1$ this yields the product form solution to the MSCCC queue. Note that the number of type $c (\in \mathcal{C})$ jobs that is being processed at a certain time strongly depends on the order of the jobs in the queue, while the total number of jobs being processed doesn't. This implies that the function k becomes dependent of the order of the jobs in the queue if the processing times become job type dependent and therefore this result does not hold for job type dependent processing times in general.

3 Job type dependent processing times

The result of Berezner et al. holds assuming job type *independent* processing times. In some production systems however the processing times are job type dependent, not every process takes the same amount of time. MSCCC queues cannot be used to model these production systems; therefore our research focuses on MSCCC queues with job type dependent exponentially distributed processing times.

As was noted before the order in which the jobs are processed strongly depends on the order of arrival. Because of this complicated processing order a detailed state description is needed in the analysis of these queueing systems. A possible state description would be the queue of jobs in order of arrival as we defined in the previous section: $c_n \dots c_1$ with $c_i \in \mathcal{C}$ being the i^{th} job in the queue. However using this state description, the state space becomes very large and analysing it becomes difficult. By using an aggregated state description this problem can be avoided, but only with some special models it is possible to aggregate the state space without losing the Markov property.

To get more insight in the solution of these queueing systems in general we studied two of these special MSCCC queues. For these two models we determined the solution using a detailed state description and an aggregated state description. The first model is a model with only one operator, the second with as many operators as machine. For these two models a solution is already known using an aggregated state description, which gives some information about the structure of the solution in a detailed state description.

3.1 Model I: only one operator

First an MSCCC queue will be discussed with m job types, each with one machine available, and only one operator. Clearly this system is very similar to the $M|H_m|1$ queue. The only difference is that in this model the job type becomes known at the arrival of the job; at the $M|H_n|1$ queue the job type becomes known when the job is taken into service.

3.1.1 Analysis of the $M|H_m|1$ queue

At an $M|H_m|1$ queue jobs arrive according to a Poisson process with intensity λ . When a job is finished, the operator takes a new job from the queue, which is with probability $p_c = \frac{\lambda_c}{\lambda}$ ($c \in \mathcal{C}$ and $|\mathcal{C}| = m$) of type c . Of course the following must hold:

$$\sum_{c \in \mathcal{C}} \lambda_c = \lambda$$

The following state description is used: (n, c) with n the number of jobs in the queue and $c \in \mathcal{C}$ the type of the job currently in service. The empty state is denoted by 0. The states and the state transitions can be visualised as is done in figure 2 for the $M|H_2|1$ queue.

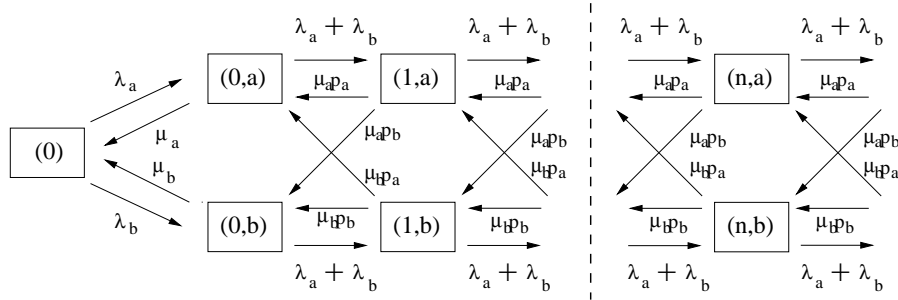


Figure 2: State transitions of the $M|H_2|1$ queue

The procedure used for solving the equilibrium equation and the normalisation equation looks like this:

- Divide the equilibrium equations into two different sets: the interior equations and the boundary equations.
- Choose a product form for the equilibrium probabilities.
- Substitute the chosen product form into the interior equations.
- Solve the interior equations. This yields one or more solutions.
- Substitute a linear combination of these solutions into the boundary and the normalisation equations.
- Solve the boundary and the normalisation equations.

For this queueing systems the equilibrium and the normalisation equations are as follows:

Normalisation equation

$$1 = \sum_{n=0}^{\infty} \sum_{c \in \mathcal{C}} \pi(n, c)$$

Boundary equations

$$\lambda \pi(0) = \sum_{c \in \mathcal{C}} \mu_c \pi(0, c) \quad (2)$$

$$(\lambda + \mu_c) \pi(0, c) = \lambda p_c \pi(0) + \sum_{d \in \mathcal{C}} \mu_d p_c \pi(1, d) \quad (\forall c \in \mathcal{C}) \quad (3)$$

Interior equations

$$(\lambda + \mu_c)\pi(n, c) = \lambda\pi(n-1, c) + p_c \sum_{d \in \mathcal{C}} \mu_d \pi(n+1, d) \quad (\forall c \in \mathcal{C}) \quad (4)$$

The product form $\pi(n, c) = x^n k_c$ is substituted into the interior equations. This yields these equations:

$$(\lambda + \mu_c)xk_c = \lambda k_c + p_c \sum_{d \in \mathcal{C}} \mu_d x^2 k_d \quad (\forall c \in \mathcal{C}) \quad (5)$$

Clearly $x = \frac{\lambda}{\lambda + \mu_c}$ ($c \in \mathcal{C}$) cannot be part of a solution satisfying all the equations. A job type $c \in \mathcal{C}$ is chosen and with this job type these equations can be reduced to the following set of equations (using $x \neq \frac{\lambda}{\lambda + \mu_c}$):

$$k_d = \frac{p_d (\lambda + \mu_c)x - \lambda}{p_c (\lambda + \mu_d)x - \lambda} k_c \quad (\forall d \in \mathcal{C}) \quad (6)$$

$$(\lambda + \mu_c)xk_c = \lambda k_c + p_c \sum_{d \in \mathcal{C}} \mu_d x^2 \left(\frac{p_d (\lambda + \mu_c)x - \lambda}{p_c (\lambda + \mu_d)x - \lambda} k_c \right) \quad (7)$$

The last equation can be rewritten, yielding

$$\prod_{d \in \mathcal{C}} ((\lambda + \mu_d)x - \lambda) = \sum_{d \in \mathcal{C}} (\mu_d p_d x^2) \prod_{e \in \mathcal{C} \setminus \{d\}} ((\lambda + \mu_e)x - \lambda) \quad (8)$$

This is an equation in x of order $m+1$, for which $x=1$ is a solution. There are m other solutions to this equation. For these solution the stability condition $|x| < 1$ holds. This can be proved using Rouché's theorem.

We define

$$f(x) := \sum_{d \in \mathcal{C}} (\mu_d p_d x^2) \prod_{e \in \mathcal{C} \setminus \{d\}} ((\lambda + \mu_e)x - \lambda)$$

and

$$g(x) := - \prod_{d \in \mathcal{C}} ((\lambda + \mu_d)x - \lambda)$$

There are m solutions to the equation $g(x) = 0$ for which $|x| < 1$ holds. These solutions are

$$x = \frac{\lambda}{\lambda + \mu_c} \quad \forall c \in \mathcal{C}$$

There exists an $\epsilon > 0$ for which

$$\frac{\lambda}{\lambda + \mu_c} < 1 - \epsilon \quad \forall c \in \mathcal{C}$$

We need to prove that there exists an ϵ for which also $|f(1-\epsilon)| < |g(1-\epsilon)|$ holds. For $f(x)$ and $g(x)$ the following holds

$$\begin{aligned} \frac{|f(x)|}{|g(x)|} &= \sum_{c \in \mathcal{C}} \left| \frac{\mu_c p_c x^2}{(\lambda + \mu_c)x - \lambda} \right| \\ &\leq \sum_{c \in \mathcal{C}} \frac{\mu_c p_c |x|^2}{(\lambda + \mu_c)|x| - \lambda} \end{aligned}$$

For $x \geq 0$ we define

$$h(x) = \sum_{c \in \mathcal{C}} \frac{\mu_c p_c x^2}{(\lambda + \mu_c)x - \lambda}$$

Note that $h(1) = 1$. If $h'(1) > 0$ then for an ϵ that is small enough $h(1 - \epsilon) < 1$ holds. For $h'(x)$ the following holds:

$$\begin{aligned} h'(x) &= \sum_{c \in \mathcal{C}} \frac{\mu_c p_c x^2 (\lambda + \mu_c) - \mu_c p_c x 2\lambda}{((\lambda + \mu_c)x - \lambda)^2} \\ h'(1) &= \sum_{c \in \mathcal{C}} \frac{p_c (\mu_c - \lambda)}{\mu_c} \\ &= 1 - \sum_{c \in \mathcal{C}} \frac{\lambda_c}{\mu_c} \end{aligned}$$

Because of the stability condition of this queueing model being

$$\sum_{c \in \mathcal{C}} \frac{\lambda_c}{\mu_c} < 1$$

it is clear that $h'(1) > 0$ and therefore there is an ϵ such that $h(1 - \epsilon) < 1$. Thus there is an ϵ such that

$$\frac{|f(1 - \epsilon)|}{|g(1 - \epsilon)|} \leq h(1 - \epsilon) < 1 - \epsilon$$

Rouché's theorem now states that the equation $f(x) + g(x) = 0$ has as many solutions x , for which $|x| < 1$ holds, as the function $g(x)$ has. The function $g(x)$ has m solutions with $|x| < 1$. Thus there are also m solutions to the equation $f(x) = -g(x)$ with $|x| < 1$.

With a linear combination of these m solutions (assuming these solutions are all different) the m boundary equations and the normalisation equation can be satisfied.

3.1.2 Analysis using the detailed state description

Now a detailed state description is used: states denoted by a sequence of jobs in order of arrival $c_n \dots c_1$ with $c_i \in \mathcal{C}$. Possible state transitions are the arrival of a new job and finishing the service c_1 . This yields these equations (with $P^{\mathcal{C}}$ the set of all possible sequences):

Normalisation equation

$$1 = \sum_{\mathbf{c} \in P^{\mathcal{C}}} \pi(\mathbf{c})$$

Boundary equations

$$\lambda \pi(0) = \sum_{c \in \mathcal{C}} \mu_c \pi(c) \tag{9}$$

$$(\lambda + \mu_c) \pi(c) = \lambda_c \pi(0) + \sum_{d \in \mathcal{C}} \mu_d \pi(cd) \quad (\forall c \in \mathcal{C}) \tag{10}$$

Interior equations ($n \geq 2$)

$$(\lambda + \mu_{c_1}) \pi(c_n \dots c_1) = \lambda_{c_n} \pi(c_{n-1} \dots c_1) + \sum_{c_0 \in \mathcal{C}} \mu_{c_0} \pi(c_{n-1} \dots c_1 c_0) \tag{11}$$

Because of the structure of the solution to the MSCCC queue with type independent processing types that was presented by Berezner et al.[1], the product form $\pi(c_n \dots c_1) = \prod_{i=1}^n x_{c_i} k_{c_i}$ is chosen and substitute it into the interior equations. This yields the following equations:

$$(\lambda + \mu_c)x_{c_n}k_c = \lambda_{c_n}k_c + \sum_{d \in \mathcal{C}} \mu_d x_{c_n} x_c k_d \quad (\forall c, c_n \in \mathcal{C}) \quad (12)$$

If these equations are divided by x_{c_n} ($x_{c_n} \neq 0$) it becomes clear that for every $c, d \in \mathcal{C}$ holds:

$$\frac{\lambda_c}{x_c} = \frac{\lambda_d}{x_d} \quad (13)$$

If we substitute $x_c = p_c x$ with $p_c = \frac{\lambda_c}{\lambda}$ and thus satisfying equation (13) the equations (12) become:

$$(\lambda + \mu_c)xk_c = \lambda k_c + \sum_{d \in \mathcal{C}} \mu_d p_c x^2 k_d \quad (\forall c \in \mathcal{C}) \quad ((12)')$$

These are the same equations as the equations (5) and thus there are also m solutions to these equations.

With these m solutions it is clear that the boundary equations (2) and (3) are the same as the boundary equations (9) and (10) and that the solution to the equilibrium equations of this queue is a linear combination of m product forms, $\prod_{i=1}^m x_{c_i} k_{c_i}$, with $x_c = p_c x$ and x a solution to equation (8).

3.2 Model II: as many operators as machines

In the second model there are as many operators as machines. For every machine there is an operator available; thus for the analysis it is no longer necessary to consider the assignment of the operators.

3.2.1 Analysis using an aggregated state space

Because the number of operators available is no longer a restriction, the number of jobs of type c in progress no longer depends on the jobs of other types in the queue; only the number of machines determines the number of jobs in progress. This system is similar to a system consisting a number of parallel $M|M|s$ queues, one for ever job type (for type c holds $s = B_c$). For further analysis we can use the following aggregated state space: (n_a, n_b, \dots) , with $\mathcal{C} = \{a, b, \dots\}$, and n_c the number of type c costumers in the queue ($c \in \mathcal{C}$).

The solution to the equilibrium equations of a $M|M|s$ queue is a product form. If states of this queue are denoted by n being the number of costumers in the system and customers arrive with intensity λ and the mean processing time equals $1/\mu$, the solution looks as follows:

$$\pi(n) = \prod_{i=1}^n \frac{\lambda}{\min\{i, s\} \cdot \mu} \pi(0)$$

With this solution to the $M|M|s$ queue the solution to the MSCCC queue using the aggregated state space is also clear:

$$\pi(n_a, n_b, \dots) = \pi^a(n_a) \cdot \pi^b(n_b) \dots$$

$\pi^c(n_c)$ is the solution to the $M|M|s$ queue as described above with $s = B_c$, $\lambda = \lambda_c$ and $\mu = \mu_c$ for all $c \in \mathcal{C}$.

3.2.2 Analysis using the detailed state description

If we use the detailed state description the solution of the equilibrium equations can be found using the result of Berezner et al.[1], because this queue is an Order Independent queue; only if there are as many operators as machines ($K = \sum B_c$) an explicit expression of the function k can be found that is independent of the order of the jobs in the queue:

$$k(c_i \cdots c_1) = \sum_{c \in \mathcal{C}} \mu_c \cdot \min\{n_c(c_i \dots c_1), B_c\}$$

3.2.3 Aggregation

The relation between the solution using the aggregated state description and the detailed state description is examined using the following chance experiment.

Consider a bag of n marbles of different colours. There are n_c marbles of colour $c \in \mathcal{C}$ in the bag. Of every colour c we put B_c marbles in the urn if there are that many in the bag; otherwise we put them all in the bag. From the urn we will randomly pick a marble. Let us assume that the probability of picking a marble of colour c from the urn if there are k_c marbles of colour c in the urn ($c \in \mathcal{C}$) equals

$$\frac{k_c \mu_c}{k_a \mu_a + k_b \mu_b + \cdots}$$

If we pick a marble of colour c from the urn take a marble of the same colour from the bag (if there is one left) and put it in the urn. We keep picking marbles from the urn until there are no marbles left in both the bag and the urn.

Let $c_n \cdots c_1$ be a sequence of colours denoting a realisation of this experiment. With $k_c(c_i \dots c_1)$ we denote the number of colour c marbles that are in the urn after the first i picks ($c_i \dots c_1$). The probability that the marbles are picked in the order $c_n \dots c_1$ equals:

$$\prod_{i=1}^n \frac{k_{c_i}(c_i \dots c_1) \mu_{c_i}}{\sum_{c \in \mathcal{C}} k_c(c_i \dots c_1) \mu_c} = \frac{\prod_{c \in \mathcal{C}} \prod_{j=1}^{n_c} \min\{B_c, j\} \mu_c}{\prod_{i=1}^n \sum_{c \in \mathcal{C}} k_c(c_i \dots c_1) \mu_c}$$

It is clear that the sum of the probabilities over all possible realisations should equal 1. We define \mathcal{X} as the set of all possible realisations of this experiment. Using this we obtain the following:

$$\sum_{(c_n \dots c_1) \in \mathcal{X}} \frac{\prod_{c \in \mathcal{C}} \prod_{j=1}^{n_c} \min\{B_c, j\} \mu_c}{\prod_{i=1}^n \sum_{c \in \mathcal{C}} k_c(c_i \dots c_1) \mu_c} = 1$$

If we then multiply the last equation with $\prod_{c \in \mathcal{C}} \lambda_c^{n_c} \pi(0)$ it can be shown that

$$\prod_{c \in \mathcal{C}} \frac{\lambda_c^{n_c}}{\prod_{j=1}^{n_c} \min\{B_c, j\} \mu_c} \pi(0) = \sum_{(c_n \dots c_1) \in \mathcal{X}} \frac{\prod_{c \in \mathcal{C}} \lambda_c^{n_c}}{\sum_{c \in \mathcal{C}} k_c(c_i \dots c_1) \mu_c} \pi(0)$$

which gives the relations between the probability of being in the state n_a, n_b, \dots in the aggregated state description and the sum of the probabilities of being in the states c_n, \dots, c_1 in the detailed state description with n_a jobs of type a , n_b of type b , etc. ($n = n_a + n_b + \dots$).

4 Suggestions for further research

This paper shows that for two special models the solution to the stationary probabilities is a sum of one or more product forms. These models form a subclass of the class of MSCCC queues with job type dependent processing times. Several important characteristics of these MSCCC queues in general can also be found in these special models, like for example the existence of several job types, the assignment of operators to machines of different job types and the possibility that jobs can be overtaken by jobs of other types because there is no machine available of their own type.

These two models, however, lack one special characteristic; in general the order in which the jobs are processed strongly depends on the order of the jobs of the different types in the queue. In the model examined in section 3.2 the number of jobs in service of job type $c \in \mathcal{C}$ only depends on the number of jobs of type c in the system and the number of machines of type c , while the jobs of other types are not important. When there are fewer operators than machines, the jobs of other types become important, because these jobs can occupy an operator that could have processed a job of type c . This job of type c has to wait until an operator becomes free.

This characteristic makes it necessary to use a detailed state description when analysing these MSCCC queues in general. The results in this paper give some insight in the structure of such a solution. The structure of the solution chosen in the first model and the structure in the second model suggest that the solution to these models in general may have the same kind of structure. In general however the solution is not a product form but it may be a linear combination of product forms each with this structure. With these results and the procedure of analysing presented in this paper further analysis is possible.

References

- [1] S.A. Berezner, C.F. Kriel, and A.E. Krzesinski. Quasi-reversible multiclass queues with order independent departure rates. *Queueing Systems*, 19:345–359, 1995.