van Dijk, Nico M.; van der Sluis, Erik

Working Paper

Simple Product-Form Bounds for Queueing Networks with Finite Clusters

Tinbergen Institute Discussion Paper, No. 01-107/4

Provided in Cooperation with:
Tinbergen Institute, Amsterdam and Rotterdam

Suggested Citation: van Dijk, Nico M.; van der Sluis, Erik (2001) : Simple Product-Form Bounds for Queueing Networks with Finite Clusters, Tinbergen Institute Discussion Paper, No. 01-107/4, Tinbergen Institute, Amsterdam and Rotterdam

This Version is available at:
https://hdl.handle.net/10419/86039

TI 2001-107/4
Tinbergen Institute Discussion Paper

Simple Product-Form Bounds for Queueing Networks with Finite Clusters

Nico M. van Dijk
Erik van der Sluis

Department of Quantitative Economics/Operational Research, Faculty of Economics and Econometrics, University of Amsterdam, and Tinbergen Institute

Tinbergen Institute
The Tinbergen Institute is the institute for economic research of the Erasmus Universiteit Rotterdam, Universiteit van Amsterdam and Vrije Universiteit Amsterdam.

Tinbergen Institute Amsterdam
Keizersgracht 482
1017 EG Amsterdam
The Netherlands
Tel.: +31.(0)20.5513500
Fax: +31.(0)20.5513555

Tinbergen Institute Rotterdam
Burg. Oudlaan 50
3062 PA Rotterdam
The Netherlands
Tel.: +31.(0)10.4088900
Fax: +31.(0)10.4089031

SIMPLE PRODUCT-FORM BOUNDS FOR QUEUEING NETWORKS WITH FINITE CLUSTERS

Nico M. van Dijk & Erik van der Sluis

Faculty of Economics and Econometrics
University of Amsterdam
The Netherlands

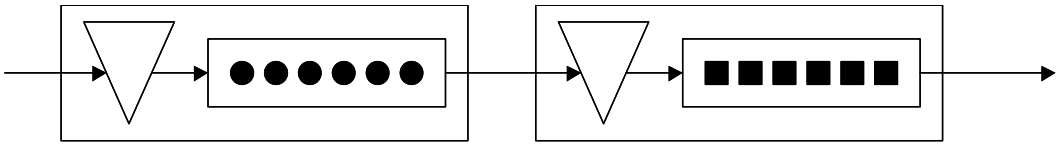
Abstract

Queueing networks are studied with finite capacity constraints for clusters of stations. First, by an instructive tandem cluster example it is shown how a product-form modification method for networks with *finite stations* can be extended to networks with *finite clusters*. Next, a general result is established by which networks with finite clusters can be studied at cluster level by merely keeping track of the total number of jobs at these clusters, that is, by regarding these clusters as aggregate stations. This result is of practical interest to conclude simple performance bounds at global network level. A number of illustrative examples with numerical support are provided.

1 INTRODUCTION

Background

Finite capacity limitations, such as on storage buffers or number of machines, are most natural in practical manufacturing or assembly line systems. These finite constraints might not only be imposed upon a single station but also on a group (cluster) of stations simultaneously, for example due to common storage buffers for clusters of stations.



In order to evaluate the performance of such systems, over the last two decades *queueing network descriptions* have widely been used and shown to provide a powerful modeling tool. Typical performance measures of interest are:

- a throughput
- a workstation utilization
- a delay or blocking probability
- a total process or response time.

Motivation

Unfortunately, closed form expressions for queueing networks, most notably *product-form* expressions, are usually not obtainable under finite constraints. Numerous approximation techniques have therefore been developed over the last decade (see [1] and references therein). Despite the usefulness of these approximations however, as disadvantages they

- are computationally expensive
- provide quantitative rather than qualitative results
- heavily rely upon detailed underlying distributional assumptions
- do not generally provide a strict confidence on the accuracy of the results.

In practical situations, in contrast, one might just be interested in rough but guaranteed indications of orders of magnitude. In [9] and related earlier references, therefore, a bounding technique was developed for networks, such as assembly lines, with finite (individual) stations.

This technique was based on modifying a non-product-form network into a product-form network by enforcing a notion of balance per individual station. Numerical support indicated a practical usefulness for both quick engineering and design purposes, in combination with the fact that it could be applied at *down-to-earth physical* basis.

Objective and results

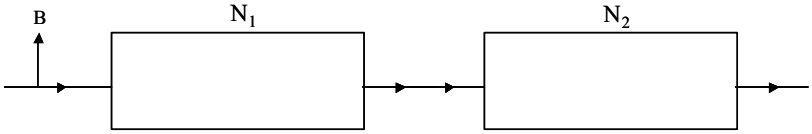
In view of the above motivation, this paper aims to extend the application of this technique to assembly line queueing network structures with finite constraints for *cluster of stations*. This extension is not evident, as it requires *station balance* in combination with a new notion, which will be called *cluster balance*.

It will be shown how these two notions are to be and can be combined at operational basis in order to extend the bounding technique to assembly line structures with **finite clusters** of stations. To this end some generic examples will be studied and be supported by numerical results. These results indicate a potential practical usefulness for quick evaluation purposes.

2 AN INSTRUCTIVE EXAMPLE

2.1 THE FINITE STATION CASE

First, let us consider a most simple but nevertheless unsolvable system, which can be regarded as representative for finite assembly or production (manufacturing) lines.



This concerns a finite tandem queue with Poisson arrival rate λ and exponential single servers with rates μ_1 and μ_2 at stations 1 and 2 respectively. Furthermore, station i cannot contain more than N_i jobs (the one in service included). When station 1 is saturated, an arriving job is rejected and lost. When station 2 is saturated the service at station 1 is effectively stopped (cf. [3] for equivalence results of different blocking protocols).

As simple as the system may look to analyze, there is no simple expression for the *throughput*, that is, the mean number of finished parts per unit of time, and as of today no explicit expression at all in terms of input and service parameters appears to be available. The large number of publications purely devoted to efficient approximations for such systems, however, might indicate the practical relevance. (See e.g. [1]).

Simple bounds

As following from the results in [9], a simple explanation for this system to be unsolvable, is the lack of a so-called notion of *station balance* (or flow balance per station) at station 1. Indeed, when station 2 is saturated:

The flow out of station 1 is blocked and thus *equal to* 0. But the flow into station 1 is still possible and thus *larger than* 0.

As the notion of station balance is responsible for a product-form expression, the following artificial modification, *purely to evaluate* the system and *not to be implemented in practice*, is therefore suggested in order to restore station balance.

> When station 2 is saturated, also stop the input.
> Hence not only the *outflow*, but also the *inflow* at station 1 then becomes 0.
> In addition, by regarding the system as cyclic we should also stop station 2 when station 1 is saturated.

This modification transforms the system into an easily solvable system from which one can derive a simple closed form expression for the throughput **H** or the directly related blocking probability **B** for arriving jobs to get blocked as given by (2.1) with **c** a normalizing constant:

$$\begin{cases} \mathbf{H} = \lambda(1-\mathbf{B}) \\ \mathbf{B} = \pi(n_1 = N_1 \text{ or } n_2 = N_2) \\ \pi(n_1, n_2) = c\left[\frac{\lambda}{\mu_1}\right]^{n_1}\left[\frac{\lambda}{\mu_2}\right]^{n_2} \end{cases} \tag{2.1}$$

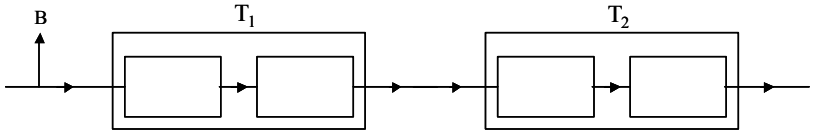
Intuitively, it is clear that this blocking probability will be larger than in the original system and thus provides an *upper bound* $\mathbf{B}_u$ for the blocking probability (and thus leading to a lower bound $\mathbf{H}_L$ for the throughput) of that for the original *non-tractable system*. Counterintuitively, however, a formal proof appears to be harder to establish, as the system is not stochastically monotone (see [15]). Nevertheless, formal proofs have been developed (see [10], [15]). In a similar fashion, also a lower bound $\mathbf{B}_L$ (and upper bound $\mathbf{H}_U$) can be obtained by rejecting jobs only when $\mathbf{N}_1 + \mathbf{N}_2$ jobs are already present while allowing up to $\mathbf{N}_1 + \mathbf{N}_2$ jobs at either station.

One cannot expect these bounds to be anywhere near too accurate as they are based on drastic distortions of the original behaviour. (The average value of the lower and the upper bound appears to be quite reasonable though, roughly within **10%** accuracy).

Nevertheless, as rough quick indicators of orders of magnitude or qualitative behaviour they can still be practically useful. In the next section we therefore aim to investigate the possible extension to more complicated situations in which constraints are imposed upon more than one station simultaneously.

2.2 THE FINITE CLUSTER CASE

In practical production environments, however, capacity constraints are often imposed upon clusters of workstations rather than individual workstations. It would thus be appealing if for such cases the same principle of station balance can also be operated at cluster level, by regarding a cluster as one *aggregated station*.



Consider for example the cluster extension of the above tandem case with four stations to be seen as a two-cluster model with finite constraints $\mathbf{T}_1$ and $\mathbf{T}_2$ for the total number of jobs in cluster 1 (stations 1 and 2) and cluster 2 (stations 3 and 4). A similar modification as in the preceding case would then seem appealing at cluster level to enforce a simple product-form expression by also rejecting jobs at cluster 1 when cluster 2 is congested, from which an *upper bound* for the blocking probability would be derived.

In other words, at first glance we would expect a similar simple product-form bounding approach by simply regarding a cluster as a station and transforming the notion of balance per station into balance per cluster by just keeping track of the total number of jobs at each cluster. This will be referred to as *global cluster balance*.

However, as the bounding approach is based upon product-forms, which in turn follow from the global balance equations for the underlying Markov process, a detailed state description is still required and one has to be careful. Indeed, by still allowing station 1 to continue to work while the second cluster is saturated ($t_2 = \mathbf{T}_2$), station balance and hence a product-form is violated as:

the inrate at station 1 is 0
but the outrate at station 1 is positive.

By also taking **station balance** into account **within the clusters** the following modification could therefore be suggested to enforce a product-form.

> When cluster 2 is saturated ($t_2 = \mathbf{T}_2$),
> *stop* the input as well as *both stations* at cluster 1.
>
> In addition, when cluster 1 is saturated ($t_1 = \mathbf{T}_1$),
> *stop* cluster 2, that is, stop *both stations* at cluster 2.

Indeed, with $\mu_j(n_j)$ the service rate of station j when n_j jobs are present and provided the station works, with S the set of admissible states as given by

$$S = \{\mathbf{n} = (n_1, n_2, n_3, n_4) \mid t_1 = n_1+n_2 \le \mathbf{T}_1;\ t_2 = n_3+n_4 \le \mathbf{T}_2;\ t_1+t_2 \ne \mathbf{T}_1+\mathbf{T}_2\}$$

and with $\mathbf{1}_{\{A\}}$ the indicator of event A, i.e. $\mathbf{1}_{\{A\}} = 1$ if A is satisfied and $\mathbf{1}_{\{A\}} = 0$ otherwise, under the above modification one easily verifies the station balance equations, for any $\mathbf{n} \in C$ and all stations j = 1, 2, 3, 4, given by:

$$\left\{\begin{array}{l} \pi(\mathbf{n})\mu_1(n_1)\mathbf{1}_{(t_2<T_2)} = \pi(\mathbf{n}-e_1)\lambda\mathbf{1}_{(t_2<T_2)} \\ \pi(\mathbf{n})\mu_2(n_2)\mathbf{1}_{(t_2<T_2)} = \pi(\mathbf{n}-e_2+e_1)\mu_1(n_1+1)\mathbf{1}_{(t_2<T_2)} \\ \pi(\mathbf{n})\mu_3(n_3)\mathbf{1}_{(t_1<T_1)} = \pi(\mathbf{n}-e_3+e_2)\mu_2(n_2+1)\mathbf{1}_{(t_1<T_1)} \\ \pi(\mathbf{n})\mu_4(n_4)\mathbf{1}_{(t_1<T_1)} = \pi(\mathbf{n}-e_4+e_3)\mu_3(n_3+1)\mathbf{1}_{(t_1<T_1)} \end{array}\right\}$$

when substituting the product-form, with c a normalizing constant at S:

$$\pi_U(\mathbf{n}) = c\,\lambda^{n_1+n_2+n_3+n_4}\prod_{i=1}^{4}\left[\prod_{k=1}^{n_i}\mu_i(k)\right]^{-1} \qquad (\mathbf{n}\in S)$$

With **B** the corresponding loss probability of the original system, a simple upper bound $\mathbf{B}_U$ is thus obtained by

$$\mathbf{B}_U = \sum_{\{n|t_1=\mathbf{T}_1 \text{ or } t_2=\mathbf{T}_2\}}\pi_U(\mathbf{n})$$

As in section 2.1, also a lower bound product-form modification $\mathbf{B}_L$ can be obtained by allowing up to $\mathbf{T}_1+\mathbf{T}_2$ jobs at either cluster (and individual station) while rejecting jobs only when $\mathbf{T}_1+\mathbf{T}_2$ jobs are already present.

Below some numerical results are given for the case of single server stations. Here μ_i represents the service speed of station i, $\mathbf{B}_L$ and $\mathbf{B}_U$ are the easily obtained lower and upper bound for the blocking probability, $\mathbf{B}_{av} = (\mathbf{B}_L + \mathbf{B}_U) / 2$ and **B** is obtained by (computationally expensive) numerical computation under the assumption of exponential services. Here one may recall that also bounds for the throughput **H** are established by virtue of the relation (with λ the arrival intensity):

$$\mathbf{H} = \lambda(1-\mathbf{B})$$

Table 1: Lower and upper bounds for two-cluster tandem example

μ_1	μ_2	μ_3	μ_4	$\mathbf{T}_1$	$\mathbf{T}_2$	$\mathbf{B}_L$	$\mathbf{B}_U$	$\mathbf{B}_{av}$	**B**
1	1	1	1	3	5	.33	.52	.43	.42
1	1	1	1	6	6	.25	.40	.33	.30
1	1	1	1	8	8	.20	.33	.27	.24
2	2	1	1	10	10	.10	.17	.14	.12
1	2	3	2	10	10	.054	.101	.078	.084
1.1	2	3	2	10	10	.021	.065	.048	.049

3 GENERAL RESULTS

The instructive example in section 2.2 illustrates that a detailed state description is still required in order to conclude a product-form also when capacity constraints are imposed only upon clusters of stations. In other words, *one cannot just simply regard a cluster as one aggregate station*. On the other hand, for practical purposes, instead of checking all station balance relations for each individual station and in all possible detailed state situations, it *seems intuitively appealing and computationally attractive* to transform the product-form conditions and insights for finite individual stations to clusters of stations as if they can be aggregated as individual stations, and thus by just considering the total numbers of jobs at these clusters.

In this section we will give a more formal treatment to resolve this theoretical and practical conflict. To this end, we will distinguish a notion of *detailed* and of *global cluster balance* (**DCB** and **GCB**), in relation to the standard notion of station balance (**SB**).

Model description

Consider a closed or open network with N service stations, with in the open case an arrival rate γ_j at station j = l, ..., N and in the closed case a fixed number of jobs. Let $\lambda = [\gamma_1 + \ldots + \gamma_N]$. The service rate at station j is $\mu_j(n_j)$ when n_j jobs are present. Furthermore, the stations are partitioned in disjoint clusters $C_l, C_2, \ldots, C_K$. Here we typically have in mind that a job arriving at a cluster can be blocked based upon the total number of jobs at this cluster.

More precisely, with a station vector $\mathbf{n} = (n_l, \ldots, n_N)$ denoting the numbers of jobs n_i at stations i and the cluster vector $\mathbf{t} = (t_1, \ldots, t_K)$ denoting the total number of jobs t_k at cluster k = 1, …, K, a job completing service at a station $i \in C_p$ will route to station $j \in C_q$ with probability:

$$p_{ij}(\mathbf{n}) = \begin{cases} a_{ij}(\mathbf{t}) & (i \neq j \in C_p) \\ p_{ij}\,\mathbf{B}_{pq}(\mathbf{t}) & (p \neq q) \\ 1 - \sum_{j \neq i} p_{ij}(\mathbf{n}) & (j = i) \end{cases} \tag{3.1}$$

where $\mathbf{B}_{pq}(\mathbf{t})$ represents a probability. In words, a routing from one cluster to another, say from station i to j, will be requested with fixed probability p_{ij}. This request is accepted with probability $\mathbf{B}_{pq}(\mathbf{t})$ where p and q represent the clusters containing i and j and it is blocked otherwise. When blocked, the job remains at station i. In addition, the routing within a cluster may depend on the cluster population **t**. For example, a finite constraint $\mathbf{T}_q$ for cluster q is modeled by:

$$\mathbf{B}_{pq}(\mathbf{t}) = \begin{cases} 1 & \text{if } t_q + 1 \leq \mathbf{T}_q \quad (\text{for } q \neq p) \\ 0 & \text{otherwise} \end{cases}$$

Furthermore, for the *open case* we also include values i = 0 and j = 0 and correspondingly p = 0 and q = 0, to reflect possible blocking upon arrival at or departure from the network, where we must read $p_{0j}(\mathbf{n}) = [\gamma_j/\gamma]\ \mathbf{B}_{0q}(\mathbf{t})$ and where the probability for leaving the network is given by $p_{i0}(\mathbf{n}) = p_{i0}\ \mathbf{B}_{i0}(\mathbf{t})$.

Remark 3.1 (Delay / stop blocking) In fact, we can also regard

$$\mathbf{D}_i(\mathbf{n}) = \sum_{j \neq i} p_{ij}(\mathbf{n}) \tag{3.2}$$

as a *delay factor* for the effective service rate of station i when the system is in state **n**. That is, the effective service rate of departing customers at station i in, state n becomes

$$\mu_i(\mathbf{n}) = \mu_i(n_i)\,\mathbf{D}_i(\mathbf{n})$$

while conditionally that a job leaves station i in state **n**, which implicitly requires that $\mathbf{D}_i(\mathbf{n}) > 0$, it will shift to station $j \neq i$ with conditional probability

$$\overline{p}_{ij}(\mathbf{n}) = \frac{p_{ij}(\mathbf{n})}{\mathbf{D}_i(\mathbf{n})} \tag{3.3}$$

Particularly, when $\mathbf{D}_i(\mathbf{n}) = 0$ the effective service rate of departing jobs from station i has become 0, which is equivalent to stating that service at station i has **completely stopped**.

In this respect, for usage later on, we also emphasize here that the routing within a cluster is allowed to be blocked with the effect that all internal transitions are blocked. Clearly, due to the underlying exponential structure we could state that station i is effectively *stopped* when

$$\mathbf{D}_i(\mathbf{n}) = 0 \qquad (\Leftrightarrow p_{ii}(\mathbf{n}) = 1)$$

Balance relations

Our interest is the steady state solution $\{\pi(\mathbf{n})\}$. To this end, let $\mathbf{n} + e_i$ or $\mathbf{n} - e_i$ denote the state equal to **n** with one job more (+) or less (-) at station i. Similarly, we define $\mathbf{t} + e_k$ and $\mathbf{t} - e_k$ for cluster vectors and by $\mathbf{n} + e_i - e_j$ we denote the vector equal to **n** with one job moved from station j to station i. Then $\{\pi(\mathbf{n})\}$ is uniquely determined, up to normalization, by the *global balance relations*:

$$\begin{aligned} &\pi(\mathbf{n})\textstyle\sum_j \mu_j(n_j)\sum_i p_{ji}(\mathbf{n}) = \\ &\textstyle\sum_j \left\{\pi(\mathbf{n}-e_j)\gamma p_{0j}(\mathbf{n}-e_j) + \sum_i \pi(\mathbf{n}-e_j+e_i)\mu_i(n_i+1)\,p_{ij}(\mathbf{n}-e_j+e_i)\right\} \end{aligned} \tag{3.4}$$

Here we note that also blocked completions at station j are included in both the left and right hand side, as we allow i = j to be included. The following result shows that a product-form can be obtained if we can reduce these global balance relations to station balance equations. In order to purely concentrate on the routing and related blocking or delay aspects, the result is presented in a form in which only the routing components remain. Herein, let S be the set of admissible states **n**. Furthermore, for a given station vector **m**, let **t**[**m**] be the corresponding cluster vector, i.e. $\mathbf{t}[\mathbf{m}] = (t_1, t_2, \ldots, t_k)$ with for any k: $t_k = \sum_{i \in C_k} n_i$.

Result 3.1 The steady state distribution exhibits the product-form

$$\pi(\mathbf{n}) = c\,\mathbf{R}(\mathbf{n})\prod_i \left[\prod_{n=1}^{n_i} \mu_i(n)\right]^{-1} \quad (\mathbf{n} \in S) \tag{3.5}$$

when there exists a function **R**(**n**) at S such that for any **m** and j with $\mathbf{m} + e_j \in S$:

$$\mathbf{R}(\mathbf{m}+e_j)\textstyle\sum_{i\neq j} p_{ji}(\mathbf{m}+e_j) = \mathbf{R}(\mathbf{m})\gamma p_{0j}(\mathbf{m}) + \sum_{i\neq j} \mathbf{R}(\mathbf{m}+e_i)\,p_{ij}(\mathbf{m}+e_i) \tag{3.6}$$

Proof

This is standard (cf [9]) and can be verified directly by substituting (3.5) in (3.4) while considering a fixed station j and underlying state $\mathbf{m} = \mathbf{n} - e_j$ for using (3.6). □

Equation (3.6) has the interpretation that for any station j the flow out of a state **n**, which we can write as $\mathbf{n} = \mathbf{m} + e_j$ due to a departing job at station j, where we could think of unit service rates $\mu_j = 1$, is balanced by the flow into that state due to an arrival at station j, as if one job is moving around with fixed underlying state $\mathbf{m} = \mathbf{n} - e_j$. Accordingly, we will refer to this equation as *station balance*. More precisely, we define the following notions:

- **Station balance (SB):**

  (3.6) holds for any **m** and j with $\mathbf{m} + e_j \in S$.

- **Detailed cluster balance (DCB):**

$$\begin{aligned} &\sum_{j\in C_k}\left\{\mathbf{R}(\mathbf{m}+e_j)\sum_{i\neq C_k} p_{ji}(\mathbf{m}+e_j)\right\} = \\ &\sum_{j\in C_k}\left\{\mathbf{R}(\mathbf{m})\gamma p_{0j}(\mathbf{m}) + \sum_{i\neq C_k}\mathbf{R}(\mathbf{m}+e_i)\, p_{ij}(\mathbf{m}+e_i)\right\} \end{aligned} \tag{3.7}$$

  for any fixed station vector **m** and cluster k.

  Relation (3.7) has the interpretation that for any underlying detailed state **m** and for any cluster k the total outflow from cluster k with one job more at that cluster is balanced by the total inflow into that cluster k due to an arriving job.

- **Global cluster balance (GCB):**

$$\begin{aligned} &\sum_{\{\mathbf{m}|\mathbf{t}[\mathbf{m}]=\mathbf{s}\}}\ \sum_{j\in C_k}\left\{\mathbf{R}(\mathbf{m}+e_j)\sum_{i\neq C_k} p_{ji}(\mathbf{m}+e_j)\right\} = \\ &\sum_{\{\mathbf{m}|\mathbf{t}[\mathbf{m}]=\mathbf{s}\}}\ \left[\sum_{j\in C_k}\left\{\mathbf{R}(\mathbf{m})\gamma p_{0j}(\mathbf{m}) + \sum_{i\neq C_k}\mathbf{R}(\mathbf{m}+e_i)\, p_{ji}(\mathbf{m}+e_i)\right\}\right] \end{aligned} \tag{3.8}$$

  for any fixed cluster vector **s** and cluster k.

  Also (3.8) has the interpretation of flow balance for each cluster k separately, but in contrast, based only upon the *global configuration* **t** for the total number of jobs at each cluster. Roughly speaking that is, **GCB** can be regarded as **SB** by regarding each cluster as an *aggregated single station*. This notion can therefore be practical for verification.

Result 3.2

$$\mathbf{SB} \overset{(1)}{\Rightarrow} \mathbf{DCB} \overset{(2)}{\Rightarrow} \mathbf{GCB} \tag{3.9}$$

Proof

Implication **(2)** is immediate by the summation over all **m** (see also the figure below). To prove implication **(1)**, consider a fixed **m** and k. Then we can write:

$$\sum_{j\in C_k}\left\{\mathbf{R}(\mathbf{m}+e_j)\sum_{i\neq j}p_{ji}(\mathbf{m}+e_j)\right\}=$$
$$\sum_{j\in C_k}\left\{\mathbf{R}(\mathbf{m}+e_j)\sum_{i\neq C_k}p_{ji}(\mathbf{m}+e_j)\right\}+\sum_{j\in C_k}\left\{\mathbf{R}(\mathbf{m}+e_j)\sum_{i\in C_k,\,i\neq j}p_{ji}(\mathbf{m}+e_j)\right\} \tag{3.10}$$

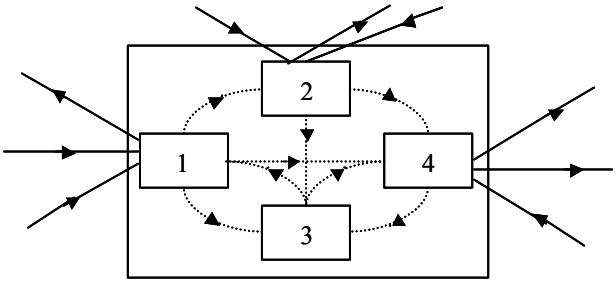
By assuming (3.6) for all j, in a similar way we can also write:

$$\sum_{j\in C_k}\left\{\mathbf{R}(\mathbf{m}+e_j)\sum_{i\neq j}p_{ji}(\mathbf{m}+e_j)\right\}=$$
$$\sum_{j\in C_k}\left\{\mathbf{R}(\mathbf{m})\gamma p_{0j}(\mathbf{m})+\sum_{i\neq C_k}\mathbf{R}(\mathbf{m}+e_i)p_{ij}(\mathbf{m}+e_i)+\sum_{i\in C_k,\,i\neq j}\mathbf{R}(\mathbf{m}+e_i)p_{ij}(\mathbf{m}+e_i)\right\} \tag{3.11}$$

As the third expression in the right hand side, by interchanging the i, j indices, can also be rewritten as:

$$\sum_{j\in C_k}\sum_{i\in C_k,\,i\neq j}\mathbf{R}(\mathbf{m}+e_i)p_{ij}(\mathbf{m}+e_i)=\sum_{j\in C_k}\mathbf{R}(\mathbf{m}+e_j)\sum_{i\in C_k,\,i\neq j}p_{ji}(\mathbf{m}+e_j),$$

equating (3.10) and (3.11) now proves (3.7) and thus implementation **(1)**.



□

Use of result 3.2

Result 3.2 can be practical in *negative and positive* sense. In *negative* sense, as it shows that when Cluster Balance (**CB**) is violated, first checked at global level (**GCB**) or otherwise at detailed level (**DGB**) also station balance (**SB**) fails so that the product-form (3.5) cannot hold.

In *positive* sense, as this failure of **GCB** or **DCB** might directly suggest a modification by which **SB** and thus the product-form (3.5) can be regained. Conditions to this end will be explored further in this section.

By the *instructive tandem cluster* from section 2.2 we already provided an example. When the second cluster was saturated ($t_2 = \mathbf{T}_2$), global cluster balance (**GCB**) was violated at the level of the total number of jobs (t_1, t_2). By also rejecting arrivals when $t_2 = \mathbf{T}_2$ and stopping service at cluster 2 when $t_1 = \mathbf{T}_1$ global as well as detailed cluster balance (**GCB** and **DCB**) were restored.

Simply stated, result 3.2 justifies a first essential check and possible modification in order to conclude a product-form at the easiest physical level, that is *the level at which blocking arises*.

Whether station balance **SB** and thus the product-form (3.5) can be obtained will still depend on the actual blocking protocol, as was also illustrated in section 2.2. Furthermore, while **GCB** is intuitively expected to be easier to verify as based upon just total cluster population, the technical form as presented by (3.8) appears less so as it still contains summations over detailed states.

The following result will provide a general structural condition by which this detailed check can be avoided so that the product-form (3.5) can be concluded directly purely at global cluster level.

Result 3.3 Consider transition probabilities p_{ij} of the form:

$$\begin{cases} p_{0i} = \beta_p \alpha_i & (\text{for } i \in C_p) \\ p_{ij} = p_{ij} & (\text{for } i \in C_p,\ j \in C_p) \\ p_{ij} = \delta_i \mathbf{R}_{pq} \alpha_i & (\text{for } i \in C_p,\ j \in C_q,\ q \neq p), \end{cases} \tag{3.12}$$

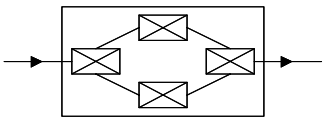
where for motivational simplicity we assume that $p_{ii} = 0$, and suppose that for some function $\mathbf{H}(.)$ at S_C, the set of admissible cluster configurations, all cluster configurations $\mathbf{t}$ and $\mathbf{s}$ and clusters p with $\mathbf{t} = \mathbf{s} + e_p \in S_C$:

$$\begin{aligned} &\mathbf{H}(\mathbf{s}+e_p)\textstyle\sum_{q\neq p}\mathbf{B}_{pq}(\mathbf{s}+e_p) = \\ &\mathbf{H}(\mathbf{s})\beta_p \mathbf{B}_{0p}(\mathbf{s}) + \textstyle\sum_{q\neq p}\mathbf{H}(\mathbf{s}+e_q)\mathbf{R}_{qp}\mathbf{B}_{qp}(\mathbf{s}+e_q) \end{aligned} \tag{3.13}$$

Further, the routing probabilities from one cluster to another are specified as by (3.1) with p_{ij} as by (3.12). The routing probabilities between stations within one cluster, however are specified by either one of the following two possible blocking protocols. Herein, we use the notation:

$$\mathbf{D}_p(\mathbf{t}) = \textstyle\sum_{q\neq p}\mathbf{R}_{pq}\mathbf{B}_{qp}(\mathbf{t}) \tag{3.14}$$

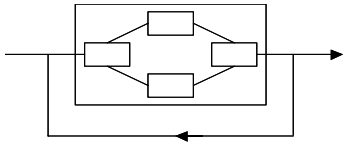
(i) (*Delay protocol*)



$\mathbf{D}_p(\mathbf{t}) = 0$

In cluster state $\mathbf{t}$, at any cluster p the service at all stations i within that cluster ($i \in C_p$) is delayed by the factor $\mathbf{D}_p(\mathbf{t})$. Particularly, when $\mathbf{D}_p(\mathbf{t}) = 0$, all stations at cluster p are stopped. Conditional that a departure takes place at station i, so that $\mathbf{D}_p(\mathbf{t}) > 0$, the job will route to one of the stations $j \neq i$ with probability $[p_{ij}(\mathbf{n}) / \mathbf{D}_p(\mathbf{t})]$.

(ii) (*Recirculate protocol*)



$\mathbf{D}_p(\mathbf{t}) = 0$

Services are not delayed. That is, any station i at a cluster p always services at a rate $\mu_i(n_i)$ when n_i jobs are present. However, upon aiming to depart cluster p after a completion at a station i, which occurs with probability p_{i0}, a job is blocked with probability $[1 - \mathbf{D}_p(\mathbf{t})]$. In that case it has to recirculate within cluster p and with probability α_j routes to station $j \in C_p$.

Then **SB** and the product-form (3.5) hold with

$$\mathbf{R(n)} = \mathbf{H(t)}\Lambda\mathbf{(n)} \qquad \text{with} \quad \Lambda\mathbf{(n)} = \prod_i \left[\lambda_i\right]^{n_i} \quad \text{where} \tag{3.15}$$

$$\lambda_j = \alpha_j + \sum_{i\in C_p} \lambda_i p_{ij} \quad \text{(for all } j \in C_p\text{, and any cluster p)} \tag{3.16}$$

Proof First note that (3.16) represents the traffic or steady state equations for the routing probabilities within one cluster. As we implicitly assumed that any station can be visited, otherwise we did not have to include it, this routing matrix with p_{0i} and p_{i0} included is necessarily irreducible, so that a unique solution $\{\lambda_i\}_{i \in Cp}$ exists. Furthermore, we have,

$$\begin{aligned}
&\textstyle\sum_{i\in C_p} \lambda_i \delta_i = \\
&\textstyle\sum_{i\in C_p} \lambda_i \left[1 - \sum_{j\in C_p} p_{ij}\right] = \\
&\textstyle\sum_{i\in C_p} \lambda_i - \sum_{j\in C_p} \sum_{i\in C_p} \lambda_i p_{ij} = \\
&\textstyle\sum_{i\in C_p} \lambda_i - \left[\sum_{j\in C_p} \lambda_j - \sum_{j\in C_p} \alpha_j\right] = 1
\end{aligned} \tag{3.17}$$

We will first consider the delay protocol i)

We need to verify the station balance relation (3.6). First note that the delay protocol is contained by the formulation of transition probabilities (3.1) by:

$$p_{ij}\mathbf{(n)} = \mathbf{D}_p\mathbf{(t)}\, p_{ij} = a_{ij}\mathbf{(t)} \qquad \text{(for } i,\ j \in\ C_p,\ j \neq i\text{)}$$

$$\mathbf{D}_p\mathbf{(t)}\frac{p_{ij}\mathbf{(n)}}{\mathbf{D}_p\mathbf{(t)}} = \delta_i \alpha_j\, \mathbf{R}_{pq}\mathbf{B}_{pq}\mathbf{(t)} = p_{ij}\, \mathbf{B}_{pq}\mathbf{(t)}$$

(for $i \in C_p$ and $j \in C_q$ with $q \neq p$, provided $\mathbf{R}_{pq}\mathbf{B}_{pq}\mathbf{(t)} > 0$)

By substituting these probabilities into the station balance relation (3.6) and using (3.14), for $j \in C_q$ and with $\mathbf{n} = \mathbf{m} + e_j$ and $\mathbf{t} = \mathbf{s} + e_p$, in order to verify (3.6) it should thus hold that (where we also implicitly assumed that $\mathbf{R}_{pp} = 0$ by $p_{ii} = 0$ and (3.12))

$$\begin{aligned}
&\mathbf{H(s}+e_p\mathbf{)}\lambda_j \Lambda\mathbf{(m)}\,\mathbf{D}_p\mathbf{(s}+e_p\mathbf{)} = \\
&\qquad \mathbf{H(s)}\Lambda\mathbf{(m)}\beta_p\alpha_j\, \mathbf{B}_{0p}\mathbf{(s)} + \\
&\qquad \textstyle\sum_{i\in C_p} \mathbf{H(s}+e_p\mathbf{)}\Lambda\mathbf{(m)}\lambda_i p_{ij}\, \mathbf{D}_p\mathbf{(s}+e_p\mathbf{)} + \\
&\qquad \textstyle\sum_{q\neq p}\sum_{i\in C_p} \mathbf{H(s}+e_q\mathbf{)}\Lambda\mathbf{(m)}\lambda_i\delta_i\, \mathbf{R}_{qp}\, \mathbf{B}_{qp}\mathbf{(s}+e_q\mathbf{)}\alpha_j
\end{aligned} \tag{3.18}$$

By cancelling $\Lambda(\mathbf{m})$ and using (3.16) and (3.17), relation (3.18) can be reduced to:

$$\mathbf{H}(\mathbf{s}+e_p)\lambda_j\,\mathbf{D}_p(\mathbf{s}+e_p) = \left[\mathbf{H}(\mathbf{s}+e_p)\sum_{i\in C_p}\lambda_i p_{ij}\,\mathbf{D}_p(\mathbf{s}+e_p)\right] + \left[\sum_{q\neq p}\mathbf{H}(\mathbf{s}+e_q)\mathbf{R}_{qp}\,\mathbf{B}_{qp}(\mathbf{s}+e_q) + \mathbf{H}(\mathbf{s})\beta_p\,\mathbf{B}_{0p}(\mathbf{s})\right]\alpha_j \tag{3.19}$$

By using the cluster balance relation (3.13), relation (3.14) and (3.16), relation (3.19) and thus (3.18) is hereby verified, which proves (3.6). The proof for the delay protocol is hereby completed.

Next consider the recirculate protocol ii)

As before, first note that this protocol is contained by the formulation of transition probabilities (3.1), by

$$p_{ij}(\mathbf{n}) = p_{ij} + \delta_i[1-\mathbf{D}_p(\mathbf{t})]\alpha_j, \qquad (\text{for } i, j\in C_p,\ j\neq i),$$
$$p_{ij}(\mathbf{n}) = p_{ij}\,\mathbf{B}_{pq}(\mathbf{t}) = \delta_i\,\mathbf{R}_{pq}\mathbf{B}_{pq}(\mathbf{t})\alpha_j \qquad (\text{for } i\in C_p, j\in C_q,\ q\neq p).$$

Again, by substituting these probabilities into the station balance relation (3.6) and fitting in (3.14), for $j \in C_p$ and with $\mathbf{n} = \mathbf{m} + e_j$ and $\mathbf{t} = \mathbf{s} + e_p$, in order to verify it should hold that (3.6) (note that $\sum_{j\neq i} p_{ij}(\mathbf{n}) = 1$)

$$\begin{aligned}\mathbf{H}(\mathbf{s}+e_p)\Lambda(\mathbf{m})\lambda_j = {}& \mathbf{H}(\mathbf{s})\Lambda(\mathbf{m})\beta_p\alpha_j\,\mathbf{B}_{0p}(\mathbf{s}) + \\ & \sum_{i\in C_p}\mathbf{H}(\mathbf{s}+e_p)\Lambda(\mathbf{m})\lambda_i p_{ij} + \\ & \sum_{i\in C_p}\mathbf{H}(\mathbf{s}+e_p)\Lambda(\mathbf{m})\lambda_i\delta_i[1-\mathbf{D}_p(\mathbf{t})]\alpha_j + \\ & \sum_{q\neq p}\sum_{i\in C_p}\mathbf{H}(\mathbf{s}+e_q)\Lambda(\mathbf{m})\lambda_i\delta_i\,\mathbf{R}_{qp}\,\mathbf{B}_{qp}(\mathbf{s}+e_q)\alpha_j.\end{aligned} \tag{3.20}$$

By canceling $\Lambda(\mathbf{m})$ and using (3.17), both for cluster p and clusters $q \neq p$, (3.20) reduces to:

$$\mathbf{H}(\mathbf{s}+e_p)\lambda_j = \left[\mathbf{H}(\mathbf{s})\beta_p\,\mathbf{B}_{0p}(\mathbf{s}) + \sum_{q\neq p}\mathbf{H}(\mathbf{s}+e_q)\mathbf{R}_{qp}\,\mathbf{B}_{qp}(\mathbf{s}+e_q)\right]\alpha_j + \mathbf{H}(\mathbf{s}+e_p)\sum_{i\in C_p}\lambda_i p_{ij} + \mathbf{H}(\mathbf{s}+e_p)[1-\mathbf{D}_p(\mathbf{t})]\alpha_j. \tag{3.21}$$

By the cluster balance relation (3.13), with (3.14) substituted, and the traffic equations (3.16) for cluster p, relation (3.21) and thus (3.20) is hereby verified, which proves (3.6). Also the proof for the recirculate protocol is hereby completed. □

Remarks 3.2

1. Condition (3.12) implies that arrivals at a cluster and departures from that cluster can be seen as taking place by one *(possibly virtual) entry node* and one *(possibly virtual) departure node*, while routing from one cluster to another is independent of the internal

source and destination nodes within the clusters. This condition is *most natural* in practical conditions.

2. Condition (3.13) can be seen as the cluster balance analogue of the station balance relations (3.2), purely based on the cluster populations with each cluster as an aggregate station. As conditions and insights for station balance to be satisfied are known in the literature (see [9], chapter 5) result 3.3 has hereby responded to the primary purpose of this paper, i.e. a way of detecting closed form solutions for networks with limited clusters to obtain bounds, merely based on global states. This is summarized by corollary 3.4.

3. A number of references have investigated general aggregation and decomposition results for Markov chains in order to simplify computations of steady state distributions (e.g. [2, 3, 6, 7, 8]). Most notably, the famous Norton result has been known for product-form type networks without capacity constraints [2], and been generalized to general product-form type networks in [4, 5].

   Result 3.2 is in fact related to results in the latter references, but written out more explicitly for the situation of finite clusters. More precisely, it is written in terms for physical interpretation in order to establish product-form modifications similarly to the finite station case (cf. [9] and [13]), as will be used in section 4. Furthermore, the specific forms as in result 3.3 under a delay or recirculate blocking protocol are not considered in these references.

Corollary 3.4 (Product-form detection)

Under condition (3.12) and by assuming either of the two blocking protocols as in result 3.3, closed product-form results can be found purely based on total cluster populations as per relation (3.13), regarding each cluster as an *aggregate station*.

Corollary 3.5 (Equivalence result)

Under the conditions (3.12) and (3.13), the delay and recirculate blocking protocol are equivalent.

4. Particularly, with $\mathbf{B}_{pq}(\mathbf{t})$ taking on strict 0-1 values, the candidate form for $\mathbf{H}(\mathbf{t})$ is:

$$\mathbf{H(t)} = \prod_k [\sigma_k]^{t_k} \quad \text{where } \left\{ \sigma_p = \beta_p + \sum_{q \neq p} \sigma_q \, \mathbf{R}_{qp} \ \textbf{(}\text{for all } p\textbf{)} \right\}$$

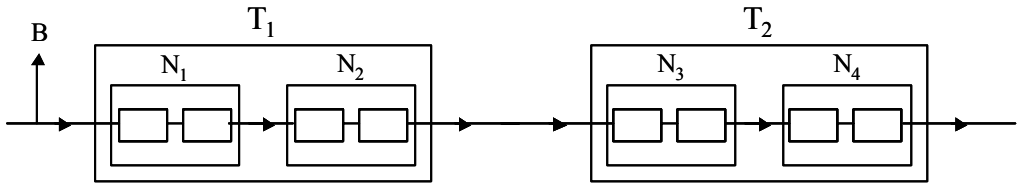
4 ILLUSTRATIVE EXAMPLES

In addition to the motivating tandem cluster example in section 2.2, in this section we provide some further illustration of how the general result from section 3 reduces the search for product-form bounds of nón-product-form networks, simply by regarding a cluster as one aggregate station. Numerical support will be provided which shows a practical usefulness for quick engineering purposes.

Here it is stated that formal proofs for product-form modifications that will be provided in these examples to lead to (formal) bounds of the loss probability, can be expected along the same lines as for the case of finite stations (e.g. [10, 11, 12, 14, 15]).

4.1 A NESTED BLOCKING STRUCTURE

As nested analogue of the example from section 2.2, in addition to the total cluster constraints $\mathbf{T}_1$ and $\mathbf{T}_2$, we also allow finite constraints $\mathbf{N}_i$ for each individual station i, i = 1, ..., 4.



Now, both station balance (SB) is violated when any of the stations is saturated ($n_i = \mathbf{N}_i$). E.g. consider a state (n_1, $\mathbf{N}_2$, n_3, n_4) with $n_4 < \mathbf{N}_4$, the rate out of this state due to a departure at station 3, that is, by (n_1, $\mathbf{N}_2$, n_3, n_4) $\rightarrow$ (n_1, $\mathbf{N}_2$, n_3-1, n_4+1) is positive, but the rate into this state due to an arrival at station 3, that is, by (n_1, $\mathbf{N}_2$+1, n_3-1, n_4) $\rightarrow$ (n_1, $\mathbf{N}_2$, n_3, n_4) is zero, while also global cluster balance (GCB) is violated as in section 2.2 when either $n_1 + n_2 = \mathbf{T}_1$ or $n_3 + n_4 = \mathbf{T}_2$. The following modification can therefore be suggested to restore both **SB** and **GCB**.

> When $n_i = \mathbf{N}_i$, stop arrivals and all other stations $j \neq i$.
>
> When $t_1 = \mathbf{T}_1$ stop both stations 3 and 4.
> When $t_2 = \mathbf{T}_2$ stop both stations 1 and 2 as well as arrivals.

In a similar fashion as in section 2.2 the station balance relations can then be verified easily which guarantee the product-form (3.5) with $\mathbf{R}(\mathbf{n}) = \lambda^t$ where $t = n_1 + n_2 + n_3 + n_4$ at the set of admissible states:

$$S_U = \{\mathbf{n} \mid t_1 \leq \mathbf{T}_1,\ t_2 \leq \mathbf{T}_2,\ t_1 + t_2 \neq \mathbf{T}_1 + \mathbf{T}_2,$$
$$n_i \leq \mathbf{N}_i,\ i = 1, ..., 4;\ n_i + n_j \neq \mathbf{N}_i + \mathbf{N}_j \text{ for all } i, j \text{ with } j \neq i\}$$

Clearly, this modification leads to an upper bound $\mathbf{B}_U$ for the loss probability **B**. Conversely, a lower bound $\mathbf{B}_L$ is obtained by the modification:

> Only reject arrivals when the total number of jobs
> $t = n_1 + n_2 + n_3 + n_4 = \mathbf{T}_1 + \mathbf{T}_2$, while any station can
> accommodate up to this number of jobs.

In this case again the station balance relations are readily verified with the same product-form (3.5) with $\mathbf{R}(\mathbf{n}) = \lambda^t$ at the set of admissible states:

$$S_L = \{\mathbf{n} \mid t \leq \mathbf{T}_1 + \mathbf{T}_2,\ n_i \leq \mathbf{T}_1 + \mathbf{T}_2 \text{ for } i = 1, ..., 4\}$$

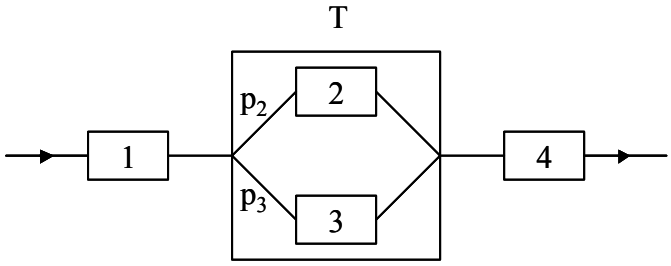
Some numerical results are presented below.

Table 2: Results for the nested constraints model

μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_7	μ_8	$\mathbf{N}_1$	$\mathbf{N}_2$	$\mathbf{N}_3$	$\mathbf{N}_4$	$\mathbf{T}_1$	$\mathbf{T}_2$	$\mathbf{B}_L$	$\mathbf{B}_U$	$\mathbf{B}_{av}$	$\mathbf{B}$
1	1	1	1	1	1	1	1	2	2	2	2	2	2	.667	.800	.733	.713
1	1	1	1	1	1	1	1	3	3	4	4	4	4	.500	.700	.600	.553
1	1	1	1	1	1	1	1	3	2	4	2	4	5	.471	.724	.598	.572
2	3	4	5	1	2	3	4	3	2	4	2	4	5	.158	.398	.278	.204

4.2 A CLUSTER WITH PARALLEL STATIONS (random routing)

In this example we allow a simple but yet random routing to either one of two stations in parallel within one cluster (as satisfying the form 3.12), as part of a production line.



With a total capacity constraint **T** for the total number of jobs at stations 2 and 3, next to finite capacity constraints $\mathbf{N}_i$ at each station i, i = 1, ..., 4.

SB is violated as before when either one of the individual stations is saturated ($n_i = \mathbf{N}_i$), while in addition when $n_2 + n_3 = \mathbf{T}$ the outflow from station 1 is 0 (even if $n_1 > 0$) while its inflow is still positive (when $n_1 < \mathbf{N}_1$), and conversely, the rate into a state n with $n_2 + n_3 = \mathbf{T}$ due to an arrival at station 4 is 0 (as there is no feasible state with $n_2 + n_3 = \mathbf{T}+1$) while the outrate from this state due to a departure from station 4 will be positive.

By regarding the cluster as one aggregated station as in section 2, product-form modifications, for which result 3.3 applies with $\mathbf{H}(.) \equiv 1$ and $\lambda_1 = \lambda$, $\lambda_2 = \lambda p_2$, $\lambda_3 = \lambda p_3$, $\lambda_4 = \lambda$, by either of the following two modifications:

> Stop arrivals and all stations either when one of stations ($n_i = \mathbf{N}_i$) or the cluster ($n_2 + n_3 = \mathbf{T}$) is saturated, or
>
> Stop arrivals when the total number of jobs is equal to $\mathbf{N}_1 + \mathbf{T} + \mathbf{N}_4 = S$, while each station may contain up to S jobs.

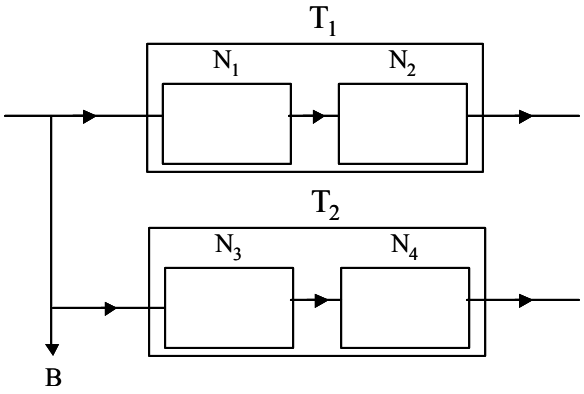
Clearly, the first modification leads to an upper bound $\mathbf{B}_U$ and the second to a lower bound $\mathbf{B}_L$ for the loss probability **B** of the original system. Some numerical results are shown below.

Table 3: Results for clusters with parallel stations

μ_1	μ_2	μ_3	μ_4	$\mathbf{N}_1$	$\mathbf{N}_2$	$\mathbf{N}_3$	$\mathbf{N}_4$	$\mathbf{T}$	p_2	p_3	$\mathbf{B}_L$	$\mathbf{B}_U$	$\mathbf{B}_{av}$	$\mathbf{B}$
1	1	1	1	1	1	1	1	1	0.5	0.5	0.48	0.75	0.62	0.64
1	1	1	1	1	1	1	1	2	0.5	0.5	0.40	0.75	0.58	0.59
2	2	2	2	2	2	2	2	3	0.5	0.5	0.03	0.30	0.16	0.16
10	10	10	10	2	2	2	2	4	0.5	0.5	0.00	0.02	0.01	0.01
1	1	1	1	5	5	5	5	10	0.5	0.5	0.10	0.30	0.20	0.18
1	1	1	1	10	5	5	10	10	0.75	0.75	0.06	0.17	0.12	0.10

4.3 OVERFLOW MODEL

In this model we have two finite clusters in parallel with arrivals at cluster 1. If a job cannot enter cluster 1 it is rerouted to cluster 2. Each cluster consists of two finite stations in tandem. In addition to the total cluster constraints $\mathbf{T}_1$ and $\mathbf{T}_2$, we also allow finite constraints $\mathbf{N}_i$ for each individual station i, i = 1, ..., 4. We assume that $\mu_1 \leq \mu_3$ and $\mu_2 \leq \mu_4$.



For this example cluster balance is violated when cluster 2 is busy while cluster 1 is not saturated. As in that case the outflow at cluster 2 is positive, but the inrate is 0. The following two modifications are therefore suggested:

> Stop both stations in cluster 2
> when cluster 1 is not saturated ($t_1 < \mathbf{T}_1$), or
>
> Assign arriving jobs randomly to either one of the clusters
> proportional to the free buffer capacity at the two clusters.

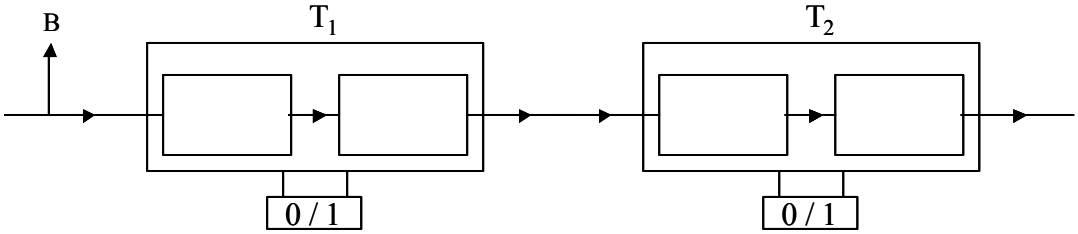
With the first modification cluster 2 is slowed down and is kept more congested, thus the arrival loss probability will be enlarged and thus leads to an upper bound $\mathbf{B}_U$ for the loss probability **B** of the original system. With the second modification, the faster overflow cluster is used more frequently than in the original system, which leads to a lower bound $\mathbf{B}_L$.

Table 4: Results for clusters with overflow

μ_1	μ_2	μ_3	μ_4	N_1	N_2	N_3	N_4	T_1	T_2	B_L	B_U	B_{av}	B
1	1	1	1	1	1	1	1	2	2	0.095	0.444	0.270	0.300
1	1	4	4	1	1	1	1	2	2	0.013	0.222	0.123	0.108
1	1	2	2	3	3	1	1	3	1	0.251	0.301	0.269	0.251
1	1	2	2	3	3	1	1	6	2	0.067	0.386	0.244	0.227
1	1	4	4	3	3	1	1	3	2	0.117	0.343	0.236	0.258
1	1	4	4	3	3	2	2	6	4	0.023	0.126	0.075	0.063
3	1	6	4	3	5	1	2	6	2	0.007	0.067	0.037	0.035

4.4 BREAKDOWN MODEL

In this model we consider two finite clusters is tandem both subject to breakdowns. In addition to the total cluster constraints $\mathbf{T}_1$ and $\mathbf{T}_2$ we assume repair and breakdown rates for cluster 1: γ_{10} and γ_{11} and, similarly, γ_{20} and γ_{21} for cluster 2.



Clearly, cluster balance is violated when either cluster is down. The following two modifications are therefore suggested:

> Stop both stations in cluster i when cluster j is down ($j \neq i$), or
>
> The breakdown rate for both clusters is 0 (breakdowns do not take place).

Clearly, the first modification leads to an upper bound $\mathbf{B}_U$ and the second to a lower bound $\mathbf{B}_L$ for the loss probability **B** of the original system. Some numerical results are shown below.

Table 5: Results for clusters with breakdowns

μ_1	μ_2	μ_3	μ_4	N_1	N_2	N_3	N_4	T_1	T_2	γ_{10}	γ_{11}	γ_{20}	γ_{21}	B_L	B_U	B_{av}	B
1	1	1	1	1	1	1	1	1	1	5	1	5	1	0.67	0.86	0.76	0.78
2	2	2	2	2	2	1	1	4	4	50	1	50	1	0.04	0.42	0.23	0.20
2	1	2	1	2	4	2	4	6	6	50	1	50	1	0.16	0.48	0.32	0.28
1	1	1	1	4	4	4	4	4	4	50	1	50	1	0.33	0.52	0.43	0.41

References

[1] Altiok, T and H.G. Perros, "*Queueing Networks with Blocking*", North-Holland (1989).

[2] Balsamo, S. and G. Iazeolla (1984), "Aggregation and disaggregation in queueing networks: The principle of product-form synthesis", in: *Mathematical Computer Performance and Reliability*, G. Iazeolla, P.J. Courtois, and A. Hordijk (eds.), North-Holland, Amsterdam, 95-109.

[3] Balsamo, S. and B. Pandolfi (1988), "Bounded aggregation in Markovian networks", in: *Mathematical Computer Performance and Reliability*, G. Iazeolla, P.J. Courtois, and A. Hordijk (eds.), North-Holland, Amsterdam, 73-92.

[4] Boucherie, R.J. (1998), "Norton's equivalent for queueing networks comprised of quasi-reversible components linked by state-dependent routing", *Performance Evaluation* 32, 83-99.

[5] Boucherie, R.J. and N.M. van Dijk (1993), "A Generalization of Norton's Theorem", *Queueing Systems* 13, 251-287.

[6] Dallery, Y. and Y. Frein (1989), "A decomposition method for the approximate analysis of closed queueing networks with blocking", in: *Queueing Networks with Blocking*, H.G. Perros and T. Altiok (eds.), North-Holland, Amsterdam, 33-46.

[7] Schweitzer, P.J. and T. Altiok (1988), "Aggregate modelling of tandem queueus with blocking", in: *Computer Performance and Reliability*, G. Iazeolla, P.J. Courtois and O. Boxma (eds.), North-Holland, Amsterdam, 135-149.

[8] Takahashi, Y. (1989), "Aggregate approximation for acyclic queueing networks with communication blocking", in: *Queueing Networks with Blocking*, H.G. Perros and T. Altiok (eds.), North-Holland, Amsterdam, 33-46.

[9] Van Dijk, N. M. "*Product forms for Queueing Networks: A system's Approach*", Wiley (1993).

[10] Van Dijk, N.M. (1988), "A Formal Proof for the Insensitivity of Simple Bounds for Multi-Server Non-Exponential Tandem Queues Based on Monotonicity Results", *Stochastic Proc. Appl*. 27, 261-277.

[11] Van Dijk, N.M. (1988), "Simple Bounds for Queueing Systems with Breakdowns", *Performance Evaluation* 8, 117-128.

[12] Van Dijk, N.M. (1998), "Bounds and error bounds for queueing networks", *Annuals of Operations Research* 79, 295-319.

[13] Van Dijk, N.M. and W.K. Grassmann (1999), "The product form tool for queueing networks", in: *Computational Probability,* W.K. Grassmann (ed.), Kluwer Academic Publishers, 409-444.

[14] Van Dijk, N.M. and P.G. Taylor (1998), "Strong stochastic bounds for the stationary distribution of a class of multicomponent performability models", *Operations Research 46,* 665-674.

[15] Van Dijk, N.M. and J. Van der Wal (1989), "Simple Bounds and Monotonicity Results for Multi-Server Exponential Tandem Queues", *Queueing Systems* 4, 1-16.