Available online at www.sciencedirect.com
ScienceDirect
Computers & Industrial Engineering 50 (2006) 466–487
computers & industrial engineering
www.elsevier.com/locate/dsw

Markovian analysis of a discrete material manufacturing system with merge operations, operation-dependent and idleness failures

A.C. Diamantidis, C.T. Papadopoulos *

Department of Economics, Aristotle University of Thessaloniki, Law School Building, GR-541 24 Thessaloniki, Greece

Abstract

This paper examines a discrete material manufacturing system consisting of three machines that are subject to breakdown and one buffer of finite capacity. It is assumed that the buffer has two immediate preceding machines performing the same operations and one immediate succeeding machine receiving material from the buffer. When the buffer reaches its own capacity, one of the two preceding machines has priority over the other to dispose its processed part into the buffer.

It is also assumed that there is a new way of the machines reaching failure, by allowing the machines to fail not only when they are operational but also when are either blocked or starved. The latter gives rise to the possibility of modeling the production of more than one part types. The model is solved analytically by developing a recursive algorithm that generates the transition matrix for any value C of the intermediate buffer capacity. Then various performance measures of the system (e.g., throughput) can be easily evaluated. Numerical results for the throughput are also given and these are compared against simulation. The proposed model may be used as a decomposition block to solve large flow lines with merge/split operations (for example, flow lines with quality inspections and rework loops) and multiple part types.

Keywords: Flow/production lines; Merge operations; Operation-dependent failures; Idleness failures; Performance evaluation

1. Introduction and literature review

Flow or production lines may be modelled as queuing networks. There are various approaches for analyzing such production systems. One approach is the classical Markovian analysis which is appropriate to solving small systems with up to six stations in series (see Heavey, Papadopoulos, & Browne, 1993; Hillier & Boling, 1967; Papadopoulos, Heavey, & O'Kelly, 1989).

Another solution approach is the classical decomposition method as introduced in the pioneering work of Gershwin (1987) and further analyzed by Dallery, David, and Xie (1988), Altiok (1997) and Helber (1999), among others. Dallery et al. (1988) actually improved the convergence of the decomposition algorithm

* Corresponding author. Tel.: +30 2310 997466; fax: +30 2310 996452.
E-mail addresses: adiama@econ.auth.gr (A.C. Diamantidis), hpap@econ.auth.gr (C.T. Papadopoulos).

0360-8352/$ - see front matter
doi:10.1016/j.cie.2005.04.004

proposed originally by Gershwin (1987). Yeralan and Tan (1997) used a matrix polynomial method to solve single-buffer discrete material flow production systems and Tan and Yeralan (1997) studied heterogeneous production systems using decomposition.

Another approach, different than the above two, is based on the aggregation method and was applied by Meerkov and his colleagues (Lim, Meerkov, & Top, 1990, among others). Meerkov has also introduced the mathematical theory of improvability for production systems (Jacobs & Meerkov, 1995a, 1995b).

There is a rich body of literature on the analysis of flow lines with linear flow of material (see the excellent review paper by Dallery & Gershwin, 1992 and the books of Gershwin, 1994; Papadopoulos, Heavey, & Browne, 1993) and a relatively scarce literature on the analysis of lines with non-linear flow of material (mainly covered in the book of Helber, 1999, where a thorough analysis of systems with split and merge operations was given). Unreliable transfer lines and also assembly/disassembly (A/D) systems with finite buffer capacities were examined by Gershwin (1991), Di Mascolo, David, and Dallery (1991) and Helber (1999), among others. An extension of the original decomposition method given in Gershwin (1987), is presented in Gershwin, Matta, Tolio, and Werner (2001), in order to evaluate the throughput and the distribution of inventory of a closed-loop production system.

The majority of the decomposition models are concerned with the production of just one part type. An extension of the decomposition approach for the performance evaluation of a serial flow line with linear flow of material and two part types was presented by Nemec (1999). Therein, Nemec extended the decomposition method originally proposed by Gershwin (1987) to cover the case where the machines are capable of producing two part type products, but it was still assumed that the machines were serially arranged and there were no merge operations in his model. Another decomposition analysis of serial flow lines that can process multiple part type products and multiple failure modes was described by Syrowicz (1999).

A different elegant decomposition analysis for serial flow lines with two part types, deterministic identical processing times and multiple failure modes was proposed by Colledani, Matta, and Tolio (2003). A major characteristic of all the decomposition approaches mentioned above is that the original production lines were decomposed only into sub-lines consisting of two machines and one intermediate buffer.

Merge operations in the flow of material are modeled through a system like the one depicted in Fig. 1. In such a system the flow of material is non-linear. Machines M_1 and M_2 that are upstream of buffer $B_{(1,2),3}$ feed this buffer with machine M_1 having priority over machine M_2 to be the next to fill the buffer when this is full.

Two other models similar to the one depicted in Fig. 1, with three stations and a limited buffer capacity with *continuous flow* of material were analyzed by Helber and Mehrtens (2003) and Tan (2001). The system examined in this paper is also similar as that given in Diamantidis, Papadopoulos, and Vidalis (2004). The latter was used as a decomposition block for the analysis of flow lines with merge operations and one part type.

In the model of the present work an extra assumption is made that the machines may break down not only when they are operational (operation-dependent failure), but also when are either blocked or starved (this is called *idleness failure* and was first introduced in Nemec (1999)). This assumption gives rise to the possibility of modeling the production of more than one part type and along with the assumption of discrete flow of material differentiates the proposed model from the models examined by Helber and Mehrtens (2003), Tan (2001) and Diamantidis et al. (2004).

The system depicted in Fig. 1, which models the merge operations, will play the role of a new decomposition block and is going to be used along with the classical two-machine one-buffer block

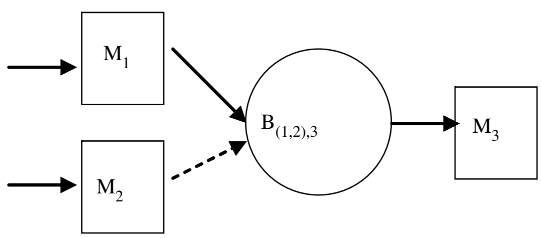

Fig. 1. A discrete material merge system.

for solving larger production lines with merge operations and two part types. This model represents a part of the production line of a Greek company that produces brass products (flat sections, rods, and tubes), copper products (sections) and aluminium products (wire) and this has been the motivation of this work.

The remainder of this paper is organized as follows: Section 2 presents the assumptions that were made in the proposed model and the notation and terminology, while Section 3 presents the derivation of the transition probabilities of the examined model. In Section 4, the steps for solving numerically the model depicted in Fig. 1 are presented. More specifically, Section 4.1 presents a formula for calculating the total number of Markovian states whereas Section 4.2 presents a special case of the transition matrix, when the buffer capacity equals two. Section 4.3 presents the steps of the proposed algorithm, while in Section 4.4 some performance measures are given. In Section 4.5 numerical results obtained from the proposed algorithm as well as simulation are presented for comparison reasons and finally Section 5 concludes this study and proposes some areas for further research.

2. Model assumptions

This section presents the assumptions of the proposed model (depicted in Fig. 1) as well as the notation and terminology that were made. The majority of the assumptions are similar to those of the model examined in Diamantidis et al. (2004).

1. An infinite supply of parts is available upstream of machines M_1 and M_2, and an unlimited storage area is present downstream of machine M_3. Thus the first two machines are never starved and the last machine is never blocked.
2. All machines have equal and constant service times. Time is scaled so that the machine cycle takes one time unit. Thus processing times are assumed to be deterministic and identical for all machines and are taken as the time unit.
3. The machines are no longer restricted to fail only when they are operating on a part (operation-dependent failures) but also it is possible for a machine to fail when it is starved or blocked (idleness failure). This assumption gives rise to the possibility of modeling the production of more than one part types in systems like the one depicted in Fig. 1. This is because the idleness failure may be used to model the case where a specific machine fails producing a part type two product while being blocked or starved for (producing) part type one products. In that case the machine seems to fail while being idle for part type one products (for this reason it is named idleness failure).
4. The probability that machine M_i, $i = 1,2$, fails while is operational, given it is not blocked is p_i, $i = 1,2$, and the probability that machine M_i fails while is blocked is v_i, $i = 1,2$. For the downstream machine M_3, the probability that it fails while is operational, given it is not starved is p_3, and the probability that it fails while is starved is v_3. The probabilities that M_i, $i = 1,2$ and M_3 are repaired while they are down are denoted by r_i, $i = 1,2,3$, respectively.
5. The failure and repair probabilities are assumed to be geometrically distributed.
6. When the buffer is full, machine M_1 has priority over machine M_2 to be next to fill the buffer. When the buffer is not full both machines feed the buffer simultaneously.
7. The current storage level of buffer $B_{(1,2),3}$ between machines M_1, M_2 and M_3 at time t is $n(t)$, $0 \leqslant n(t) \leqslant C$. The physical buffer between the machines is distinguished from the storage of the whole system. The storage of the whole system called extended buffer capacity includes the work areas of M_1, M_2 and M_3. Therefore the total storage capacity of the system is $N = C + 3$. The number of units in the system, b, varies from 0 to N.
8. The state of the three-machine line is $s = (b(t), \alpha_1(t), \alpha_2(t), \alpha_3(t))$, where $b(t)$ is the number of units in the system at time t and $\alpha_i(t)$, $i = 1,2,3$ is 0 or 1 if M_i, is down or up at time t, respectively ($b(t) = 0,\ldots, N$).
9. Transportation takes negligible time compared to processing times. All operating machines start their operations at the same time.
10. Machine states change due to failures and repairs at the beginning of time periods whereas buffer levels change due to completion of processing at the end of time periods.

11. It is assumed that $C \geqslant 2$ because systems with this property have a common mathematical structure that can be exploited.

3. Derivation of the transition equations

The assumptions of the model imply that certain states have zero steady-state probability. Such states are called *transient states* and cannot be reached from any other state except from other transient states. These states are all on the boundaries, (the reader is addressed to Gershwin (1994)). Take as an example state $(0, 0, 1, 1)$. This is a transient state because it cannot be reached from any other state. Since $a_1(t+1) = 0$, $a_2(t+1) = 1$, $a_3(t+1) = 1$ and $b(t) = 0$, machine M_3 will be unable to produce a part at time $t+1$ because it was starved at time t. However, machine M_2 will produce a part at time $t+1$ because it was not blocked at time t and, by assumption; an upstream machine is never starved. Machine M_1 is unable to produce a part at time $t+1$ since it is not operational and therefore $b(t+1)$ should be 1, which is not the case. In the case where $b(t) > 0$ and $a_1(t+1) = 0$, $a_2(t+1) = 1$, $a_3(t+1) = 1$, and following an analogous explanation one is led to the conclusion that $b(t+1)$ should equal $b(t)$. Therefore

$$p[0, 0, 1, 1] = 0. \tag{1}$$

A similar explanation applies to all the remaining transient states which are the states (0,0,1,0), (0,1,0,0), (0,1,0,1), (0,1,1,0), (0,1,1,1), (1,1,1,0), (1,1,1,1), (N,0,0,1), (N,0,1,1), (N,1,0,1) and (N,1,1,1).

The idleness failures affect the transition equations and this may be easily observed by comparing the states derived in the case where all machines have only operation-dependent failures (such a system was examined in Diamantidis et al. (2004)) with the states derived in this work. One may see that states which were transient in the system considered in Diamantidis et al. (2004) now have positive transition probability. Such a state is the (0,0,0,0), among others.

The system states that are non-transient can be divided with respect to the number of units in the system b into three sets: "*lower boundary states*", "*internal states*" and "*upper boundary states*". Due to space limitation, only a couple of sample transition equations for all three categories are given below. The remaining transition equations are given in Appendix A.

3.1. Internal state equations

A couple of sample internal state equations are given below. These equations have the following structure for probabilities $p[b, \alpha_1, \alpha_2, \alpha_3]$ of states $(b, \alpha_1, \alpha_2, \alpha_3)$, with $2 < b < N-2$, where, $N \geqslant 5$, and $\alpha_1 = \alpha_2 = 0$ and $\alpha_3 = 0, 1$

$$\begin{aligned} p[b,0,0,0] = {} & (1-r_1)(1-r_2)(1-r_3)p[b,0,0,0] + (1-r_1)(1-r_2)p_3p[b,0,0,1] \\ & + (1-r_1)p_2(1-r_3)p[b,0,1,0] + (1-r_1)p_2p_3p[b,0,1,1] + p_1(1-r_2)(1-r_3)p[b,1,0,0] \\ & + p_1(1-r_2)p_3p[b,1,0,1] + p_1p_2(1-r_3)p[b,1,1,0] + p_1p_2p_3p[b,1,1,1] \end{aligned} \tag{2}$$

$$\begin{aligned} p[b,0,0,1] = {} & (1-r_1)(1-r_2)r_3p[b+1,0,0,0] + (1-r_1)(1-r_2)(1-p_3)p[b+1,0,0,1] \\ & + (1-r_1)p_2r_3p[b+1,0,1,0] + (1-r_1)p_2(1-p_3)p[b+1,0,1,1] \\ & + p_1(1-r_2)r_3p[b+1,1,0,0] + p_1(1-r_2)(1-p_3)p[b+1,1,0,1] + p_1p_2r_3p[b+1,1,1,0] \\ & + p_1p_2(1-p_3)p[b+1,1,1,1] \end{aligned} \tag{3}$$

3.2. Lower boundary state equations

States with $b = 0$, 1, 2 are called lower boundary states. A couple of these transition equations are given below for $\alpha_1 = \alpha_2 = 0$ and $\alpha_3 = 1$.

$$\begin{aligned}
p[0,0,0,1] = {} & (1-r_1)(1-r_2)r_3p[0,0,0,0] + (1-r_1)(1-r_2)(1-v_3)p[0,0,0,1] \\
& + (1-r_1)p_2r_3p[0,0,1,0] + (1-r_1)p_2(1-v_3)p[0,0,1,1] + p_1(1-r_2)r_3p[0,1,0,0] \\
& + p_1(1-r_2)(1-v_3)p[0,1,0,1] + p_1p_2r_3p[0,1,1,0] + p_1p_2(1-v_3)p[0,1,1,1] \\
& + (1-r_1)(1-r_2)r_3p[1,0,0,0] + (1-r_1)(1-r_2)(1-p_3)p[1,0,0,1] \\
& + (1-r_1)p_2r_3p[1,0,1,0] + (1-r_1)p_2(1-p_3)p[1,0,1,1] + p_1(1-r_2)r_3p[1,1,0,0] \\
& + p_1(1-r_2)(1-p_3)p[1,1,0,1] + p_1p_2r_3p[1,1,1,0] + p_1p_2(1-p_3)p[1,1,1,1] \qquad (4)
\end{aligned}$$

$$\begin{aligned}
p[1,0,0,1] = {} & (1-r_1)(1-r_2)r_3p[2,0,0,0] + (1-r_1)(1-r_2)(1-p_3)p[2,0,0,1] \\
& + (1-r_1)p_2r_3p[2,0,1,0] + (1-r_1)p_2(1-p_3)p[2,0,1,1] + p_1(1-r_2)r_3p[2,1,0,0] \\
& + p_1(1-r_2)(1-p_3)p[2,1,0,1] + p_1p_2r_3p[2,1,1,0] + p_1p_2(1-p_3)p[2,1,1,1] \qquad (5)
\end{aligned}$$

3.3. Upper boundary state equations

States with $b = N-1$, $N-2$, N, are called upper boundary states. A couple of these transition equations are also given below for $\alpha_1 = \alpha_2 = 0, 1$ and $\alpha_3 = 0$.

$$\begin{aligned}
p[N,0,0,0] = {} & (1-r_1)(1-r_2)(1-r_3)p[N,0,0,0] + (1-r_1)(1-r_2)p_3p[N,0,0,1] \\
& + (1-r_1)v_2(1-r_3)p[N,0,1,0] + (1-r_1)v_2p_3p[N,0,1,1] + v_1(1-r_2)(1-r_3)p[N,1,0,0] \\
& + v_1(1-r_2)p_3p[N,1,0,1] + v_1v_2(1-r_3)p[N,1,1,0] + v_1v_2p_3p[N,1,1,1] \qquad (6)
\end{aligned}$$

$$\begin{aligned}
p[N-1,1,1,0] = {} & r_1r_2(1-r_3)p[N-3,0,0,0] + r_1r_2p_3p[N-3,0,0,1] \\
& + r_1(1-p_2)(1-r_3)p[N-3,0,1,0] + r_1(1-p_2)p_3p[N-3,0,1,1] \\
& + (1-p_1)r_2(1-r_3)p[N-3,1,0,0] + (1-p_1)r_2p_3p[N-3,1,0,1] \\
& + (1-p_1)(1-p_2)(1-r_3)p[N-3,1,1,0] + (1-p_1)(1-p_2)p_3p[N-3,1,1,1] \qquad (7)
\end{aligned}$$

4. Steps of the algorithm for solving the system

Although the Markovian property is lost in models with deterministic and machine specific processing times, in our case this property does hold because all three machines are assumed to have identical deterministic processing times and the failure and repair probabilities are distributed according to the memoryless geometric distribution.

To analyze the model of Fig. 1 in order to evaluate the throughput of the system, the steps given below have been followed:

Step 1: Derivation of the transition equations (internal, lower boundary and upper boundary) of the Markovian model.
Step 2: Development of a recursive algorithm for generating the transition matrix for any value N of the extended storage level ($N = C + 3$) of buffer $B_{(1,2),3}$.
Step 3: Numerical computation of the transition probabilities and the various performance measures of the system, such as throughput.

4.1. The number of states

A formula that gives the number of the non-transient states for any value $N \geqslant 5$ of the extended storage level of buffer $B_{(1,2),3}$ is derived. For each number b of units in the system there are 8 states, except when $b = 0$, $b = 1$ and $b = N$. When $b = 0$ there are 2 states, when $b = 1$ there are 6 states and when $b = N$ there are 4 states for the model under consideration.

Let $S_{j,N}$ denote the set of states when the units in the system equal j and the extended buffer capacity equals N. Then $S_{J,N}$ takes the following form for $j = 0, 1, \ldots, N$.

$$S_{0,N} = \{(0,0,0,0),(0,0,0,1)\}$$
$$S_{1,N} = \{(1,0,0,0),(1,0,0,1),(1,0,1,0),(1,0,1,1),(1,1,0,0),(1,1,0,1)\}$$
$$S_{i,N} = \{(i,0,0,0),(i,0,0,1),(i,0,1,0),(i,0,1,1),(i,1,0,0),(i,1,0,1),(i,1,1,0),(i,1,1,1)\}, \quad i = 2,\ldots,N-1$$
$$S_{N,N} = \{(N,0,0,0),(N,0,1,0),(N,1,0,0),(N,1,1,0)\}.$$

The total number of states equals to

$$S_{0,N} + S_{1,N} + S_{2,N} + \cdots + S_{N-1,N} + S_{N,N} = 2+6+8+\cdots+8+4 = 2+6+[(N-2)\times 8]+4$$
$$= 12+[(N-2)\times 8] = 12+(8\times N)-16 = (8\times N)-4.$$

Therefore for any value N of the extended buffer storage the total number of states with non-zero probability, m_N is given by

$$m_N = (8\times N) - 4. \tag{8}$$

4.2. Generation of the transition matrix for N = 5

The form of the transition matrix when the total system storage N is equal to 5 is explicitly described in this section. The transition matrix, P, is of size $[(N\times 8)-4]\times[(N\times 8)-4] = (36\times 36)$ and has the form given in Table 1.

Sub-matrices $S_{i,5}S_{j,5}$, $i,j = 0,1,2,3,4,5$, are $(k\times \ell)$ sub-matrices describing the transitions from states $\{i, a_1, a_2, a_3\}$ to states $\{j, a_1, a_2, a_3\}$, where k is equal to the number of states of set $S_{i,5}$ and ℓ is equal to the number of states of set $S_{j,5}$. For example, $S_{1,5}S_{1,5}$ is an (6×6) sub-matrix since set $S_{1,5}$ consists of 6 states, describing all the transitions from states $\{1, a_1, a_2, a_3\}$ to states $\{1, a_1, a_2, a_3\}$ and is presented in Table 2. All the rest sub-matrices $S_{i,5}S_{j,5}$, $i,j = 0,1,2,3,4,5$, are explicitly given in Appendix B.

Table 1
The transition matrix when $N = 5$

$$P = \begin{vmatrix} S_{0,5}S_{0,5} & S_{0,5}S_{1,5} & S_{0,5}S_{2,5} & S_{0,5}S_{3,5} & S_{0,5}S_{4,5} & S_{0,5}S_{5,5} \\ S_{1,5}S_{0,5} & S_{1,5}S_{1,5} & S_{1,5}S_{2,5} & S_{1,5}S_{3,5} & S_{1,5}S_{4,5} & S_{1,5}S_{5,5} \\ S_{2,5}S_{0,5} & S_{2,5}S_{1,5} & S_{2,5}S_{2,5} & S_{2,5}S_{3,5} & S_{2,5}S_{4,5} & S_{2,5}S_{5,5} \\ S_{3,5}S_{0,5} & S_{3,5}S_{1,5} & S_{3,5}S_{2,5} & S_{3,5}S_{3,5} & S_{3,5}S_{4,5} & S_{3,5}S_{5,5} \\ S_{4,5}S_{0,5} & S_{4,5}S_{1,5} & S_{4,5}S_{2,5} & S_{4,5}S_{3,5} & S_{4,5}S_{4,5} & S_{4,5}S_{5,5} \\ S_{5,5}S_{0,5} & S_{5,5}S_{1,5} & S_{5,5}S_{2,5} & S_{5,5}S_{3,5} & S_{5,5}S_{4,5} & S_{5,5}S_{5,5} \end{vmatrix}$$

Table 2
Sub-matrix $S_{1,5}S_{1,5}$

States	(1,0,0,0)	(1,0,0,1)	(1,0,1,0)	(1,0,1,1)	(1,1,0,0)	(1,1,0,1)
(1,0,0,0)	$(1-r_1)(1-r_2)(1-r_3)$	0	0	$(1-r_1)r_2r_3$	0	$r_1(1-r_2)r_3$
(1,0,0,1)	$(1-r_1)(1-r_2)p_3$	0	0	$(1-r_1)r_2(1-p_3)$	0	$r_1(1-r_2)(1-p_3)$
(1,0,1,0)	$(1-r_1)p_2(1-r_3)$	0	0	$(1-r_1)(1-p_2)r_3$	0	$r_1p_2r_3$
(1,0,1,1)	$(1-r_1)p_2p_3$	0	0	$(1-r_1)(1-p_2)(1-p_3)$	0	$r_1p_2(1-p_3)$
(1,1,0,0)	$p_1(1-r_2)(1-r_3)$	0	0	$p_1r_2r_3$	0	$(1-p_1)(1-r_2)r_3$
(1,1,0,1)	$p_1(1-r_2)p_3$	0	0	$p_1r_2(1-p_3)$	0	$(1-p_1)(1-r_2)(1-p_3)$

4.3. The algorithm for generating the transition matrix

For the three-machine, one-buffer merge system depicted in Fig. 1, a recursive algorithm that generates the transition matrix corresponding to the total system storage N from the transition matrix corresponding to the buffer level $N = 5$, for any value of N ($N > 5$) has been developed.

The algorithm takes as initial matrix the transition matrix that corresponds to the value $N = 5$ of the extended buffer level, i.e., all sub-matrices $S_{i,5}S_{j,5}$, $i, j = 0,1,2,3,4,5$, are known and used as initial values by the algorithm.

The steps of the proposed algorithm mimic the respective steps of a recursive algorithm developed by Papadopoulos and O'Kelly (1989) for generating the transition matrices of multi-station exponential series production lines. The algorithm consists of three steps, each one corresponding to the creation of the sub-matrices that contain the transition probabilities *leading to*: lower boundary states (Step 1), internal states (Step 2) and upper boundary states (Step 3), respectively and is presented in pseudo-code format in Appendix C.

4.4. Performance measures

The expected in-process inventory (average buffer level) $\overline{n}$ of the system of Fig. 1 can be written as follows (the reader is addressed to Gershwin (1994) and Helber (1999) for further reading):

$$\bar{b} = \sum_{b=0}^{N} \sum_{a_1=0}^{1} \sum_{a_2=0}^{1} \sum_{a_3=0}^{1} b\ p[b, a_1, a_2, a_3]. \tag{9}$$

The mean production rates related to each one of the three machines can be easily determined.

Recall that if p_i and r_i are the rates of failure and repair, respectively, of machine M_i, then $e_i = \frac{r_i}{r_i + p_i}$, $i = 1,2,3$, represents the fraction of time that machine M_i is operational. Since all processing times are identical and are taken as the time unit, it is obvious that e_i, $i = 1,2,3$, is the isolated mean production rate of machine M_i, i.e., the mean production rate of machine M_i, if it were never embedded by other machines or buffers.

The mean production rate of the system depicted in Fig. 1 is simply given by the mean production rate of machine M_3.

Therefore, the mean production rate of the system is

$$PR = e_3(1 - p_{s3}) + e_3 v_3 p_{s3}\left[\frac{r_3 + p_3 - 1}{r_3} - \frac{(1 - r_1)(1 - r_2)(p_3 + r_3)}{1 - (1 - r_1)(1 - r_2)(1 - r_3)}\right], \tag{10}$$

where, $p_{s3} = p[0,0,0,1]$ is the starvation probability of machine M_3 and v_3 is the idleness failure of machine M_3. The derivation of this formula is given in Appendix D.

4.5. Numerical results

The transition equations for any value of the extended buffer level N of a system with three-machines and one-buffer (depicted in Fig. 1) were solved and the steady state probabilities were computed using the Gaussian elimination method. A C++ program that generates the transition matrix and solves numerically the transition equations computing all transition probabilities was developed.

The CPU time that the numerical solution takes for each case was also obtained. For comparison purposes, a simulation model was developed in ARENA V3.0 and the simulation results were found to be close enough to those obtained from our analytical model. Statistical 95% confidence intervals were computed for any value N.

The first column in all tables gives the total system storage, N, the second column gives the throughput (mean production rate) of the system obtained by the proposed algorithm and the third column gives the estimated 95% confidence interval for the simulated throughput (mean production rate). The last column provides the CPU time (in seconds) taken by the proposed algorithm.

For the experiments presented in Tables 3 and 4, failure rates, idleness failure rates and repair rates were taken equal to $p_i = 0.01$, $v_i = 0.01$ and $r_i = 0.1$, $i = 1, 2, 3$, respectively. Since the case of all machines having identical failure and repair probabilities is not quite realistic because such a system is very balanced, three sets of experiments have been conducted in order to see how the system behaves when one of these parameters changes systematically. For the experiments shown in Table 5 (Tables 6 and 7, respectively) the failure parameter p_1 (p_2 and p_3, respectively) varies from 0.02 to 0.06 with step 0.01, while the failure parameters of the other two machines remain constant and equal to 0.01 and $r_i = 0.1$, $v_i = 0.01$, for $i = 1, 2, 3$. Similarly, for the cases

Table 3
Throughput of a system with $N = 5$, 10, 15, 20, 30, 40

Extended buffer Size (N)	Throughput obtained from the algorithm ($PR_{algorithm}$)	Confidence interval for simulated throughput ($PR_{simulated}$)	CPU time in seconds
5	0.90461	(0.90422, 0.91155)	0.0
10	0.90743	(0.90549, 0.91255)	0.0099
15	0.90848	(0.90423, 0.91160)	0.0099
20	0.90886	(0.90549, 0.91255)	0.04
30	0.90906	(0.90595, 0.91378)	0.0899
40	0.90908	(0.90595, 0.91378)	0.23

Table 4
Throughput of a system with $N = 50, 60, \ldots, 100, 150, 200, 250$

Extended buffer size (N)	Throughput obtained from the algorithm ($PR_{algorithm}$)	Confidence interval for simulated throughput ($PR_{simulated}$)	CPU time in seconds
50	0.90909000000000000	(0.90601, 0.91399)	0.460
60	0.90909080000000000	(0.90720, 0.91470)	0.780
70	0.90909090000000000	(0.90654, 0.91379)	1.301
80	0.90909093400000000	(0.90641, 0.91415)	1.826
90	0.90909093500000000	(0.90641, 0.91415)	2.613
100	0.90909093600000000	(0.90641, 0.91415)	3.574
150	0.90909093618392800	(0.90599, 0.91277)	11.915
200	0.90909093618392900	(0.90770, 0.91447)	28.418
250	0.90909093618392944	(0.90794, 0.91475)	55.454

Table 5
Throughput of a system with $p_1 = 0.02, \ldots, 0.06$, $p_2 = p_3 = v_i = 0.01$, $i = 1, 2, 3$

Extended buffer Size (N)	Throughput obtained from the algorithm ($PR_{algorithm}$)	Confidence interval for simulated throughput ($PR_{simulated}$)	p_1	CPU time in seconds
10	0.90592	(0.90109, 0.90670)	0.02	0.02
10	0.90447	(0.90202, 0.9102)	0.03	0.03
10	0.90309	(0.88854, 0.90748)	0.04	0.03
10	0.90179	(0.88684, 0.90166)	0.05	0.04
10	0.90056	(0.88270, 0.90864)	0.06	0.05

Table 6
Throughput of a system with $p_2 = 0.02, \ldots, 0.06$, $p_1 = p_3 = v_i = 0.01$, $i = 1, 2, 3$

Extended buffer size (N)	Throughput obtained from the algorithm ($PR_{algorithm}$)	Confidence interval for simulated throughput ($PR_{simulated}$)	p_2	CPU time in seconds
10	0.90649	(0.89638, 0.91138)	0.02	0.1
10	0.90554	(0.89553, 0.90863)	0.03	0.1
10	0.90460	(0.88810, 0.90641)	0.04	0.1
10	0.90367	(0.89000, 0.90440)	0.05	0.1
10	0.90270	(0.88150, 0.90270)	0.06	0.1

Table 7
Throughput of a system with $p_3 = 0.02, \ldots, 0.06$, $p_1 = p_2 = v_i = 0.01$, $i = 1,2,3$

Extended buffer Size (N)	Throughput obtained from the algorithm ($PR_{algorithm}$)	Confidence interval for simulated throughput ($PR_{simulated}$)	p_3	CPU time in seconds
20	0.83316	(0.83199, 0.84529)	0.02	0.1
20	0.76910	(0.75637, 0.forthcoming0)	0.03	0.1
20	0.71418	(0.70549, 0.72632)	0.04	0.1
20	0.66659	(0.66202, 0.68523)	0.05	0.1
20	0.62494	(0.61260, 0.63856)	0.06	0.1

Table 8
Throughput of a system with $v_1 = 0.02, \ldots, 0.06$, $p_i = v_2 = v_3 = 0.01$, $i = 1,2,3$

Extended buffer Size (N)	Throughput obtained from the algorithm ($PR_{algorithm}$)	Confidence interval for simulated throughput ($PR_{simulated}$)	v_1	CPU time in seconds
10	0.90737	(0.90549, 0.91255)	0.02	0.05
15	0.90843	(0.90423, 0.91160)	0.03	0.05
20	0.90884	(0.90549, 0.91255)	0.04	0.06
30	0.90905	(0.90595, 0.91378)	0.05	0.08
40	0.90908	(0.90595, 0.91378)	0.06	1.36

Table 9
Throughput of a system with $v_2 = 0.02, \ldots, 0.06$, $p_i = v_1 = v_3 = 0.01$, $i = 1,2,3$

Extended buffer size (N)	Throughput obtained from the algorithm ($PR_{algorithm}$)	Confidence interval for simulated throughput ($PR_{simulated}$)	v_2	CPU time in seconds
10	0.90682	(0.90549, 0.91255)	0.02	0.2
15	0.90806	(0.90423, 0.91160)	0.03	0.2
20	0.90865	(0.90549, 0.91255)	0.04	0.39
30	0.90902	(0.90595, 0.91378)	0.05	0.88
40	0.90908	(0.90595, 0.91378)	0.06	1.59

Table 10
Throughput of a system with $v_3 = 0.02\ (0.01)\ 0.06$, $p_i = v_1 = v_3 = 0.01$, $i = 1,2,3$

Extended buffer size (N)	Throughput obtained from the algorithm ($PR_{algorithm}$)	Confidence interval for simulated throughput ($PR_{simulated}$)	v_3	CPU time in seconds
10	0.90735	(0.90549, 0.91255)	0.02	0.2
15	0.90841	(0.90423, 0.91160)	0.03	0.2
20	0.90883	(0.90549, 0.91255)	0.04	0.39
30	0.909054	(0.90595, 0.91378)	0.05	0.88
40	0.909085	(0.90595, 0.91378)	0.06	1.59

presented in Table 8 (Tables 9 and 10, respectively) the idleness failure parameter v_1 (v_2 and v_3, respectively) varies from 0.02 to 0.06 with step 0.01, while the idleness failure parameters of the other two machines remain constant and equal to 0.01 and $r_i = 0.1$, $p_i = 0.01$, for $i = 1,2,3$.

The reader may see that in all cases examined the values of the throughput obtained from the proposed algorithm falls within the bounds of the simulated average throughput.

5. Conclusions and further research

A Markov process model of a three-machine, one-buffer merge manufacturing system with finite buffer capacity, discrete material, and operation-dependent as well as idleness failures was analyzed in this paper.

The contribution of this study is the exact analysis of the above system which may be used to model systems with more than part types in a decomposition approximation.

The motivation for this work has been the analysis of a real production line of a Greek company that produces brass, copper and aluminium products.

A similar three-machine, one-buffer merge system was also studied by Helber and Mehrtens (2003) and Tan (2001), but for the case of machine specific processing times, continuous flow of material and operation-dependent failures.

The transition equations of the Markovian model were derived and analytically solved and a procedure that generates the transition matrix for any value N of the (extended) storage level was developed.

The algorithm was programmed and implemented in the C++ programming language.

A simulation model was also developed in ARENA V3.0 and the simulation results matched perfectly well those obtained from the analytical model.

This study could be extended by using the proposed model as a building block in a decomposition method to evaluate performance measures of large production systems with split/merge operations, for example in flow lines with quality inspections and rework loops. Another possible extension might be the use of this new decomposition block to evaluate performance measures of production systems with different topologies (e.g., multi-station series production lines with parallel machines at each workstation, etc.).

Appendix A. The remaining lower-boundary, internal and upper-boundary state equations

Internal state equations

$$\begin{aligned}p[b,0,1,0,] = {} & (1-r_1)r_2(1-r_3)p[b-1,0,0,0] + (1-r_1)r_2p_3p[b-1,0,0,1] \\ & + (1-r_1)(1-p_2)(1-r_3)p[b-1,0,1,0] + (1-r_1)(1-p_2)p_3p[b-1,0,1,1] \\ & + p_1r_2(1-r_3)p[b-1,1,0,0] + p_1r_2p_3p[b-1,1,0,1] + p_1(1-p_2)(1-r_3)p[b-1,1,1,0] \\ & + p_1(1-p_2)p_3p[b-1,1,1,1]\end{aligned} \tag{11}$$

$$\begin{aligned}p[b,0,1,1] = {} & (1-r_1)r_2r_3p[b,0,0,0] + (1-r_1)r_2(1-p_3)p[b,0,0,1] + (1-r_1)(1-p_2)r_3p[b,0,1,0] \\ & + (1-r_1)(1-p_2)(1-p_3)p[r,0,1,1] + p_1r_2r_3p[b,1,0,0] + p_1r_2(1-p_3)p[b,1,0,1] \\ & + p_1(1-p_2)r_3p[b,1,1,0] + p_1(1-p_2)(1-p_3)p[b,1,1,1]\end{aligned} \tag{12}$$

$$\begin{aligned}p[b,1,0,0] = {} & r_1(1-r_2)(1-r_3)p[b-1,0,0,0] + r_1(1-r_2)p_3p[b-1,0,0,1] \\ & + r_1p_2(1-r_3)p[b-1,0,1,0] + r_1p_2p_3p[b-1,0,1,1] \\ & + (1-p_1)(1-r_2)(1-r_3)p[b-1,1,0,0,] + (1-p_1)(1-r_2)p_3p[b-1,1,0,1] \\ & + (1-p_1)p_2(1-r_3)p[b-1,1,1,0] + (1-p_1)p_2p_3p[b-1,1,1,1]\end{aligned} \tag{13}$$

$$\begin{aligned}p[b,1,0,1] = {} & r_1(1-r_2)r_3p[b,0,0,0] + r_1(1-r_2)(1-p_3)p[b,0,0,1] + r_1p_2r_3p[b,0,1,0] \\ & + r_1p_2(1-p_3)p[b,0,1,1] + (1-p_1)(1-r_2)r_3p[b,1,0,0] \\ & + (1-p_1)(1-r_2)(1-p_3)p[b,1,0,1] + (1-p_1)p_2r_3p[b,1,1,0] \\ & + (1-p_1)p_2(1-p_3)p[b,1,1,1]\end{aligned} \tag{14}$$

$$\begin{aligned}p[b,1,1,0] = {} & r_1r_2(1-r_3)p[b-2,0,0,0] + r_1r_2p_3p[b-2,0,0,1] + r_1(1-p_2)(1-r_3)p[b-2,0,1,0] \\ & + r_1(1-p_2)p_3p[b-2,0,1,1] + (1-p_1)r_2(1-r_3)p[b-2,1,0,0] \\ & + (1-p_1)r_2p_3p[b-2,1,0,1] + (1-p_1)(1-p_2)(1-r_3)p[b-2,1,1,0] \\ & + (1-p_1)(1-p_2)p_3p[b-2,1,1,1]\end{aligned} \tag{15}$$

$$\begin{aligned}p[b,1,1,1] = {} & r_1r_2r_3p[b-1,0,0,0] + r_1r_2(1-p_3)p[b-1,0,0,1] + r_1(1-p_2)r_3p[b-1,0,1,0] \\ & + r_1(1-p_2)(1-p_3)p[b-1,0,1,1] + (1-p_1)r_2r_3p[b-1,1,0,0] \\ & + (1-p_1)r_2(1-p_3)p[b-1,1,0,1] + (1-p_1)(1-p_2)r_3p[b-1,1,1,0] \\ & + (1-p_1)(1-p_2)(1-p_3)p[b-1,1,1,1]\end{aligned} \tag{16}$$

Lower boundary state equations

$$\begin{aligned} p[0,0,0,0] = {} & (1-r_1)(1-r_2)(1-r_3)p[0,0,0,0] + (1-r_1)(1-r_2)v_3p[0,0,0,1] \\ & + (1-r_1)p_2(1-r_3)p[0,0,1,0] + (1-r_1)p_2v_3p[0,0,1,1] + p_1(1-r_2)(1-r_3)p[0,1,0,0] \\ & + p_1(1-r_2)v_3p[0,1,0,1] + p_1p_2(1-r_3)p[0,1,1,0] + p_1p_2v_3p[0,1,1,1] \end{aligned} \tag{17}$$

$$\begin{aligned} p[1,0,0,0] = {} & (1-r_1)(1-r_2)(1-r_3)p[1,0,0,0] + (1-r_1)(1-r_2)p_3p[1,0,0,1] \\ & + (1-r_1)p_2(1-r_3)p[1,0,1,0] + (1-r_1)p_2p_3p[1,0,1,1] + p_1(1-r_2)(1-r_3)p[1,1,0,0] \\ & + p_1(1-r_2)p_3p[1,1,0,1] + p_1p_2(1-r_3)p[1,1,1,0] + p_1p_2p_3p[1,1,1,1] \end{aligned} \tag{18}$$

$$\begin{aligned} p[1,0,1,0] = {} & (1-r_1)r_2(1-r_3)p[0,0,0,0] + (1-r_1)r_2v_3p[0,0,0,1] + (1-r_1)(1-p_2)(1-r_3)p[0,0,1,0] \\ & + (1-r_1)(1-p_2)v_3p[0,0,1,1] + p_1r_2(1-r_3)p[0,1,0,0] + p_1r_2v_3p[0,1,0,1] \\ & + p_1(1-p_2)(1-r_3)p[0,1,1,0] + p_1(1-p_2)v_3p[0,1,1,1] \end{aligned} \tag{19}$$

$$\begin{aligned} p[1,0,1,1] = {} & (1-r_1)r_2r_3p[0,0,0,0] + (1-r_1)r_2(1-v_3)p[0,0,0,1] + (1-r_1)(1-p_2)r_3p[0,0,1,0] \\ & + (1-r_1)(1-p_2)(1-v_3)p[0,0,1,1] + p_1r_2r_3p[0,1,0,0] + p_1r_2(1-v_3)p[0,1,0,1] \\ & + p_1(1-p_2)r_3p[0,1,1,0] + p_1(1-p_2)(1-v_3)p[0,1,1,1] + (1-r_1)r_2r_3p[1,0,0,0] \\ & + (1-r_1)r_2(1-p_3)p[1,0,0,1] + (1-r_1)(1-p_2)r_3p[1,0,1,0] \\ & + (1-r_1)(1-p_2)(1-p_3)p[1,0,1,1] + p_1r_2r_3p[1,1,0,0] + p_1r_2(1-p_3)p[1,1,0,1] \\ & + p_1(1-p_2)r_3p[1,1,1,0] + p_1(1-p_2)(1-p_3)p[1,1,1,1] \end{aligned} \tag{20}$$

$$\begin{aligned} p[1,1,0,0] = {} & r_1(1-r_2)(1-r_3)p[0,0,0,0] + r_1(1-r_2)v_3p[0,0,0,1] + r_1p_2(1-r_3)p[0,0,1,0] \\ & + r_1p_2v_3p[0,0,1,1] + (1-p_1)(1-r_2)(1-r_3)p[0,1,0,0,] + (1-p_1)(1-r_2)v_3p[0,1,0,1] \\ & + (1-p_1)p_2(1-r_3)p[0,1,1,0] + (1-p_1)p_2v_3p[0,1,1,1] \end{aligned} \tag{21}$$

$$\begin{aligned} p[1,1,0,1] = {} & r_1(1-r_2)r_3p[0,0,0,0] + r_1(1-r_2)(1-v_3)p[0,0,0,1] + r_1p_2r_3p[0,0,1,0] \\ & + r_1p_2(1-v_3)p[0,0,1,1] + (1-p_1)(1-r_2)r_3p[0,1,0,0] \\ & + (1-p_1)(1-r_2)(1-v_3)p[0,1,0,1] + (1-p_1)p_2r_3p[0,1,1,0] \\ & + (1-p_1)p_2(1-v_3)p[0,1,1,1] + r_1(1-r_2)r_3p[1,0,0,0] + r_1(1-r_2)(1-p_3)p[1,0,0,1] \\ & + r_1p_2r_3p[1,0,1,0] + r_1p_2(1-p_3)p[1,0,1,1] + (1-p_1)(1-r_2)r_3p[1,1,0,0] \\ & + (1-p_1)(1-r_2)(1-p_3)p[1,1,0,1] + (1-p_1)p_2r_3p[1,1,1,0] \\ & + (1-p_1)p_2(1-p_3)p[1,1,1,1] \end{aligned} \tag{22}$$

The probability transition equations of states (2,0,0,0), (2,0,0,1), (2,0,1,0), (2,0,1,1), (2,1,0,0), (2,1,0,1) have exactly the same form as those of the internal states (n,0,0,0), (n,0,0,1), (n,0,1,0), (n,0,1,1), (n,1,0,0), (n,1,0,1), given above, substituting $n = 2$.

$$\begin{aligned} p[2,1,1,0] = {} & r_1r_2(1-r_3)p[0,0,0,0] + r_1r_2v_3p[0,0,0,1] + r_1(1-p_2)(1-r_3)p[0,0,1,0] \\ & + r_1(1-p_2)v_3p[0,0,1,1] + (1-p_1)r_2(1-r_3)p[0,1,0,0] + (1-p_1)r_2v_3p[0,1,0,1] \\ & + (1-p_1)(1-p_2)(1-r_3)p[0,1,1,0] + (1-p_1)(1-p_2)v_3p[0,1,1,1] \end{aligned} \tag{23}$$

$$\begin{aligned} p[2,1,1,1] = {} & r_1r_2r_3p[0,0,0,0] + r_1r_2(1-v_3)p[0,0,0,1] + r_1(1-p_2)r_3p[0,0,1,0] \\ & + r_1(1-p_2)(1-v_3)p[0,0,1,1] + (1-p_1)r_2r_3p[0,1,0,0] + (1-p_1)r_2(1-v_3)p[0,1,0,1] \\ & + (1-p_1)(1-p_2)r_3p[0,1,1,0] + (1-p_1)(1-p_2)(1-v_3)p[0,1,1,1] \end{aligned} \tag{24}$$

Upper boundary state equations

The probability transition equations of states ($N-2$,0,0,0), ($N-2$,0,1,0), ($N-2$,1,0,0), ($N-2$,1,0,1), ($N-2$,1,1,0) and ($N-2$,1,1,1) have exactly the same form as those of the internal states (n,0,0,0), (n,0,1,0), (n,1,0,0), (n,1,0,1), (n,1,1,0), (n,1,1,1), given above, substituting $n = N-2$.

$$\begin{aligned} p[N-2,0,0,1] = {} & (1-r_1)(1-r_2)r_3p[N-1,0,0,0] + (1-r_1)(1-r_2)(1-p_3)p[N-1,0,0,1] \\ & + (1-r_1)v_2r_3p[N-1,0,1,0] + (1-r_1)v_2(1-p_3)p[N-1,0,1,1] \\ & + p_1(1-r_2)r_3p[N-1,1,0,0] + p_1(1-r_2)(1-p_3)p[N-1,1,0,1] \\ & + p_1v_2r_3p[N-1,1,1,0] + p_1v_2(1-p_3)p[N-1,1,1,1] \end{aligned} \tag{25}$$

$$\begin{aligned} p[N-2,0,1,1] = {} & (1-r_1)r_2r_3p[N-2,0,0,0] + (1-r_1)r_2(1-p_3)p[N-2,0,0,1] \\ & + (1-r_1)(1-p_2)r_3p[N-2,0,1,0] + (1-r_1)(1-p_2)(1-p_3)p[N-2,0,1,1] \\ & + p_1r_2r_3p[N-2,1,0,0] + p_1r_2(1-p_3)p[N-2,1,0,1] + p_1(1-p_2)r_3p[N-2,1,1,0] \\ & + p_1(1-p_2)(1-p_3)p[N-2,1,1,1] + (1-r_1)r_2r_3p[N-1,0,0,0] \\ & + (1-r_1)r_2(1-p_3)p[N-1,0,0,1] + (1-r_1)(1-v_2)r_3p[N-1,0,1,0] \\ & + (1-r_1)(1-v_2)(1-p_3)p[N-1,0,1,1] + p_1r_2r_3p[N-1,1,0,0] \\ & + p_1r_2(1-p_3)p[N-1,1,0,1] + p_1(1-v_2)r_3p[N-1,1,1,0] \\ & + p_1(1-v_2)(1-p_3)p[N-1,1,1,1] \end{aligned} \tag{26}$$

$$\begin{aligned} p[N-1,0,0,0] = {} & (1-r_1)(1-r_2)(1-r_3)p[N-1,0,0,0] + (1-r_1)(1-r_2)p_3p[N-1,0,0,1] \\ & + (1-r_1)v_2(1-r_3)p[N-1,0,1,0] + (1-r_1)v_2p_3p[N-1,0,1,1] \\ & + p_1(1-r_2)(1-r_3)p[N-1,1,0,0] + p_1(1-r_2)p_3p[N-1,1,0,1] \\ & + p_1v_2(1-r_3)p[N-1,1,1,0] + p_1v_2p_3p[N-1,1,1,1] \end{aligned} \tag{27}$$

$$\begin{aligned} p[N-1,0,0,1] = {} & (1-r_1)(1-r_2)r_3p[N,0,0,0] + (1-r_1)(1-r_2)(1-p_3)p[N,0,0,1] \\ & + (1-r_1)v_2r_3p[N,0,1,0] + (1-r_1)v_2(1-v_3)p[N,0,1,1] \\ & + v_1(1-r_2)r_3p[N,1,0,0] + v_1(1-r_2)(1-p_3)p[N,1,0,1] + v_1v_2r_3p[N,1,1,0] \\ & + v_1v_2(1-p_3)p[n+1,1,1,1] \end{aligned} \tag{28}$$

$$\begin{aligned} p[N-1,0,1,0,] = {} & (1-r_1)r_2(1-r_3)p[N-1,0,0,0] + (1-r_1)r_2p_3p[N-1,0,0,1] \\ & + (1-r_1)(1-v_2)(1-r_3)p[N-1,0,1,0] + (1-r_1)(1-v_2)p_3p[N-1,0,1,1] \\ & + p_1r_2(1-r_3)p[N-1,1,0,0] + p_1r_2p_3p[N-1,1,0,1] \\ & + p_1(1-v_2)(1-r_3)p[N-1,1,1,0] + p_1(1-v_2)p_3p[N-1,1,1,1] \\ & + (1-r_1)r_2(1-r_3)p[N-2,0,0,0] + (1-r_1)r_2p_3p[N-2,0,0,1] \\ & + (1-r_1)(1-p_2)(1-r_3)p[N-2,0,1,0] + (1-r_1)(1-p_2)p_3p[N-2,0,1,1] \\ & + p_1r_2(1-r_3)p[N-2,1,0,0] + p_1r_2p_3p[N-2,1,0,1] \\ & + p_1(1-p_2)(1-r_3)p[N-2,1,1,0] + p_1(1-p_2)p_3p[N-2,1,1,1] \end{aligned} \tag{29}$$

$$\begin{aligned} p[N-1,0,1,1] = {} & (1-r_1)r_2r_3p[N,0,0,0] + (1-r_1)r_2(1-p_3)p[N,0,0,1] \\ & + (1-r_1)(1-v_2)r_3p[N,0,1,0] + (1-r_1)(1-v_2)(1-p_3)p[N,0,1,1] \\ & + v_1r_2r_3p[N,1,0,0] + v_1r_2(1-p_3)p[N,1,0,1] + v_1(1-v_2)r_3p[N,1,1,0] \\ & + v_1(1-v_2)(1-p_3)p[N,1,1,1] \end{aligned} \tag{30}$$

$$\begin{aligned} p[N-1,1,0,0] = {} & r_1(1-r_2)(1-r_3)p[N-2,0,0,0] + r_1(1-r_2)p_3p[N-2,0,0,1] \\ & + r_1p_2(1-r_3)p[N-2,0,1,0] + r_1p_2p_3p[N-2,0,1,1] \\ & + (1-p_1)(1-r_2)(1-r_3)p[N-2,1,0,0,] + (1-p_1)(1-r_2)p_3p[N-2,1,0,1] \\ & + (1-p_1)p_2(1-r_3)p[N-2,1,1,0] + (1-p_1)p_2p_3p[N-2,1,1,1] \end{aligned} \tag{31}$$

$$\begin{aligned}p[N-1,1,0,1] = {} & r_1(1-r_2)r_3p[N-1,0,0,0]+r_1(1-r_2)(1-p_3)p[N-1,0,0,1]\\ & +r_1v_2r_3p[N-1,0,1,0]+r_1v_2(1-p_3)p[N-1,0,1,1]\\ & +(1-p_1)(1-r_2)r_3p[N-1,1,0,0]+(1-p_1)(1-r_2)(1-p_3)p[N-1,1,0,1]\\ & +(1-p_1)v_2r_3p[N-1,1,1,0]+(1-p_1)v_2(1-p_3)p[N-1,1,1,1]\\ & +r_1(1-r_2)r_3p[N,0,0,0]+r_1(1-r_2)(1-p_3)p[N,0,0,1]+r_1v_2r_3p[N,0,1,0]\\ & +r_1v_2(1-p_3)p[N,0,1,1]+(1-v_1)(1-r_2)r_3p[N,1,0,0]\\ & +(1-v_1)(1-r_2)(1-p_3)p[N,1,0,1]+(1-v_1)v_2r_3p[N,1,1,0]\\ & +(1-v_1)v_2(1-p_3)p[N,1,1,1]\end{aligned} \tag{32}$$

$$\begin{aligned}p[N-1,1,1,1] = {} & r_1r_2r_3p[N-1,0,0,0]+r_1r_2(1-p_3)p[N-1,0,0,1]+r_1(1-v_2)r_3p[N-1,0,1,0]\\ & +r_1(1-v_2)(1-p_3)p[N-1,0,1,1]+(1-p_1)r_2r_3p[N-1,1,0,0]\\ & +(1-p_1)r_2(1-p_3)p[N-1,1,0,1]+(1-p_1)(1-v_2)r_3p[N-1,1,1,0]\\ & +(1-p_1)(1-v_2)(1-p_3)p[N-1,1,1,1]+r_1r_2r_3p[N-2,0,0,0]\\ & +r_1r_2(1-p_3)p[N-2,0,0,1]+r_1(1-p_2)r_3p[N-2,0,1,0]\\ & +r_1(1-p_2)(1-p_3)p[N-2,0,1,1]+(1-p_1)r_2r_3p[N-2,1,0,0]\\ & +(1-p_1)r_2(1-p_3)p[N-2,1,0,1]+(1-p_1)(1-p_2)r_3p[N-2,1,1,0]\\ & +(1-p_1)(1-p_2)(1-p_3)p[N-2,1,1,1]+r_1r_2r_3p[N,0,0,0]\\ & +r_1r_2(1-p_3)p[N,0,0,1]+r_1(1-v_2)r_3p[N,0,1,0]+r_1(1-v_2)(1-p_3)p[N,0,1,1]\\ & +(1-v_1)r_2r_3p[N,1,0,0]+(1-v_1)r_2(1-p_3)p[N,1,0,1]\\ & +(1-v_1)(1-v_2)r_3p[N,1,1,0]+(1-v_1)(1-v_2)(1-p_3)p[N,1,1,1]\end{aligned} \tag{33}$$

$$\begin{aligned}p[N,0,1,0] = {} & (1-r_1)r_2(1-r_3)p[N,0,0,0]+(1-r_1)r_2p_3p[N,0,0,1]\\ & +(1-r_1)(1-v_2)(1-r_3)p[N,0,1,0]+(1-r_1)(1-v_2)p_3p[N,0,1,1]\\ & +v_1r_2(1-r_3)p[N,1,0,0]+v_1r_2p_3p[n-1,1,0,1]+v_1(1-v_2)(1-r_3)p[N,1,1,0]\\ & +v_1(1-v_2)p_3p[N,1,1,1]\end{aligned} \tag{34}$$

$$\begin{aligned}p[N,1,0,0] = {} & r_1(1-r_2)(1-r_3)p[N-1,0,0,0]+r_1(1-r_2)p_3p[N-1,0,0,1]\\ & +r_1v_2(1-r_3)p[N-1,0,1,0]+r_1v_2p_3p[N-1,0,1,1]\\ & +(1-p_1)(1-r_2)(1-r_3)p[N-1,1,0,0,]+(1-p_1)(1-r_2)p_3p[N-1,1,0,1]\\ & +(1-p_1)v_2(1-r_3)p[N-1,1,1,0]+(1-p_1)v_2p_3p[N-1,1,1,1]\\ & +r_1(1-r_2)(1-r_3)p[N,0,0,0]+r_1(1-r_2)p_3p[N,0,0,1]+r_1v_2(1-r_3)p[N,0,1,0]\\ & +r_1v_2p_3p[N,0,1,1]+(1-v_1)(1-r_2)(1-r_3)p[N,1,0,0,]+(1-v_1)(1-r_2)p_3p[N,1,0,1]\\ & +(1-v_1)v_2(1-r_3)p[N,1,1,0]+(1-v_1)v_2p_3p[N,1,1,1]\end{aligned} \tag{35}$$

$$\begin{aligned}p[N,1,1,0] = {} & r_1r_2(1-r_3)p[N-2,0,0,0]+r_1r_2p_3p[N-2,0,0,1]+r_1(1-p_2)(1-r_3)p[N-2,0,1,0]\\ & +r_1(1-p_2)p_3p[N-2,0,1,1]+(1-p_1)r_2(1-r_3)p[N-2,1,0,0]\\ & +(1-p_1)r_2p_3p[N-2,1,0,1]+(1-p_1)(1-p_2)(1-r_3)p[N-2,1,1,0]\\ & +(1-p_1)(1-p_2)p_3p[N-2,1,1,1]+r_1r_2(1-r_3)p[N-1,0,0,0]\\ & +r_1r_2p_3p[N-1,0,0,1]+r_1(1-v_2)(1-r_3)p[N-1,0,1,0]\\ & +r_1(1-v_2)p_3p[N-1,0,1,1]+(1-p_1)r_2(1-r_3)p[N-1,1,0,0]\\ & +(1-p_1)r_2p_3p[N-1,1,0,1]+(1-p_1)(1-v_2)(1-r_3)p[N-1,1,1,0]\\ & +(1-p_1)(1-v_2)p_3p[N-1,1,1,1]+r_1r_2(1-r_3)p[N,0,0,0]+r_1r_2p_3p[N,0,0,1]\\ & +r_1(1-v_2)(1-r_3)p[N,0,1,0]+r_1(1-v_2)p_3p[N,0,1,1]+(1-v_1)r_2(1-r_3)p[N,1,0,0]\\ & +(1-v_1)r_2p_3p[N,1,0,1]+(1-v_1)(1-v_2)(1-r_3)p[N,1,1,0]\\ & +(1-v_1)(1-v_2)p_3p[N,1,1,1]\end{aligned} \tag{36}$$

Appendix B. Derivation of sub-matrices $S_{i,5}S_{j,5}$, $i,j = 0,1, \ldots, 5$ for $N = 5$

Sub-matrix $S_{0,5}S_{0,5}$ is a (2×2) matrix, which describes the transitions from states $\{0, a_1, a_2, a_3\}$ to states $\{0, a_1, a_2, a_3\}$, and is of the form as shown in Table 11.

Sub-matrix $S_{0,5}S_{1,5}$ is a (2×6) matrix, which describes the transitions from states $\{0, a_1, a_2, a_3\}$ to states $\{1, a_1, a_2, a_3\}$, and is of the form as shown in Table 12.

Sub-matrix $S_{0,5}S_{2,5}$ is a (2×8) matrix, which describes the transitions from states $\{0, a_1, a_2, a_3\}$ to states $\{2, a_1, a_2, a_3\}$, and is of the form as shown in Table 13.

All the elements of sub-matrices $S_{0,5}S_{j,5}$, $j = 3,4,5$ are equal to zero and thus $S_{0,5}S_{j,5} = 0$.

Sub-matrix $S_{1,5}S_{0,5}$ is a (6×2) matrix, which describes the transitions from states $\{1, a_1, a_2, a_3\}$ to states $\{0, a_1, a_2, a_3\}$, and is of the form as shown in Table 14.

Sub-matrix $S_{1,5}S_{1,5}$ is a (6×6) matrix, which describes the transitions from states $\{1, a_1, a_2, a_3\}$ to states $\{1, a_1, a_2, a_3\}$, and is of the form as shown in Table 15.

Sub-matrix $S_{1,5}S_{2,5}$ is a (6×8) matrix, which describes the transitions from states $\{1, a_1, a_2, a_3\}$ to states $\{2, a_1, a_2, a_3\}$, and is of the form as shown in Table 16.

Sub-matrix $S_{1,5}S_{3,5}$ is a (6×8) matrix, which describes the transitions from states $\{1, a_1, a_2, a_3\}$ to states $\{3, a_1, a_2, a_3\}$, and is of the form as shown in Table 17.

All the elements of sub-matrices $S_{1,5}S_{j,5}$, $j = 4,5$ are equal to zero and thus $S_{1,5}S_{j,5} = 0$.

Sub-matrix $S_{2,5}S_{1,5}$ is a (8×6) matrix, which describes the transitions from states $\{2, a_1, a_2, a_3\}$ to states $\{1, a_1, a_2, a_3\}$, and is of the form as shown in Table 18.

Sub-matrix $S_{2,5}S_{2,5}$ is a (8×8) matrix, which describes the transitions from states $\{2, a_1, a_2, a_3\}$ to states $\{2, a_1, a_2, a_3\}$, and is of the form as shown in Table 19.

Sub-matrix $S_{2,5}S_{3,5}$ is a (8×8) matrix, which describes the transitions from states $\{2, a_1, a_2, a_3\}$ to states $\{3, a_1, a_2, a_3\}$, and is of the form as shown in Table 20.

Sub-matrix $S_{2,5}S_{4,5}$ is a (8×8) matrix, which describes the transitions from states $\{2, a_1, a_2, a_3\}$ to states $\{4, a_1, a_2, a_3\}$, and is of the form as shown in Table 21.

All the elements of sub matrices $S_{2,5}S_{0,5}$ and $S_{2,5}S_{5,5}$ are equal to zero.

Sub-matrix $S_{3,5}S_{2,5}$ is a (8×8) matrix, which describes the transitions from states $\{3, a_1, a_2, a_3\}$ to states $\{2, a_1, a_2, a_3\}$, and is of the form as shown in Table 22.

Sub-matrix $S_{3,5}S_{3,5}$ is a (8×8) matrix, which describes the transitions from states $\{3, a_1, a_2, a_3\}$ to states $\{3, a_1, a_2, a_3\}$, and is of the form as shown in Table 23.

Sub-matrix $S_{3,5}S_{4,5}$ is a (8×8) matrix, which describes the transitions from states $\{3, a_1, a_2, a_3\}$ to states $\{4, a_1, a_2, a_3\}$, and is of the form as shown in Table 24.

Sub-matrix $S_{3,5}S_{5,5}$ is a (8×4) matrix, which describes the transitions from states $\{3, a_1, a_2, a_3\}$ to states $\{5, a_1, a_2, a_3\}$, and is of the form as shown in Table 25.

All the elements of sub-matrices $S_{3,5}S_{j,5}$, $j = 0,1$ are equal to zero.

Sub-matrix $S_{4,5}S_{3,5}$ is a (8×8) matrix, which describes the transitions from states $\{4, a_1, a_2, a_3\}$ to states $\{3, a_1, a_2, a_3\}$, and is of the form as shown in Table 26.

Sub-matrix $S_{4,5}S_{4,5}$ is a (8×8) matrix, which describes the transitions from states $\{4, a_1, a_2, a_3\}$ to states $\{4, a_1, a_2, a_3\}$, and is of the form as shown in Table 27.

Sub-matrix $S_{4,5}S_{5,5}$ is a (8×4) matrix, which describes the transitions from states $\{4, a_1, a_2, a_3\}$ to states $\{5, a_1, a_2, a_3\}$, and is of the form as shown in Table 28.

All the elements of sub-matrices $S_{4,5}S_{j,5}$, $j = 0,1,2$ are equal to zero.

Sub-matrix $S_{5,5}S_{4,5}$ is a (4×8) matrix, which describes the transitions from states $\{5, a_1, a_2, a_3\}$ to states $\{4, a_1, a_2, a_3\}$, and is of the form as shown in Table 29.

Sub-matrix $S_{5,5}S_{5,5}$ is a (4×4) matrix, which describes the transitions from states $\{5, a_1, a_2, a_3\}$ to states $\{5, a_1, a_2, a_3\}$, and is of the form as shown in Table 30.

All the elements of sub-matrices $S_{5,5}S_{j,5}$, $j = 0,1,2,3$ are equal to zero.

Table 11
Sub-matrix $S_{0,5}S_{0,5}$

States	(0,0,0,0)	(0,0,0,1)
(0,0,0,0)	$(1\text{-}r_1)(1\text{-}r_2)(1\text{-}r_3)$	$(1\text{-}r_1)(1\text{-}r_2)r_3$
(0,0,0,1)	$(1\text{-}r_1)(1\text{-}r_2)v_3$	$(1\text{-}r_1)(1\text{-}r_2)(1\text{-}v_3)$

Table 12
Sub-matrix $S_{0,5}S_{1,5}$

States	(1,0,0,0)	(1,0,0,1)	(1,0,1,0)	(1,0,1,1)	(1,1,0,0)	(1,1,0,1)
(0,0,0,0)	0	0	$(1\text{-}r_1)r_2(1\text{-}r_3)$	$(1\text{-}r_1)r_2r_3$	$r_1(1\text{-}r_2)(1\text{-}r_3)$	$r_1(1\text{-}r_2)r_3$
(0,0,0,1)	0	0	$(1\text{-}r_1)r_2v_3$	$(1\text{-}r_1)r_2(1\text{-}v_3)$	$r_1(1\text{-}r_2)v_3$	$r_1(1\text{-}r_2)(1\text{-}v_3)$

Table 13
Sub-matrix $S_{0,5}S_{2,5}$

States	(2,0,0,0)	(2,0,0,1)	(2,0,1,0)	(2,0,1,1)	(2,1,0,0)	(2,1,0,1)	(2,1,1,0)	(2,1,1,1)
(0,0,0,0)	0	0	0	0	0	0	$r_1r_2(1\text{-}r_3)$	$r_1r_2r_3$
(0,0,0,1)	0	0	0	0	0	0	$r_1r_2v_3$	$r_1r_2(1\text{-}v_3)$

Table 14
Sub-matrix $S_{1,5}S_{0,5}$

States	(0,0,0,0)	(0,0,0,1)
(1,0,0,0)	0	$(1\text{-}r_1)(1\text{-}r_2)r_3$
(1,0,0,1)	0	$(1\text{-}r_1)(1\text{-}r_2)(1\text{-}p_3)$
(1,0,1,0)	0	$(1\text{-}r_1)p_2r_3$
(1,0,1,1)	0	$(1\text{-}r_1)p_2(1\text{-}p_3)$
(1,1,0,0)	0	$P_1(1\text{-}r_2)r_3$
(1,1,0,1)	0	$P_1(1\text{-}r_2)(1\text{-}p_3)$

Table 15
Sub-matrix $S_{1,5}S_{1,5}$

States	(1,0,0,0)	(1,0,0,1)	(1,0,1,0)	(1,0,1,1)	(1,1,0,0)	(1,1,0,1)
(1,0,0,0)	$(1\text{-}r_1)(1\text{-}r_2)(1\text{-}r_3)$	0	0	$(1\text{-}r_1)r_2r_3$	0	$r_1(1\text{-}r_2)r_3$
(1,0,0,1)	$(1\text{-}r_1)(1\text{-}r_2)p_3$	0	0	$(1\text{-}r_1)r_2(1\text{-}p_3)$	0	$r_1(1\text{-}r_2)(1\text{-}p_3)$
(1,0,1,0)	$(1\text{-}r_1)p_2(1\text{-}r_3)$	0	0	$(1\text{-}r_1)(1\text{-}p_2)r_3$	0	$r_1p_2r_3$
(1,0,1,1)	$(1\text{-}r_1)p_2p_3$	0	0	$(1\text{-}r_1)(1\text{-}p_2)(1\text{-}p_3)$	0	$r_1p_2(1\text{-}p_3)$
(1,1,0,0)	$p_1(1\text{-}r_2)(1\text{-}r_3)$	0	0	$p_1r_2r_3$	0	$(1\text{-}p_1)(1\text{-}r_2)r_3$
(1,1,0,1)	$p_1(1\text{-}r_2)p_3$	0	0	$p_1r_2(1\text{-}p_3)$	0	$(1\text{-}p_1)(1\text{-}r_2)(1\text{-}p_3)$

Table 16
Sub-matrix $S_{1,5}S_{2,5}$

States	(2,0,0,0)	(2,0,0,1)	(2,0,1,0)	(2,0,1,1)	(2,1,0,0)	(2,1,0,1)	(2,1,1,0)	(2,1,1,1)
(1,0,0,0)	0	0	$(1-r_1)r_2(1-r_3)$	0	$r_1(1-r_2)(1-r_3)$	0	0	$r_1r_2r_3$
(1,0,0,1)	0	0	$(1-r_1)r_2p_3$	0	$r_1(1-r_2)p_3$	0	0	$r_1r_2(1-p_3)$
(1,0,1,0)	0	0	$(1-r_1)(1-p_2)(1-r_3)$	0	$r_1p_2(1-r_3)$	0	0	$r_1(1-p_2)r_3$
(1,0,1,1)	0	0	$(1-r_1)(1-p_2)p_3$	0	$r_1p_2p_3$	0	0	$r_1(1-p_2)(1-p_3)$
(1,1,0,0)	0	0	$p_1r_2(1-r_3)$	0	$(1-p_1)(1-r_2)(1-r_3)$	0	0	$(1-p_1)r_2r_3$
(1,1,0,1)	0	0	$p_1r_2p_3$	0	$(1-p_1)(1-r_2)p_3$	0	0	$(1-p_1)r_2(1-p_3)$

Table 17
Sub-matrix $S_{1,5}S_{3,5}$

States	(3,0,0,0)	(3,0,0,1)	(3,0,1,0)	(3,0,1,1)	(3,1,0,0)	(3,1,0,1)	(3,1,1,0)	(3,1,1,1)
(1,0,0,0)	0	0	0	0	0	0	$r_1r_2(1-r_3)$	0
(1,0,0,1)	0	0	0	0	0	0	$r_1r_2p_3$	0
(1,0,1,0)	0	0	0	0	0	0	$r_1(1-p_2)(1-r_3)$	0
(1,0,1,1)	0	0	0	0	0	0	$r_1(1-p_2)p_3$	0
(1,1,0,0)	0	0	0	0	0	0	$(1-p_1)r_2(1-r_3)$	0
(1,1,0,1)	0	0	0	0	0	0	$(1-p_1)r_2p_3$	0

Table 18
Sub-matrix $S_{2,5}S_{1,5}$

States	(1,0,0,0)	(1,0,0,1)	(1,0,1,0)	(1,0,1,1)	(1,1,0,0)	(1,1,0,1)
(2,0,0,0)	0	$(1-r_1)(1-r_2)r_3$	0	0	0	0
(2,0,0,1)	0	$(1-r_1)(1-r_2)(1-p_3)$	0	0	0	0
(2,0,1,0)	0	$(1-r_1)p_2r_3$	0	0	0	0
(2,0,1,1)	0	$(1-r_1)p_2(1-p_3)$	0	0	0	0
(2,1,0,0)	0	$p_1(1-r_2)r_3$	0	0	0	0
(2,1,0,1)	0	$p_1(1-r_2)(1-p_3)$	0	0	0	0
(2,1,1,0)	0	$p_1p_2r_3$	0	0	0	0
(2,1,1,1)	0	$p_1p_2(1-p_3)$	0	0	0	0

Table 19
Sub-matrix $S_{2,5}S_{2,5}$

States	(2,0,0,0)	(2,0,0,1)	(2,0,1,0)	(2,0,1,1)	(2,1,0,0)	(2,1,0,1)	(2,1,1,0)	(2,1,1,1)
(2,0,0,0)	$(1-r_1)(1-r_2)(1-r_3)$	0	0	$(1-r_1)r_2r_3$	0	$r_1(1-r_2)r_3$	0	0
(2,0,0,1)	$(1-r_1)(1-r_2)p_3$	0	0	$(1-r_1)r_2(1-p_3)$	0	$r_1(1-r_2)(1-p_3)$	0	0
(2,0,1,0)	$(1-r_1)p_2(1-r_3)$	0	0	$(1-r_1)(1-p_2)r_3$	0	$r_1p_2r_3$	0	0
(2,0,1,1)	$(1-r_1)p_2p_3$	0	0	$(1-r_1)(1-p_2)(1-p_3)$	0	$r_1p_2(1-p_3)$	0	0
(2,1,0,0)	$p_1(1-r_2)(1-r_3)$	0	0	$p_1r_2r_3$	0	$(1-p_1)(1-r_2)r_3$	0	0
(2,1,0,1)	$p_1(1-r_2)p_3$	0	0	$p_1r_2(1-p_3)$	0	$(1-p_1)(1-r_2)(1-p_3)$	0	0
(2,1,1,0)	$p_1p_2(1-r_3)$	0	0	$p_1(1-p_2)r_3$	0	$(1-p_1)p_2r_3$	0	0
(2,1,1,1)	$p_1p_2p_3$	0	0	$p_1(1-p_2)(1-p_3)$	0	$(1-p_1)p_2(1-p_3)$	0	0

Table 20
Sub-matrix $S_{2,5}S_{3,5}$

States	(3,0,0,0)	(3,0,0,1)	(3,0,1,0)	(3,0,1,1)	(3,1,0,0)	(3,1,0,1)	(3,1,1,0)	(3,1,1,1)
(2,0,0,0)	0	0	$(1-r_1)r_2(1-r_3)$	0	$r_1(1-r_2)(1-r_3)$	0	0	$r_1r_2r_3$
(2,0,0,1)	0	0	$(1-r_1)r_2p_3$	0	$r_1(1-r_2)p_3$	0	0	$r_1r_2(1-p_3)$
(2,0,1,0)	0	0	$(1-r_1)(1-p_2)(1-r_3)$	0	$r_1p_2(1-r_3)$	0	0	$r_1(1-p_2)r_3$
(2,0,1,1)	0	0	$(1-r_1)(1-p_2)p_3$	0	$r_1p_2p_3$	0	0	$r_1(1-p_2)(1-p_3)$
(2,1,0,0)	0	0	$p_1r_2(1-r_3)$	0	$(1-p_1)(1-r_2)(1-r_3)$	0	0	$(1-p_1)r_2(1-r_3)$
(2,1,0,1)	0	0	$p_1r_2p_3$	0	$(1-p_1)(1-r_2)p_3$	0	0	$(1-p_1)r_2(1-p_3)$
(2,1,1,0)	0	0	$p_1(1-p_2)(1-r_3)$	0	$(1-p_1)p_2(1-r_3)$	0	0	$(1-p_1)(1-p_2)r_3$
(2,1,1,1)	0	0	$p_1(1-p_2)p_3$	0	$(1-p_1)p_2p_3$	0	0	$(1-p_1)(1-p_2)(1-p_3)$

Table 21
Sub-matrix $S_{2,5}S_{4,5}$

States	(4,0,0,0)	(4,0,0,1)	(4,0,1,0)	(4,0,1,1)	(4,1,0,0)	(4,1,0,1)	(4,1,1,0)	(4,1,1,1)
(2,0,0,0)	0	0	0	0	0	0	$r_1r_2(1-r_3)$	0
(2,0,0,1)	0	0	0	0	0	0	$r_1r_2p_3$	0
(2,0,1,0)	0	0	0	0	0	0	$r_1(1-p_2)(1-r_3)$	0
(2,0,1,1)	0	0	0	0	0	0	$r_1(1-p_2)p_3$	0
(2,1,0,0)	0	0	0	0	0	0	$(1-p_1)r_2(1-r_3)$	0
(2,1,0,1)	0	0	0	0	0	0	$(1-p_1)r_2p_3$	0
(2,1,1,0)	0	0	0	0	0	0	$(1-p_1)(1-p_2)(1-r_3)$	0
(2,1,1,1)	0	0	0	0	0	0	$(1-p_1)(1-p_2)p_3$	0

Table 22
Sub-matrix $S_{3,5}S_{2,5}$

States	(2,0,0,0)	(2,0,0,1)	(2,0,1,0)	(2,0,1,1)	(2,1,0,0)	(2,1,0,1)	(2,1,1,0)	(2,1,1,1)
(3,0,0,0)	0	$(1-r_1)(1-r_2)r_3$	0	0	0	0	0	0
(3,0,0,1)	0	$(1-r_1)(1-r_2)(1-p_3)$	0	0	0	0	0	0
(3,0,1,0)	0	$(1-r_1)p_2r_3$	0	0	0	0	0	0
(3,0,1,1)	0	$(1-r_1)p_2(1-p_3)$	0	0	0	0	0	0
(3,1,0,0)	0	$p_1(1-r_2)r_3$	0	0	0	0	0	0
(3,1,0,1)	0	$p_1(1-r_2)(1-p_3)$	0	0	0	0	0	0
(3,1,1,0)	0	$p_1p_2r_3$	0	0	0	0	0	0
(3,1,1,1)	0	$p_1p_2(1-p_3)$	0	0	0	0	0	0

Table 23
Sub-matrix $S_{3,5}S_{3,5}$

States	(3,0,0,0)	(3,0,0,1)	(3,0,1,0)	(3,0,1,1)	(3,1,0,0)	(3,1,0,1)	(3,1,1,0)	(3,1,1,1)
(3,0,0,0)	$(1-r_1)(1-r_2)(1-r_3)$	0	0	$(1-r_1)r_2r_3$	0	$r_1(1-r_2)r_3$	0	0
(3,0,0,1)	$(1-r_1)(1-r_2)p_3$	0	0	$(1-r_1)r_2(1-p_3)$	0	$r_1(1-r_2)(1-p_3)$	0	0
(3,0,1,0)	$(1-r_1)p_2(1-r_3)$	0	0	$(1-r_1)(1-p_2)r_3$	0	$r_1p_2r_3$	0	0
(3,0,1,1)	$(1-r_1)p_2p_3$	0	0	$(1-r_1)(1-p_2)(1-p_3)$	0	$r_1p_2(1-p_3)$	0	0
(3,1,0,0)	$p_1(1-r_2)(1-r_3)$	0	0	$p_1r_2r_3$	0	$(1-p_1)(1-r_2)r_3$	0	0
(3,1,0,1)	$p_1(1-r_2)p_3$	0	0	$p_1r_2(1-p_3)$	0	$(1-p_1)(1-r_2)(1-p_3)$	0	0
(3,1,1,0)	$p_1p_2(1-r_3)$	0	0	$p_1(1-p_2)r_3$	0	$(1-p_1)p_2r_3$	0	0
(3,1,1,1)	$p_1p_2p_3$	0	0	$p_1(1-p_2)(1-p_3)$	0	$(1-p_1)p_2(1-p_3)$	0	0

Table 24
Sub-matrix $S_{3,5}S_{4,5}$

States	(4,0,0,0)	(4,0,0,1)	(4,0,1,0)	(4,0,1,1)	(4,1,0,0)	(4,1,0,1)	(4,1,1,0)	(4,1,1,1)
(3,0,0,0)	0	0	$(1-r_1)r_2(1-r_3)$	0	$r_1(1-r_2)(1-r_3)$	0	0	$r_1r_2r_3$
(3,0,0,1)	0	0	$(1-r_1)r_2p_3$	0	$r_1(1-r_2)p_3$	0	0	$r_1r_2(1-p_3)$
(3,0,1,0)	0	0	$(1-r_1)(1-p_2)(1-r_3)$	0	$r_1p_2(1-r_3)$	0	0	$r_1(1-p_2)r_3$
(3,0,1,1)	0	0	$(1-r_1)(1-p_2)p_3$	0	$r_1p_2p_3$	0	0	$r_1(1-p_2)(1-p_3)$
(3,1,0,0)	0	0	$p_1r_2(1-r_3)$	0	$(1-p_1)(1-r_2)(1-r_3)$	0	0	$(1-p_1)r_2r_3$
(3,1,0,1)	0	0	$p_1r_2p_3$	0	$(1-p_1)(1-r_2)p_3$	0	0	$(1-p_1)r_2(1-p_3)$
(3,1,1,0)	0	0	$p_1(1-p_2)(1-r_3)$	0	$(1-p_1)p_2(1-r_3)$	0	0	$(1-p_1)(1-p_2)r_3$
(3,1,1,1)	0	0	$p_1(1-p_2)p_3$	0	$(1-p_1)p_2p_3$	0	0	$(1-p_1)(1-p_2)(1-p_3)$

Table 25
Sub-matrix $S_{3,5}S_{5,5}$

States	(5,0,0,0)	(5,0,1,0)	(5,1,0,0)	(5,1,1,0)
(3,0,0,0)	0	0	0	$r_1r_2(1-r_3)$
(3,0,0,1)	0	0	0	$r_1r_2p_3$
(3,0,1,0)	0	0	0	$r_1(1-p_2)(1-r_3)$
(3,0,1,1)	0	0	0	$r_1(1-p_2)p_3$
(3,1,0,0)	0	0	0	$(1-p_1)r_2(1-r_3)$
(3,1,0,1)	0	0	0	$(1-p_1)r_2p_3$
(3,1,1,0)	0	0	0	$(1-p_1)(1-p_2)(1-r_3)$
(3,1,1,1)	0	0	0	$(1-p_1)(1-p_2)p_3$

Table 26
Sub-matrix $S_{4,5}S_{3,5}$

States	(3,0,0,0)	(3,0,0,1)	(3,0,1,0)	(3,0,1,1)	(3,1,0,0)	(3,1,0,1)	(3,1,1,0)	(3,1,1,1)
(4,0,0,0)	0	$(1-r_1)(1-r_2)r_3$	0	$(1-r_1)r_2r_3$	0	0	0	0
(4,0,0,1)	0	$(1-r_1)(1-r_2)(1-p_3)$	0	$(1-r_1)r_2(1-p_3)$	0	0	0	0
(4,0,1,0)	0	$(1-r_1)p_2r_3$	0	$(1-r_1)(1-v_2)r_3$	0	0	0	0
(4,0,1,1)	0	$(1-r_1)p_2(1-p_3)$	0	$(1-r_1)(1-v_2)(1-p_3)$	0	0	0	0
(4,1,0,0)	0	$p_1(1-r_2)r_3$	0	$p_1r_2r_3$	0	0	0	0
(4,1,0,1)	0	$p_1(1-r_2)(1-p_3)$	0	$p_1r_2(1-p_3)$	0	0	0	0
(4,1,1,0)	0	$p_1p_2r_3$	0	$p_1(1-v_2)r_3$	0	0	0	0
(4,1,1,1)	0	$p_1p_2(1-p_3)$	0	$p_1(1-v_2)(1-p_3)$	0	0	0	0

Table 27
Sub-matrix $S_{4,5}S_{4,5}$

States	(4,0,0,0)	(4,0,0,1)	(4,0,1,0)	(4,0,1,1)	(4,1,0,0)	(4,1,0,1)	(4,1,1,0)	(4,1,1,1)
(4,0,0,0)	$(1-r_1)(1-r_2)(1-r_3)$	0	$(1-r_1)r_2(1-r_3)$	0	0	$r_1(1-r_2)r_3$	0	$r_1r_2r_3$
(4,0,0,1)	$(1-r_1)(1-r_2)p_3$	0	$(1-r_1)r_2p_3$	0	0	$r_1(1-r_2)(1-p_3)$	0	$r_1r_2(1-p_3)$
(4,0,1,0)	$(1-r_1)v_2(1-r_3)$	0	$(1-r_1)(1-v_2)(1-r_3)$	0	0	$r_1v_2r_3$	0	$r_1(1-v_2)r_3$
(4,0,1,1)	$(1-r_1)v_2p_3$	0	$(1-r_1)(1-v_2)p_3$	0	0	$r_1v_2(1-p_3)$	0	$r_1(1-v_2)(1-p_3)$
(4,1,0,0)	$p_1(1-r_2)(1-r_3)$	0	$p_1r_2(1-r_3)$	0	0	$(1-p_1)(1-r_2)r_3$	0	$(1-p_1)r_2r_3$
(4,1,0,1)	$p_1(1-r_2)p_3$	0	$p_1r_2p_3$	0	0	$(1-p_1)(1-r_2)(1-p_3)$	0	$(1-p_1)r_2(1-p_3)$
(4,1,1,0)	$p_1v_2(1-r_3)$	0	$p_1(1-v_2)(1-r_3)$	0	0	$(1-p_1)v_2r_3$	0	$(1-p_1)(1-v_2)r_3$
(4,1,1,1)	$p_1v_2p_3$	0	$p_1(1-v_2)p_3$	0	0	$(1-p_1)v_2(1-p_3)$	0	$(1-p_1)(1-v_2)(1-p_3)$

Table 28
Sub-matrix $S_{4,5}S_{5,5}$

States	(5,0,0,0)	(5,0,1,0)	(5,1,0,0)	(5,1,1,0)
(4,0,0,0)	0	0	$r_1(1-r_2)(1-r_3)$	$r_1r_2(1-r_3)$
(4,0,0,1)	0	0	$r_1(1-r_2)p_3$	$r_1r_2p_3$
(4,0,1,0)	0	0	$r_1v_2(1-r_3)$	$r_1(1-v_2)(1-r_3)$
(4,0,1,1)	0	0	$r_1v_2p_3$	$r_1(1-v_2)p_3$
(4,1,0,0)	0	0	$(1-p_1)(1-r_2)(1-r_3)$	$(1-p_1)r_2(1-r_3)$
(4,1,0,1)	0	0	$(1-p_1)(1-r_2)p_3$	$(1-p_1)r_2p_3$
(4,1,1,0)	0	0	$(1-p_1)v_2(1-r_3)$	$(1-p_1)(1-v_2)(1-r_3)$
(4,1,1,1)	0	0	$(1-p_1)v_2p_3$	$(1-p_1)(1-v_2)p_3$

Table 29
Sub-matrix $S_{5,5}S_{4,5}$

States	(4,0,0,0)	(4,0,0,1)	(4,0,1,0)	(4,0,1,1)	(4,1,0,0)	(4,1,0,1)	(4,1,1,0)	(4,1,1,1)
(5,0,0,0)	0	$(1-r_1)(1-r_2)r_3$	0	$(1-r_1)r_2r_3$	0	$r_1(1-r_2)r_3$	0	$r_1r_2r_3$
(5,0,1,0)	00	$(1-r_1)v_2r_3$	0	$(1-r_1)(1-v_2)r_3$	0	$r_1v_2r_3$	0	$r_1(1-v_2)r_3$
(5,1,0,0)	0	$v_1(1-r_2)r_3$	0	$V_1r_2r_3$	0	$(1-v_1)(1-r_2)r_3$	0	$(1-v_1)r_2r_3$
(5,1,1,0)	0	$v_1v_2r_3$	0	$V_1(1-v_2)r_3$	0	$(1-v_1)v_2r_3$	0	$(1-v_1)(1-v_2)r_3$

Table 30
Sub-matrix $S_{5,5}S_{5,5}$

States	(5,0,0,0)	(5,0,1,0)	(5,1,0,0)	(5,1,1,0)
(5,0,0,0)	$(1-r_1)(1-r_2)(1-r_3)$	$(1-r_1)r_2(1-r_3)$	$r_1(1-r_2)(1-r_3)$	$r_1r_2(1-r_3)$
(5,0,1,0)	$(1-r_1)v_2(1-r_3)$	$(1-r_1)(1-v_2)(1-r_3)$	$r_1v_2(1-r_3)$	$r_1(1-v_2)(1-r_3)$
(5,1,0,0)	$v_1(1-r_2)(1-r_3)$	$v_1r_2(1-r_3)$	$(1-v_1)(1-r_2)(1-r_3)$	$(1-v_1)r_2(1-r_3)$
(5,1,1,0)	$v_1v_2(1-r_3)$	$v_1(1-v_2)(1-r_3)$	$(1-v_1)v_2(1-r_3)$	$(1-v_1)(1-v_2)(1-r_3)$

Appendix C. Pseudo-code of the algorithm

Step 1: (Creation of the transition probabilities leading to states with $b = 0, 1$) Create sub-matrices $S_{0,N}S_{i,N}$, $S_{1,N}S_{i,N}$, $0 \leqslant i \leqslant 5$, as follows:

$$S_{j,N}S_{i,N} = S_{j,5}S_{i,5} \ \forall \ N \geqslant 6, \ 0 \leqslant i \leqslant 5, \ j = 0, 1$$

$$\forall \ i \geqslant 6 \ S_{j,N}S_{i,N} = 0, N \geqslant 6 \ \mathit{and} \ j = 0, 1$$

Step 2: (Creation of the transition probabilities leading to states with $b = 2, \ldots, N-2$) $\forall \ N \geqslant 6$, $2 \leqslant K \leqslant N-2$, $M = 0, \ldots, N$:

For $K = 2$ **to** $N-2$ **do**

  For $M = 0$ **to** M = N **do**

    Create sub-matrix $S_{K,N}S_{M,N}$

Case 1:

$$K > M \quad \text{and} \quad K - M > 1 \quad \text{set} \quad S_{K,N}S_{M,N} = 0$$

Case 2:

$$K = 2 \quad \text{and} \quad M = 1 \quad \text{set} \quad S_{K,N}S_{M,N} = S_{2,5}S_{1,5}$$

Case 3:

$K > M$ and $K - M = 1$ and $K + M \neq 3$ set $S_{K,N}S_{M,N} = S_{3,5}S_{2,5}$

Case 4:

$K = M$ set $S_{K,N}S_{M,N} = S_{3,5}S_{3,5}$

Case 5:

$M > K$ and $M - K = 1$ and $N - M = 1$ set $S_{K,N}S_{M,N} = S_{3,5}S_{4,5}$

Case 6:

$M > K$ and $M - K = 1$ and $N - M > 1$ set $S_{K,N}S_{M,N} = S_{2,5}S_{3,5}$

Case 7:

$M > K$ and $M - K = 2$ and $N - M = 0$ set $S_{K,N}S_{M,N} = S_{3,5}S_{5,5}$

Case 8:

$M > K$ and $M - K = 2$ and $N - M = 1$ set $S_{K,N}S_{M,N} = S_{2,5}S_{4,5}$

Case 9:

$M > K$ and $M - K = 2$ and $N - M > 1$ set $S_{K,N}S_{M,N} = S_{2,5}S_{4,5}$

Case 10:

$M - K > 2$ set $S_{K,N}S_{M,N} = 0$

End for

End For

Step 3: (Creation of the transition probabilities leading to states with $b = N - 1, N$)

$\forall N \geqslant 6, M = 0, 1, 2, \ldots, N$:

For M=0 **to** M=N **do**Create sub-matrix $S_{N-1,N}S_{M,N}$

Case 1:

$(N - 1) - M \geqslant 2$ set $S_{N-1,N}S_{M,N} = 0$

Case 2:

$(N - 1) - M = 1$ set $S_{N-1,N}S_{M,N} = S_{4,5}S_{3,5}$

Case 3:

$N - 1 = M$ set $S_{N-1,N}S_{M,N} = S_{4,5}S_{4,5}$

Case 4:

$M > N - 1$ and $M - (N - 1) = 1$ then $S_{N-1,N}S_{M,N} = S_{4,5}S_{5,5}$

End For

For $M = 0$ **to** M = N **do**

Create sub-matrix $S_{N,N}S_{M,N}$

Case 1:

$N - M \geqslant 2$ set $S_{N,N}S_{M,N} = 0$

Case 2:

$N - M = 1$ set $S_{N,N}S_{M,N} = S_{5,5}S_{4,5}$

Case 3:

$N = M$ set $S_{N,N}S_{M,N} = S_{5,5}S_{5,5}$

End For

Appendix D. Proof of Eq. (10)

For the three-machine, one-buffer system depicted in Fig. 1 define the following quantities:

$$A_3 = \Pr[\alpha_3(t) = 1 \cap b(t) > 0],\ D_3 = \Pr[\alpha_3(t) = 0 \cap b(t) > 0],\ F_3 = \Pr[\alpha_3(t) = 0 \cap b(t) = 0] \text{ and } p_s = \Pr[\alpha_3(t) = 1 \cap b(t) = 0],$$

where p_s is the starvation probability of machine M_3.

It is obvious that

$$A_3 + D_3 + F_3 + p_s = 1. \tag{37}$$

Because for every machine failure that takes place there is a repair, the following equation holds expressing that the repair frequency equals the failure frequency which is expressed as follows:

$$r_3\Pr[\alpha_3(t) = 0 \cap b(t) > 0] + r_3\Pr[\alpha_3(t) = 0 \cap b(t) = 0] = p_3\Pr[\alpha_3(t) = 1 \cap b(t) > 0] + v_3\Pr[\alpha_3(t) = 1 \cap b(t) = 0]. \tag{38}$$

The mean production rate of the system, PR, is given by the following formula:

$$\begin{aligned} PR &= \Pr[\alpha_3(t+1) = 1 \cap b(t) > 0] = \Pr[\alpha_3(t+1) = 1/\alpha_3(t) = 1 \cap b(t) > 0]\Pr[\alpha_3(t) = 1 \cap b(t) > 0] \\ &\quad + \Pr[\alpha_3(t+1) = 1/\alpha_3(t) = 0 \cap b(t) > 0]\Pr[\alpha_3(t) = 0 \cap b(t) > 0] = (1 - p_3)\Pr[\alpha_3(t) \\ &= 1 \cap b(t) > 0] + r_3\Pr[\alpha_3(t) = 0 \cap b(t) > 0] = \Pr[\alpha_3(t) = 1 \cap b(t) > 0] - p_3\Pr[\alpha_3(t) \\ &= 1 \cap b(t) > 0] + r_3\Pr[\alpha_3(t) = 0 \cap b(t) > 0]. \end{aligned}$$

Using equation repair (37) one may obtain

$$\begin{aligned} PR &= \Pr[\alpha_3(t) = 1 \cap b(t) > 0] + v_3\ \Pr[\alpha_3(t) = 1 \cap b(t) = 0] - r_3\ \Pr[\alpha_3(t) = 0 \cap b(t) = 0] \\ &= A_3 + v_3 p_s - r_3 F_3 \Rightarrow A_3 = PR - v_3 p_s + r_3 F_3. \end{aligned} \tag{39}$$

Rewriting Eq. (38) using the above notation:

$$r_3 D_3 + r_3 F_3 = p_3 A_3 + v_3 p_s. \tag{40}$$

Substituting Eq. (39) into (40) and after some algebraic manipulation

$$D_3 = \frac{p_3}{r_3} PR + \frac{(1 - p_3)}{r_3} v_3 p_s - (1 - p_3) F_3. \tag{41}$$

Rewriting Eq. (37) yields

$$A_3 + D_3 = 1 - p_s - F_3. \tag{42}$$

By substituting Eqs. (39), (41) into (42) and using Eq. (17) and some algebraic manipulation, Eq. (10) is obtained for the mean production rate of the system depicted in Fig. 1.

References

Altiok, T. (1997). *Performance analysis of manufacturing systems*. New York: Springer.

Colledani, M., Matta, A. & Tolio, T. (2003). Performance evaluation of production lines with finite buffer capacity producing two different products, In *Proceedings of the fourth aegean international conference on "analysis of manufacturing systems"* (pp. 231—240). 1–4 July 2003, Samos Island, Greece.

Dallery, Y., David, R., & Xie, X. (1988). An efficient algorithm for the analysis of transfer lines with unreliable machines and finite buffers. *IIE Transactions, 20*(3), 280–283.

Dallery, Y., & Gershwin, S. B. (1992). Manufacturing flow line systems: a review of models and analytical results. *Queueing Systems Theory and Applications, 12*, 3–94, Special issue on queuing models of manufacturing systems.

Diamantidis, A. C., Papadopoulos, C. T., & Vidalis, M. I. (2004). Exact analysis of a discrete material three station one buffer merge system with unreliable machines. *International Journal of Production Research, 42*(4), 651–675.

Di Mascolo, M., David, R., & Dallery, Y. (1991). Modeling and analysis of assembly systems with unreliable machines and finite buffers. *IIE Transactions, 23*(4), 315–331.

Gershwin, S. B. (1987). An efficient decomposition algorithm for the approximate evaluation of tandem queues with finite storage space and blocking. *Operations Research, 35*, 291–305.

Gershwin, S. B. (1991). Assembly/disassembly systems: an efficient decomposition algorithm for tree-structured networks. *IIE Transactions, 23*(4), 302–314.

Gershwin, S. B. (1994). *Manufacturing Systems Engineering*. Englewood Cliffs: Prentice Hall.

Gershwin, S. B. Maggio, N., Matta, A., Tolio, T. & Werner, L. (2001). Analysis of Loop Networks by decomposition. In *Proceedings of the third aegean international conference on analysis and modeling of manufacturing systems* (pp. 239–248). May 19–22, Tinos Island, Greece.

Heavey, C., Papadopoulos, H. T., & Browne, J. (1993). The throughput rate of multistation unreliable production lines. *European Journal of Operational Research, 68*, 69–89.

Helber, S. (1999). Performance analysis of flow-lines with nonlinear flow of material. *Lecture Notes in Economics and Mathematical Systems* (Vol. 473). Berlin: Springer.

Helber, S., & Mehrtens, N. (2003) Exact analysis of a continuous material merge system with limited buffer capacity and three stations. In Gershwin, Dallery, Papadopoulos & MacGregor Smith (Eds.), *Analysis and modeling of manufacturing systems* (pp. 85–121). Kluwer's International Series, Kluwer Academic Publishers.

Hillier, F. S., & Boling, R. W. (1967). Finite queues in series, with exponential or Erlang service times – a numerical approach. *Operations Research, 15*, 286–303.

Jacobs, D., & Meerkov, S. M. (1995a). A system-theoretic property of serial production lines: improvability. *International Journal of Systems Science, 26*(4), 755–785.

Jacobs, D., & Meerkov, S. M. (1995b). Mathematical of improvability of production systems. *Mathematical Problems in Engineering, 1*, 95–137.

Lim, J.-T., Meerkov, S. M., & Top, F. (1990). Homogeneous, asymptotically reliable serial production lines: theory and a case study. *IEEE Transactions On Automatic Control, 35*(5), 525–533.

Nemec, Joseph E. (1999). Diffusion and Decomposition Approximations of Stochastic Models of Multiclass Processing Networks, Ph.D. Thesis, Operations Research Center, M.I.T., USA.

Papadopoulos, H. T., Heavey, C., & Browne, J. (1993). *Queueing theory in manufacturing systems analysis and design*. London: Chapman & Hall.

Papadopoulos, H. T., Heavey, C., & O'Kelly, M. E. J. (1989). The throughput rate of multistation reliable production lines with interstation buffers: (I) Exponential case. *Computers in Industry, 13*, 229–244.

Papadopoulos, H. T., & O'Kelly, M. E. J. (1989). A recursive algorithm for generating the transition matrices of multistation series production lines. *Computers in Industry, 12*, 227–240.

Syrowicz, D. (1999). Decomposition Analysis of a Deterministic, Multiple-Part-Type, Multiple-Failure-Mode Production Line, EECS, Ph.D. Thesis, Operations Research Center, MIT, USA.

Tan, B. (2001). A three station merge system with unreliable stations and a shared buffer. *Mathematical and Computer Modelling, 33*, 1011–1026.

Tan, B., & Yeralan, S. (1997). Analysis of multistation production systems with limited buffer capacity, Part II. *Mathematical and Computer Modelling, 25*(11), 109–123.

Yeralan, S., & Tan, B. (1997). Analysis of multistation production systems with limited buffer capacity, Part I. *Mathematical and Computer Modelling, 25*(7), 109–122.