Chapter 4

Modelling of Recycling Systems

4.1 Introduction

In this chapter, a new modelling approach for recycling systems is shown. Differently from the models available in the existing literature, our proposal allows to evaluate the impact of aspects that are commonly taken into account in traditional manufacturing systems, but that have seldom been a matter of concern in the field of recycling. These aspects include the processing rates of the stations in the system, which are usually different in order to increase the separation performance, failures occurring to the stations, the flow rate of the material going out of the system, which should be in a trade-off with the separation quality and, finally, buffers, which can be designed to reduce the impact of failures. Combined with the detailed process models, the system model proposed here can become an effective supporting tool for the design and the configuration of recycling systems.

In the first part of this chapter, the main assumptions and the notation of the model are introduced. Then, the main features of the analytical method used to evaluate the main system performances are described, along with its algorithmic implementation. This analytical method is based on decomposition techniques commonly used for manufacturing systems.

Then, there is a description of the main components of the discrete-event simulator that has been developed to test the accuracy of the analytical method, along with its validation. After that, some numerical tests on different system layouts are provided; comparing the results of the analytical approach and the simulation, it is shown that the developed analytical method is accurate and reliable, especially regarding the evaluation of the system throughput.

4.2 Modelling Assumptions and Notation

4.2.1 Layout

The general architecture of the recycling systems studied in this work is shown in Figure 4.1. A summary of the adopted notation is reported in Table 4.1. The system taken into account is a continuous flow network with K asynchronous stations $M_i \in M$ that are spaced out by *K-1* buffers $B_{ij} \in E$ with finite capacity N_{ij}; in this work, we'll refer to *machines* or *stations* without distinction. The incoming material enters the system in one of the *input stations* (M_1 in Figure 4.1), then it flows across its stages, which can include separation, comminution, merging and splitting stations. Finally, the end product leaves the system in the *output stations* (M_6 and M_8 in Figure 4.1). Input stations belong to the subset IN, while output stations belong to the subset OUT. Each output station will have a different concentration of the elements that form the material stream; those that contain the result of interest, i.e. feature a high concentration of target elements, constitute the OUT_T subset. We assume that input and output stations have no more than one downstream and upstream buffers, respectively.

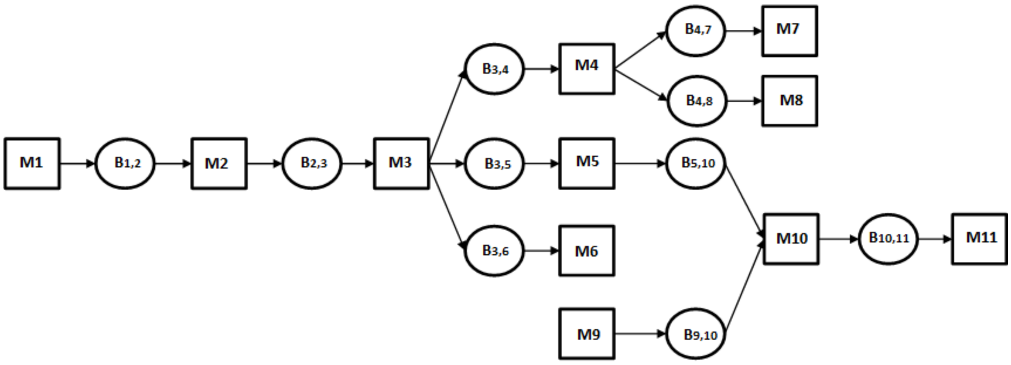


Figure 4.1: General system architecture

We also assume that each buffer B_{ij} is allowed to process only a single incoming flow from its upstream machine and to generate only a single outgoing flow towards its downstream machine. Thus, we can consider buffers as arcs of a directed graph:

$$B_{ij} = (M_i, M_j) \tag{4.1}$$

where M_i and M_j represent the upstream and downstream machines of B_{ij}, respectively.

In a similar way, stations in the system represent nodes of the underlying network, so the topology of the system can be expressed by the adjacency matrix **A** of size $K \times K$ whose a_{ud} entries are defined by:

$$a_{ud} = \begin{cases} 1 & \text{if } B_{ud} \in E \\ 0 & \text{if } \nexists\, B_{ud} \in E \end{cases} \tag{4.2}$$

From the definition of matrix **A,** it's easy to find out the sets of buffers that are directly connected with M_i. We define these sets U_i and D_i, respectively referring to the upstream and the downstream buffers that are connected with machine M_i. They can be formally expressed as follows:

$$U_i = \{B_{ki} |\, a_{ki} = 1\} \quad \forall i = 1, \ldots, K \tag{4.3}$$

$$D_i = \{B_{ik} |\, a_{ik} = 1\} \quad \forall i = 1, \ldots, K \tag{4.4}$$

The sets U_i and D_i are made of I_i and S_i elements, respectively, where

$$I_i = \sum_{k=1}^{K} a_{ki} \quad \forall i = 1, \ldots, K \tag{4.5}$$

$$S_i = \sum_{k=1}^{K} a_{ik} \quad \forall i = 1, \ldots, K \tag{4.6}$$

4.2.2 Material Characteristics

The flow of material in the system is assumed to be composed by N elements. With the term *element* we can refer to actual chemical elements, such as metals (aluminium, copper, lead, etc.), or we can refer to chemical compounds, such as polymers and plastics in general. Among these elements, we can distinguish between target and non-target materials, with the former being the components that are considered economically valuable, i.e. from which adequate revenues can be obtained after proper processing. Target elements belong to the set T.

Each of these elements has specific properties that can be exploited by separation stages in order to reach the goal of the whole process, which can be different depending on the considered system. For instance, in some cases the objective is to obtain output streams above certain levels of purity, while in other cases the recovery may be more important. Since the characteristics of the material flow are of interest when we have to evaluate the performance of these systems and since the concentration of the flow can change after visiting the stations in the system, the concentration vector $\boldsymbol{c}_{ij}$ of size N is defined for each buffer B_{ij} in the system. This vector represents the mass concentration of the material flow coming into the buffer B_{ij} from its upstream machine, so the sum of its elements must always be equal to 1. The values inside these vectors aren't known *a priori*, except for buffers that are positioned immediately after the input stations, but they can be determined by knowing how all the upstream stations treat the incoming material flows.

4.2.3 Machine Characteristics

The dynamics of each station are modelled according to a continuous-time, discrete-state Markov chain, where different processing rates are linked to each state. In states that are characterized by a processing rate equal to zero, the machine is said to be inactive, or *down*, due to a local failure, while in the other ones the machine is active, or *up*, although it may not process material due to failures that have happened on other machines. In the proposed analytical model, each machine M_i has a single up state – characterized by the processing rate μ_i – and it's subject to F_i failure modes, which depend on the type of station considered and on the technology involved.

When the station M_i is working, it can go down in failure mode $f_i = 1, \dots, F_i$ with a failure rate $p_{i,f_i} = 1/MTTF_{i,f_i}$, where $MTTF_{i,f_i}$ is the mean time between failures with the same mode f_i. Since the modelled random process is memory-less, the time to failure TTF_{i,f_i} can be considered a random variable with exponential distribution. Once the machine is down, a repairing intervention is needed in order to restart processing the incoming material flow. The repair rate linked to the specific down state is $r_{i,f_i} = 1/MTTR_{i,f_i}$, where $MTTR_{i,f_i}$ is the mean time to repair and TTR_{i,f_i} (time to repair) is exponentially distributed.

Each station works at its maximum speed unless one of the upstream buffers is empty or one of the downstream buffers is full. If the buffer between the generic stations M_k and M_i is empty, M_i is said to be:

- *partially starved*, or *slowed down*, if the current speed of the upstream machine M_k is lower than the maximum speed of the current state of M_i, but greater than zero;
- *completely starved*, if M_i is active but the current speed of M_k is zero.

On the contrary, if the same buffer is full, M_k is said to be:

- *partially blocked*, or *slowed down*, if the current speed of the downstream machine M_i is lower than the characteristic speed of the current state of M_k, but greater than zero;
- *completely blocked*, if M_k is active but the current speed of M_i is zero.

We also assume that all the considered failures are *operation dependent*, which means that failure rates are affected by the actual processing rate of the machine, which in turn is affected by the levels of the neighbouring buffers. When a machine is completely blocked or starved, it doesn't process any material, so it cannot fail; otherwise, when it is partially blocked or starved, its failure rate decreases. It is assumed that the new failure rate ψ_{i,f_i} is proportional to the original one and that their ratio is equal to the ratio between the current and the characteristic processing rates of the machine, or more formally:

$$\psi_{i,f_i} = \frac{\mu_i(t)}{\mu_i}\, p_{i,f_i} \tag{4.7}$$

It is assumed that *input stations* are never starved and that *output stations* are never blocked. If in the real system these assumptions are too strong, 'virtual' input and output stations could be included to represent the disruptions caused by what happens outside the actual system.

The assumptions proposed here are valid for every station inside the system, however, recycling systems can have different types of stations which treat the incoming material in very specific ways, as it was shown in Chapter 2. In some cases, these specific characteristics require the definition of other parameters besides processing, failure and repair rates. The features of each machine type are described in the following paragraphs.

4.2.3.1 Comminution stages

Size reduction stages break large particles into smaller ones with the aim to improve the quality of the downstream separation processes. In the proposed system model, since the particle size of the material flow isn't directly taken into account and the concentration doesn't change, these stages don't affect the characteristics of the incoming material flow. So, in this case additional parameters aren't needed.

4.2.3.2 Separators

Separation stations divide the incoming material flow into S_i output flows that have different concentrations from the input one. The fractions of material going to each downstream branch are not known *a priori* but they depend on the parameters of the separation process and on the properties of the elements in the incoming flow. For example, the magnetic separator can split ferromagnetic metals from the rest of the flow. Ideally, the separation should perfectly separate the input material in several output streams that are composed by one element – or one class of elements – each. However, due to random disturbances, non-liberated particles or impacts between particles, each output flow can contain contaminating elements. Thus, a separation process can be considered as an unreliable inspection process that is continuously performed, similarly to what happens in inspection stations in manufacturing systems [82]. So, as it has been proposed in some papers in the literature [17] [18], we can interpret this inspection as a test of hypothesis that depends on the type of separation process. For instance, for the eddy current separation, the classification criterion is the electrical conductivity of the particle and the null hypothesis is "the particle is part of the conductive flow" while the alternative hypothesis is "the particle is part of the non-conductive flow". In this case, the hypothesis is subject to two error types:

- Type I, which is the probability that a conductive particle is misclassified as non-conductive, in other words it's the probability of a *false negative*. Using the notation adopted in [17], it can be written as this conditional probability:
$$P(negative\ test \mid conductive) = 1 - r \tag{4.8}$$
- Type II, which is the probability that a non-conductive particle is misclassified as conductive, i.e. it's the probability of a *false positive*. It can be written as this conditional probability:
$$P(positive\ test \mid non\text{–}conductive) = 1 - q \tag{4.9}$$

When there is more than one conductive element, each element has a different r parameter. The same can be said for the q parameter for the non-conductive elements. These parameters can be estimated either by experimental analysis of the separation process or by using physical simulation models of the process, as the one presented in Chapter 3.

As it was already hinted in Chapter 2, the estimated parameters can be inserted in a matrix $\boldsymbol{Q}_i$ of size $N \times S_i$, which represents an additional parameter that must be defined as an input of the model. Each entry $q_i(n, s)$ of this matrix represents the mass fraction of element n that is sent to the output stream $s = 1, \dots, S_i$. Each of these flows is assigned to each buffer $B_{ik} \in D_i$ by using the routing vector $\boldsymbol{r}_i$ of size S_i: in this vector, the s^{th} entry is the index of the downstream machine where the stream determined by the s^{th} column of $\boldsymbol{Q}_i$ is routed.

By knowing the matrix $\boldsymbol{Q}_i$ and the concentration vector $\boldsymbol{c}_{ui}$ of the upstream buffer, it's possible to determine the fraction $\alpha_{r_i(s)} = \alpha_{ik}$ of the incoming material flow being directed towards each downstream buffer B_k, and their concentrations:

$$\alpha_{r_i(s)} = \sum_{n=1}^{N} q_i(n, s) \cdot c_u(n) \quad \forall s = 1, \dots, S_i \tag{4.10}$$

$$\boldsymbol{c}_{r_i(s)} = \frac{\boldsymbol{S}_{i,s} \cdot \boldsymbol{c}_{ui}}{\alpha_{r_i(s)}} \quad \forall s = 1, \dots, S_i \tag{4.11}$$

where $\boldsymbol{S}_{i,s}$ is a $N \times N$ diagonal matrix whose non-zero elements correspond to the elements of the s^{th} column of matrix $\boldsymbol{Q}_i$.

For this separation model, we assume that the entries in the matrix $\boldsymbol{Q}_i$ remain constant with the flow rate, although it isn't always true. For instance, in the eddy current separation process, the flow rate is related to the impacts between particles, so we might observe a change in the separation performance.

K	number of machines in the system
N	number of elements inside the material flow
IN	set of input stations
OUT	set of output stations
OUT_T	set of output stations with high concentrations of target elements
T	set of target materials
M_i	i^{th} machine in the system
B_{ij}	buffer between M_i and M_j
N_{ij}	capacity of B_{ij}
U_i	set of buffers upstream of machine M_i, directly connected with M_i
D_i	set of buffers downstream of machine M_i, directly connected with M_i
I_i	number of upstream buffers of the machine M_i
S_i	number of downstream buffers of the machine M_i
F_i	total number of failure modes for M_i
p_{i,f_i}	failure rate of machine M_i in mode f_i
r_{i,f_i}	repair rate of machine M_i in mode f_i
α_{ij}	fraction of material routed to B_{ij} from the machine M_i
υ_{ij}	fraction of material taken from B_{ij} by its downstream machine

Table 4.1: Notation

However, we can reasonably assume that the process parameters of each separator are set to determine a maximum processing rate that is able to guarantee the desired separation quality. Thus, since fewer impacts are expected with lower flow rates, when the separator is slowed down we are at most underestimating the quality of the separation process.

4.2.3.3 Splitters

Splitter can be modelled as a special case of separators. Since they divide the incoming flow in two or more output flows without modifying the elements' concentrations, a particular matrix $\boldsymbol{Q}_i$ can be defined, where for each column s the entries are all equal to $\alpha_{r_i(s)}$.

4.2.3.4 Merge Stations

Merge stations receive the incoming material from I_i input flows and mix them in one output stream, whose concentration is a weighted average of the concentrations of the incoming material flows. These weights are given by the column vector $\boldsymbol{U}_i$, which comprises of I_i entries υ_a whose sum must be equal to 1. Formally, the expression used to determine the concentration $\boldsymbol{c}_{id}$ directed towards the downstream buffer B_{id} is the following one:

$$\boldsymbol{c}_{id} = \boldsymbol{Z}_i \cdot \boldsymbol{U}_i \tag{4.12}$$

where $\boldsymbol{Z}_i$ is a matrix of size $N \times I_i$, whose columns are the concentration vectors of the upstream buffers. A vector $\boldsymbol{w}_i$ of size I_i is introduced to relate each weight υ_a to the corresponding upstream buffer.

In this model, the components of the vector $\boldsymbol{U}_i$ are assumed to be constant and independent from the characteristics of the incoming material flow.

4.2.3.5 Integrated Merge-Separation Stations

These machines can represent stages where there is no buffer between the merge station and the following separator. In this case, a single Markov chain is defined, which considers all the failure modes that affect both machines. For this kind of machine both matrix $\boldsymbol{Q}_i$ and vector $\boldsymbol{U}_i$ have to be given as input parameters. In this case, the new concentration obtained by merging the input flows is determined and then the outflow concentrations are evaluated for each downstream buffer.

4.3 Performance Measures

Given the system defined by the assumptions and the parameters previously introduced, the objective is to evaluate the following system performance measures:

- Total Throughput E_{ij}^{Tot}: the average rate at which material – both target and non-target – flows into the buffer B_{ij} from its upstream machine;
- Average Concentration $\bar{\boldsymbol{c}}_{ij}$: the average concentration of the mixture flowing into the buffer B_{ij} from its upstream machine;
- Effective Throughput E_{ij}^{Eff}: the average rate at which target material flows into the buffer B_{ij} from its upstream machine;
- Grade G_{ij}: the concentration of target elements flowing into the buffer B_{ij} from its upstream machine. It is equal to the ratio $\frac{E_{ij}^{Eff}}{E_{ij}^{Tot}}$;
- Recovery R_{ij}: the average fraction of input target material flowing into the buffer B_{ij} from its upstream machine. It is equal to the ratio $\frac{E_{ij}^{Eff}}{\sum_{M_k \in IN} E_{km}^{Eff}}$.

The names of these measures are chosen in order to highlight again the link between this recycling system model and the models for evaluating performances of manufacturing systems with quality control [76]. In this case, the grade corresponds to the *yield* of a manufacturing system.

These measures can be calculated for every buffer in the system, in order to understand how the material flow changes along the system. However, we're mainly interested in the end products, so we'll focus our attention on the output stations. Since we assumed that each output station has only one upstream machine, i.e. it can't be a merge station, their average concentrations $\bar{\boldsymbol{C}}_j$ will be those of their upstream buffers, thus:

$$\bar{\boldsymbol{C}}_j = \bar{\boldsymbol{c}}_{ij} \quad \forall B_{ij} \mid M_j \in OUT \tag{4.13}$$

We can observe that the average concentration vector $\bar{\boldsymbol{c}}_{ij}$ is related to the effective throughput. For the generic buffer B_{ij}, the relationship can be expressed as:

$$E_{ij}^{Eff} = E_{ij}^{Tot} \cdot \left(\sum_{e \in T} \bar{c}_{ij,e} \right) \tag{4.14}$$

If both sides of Equation 4.14 are divided by E_{ij}^{Tot}, the relationship between vector $\bar{\boldsymbol{c}}_{ij}$ and the grade G_{ij} can be obtained.

Other relations between performance measures can be derived by the *Conservation of Flow* law [83], which is the basic principle that states that the material flowing in the system cannot be neither created nor destroyed. This principle must hold for the flow of every element and for every point of observation in the system.

In the most general case, the condition can be expressed as:

$$\sum_{B_{ij} \in U_j} E_{ij}^{Tot} \cdot \bar{c}_{ij,e} = \sum_{B_{jl} \in D_j} E_{jl}^{Tot} \cdot \bar{c}_{jl,e} \quad \forall j = 1, \dots, K, \forall e = 1, \dots, N \tag{4.15}$$

When the Conservation of Flow law holds for all the elements, it can be necessarily extended to the total throughput of each machine and, then, of the overall system.

The other performance measures that we take into account are related to buffers. These measures are:

- Average Buffer Level $\bar{X}_{ij}$: the average amount of material in the buffer B_{ij};
- Work In Progress *WIP*: the average total amount of material inside the system.

Since we consider a continuous-time model, these two performances are related by the following equation:

$$WIP = \sum_{i=1}^{K-1} \bar{X}_{ij} \quad \forall B_{ij} \in E \tag{4.16}$$

4.4 Outline of the Decomposition Method

The approximate analytical method proposed here is an extension of the continuous-time, multiple failure modes decomposition technique proposed by Levantesi, Matta and Tolio [60], which was initially developed for production lines with linear layout.

The procedure allows decomposing the original system of K machines into a series of *K-1* sub-systems, called *building blocks*: for each buffer B_{ij} of the original line, a two-machine line $l(i,j)$ consisting of a buffer $B(i,j)$ of the same size of B_{ij}, an upstream pseudo-machine $M^u(i,j)$ and a downstream pseudo-machine $M^d(i,j)$ is created. These pseudo-machines mimic the behaviour of the material flow entering and leaving the buffer B_{ij} in the original system, so by reproducing in each building block a material flow that is the closest possible to the one observed in the same buffer of the system, the objective is to get a reasonable estimate of the performance of the original system. An exact analytical solution of the two-machine line has been provided in [84].

In order to replicate the dynamics of the original system, a first aspect that has to be considered is related to the fact that the flow of material into a buffer of the original system can be interrupted because the upstream machine fails in a certain mode or because it gets completely starved, owing to the emptying of one of its upstream buffers or because it gets completely blocked due to the filling of one of its downstream buffers, except the one that is being considered. Contrarily, in the two-machine line the starvation of the upstream machine cannot occur because it is the first machine of the line. Therefore, we introduce *remote failure modes* to reproduce in the two-machine line the interruptions of flow observed in the original system. Similarly, the necessity to take into account all the different causes that can determine an interruption of flow out of a buffer of the original system calls for the addition of remote failure modes also to the downstream machine of the two-machine line. So, a first problem is to find out the proper values of the parameters of the remote failure modes that have been introduced.

A second aspect is linked to the slowing down that can occur because of partial blocking and starvation phenomena, which allow the same machine to work at different processing rates. This means that the average processing rate both of real machines and pseudo-machines is different from their nominal speed. Therefore, due to the assumption of operation dependent failures made previously, we can’t set the actual failure rates of the pseudo-machines equal to the nominal failure rates of the physical machines. Hence, together with the parameters of the remote failure modes, the average processing rates and the actual failure rates of the pseudo-machines need to be found.

4.4.1 Two-machine line performance measures and notation

Before going into the details of the analytical method, it’s important and useful to understand the details of the two-machine line, and in order to do this we introduce the notation used to describe the building block. The state of the two-machine system is given by (x, α_u, α_d), where x represents the amount of material stored in the buffer $(0 \leq x \leq N)$ and α_u, α_d represent, respectively, the state of the upstream machine M^u and the downstream machine M^d. When M^u is working, $\alpha_u = 1$ while $\alpha_u = u_i$ for $i = 1, \dots, s$ means that the upstream machine is down due to failure mode *i*. Similarly, pseudo-machine M^d can get values $1, \dots, d_t$. We can classify the states of the building block in *internal states* $(0 < x < N)$ and *boundary states* $(x = 0, x = N)$. The internal states are characterized by *probability density functions* $f(x, \alpha_u, \alpha_d)$, while a *probability mass* $p(x, \alpha_u, \alpha_d)$ is associated to the boundary states.

Differently from what happens in deterministic models with synchronous machines, in the continuous two-machine line, when the upstream machine is faster than the downstream machine, the buffer can be full and both the machines can be working: as we said, in this situation the upstream machine is slowed down, so its processing rate is lower than the nominal one. So, the upstream machine efficiency W_u, which is the steady state probability that a machine is working and processing material, must include also the state probability $p(N, 1, 1)$. The expression of W^u, as provided in [60], is the following:

$$W_u = \text{prob}[\alpha_u = 1, x < N] + \frac{\mu_d}{\mu_u}\text{prob}[\alpha_u = 1, \alpha_d = 1, x = N] \qquad (4.17)$$

The correction factor μ_u/μ_d is introduced because in the state $(N, 1, 1)$, the upstream machine operates at its nominal processing rate only for that fraction of time, while remains idle for the rest of it. From the expression of the machine efficiency, the total throughput P_u of the machine can be inferred as follows:

$$P_u = \mu_u \cdot W_u \qquad (4.18)$$

In a similar fashion, when $\mu_d > \mu_u$, both the machines can operate when the buffer is empty. From this, the efficiency and the total throughput of the downstream machine are determined as follows:

$$W_d = \text{prob}[\alpha_d = 1, x > 0] + \frac{\mu_u}{\mu_d}\text{prob}[\alpha_u = 1, \alpha_d = 1, x = 0] \tag{4.19}$$

$$P_d = \mu_d \cdot W_d \tag{4.20}$$

Finally, in order to satisfy the Conservation of Flow law, the following condition must hold;

$$\mu_u \cdot W_u = \mu_d \cdot W_d \tag{4.21}$$

4.4.2 Pseudo-machine parameters

The upstream pseudo-machine $M^u(i,j)$ of the building block $l(i,j)$ features the following parameters:

- the processing rate $\mu^u(i,j)$, which represents the rate at which the material is transferred into the buffer B_{ij} by $M^u(i,j)$ when it isn't slowed down by $M^d(i,j)$;
- the failure and repair rates $p^u_{i,f_i}(i,j)$ and $r^u_{i,f_i}(i,j)$, for $f_i = 1, \dots, F_i$ of the local failure modes related to the machine M_i;
- the failure and repair rates $p^u_{t,f_t}(i,j)$ and $r^u_{t,f_t}(i,j)$, for $f_t = 1, \dots, F_t$ of the remote failure modes which model the starvation of the machine M_i, due to the failure f_t of the machine M_t.

Similarly, the downstream pseudo-machine $M^d(i,j)$ of the building block $l(i,j)$ features the following parameters:

- the processing rate $\mu^d(i,j)$, which represents the rate at which the material is taken from the buffer B_{ij} by $M^d(i,j)$ when it isn't slowed down by $M^u(i,j)$;
- the failure and repair rates $p^d_{j,f_j}(i,j)$ and $r^u_{j,f_j}(i,j)$, for $f_j = 1, \dots, F_j$ of the local failure modes related to the machine M_j;
- the failure and repair rates $p^d_{s,f_s}(i,j)$ and $r^d_{s,f_s}(i,j)$, for $f_s = 1, \dots, F_s$ of the remote failure modes which model the blocking of the machine M_j, due to the failure f_s of the machine M_s.

4.4.3 Decomposition equations

The decomposition equations are used to set the appropriate values for the unknown parameters of virtual failure modes in order to make the flow of parts in each two-machine line $l(i,j)$ as close as possible to the flow observed in the corresponding buffer of the original system. Before addressing directly the decomposition equations, it's important to understand the states where the pseudo-machines can be found and the notation used to indicate them.

More specifically, the upstream pseudo-machine $M^u(i,j)$ can be:

- processing material with probability $W^u(i,j)$, which corresponds to the efficiency of $M^u(i,j)$;
- down due to local failure f_i, with probability $D^u_{f_i}(i,j)$;
- idle due to remote failure f_t of the machine M_t, with probability $X^u_{t,f_t}(i,j)$;
- completely blocked due to the failure f_j of $M^d(i,j)$, with probability $Bl_{j,f_j}(i,j)$.

Similarly, the downstream pseudo-machine $M^d(q,i)$ can be:

- processing material with probability $W^d(q,i)$;
- down due to local failure f_j, with probability $D^d_{f_j}(q,i)$;
- idle due to remote failure f_s of the machine M_s, with probability $X^d_{s,f_s}(q,i)$;
- completely starved due to the failure mode f_i of $M^u(q,i)$, with probability $S_{i,f_i}(q,i)$.

4.4.3.1 Resumption of Flow Equations

The first set of equations is represented by the *resumption of flow* equations, which provide the repair rates of the various down states in which $M^u(i,j)$ and $M^d(q,i)$ can fail. Actually, these rates aren't unknown, either in the case of local or remote failure modes, since the repair times of the machines in the system aren't affected by the other machines' states and buffer levels. In particular, the repair rate of the remote failure mode t, f_t of a pseudo-machine is equal to the repair rate r_{f_t} of the real machine M_t from the down state f_t.

4.4.3.2 Interruption of Flow Equations

The second set of equations is represented by the *interruption of flow* equations, which provide the failure rates for each failure mode affecting the pseudo-machines $M^u(i,j)$ and $M^d(q,i)$. These equations are based on the proof given in [83] that failure frequency must equal repair frequency for each failure mode, either local or remote.

More formally, we can write for $M^u(i,j)$ the following equations:

$$W^u(i,j) \cdot p^u_{i,f_i}(i,j) = D^u_{f_i}(i,j) \cdot\ r_{i,f_i} \tag{4.22}$$

$$W^u(i,j) \cdot p^u_{t,f_t}(i,j) = X^u_{t,f_t}(i,j) \cdot\ r_{t,f_t} \tag{4.23}$$

Similarly, the following equations can be written for $M^d(q,i)$:

$$W^d(q,i) \cdot p^d_{i,f_i}(q,i) = D^d_{f_i}(q,i) \cdot\ r_{i,f_i} \tag{4.24}$$

$$W^d(q,i) \cdot p^d_{t,f_t}(q,i) = X^d_{t,f_t}(q,i) \cdot\ r_{t,f_t} \tag{4.25}$$

Local failure rates equations

Pseudo-machines $M^u(i,j)$ and $M^d(q,i)$ are associated to the same real machine M_i, so the steady state probability of both pseudo-machines being failed in a local failure mode f_i is equal:

$$D^u_{f_i}(i,j) = D^d_{f_i}(q,i) \quad f_i = 1, \dots, F_i \tag{4.26}$$

Then, substituting Equation 4.26 into Equation 4.24, we obtain:

$$p^u_{i,f_i}(i,j) = \frac{D^d_{f_i}(q,i) \cdot\ r_{i,f_i}}{W^u(i,j)} \quad f_i = 1, \dots, F_i \tag{4.27}$$

Similarly, substituting Equation 4.26 into Equation 4.25, we obtain:

$$p^d_{i,f_i}(q,i) = \frac{D^u_{f_i}(q,i) \cdot\ r_{i,f_i}}{W^d(i,j)} \quad f_i = 1, \dots, F_i \tag{4.28}$$

Remote failure rates equations

Remote failure modes of the pseudo-machines have been introduced to take into account the different causes that can bring to an interruption of flow in the system. Thus, the probabilities of the states $X^u_{t,f_t}(i,j)$ of the pseudo-machine $M^u(i,j)$ must be equal either to the probabilities of starvation of the building blocks immediately upstream of M_i or to the probabilities of blocking of the two-machine lines immediately downstream of M_i, except $l(i,j)$:

$$X^u_{t,f_t}(i,j) = S_{t,f_t}(u,i) \quad \forall(u,i)|B_{ui} \in U_i \tag{4.29}$$

$$X^u_{t,f_t}(i,j) = Bl_{t,f_t}(i,z) \quad \forall(i,z)|B_{iz} \in (D_i - B_{ij}) \tag{4.30}$$

Similarly, the probabilities of the states $X^d_{s,f_s}(q,i)$ of the pseudo-machine $M^d(q,i)$ must be equal either to the probabilities of blocking of the building blocks immediately downstream of M_i or to the probabilities of starvation of the two-machine lines immediately upstream of M_i, except $l(q,i)$:

$$X^d_{s,f_s}(q,i) = Bl_{s,f_s}(i,z) \quad \forall(i,z)|B_{iz} \in D_i \tag{4.31}$$

$$X^d_{s,f_s}(q,i) = S_{s,f_s}(u,i) \quad \forall(u,i)|B_{ui} \in (U_i - B_{qi}) \tag{4.32}$$

4.4.3.3 Processing Rate Equations

The third and final set of equations is related to the processing rates of the pseudo-machines $M^u(i,j)$ and $M^d(q,i)$. Since the real machines are affected by slowing down phenomena, their processing rates aren't equal to their nominal values. The same is valid for the pseudo-machines. For the conservation of flow, the amount of material transferred to $B(i,j)$ must be equal to the fraction α_{ij} of the material flow coming from $M^u(i,j)$. In turn, the material flow reaching $M^u(i,j)$ must be equal to the sum of the flows of all the building blocks $l(u,i)|B_{ui} \in U_i$. The equivalent formal expression is:

$$E(i,j) = \mu^u(i,j) \cdot W^u(i,j) = a_{ij} \cdot \sum_{B_{ui} \in U_i} E(u,i) \tag{4.33}$$

From this expression, it's easy to derive the expression for the processing rate of $M^u(i,j)$:

$$\mu^u(i,j) = \frac{a_{ij} \cdot \sum_{B_{ui} \in U_i} E(u,i)}{W^u(i,j)} \tag{4.34}$$

Similarly, the amount of material coming out of $B(q,i)$ must be equal to the fraction υ_{qi} of the material flow coming to $M^d(q,i)$, which must be equal to the sum of the flows of all the building blocks $l(i,z)|B_{iz} \in D_i$. The expression is the following one:

$$E(q,i) = \mu^d(q,i) \cdot W^d(q,i) = \upsilon_{qi} \cdot \sum_{B_{iz} \in D_i} E(i,z) \tag{4.35}$$

From that, the following expression can be derived for the processing rate of $M^d(q,i)$:

$$\mu^d(q,i) = \frac{\upsilon_{qi} \cdot \sum_{B_{iz} \in D_i} E(i,z)}{W^d(q,i)} \tag{4.36}$$

4.4.4 Algorithm

A modified version of the algorithm proposed by Dallery et al. [55], which is commonly used to solve the decomposition equations, has been developed. Changes have been introduced to treat systems with multiple branches and multi-element flows. The algorithm comprises the following steps:

1. *Initialization*
2. *Evaluation Sequence Generation*
3. *Update Cycle*
4. *Convergence Check*

4.4.4.1 Initialization

For each building block in which the original system is decomposed, the known parameters are provided while the unknown parameters are initialized in the following way:

$$\mu^u(i,j) = \mu_i \tag{4.37}$$

$$p^u_{i,f_i}(i,j) = p_{i,f_i} \quad f_i = 1, \dots, F_i \tag{4.38}$$

$$p^u_{t,f_t}(i,j) = 0.01 \quad f_t = 1, \dots, F_t \tag{4.39}$$

$$\mu^d(q,i) = \mu_i \tag{4.40}$$

$$p^d_{i,f_i}(q,i) = p_{i,f_i} \quad f_i = 1, \dots, F_i \tag{4.41}$$

$$p^d_{s,f_s}(q,i) = 0.01 \quad f_s = 1, \dots, F_s \tag{4.42}$$

Besides these factors, additional parameters are provided to take into account the specific features of recycling systems, in particular:

- the separation matrices $\boldsymbol{Q}_i$ and the vectors $\boldsymbol{U}_k$ for all the machines M_i and M_j that act as separation and mixing stages, respectively;
- the composition vectors $\boldsymbol{c}_{ij}$ for all the building blocks whose upstream machine is an input station.

Since rework loops aren’t taken into account, from these additional parameters it is possible to determine:

- the fractions α_{ij} and υ_{ij} for each building block. When the upstream machine M_i is neither a separator nor a splitter, then $\alpha_{ij} = 1$. When the original downstream machine $M_j$is not a mixer, $\upsilon_{ij} = 1$;
- the average Concentration $\bar{\boldsymbol{c}}_{ij}$ of all the building blocks.

4.4.4.2 Evaluation Sequence Generation

Before proceeding with the Update Cycle, an evaluation sequence must be defined in order to make the algorithm properly work. The first pseudo-machines that are included in the sequence are those whose parameters are all known. These machines correspond to the input and output stations of the original system, since they aren't affected by any remote failure. Then, the other pseudo-machines are iteratively included in the sequence only if all of their parameters can be derived from the ones of the pseudo-machines that have already been put inside the sequence, by using the previously shown decomposition equations.

4.4.4.3 Update Cycle

Following the previously generated evaluation sequence, the parameters of each two-machine line are updated.

When the upstream pseudo-machine $M^u(i,j)$ is considered, its parameters are updated making use of the following formulas:

– *processing rate* $\mu^u(i,j)$ is given by:

$$\mu^u(i,j) = \frac{a_{ij} \cdot \sum_{B_{ui} \in U_i} E(u,i)}{W^u(i,j)} \tag{4.43}$$

– *local failure rates* $p^u_{i,f_i}(i,j)$ are determined with the results of the building block $l(q,i) \in U_i$ that has been evaluated last:

$$p^u_{i,f_i}(i,j) = \frac{D^d_{f_i}(q,i) \cdot r_{i,f_i}}{W^u(i,j)} \quad f_i = 1, \ldots, F_i \tag{4.44}$$

– *remote failure rates* $p^u_{t,f_t}(i,j)$ can be caused both by starvation and blocking of $M^u(i,j)$, so we obtain:

$$p^u_{t,f_t}(i,j) = \frac{S_{t,f_t}(u,i) \cdot r_{i,f_i}}{W^u(i,j)} \quad f_t = 1, \ldots, F_t, \quad \forall (u,i) | B_{ui} \in U_i \tag{4.45}$$

$$p^u_{t,f_t}(i,j) = \frac{Bl_{t,f_t}(i,z) \cdot r_{i,f_i}}{W^u(i,j)} \quad f_t = 1, \dots, F_t, \quad \forall (i,z) | B_{iz} \in (D_i - B_{ij}) \tag{4.46}$$

Similarly, for $M^d(i,j)$ the considered equations are:

- *processing rate* $\mu^d(q,i)$ is given by:

$$\mu^u(i,j) = \frac{\upsilon_{qi} \cdot \sum_{B_{iz} \in D_i} E(i,z)}{W^d(q,i)} \tag{4.47}$$

- *local failure rates* $p^d_{i,f_i}(q,i)$ are determined with the results of the building block $l(i,j) \in D_i$ that has been evaluated last:

$$p^d_{i,f_i}(q,i) = \frac{D^u_{f_i}(i,j) \cdot r_{i,f_i}}{W^d(q,i)} \quad f_i = 1, \dots, F_i \tag{4.48}$$

- *remote failure rates* $p^d_{s,f_s}(q,i)$ can be caused by both starvation and blocking of $M^u(i,j)$, so we obtain:

$$p^d_{s,f_s}(q,i) = \frac{Bl_{s,f_s}(i,z) \cdot r_{i,f_i}}{W^d(q,i)} \quad f_s = 1, \dots, F_s, \quad \forall (i,z) | B_{iz} \in D_i \tag{4.49}$$

$$p^d_{s,f_s}(q,i) = \frac{S_{s,f_s}(u,i) \cdot r_{i,f_i}}{W^d(q,i)} \quad f_s = 1, \dots, F_s, \quad \forall (u,i) | B_{ui} \in (U_i - B_{qi}) \tag{4.50}$$

4.4.4.4 Convergence Check

The Update Cycle is repeated until the equations converge. The convergence check is performed by controlling whether the conservation of the total flow is satisfied for each machine M_i in the system. The tolerance parameter ε has been set equal to 10^{-5} in the cases that will be presented in this work.

4.5 Discrete-Event Simulation Model

In order to validate the results obtained with the analytical method, a discrete-event simulator has been created in MATLAB. Systems that have been modelled with the simulator work according to the same assumptions of the analytical method proposed, so we expect to obtain similar results with these two methods.

With the simulation we have a tool for generating sample paths of stochastic discrete-event systems. Each simulation run works up to a certain number of simulated events that is sufficient to make performance measures converge.

4.5.1 Structure Description

For each simulation run, 3 macro steps are followed:

1. *Initialization;*
2. *Main Simulation Cycle;*
3. *Performance Evaluation.*

Within the Main Simulation Cycle, the following routines are repeated until the stopping condition is satisfied:

1. *Concentration Update;*
2. *Processing Rates Update*;
3. *Scheduled Event List Generation;*
4. *System State Update;*
5. *Performance Variables Update.*

Each of these parts is briefly described in the following paragraphs.

4.5.1.1 Initialization

In this step, the simulation clock is set to 0 and the state of each system component – buffers and machines – is set to some value. In particular, it can be decided whether the machines are up or in one of their down states, while for buffers we can decide their starting levels. In this part, the stream of random numbers to be used by the Random Number Generator is defined as well.

4.5.1.2 Main Simulation Cycle

This is the core of the whole program and it's responsible for modelling the system evolution event by event.

Concentration Update

In this step, the concentration vectors of the material flows coming into each buffer are updated. Actually, when the material supplied in the input station has a constant concentration or there aren't any rework loops inside the system, the concentration inside each buffer is always the same.

Processing Rates Update

Here the current processing rates $\mu_{i,curr}$ of the machines are determined by taking into account blocking and starvation phenomena.

Scheduled Event List Generation

This is the section that is responsible for the correct progress of time in the simulation. In our case, events can be related both to buffers and machines.

Regarding buffers, if the current flow rate of its upstream machine is greater than the one of the downstream, the buffer can become full, so a "filling" event is inserted in the Scheduled Event List; on the contrary, when the upstream machine is slower, the buffer can become empty, so an "emptying" event is added. In these cases, since the buffer level and the filling and emptying rates, given by the difference of the machines' flow rates, are known, the event time is deterministic.

Machine events are quite more complex to model due to the presence of stochastic events and the assumption of operation dependent failures. When a machine has just become active after a repair event, for each failure mode a new random time to failure (TTF) is generated from an exponential distribution with mean equal to their MTTF. Then, from this number the total amount of material that can be processed before failure is obtained by multiplying this TTF by the maximum processing rate of the station. This quantity, defined as the "operations to failure" (OTF), is needed to handle properly the assumption of operation dependent failures.

If the machine is active and if it is neither completely blocked nor completely starved, all the failure events are feasible and the previous number of OTF is reduced by a quantity that depends on the actual flow rate of the machine in the previous time interval, and the length of that interval. In this way, by dividing the number of available OTF by the current processing rate of the machine, the updated time to failure can be determined. If the machine is active but it's completely blocked or starved, only the OTF is updated, but no event is inserted in the Scheduled Event List.

When the machine is down, only repair events can happen, and since they aren't affected by the assumption of operation dependence, a time to repair (TTR) is generated from an exponential distribution. Differently from failures, the repair event time is constant and doesn't need to be updated, so the old value that was inside the previous Scheduled Event List is simply copied into the new one.

State Update

In this section, the events in the new Scheduled Event List are sorted according to a *smallest-scheduled-time-first* scheme. The earliest event in the list is extracted from it, while the remaining events are copied in an Old Scheduled Event List used to check machine events in the "Scheduled Event List Generation" step of the next iteration.

If the new event is a machine event, the current flow rate of the machine is set equal to its characteristic speed. If the machine has broken down, the number of "operations to failure" is set to zero for each failure event, so a new random time to failure will be generated in the "Scheduled Event List Generation" after the station is repaired.

If the new event is related to a buffer, no particular action is performed, since in every case the buffer levels are updated.

Performance Variables Update

After the system state has been updated, some performance variables are updated. In particular, the recorded performances are:

- the total time that each machine has spent in each state. The actual flow rate in that state is recorded as well, in order to verify the impact of blocking and starvation phenomena;
- the cumulated material flow in each station, for each element;
- the cumulated amount of material that has been stored in each buffer.

4.5.1.3 Performance Evaluation

After the end of the Main Simulation Cycle, the performance measures of interest are determined from the performance variables that have been recorded event by event, simply by dividing the values of those variables by the total simulated time. If more than one simulation run has been performed, confidence intervals for these performance measures can be calculated, too.

4.5.2 Example

Let us now proceed with a simulation of the 4-machine recycling system shown below. Here the different parts of the discrete-event simulator are thoroughly described to give a better idea about how the algorithm actually works. In this system, the machine M_2 is a separator, thus a separation matrix $\boldsymbol{Q}_2$ has to be introduced, while the incoming material is made by two elements, A and B (A is the target one). belowThe input data are shown in Table 4.2, Table 4.3, Table 4.4 and Table 4.5.

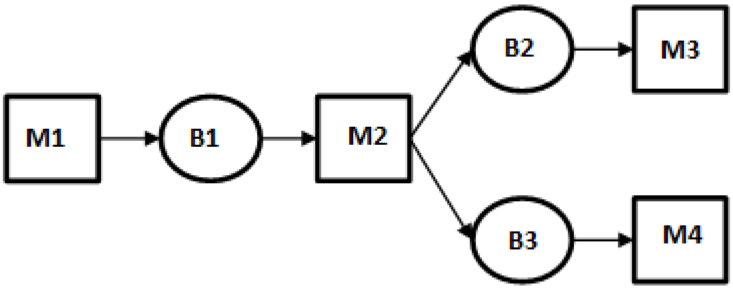


Figure 4.2: Layout considered in the example

Elements	*Concentration*
A	20%
B	80%

Table 4.2 Concentration of the Input Flow

Machines	μ_i	$p_{i,1}$	$r_{i,1}$
1	1	0.03	0.101
2	1.3	0.03	0.101
3	0.7	0.01	0.08
4	0.5	0.01	0.08

Table 4.3 Processing, Failure and Repair Rates of the Machines

Buffers	***Upstream Machine***	***Downstream Machine***	***Capacity***
1	1	2	5
2	2	3	4
3	2	4	4

Table 4.4 Buffer sizes

	M3	**M4**
A	0.95	0.05
B	0.3	0.7

Table 4.5 Separation Matrix of Machine 2

In what follows, we describe what happens in each macro step of the simulation and, in particular, we analyse in depth an iteration of the Main Simulation Cycle. More specifically, we show how the processing rates of the machines are determined, how the events are inserted and updated in the Scheduled Event List and how the variables required for estimating performance measures are recorded.

4.5.2.1 Initialization

The system state is initialized as shown in these tables:

Machine	***State***
1	UP
2	UP
3	UP
4	UP

Buffer	***Level***
1	0
2	0
3	0

The clock is set to $\boldsymbol{t_0 = 0}$.

4.5.2.2 Main Simulation Cycle

Concentration Update

The concentrations of the material flows are determined. The results are shown in the following table:

Buffer	***A***	***B***
1	20%	80%
2	44.19%	55.81%
3	1.75%	98.25%

These concentrations will be the same after every event, because of the lack of rework loops. For what regards the separator, the fractions of the input flow directed towards M_3 and M_4 are 43% and 57%, respectively.

Processing Rates Update

Since every buffer is empty, all the machines except the first one can be partially starved. In order to identify how the performance of each machine is affected by the other ones, the dependency matrix $\boldsymbol{D}$ of size $K \times K$ is created and its entries $D_{i,j}$ are initialized as follows:

$$D_{i,j} = \begin{cases} \min\left(\mu_i, \dfrac{\upsilon_n \mu_j}{\alpha_n}\right) & \text{if } a_{ij} = 1 \land X_n = N_n \\ \min\left(\mu_i, \dfrac{\alpha_n \mu_j}{\upsilon_n}\right) & \text{if } a_{ji} = 1 \land X_n = 0 \\ -1 & \text{else} \end{cases} \tag{4.51}$$

The entries $D_{i,j} = -1$ are provided to mark the independence of M_i from M_j.

Here we show the considerations that are made to initialize the dependency matrix:

- Machine M_1 can't be affected by the other machines in the system, since its downstream buffer isn't full, so we set $D_{1,j} = -1\ \forall j$. We can already set $\mu_{1,curr} = 1$;

- The upstream buffer of M_2 is empty, so it can be affected by M_1: $D_{2,1} = \min\left(1.3, \frac{1 \cdot 1}{1}\right) = 1$;
- Machine M_3 can be affected by its upstream machine M_2: $D_{3,2} = \min\left(0.7, \frac{0.43 \cdot 1.5}{1}\right) = 0.645$;
- Machine M_4 may depend on its upstream machine M_2: $D_{4,2} = \min\left(0.5, \frac{0.57 \cdot 1.5}{1}\right) = 0.5$.

We obtain the following matrix ***D***:

	M1	M2	M3	M4
M1	-1	-1	-1	-1
M2	1	-1	-1	-1
M3	-1	0.645	-1	-1
M4	-1	0.5	-1	-1

After the initialization, we don't take into account the independent machines (M_1) anymore and proceed with the following iterative steps:

1. Machine M_2 depends just on M_1. Since M_1 is independent, we can set $\mu_{2,curr} = \min(D_{2,1}, \mu_{1,curr}) = 1$. Then, we set $D_{2,1} = -1$ so M_2 will appear as an 'independent' machine from now on;
2. Machine M_3 depends solely on M_2, which has become 'independent'. As it has been done in the previous step, we set $\mu_{3,curr} = \min(D_{3,2}, 0.43 \cdot \mu_{2,curr}) = 0.43$ and then $D_{3,2} = -1$;
3. In a similar way, for M_4 we set $\mu_{4,curr} = \min(D_{4,2}, 0.57 \cdot \mu_{2,curr}) = 0.5$ and $D_{4,2} = -1$;

These steps are repeated until the matrix ***D*** is made just of '-1' entries and all the values of the current flow rates are correctly set. At the end of this section, the resulting current processing rates are:

	$\mu_{i,curr}$
M1	1
M2	1
M3	0.43
M4	0.5

Scheduled Event List Generation

As it was previously said, buffer-related events are deterministic, and can be calculated this way:

Buffer	Incoming Flow rate (IN)	Outcoming Flow Rate (OUT)	IN - OUT	Feasible Event	Event Time
1	1	1	0	/	/
2	0.43	0.43	0	/	/
3	0.57	0.5	0.07	FILLING	57.1429

The only buffer that can change its state is B_3. The event time t_e is determined by:

$$t_e = t_0 + \frac{N_3}{\alpha_3\mu_3 - v_3\mu_4} = 0 + \frac{4}{0.07} = 57.1429 \qquad (4.52)$$

Regarding the machines, since they are all active and there aren't any cases of complete starvation or blocking, they can fail. Since this is the first simulated event, new random numbers have to be generated. In order to see how the assumption of operation dependent failures is considered in the simulator, we take a look at machine M_2, which is partially starved.

First of all, a random number u is generated from a uniform distribution in the interval [0,1]. In our example, the value $u = 0.5998$ is generated for M_2. In order to convert this value in a random variate X from an exponential distribution $F(x) = 1 - e^{-\lambda x}, x \geq 0$, we set:

$$u = 1 - e^{-\lambda X} \qquad (4.53)$$

Therefore, solving for X, we obtain:

$$X = \frac{-1}{\lambda}\ln(1-u) = \frac{-1}{0.101}\ln(1-0.5998) = 30.5263 \tag{4.54}$$

In our case, X represents the potential time to failure calculated for the machine M_2; notice that $\lambda = p_{2,1}$ and that $1/\lambda = MTTF_{2,1}$.

Since the machine is partially starved, its actual time to failure $TTF_{2,1}$ will be longer than X. In order to compute the corrected "time to failure", we determine the number of "operations to failure" $OTF_{2,1}$ and then divide this number by the current processing rate of M_2:

$$OTF_{2,1} = X \cdot \mu_2 = 39.6855 \tag{4.55}$$

$$TTF_{2,1} = \frac{OTF_{2,1}}{\mu_{2,curr}} = 39.6855 \tag{4.56}$$

We proceed in the same way for the other machines in the system and obtain the following Scheduled Event List:

Entity	***Event***	***Occurrence Time***
M1	FAILURE	1.7352
M4	FAILURE	8.7335
M2	FAILURE	39.6855
M3	FAILURE	56.9589
B3	FILLING	57.1429

State Update

The first event in the list is the failure of M_1, so we update its state to "DOWN" and set $\mu_1 = 0$ and $OTF_{1,1} = 0$. The simulation clock is updated to $\boldsymbol{t_1 = 1.7352}$.

The buffer levels are updated, too. The only level change in the time interval $[t_0, t_1]$ happens in the buffer B_3 and it's determined with this formula:

$$X_3(t_1) = X_3(t_0)+(\alpha_3\mu_3 - \upsilon_3\mu_4) \cdot (t_1 - t_0) = (0.57 - 0.5) \cdot 1.7352 = 0.1215 \qquad (4.57)$$

The new system state is shown below:

Machine	*State*
1	DOWN
2	UP
3	UP
4	UP

Buffer	*Level*
1	0
2	0
3	0.1215

Performance Variables Update

After the system state has been updated, we can determine the values of the performance measures of interest that were indicated previously. Here we show how the cumulated amount of material processed by the machines, ($Q_{e,i}$, Q_i) and the cumulated amount stored in the buffers (ST_j) are updated:

$$Q_{e,i}(t_k) = Q_{e,i}(t_{k-1}) + \mu_i \cdot (t_k - t_{k-1}) \quad \forall e = 1, \dots, N \ \wedge \forall i = 1, \dots, K \qquad (4.58)$$

$$Q_i(t_k) = \sum_{e=1}^{N} Q_{e,i}(t_k) \quad \forall i = 1, \dots, K \qquad (4.59)$$

$$ST_j(t_k) = ST_j(t_{k-1}) + \frac{1}{2}\left(X_j(t_{k-1}) + X_j(t_k)\right) \cdot (t_k - t_{k-1}) \quad \forall j = 1, \dots, K-1 \qquad (4.60)$$

From these formulas, the following values are obtained:

Machine	*Cumulated A*	*Cumulated B*	*Cumulated Total*
1	0.3470	1.3881	1.7352
2	0.3470	1.3881	1.7352
3	0.3297	0.4164	0.7461
4	0.0152	0.8524	0.8676

Buffer	*Cumulated Storage*
1	0
2	0
3	0.1054

After this step, the first iteration is complete and so the simulation will proceed repeating the routines that have been shown until the stopping condition becomes true. This condition can be related to a maximum number of simulated events or to a maximum simulated time.

4.5.2.3 Performance Evaluation

In this example, the simulation is stopped after $b = 1000$ events, which in this specific case correspond to 6359 time units. The evolution of Q_i and of the buffer levels are shown below. Notice that the cumulated storage values $ST_j(t_{1000})$ correspond to the coloured areas in Figure 4.4 .

From these results, we can get the relevant performances of the system, in particular the throughputs of the machines – E_i^{Eff} and E_i^{Tot} – and the average buffer levels $\bar{X}_j$, which are computed with the following expressions:

$$E_i^{Eff} = \frac{\sum_{e \in T} Q_{i,e}(t_k)}{t_k} \quad \forall i = 1, \dots, K \tag{4.61}$$

$$E_i^{Tot} = \frac{Q_i(t_k)}{t_k} \quad \forall i = 1, \dots, K \tag{4.62}$$

$$\bar{X}_j = \frac{ST_j(t_k)}{t_k} \quad \forall j = 1, \dots, K-1 \tag{4.63}$$

In this case, after b events we obtain the following results:

	M1	**M2**	**M3**	**M4**
Effective Throughput	0.135	0.135	0.128	0.007
Total Throughput	0.673	0.673	0.290	0.383

Notice that the conservation of flow is satisfied.

	B1	**B2**	**B3**
Average Level	2.1908	0.2105	2.0890

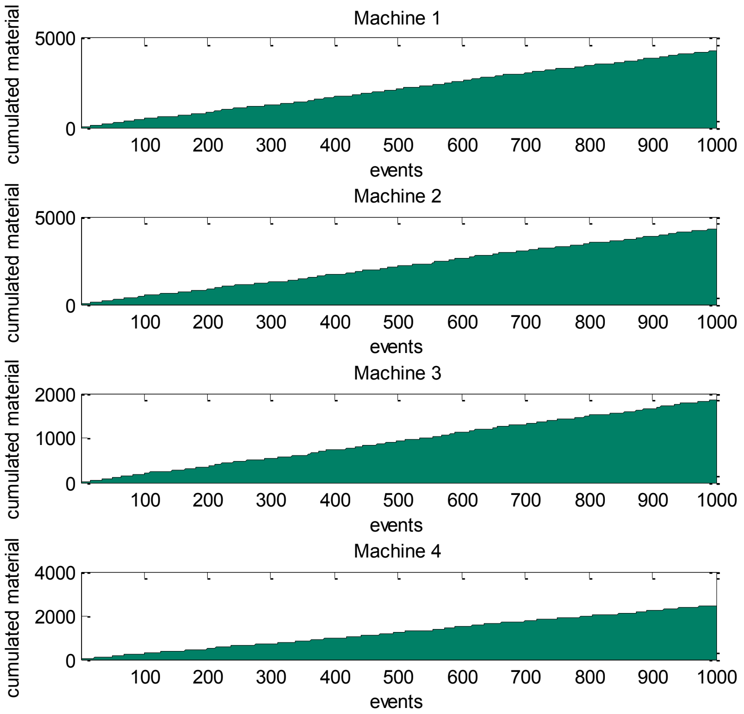


Figure 4.3: Evolution of the Cumulated Material Processed by the Machines

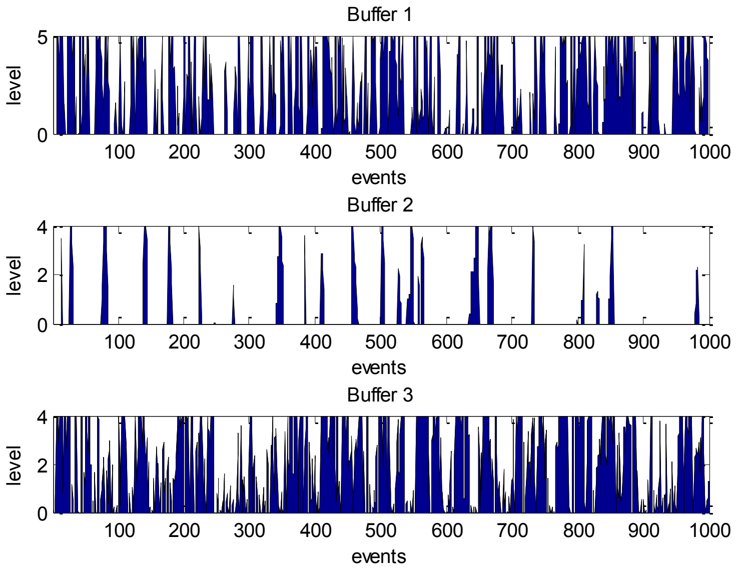


Figure 4.4: Evolution of the Buffer Levels

Finally, we can plot the performance measures against the number of events in order to check if they have converged towards stable values. In the example shown below, we have plotted the total throughputs of the machines:

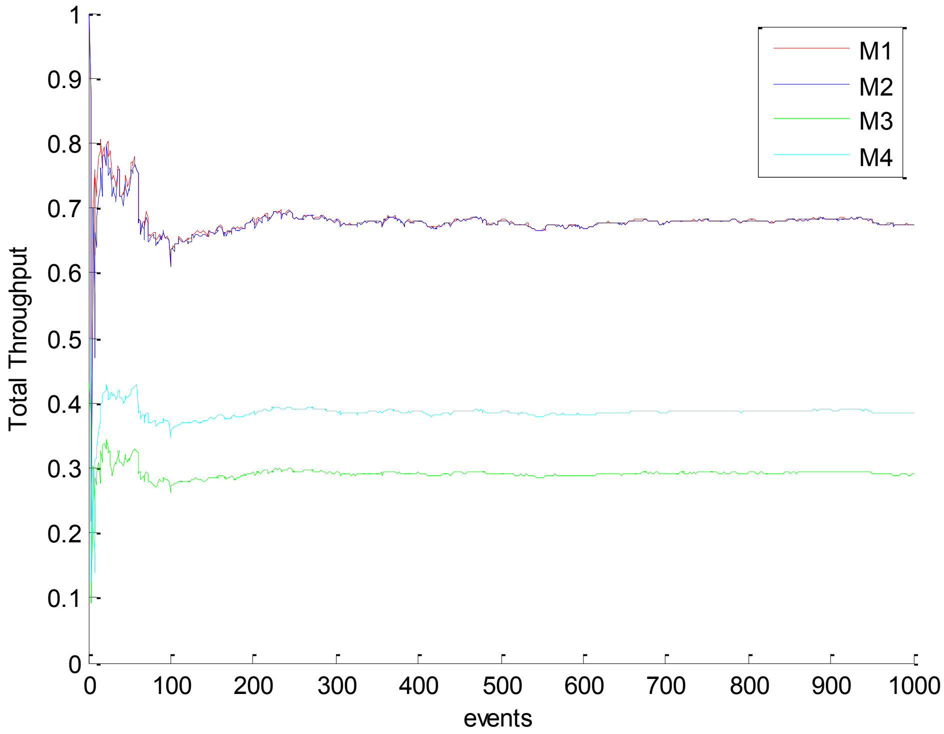


Figure 4.5: Evolution of the average throughput during 1000 events

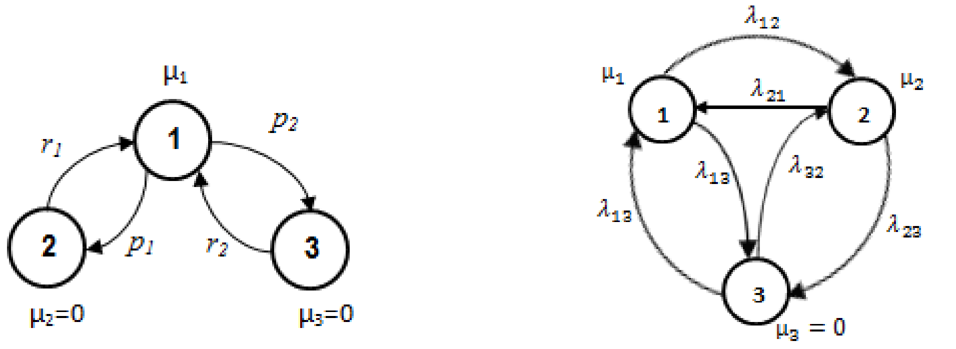
From this chart, it seems that the performances have converged towards quite stable values, although we might perform a longer simulation to be sure of this.

4.5.3 Validation of the Discrete-Event Simulator

In order to check the validity of the results obtained with our discrete-event simulator, we have compared its results with the ones obtained with the decomposition method applied to a 2-machine line. Since the analytical method provides an exact solution for these systems [84], we expect that the results obtained with the decomposition method fall within the confidence interval ($\alpha = 0.05$) of the corresponding measures determined with the simulation.

Several two-machine lines were used to check the validity of the simulator, however here just one example is reported for the sake of brevity. Notice that in this case the machine M_2 has two up states; in fact, our simulator is capable of simulating generally complex Markov chains, while our decomposition algorithm can model machines with a single up state and multiple failure modes. Machines' and simulation parameters are shown below.

4.5.3.1 System Inputs



Machine 1	
μ_1	1.2
p_1	0.3333
r_1	0.1
p_2	0.04
r_2	0.05

Machine 2			
μ_1	0.8	λ_{21}	0.1
μ_2	1	λ_{23}	0.2
λ_{12}	0.0667	λ_{31}	0.0667
λ_{13}	0.05	λ_{32}	0.04

Figure 4.6: Markov chains of Machine 1 and Machine 2

The buffer capacity is equal to 5.

4.5.3.2 Simulation inputs

For this test, 110 simulation runs were performed and in each of these runs, 30000 events have been simulated.

4.5.3.3 Results

As it can be seen in Table 4.6, for every performance measure of interest, the difference between the sample mean of the results of the 110 simulation runs and the exact solution of the analytical method is always safely below 1%. Furthermore, in every case the exact solution is within the 95% confidence interval determined by the simulation runs. Hence, taking into account also the fact that other tests have been performed with similar results, we are assured that our simulator works according to the assumptions that were previously illustrated, so it represents a suitable basis for comparison of our analytical model.

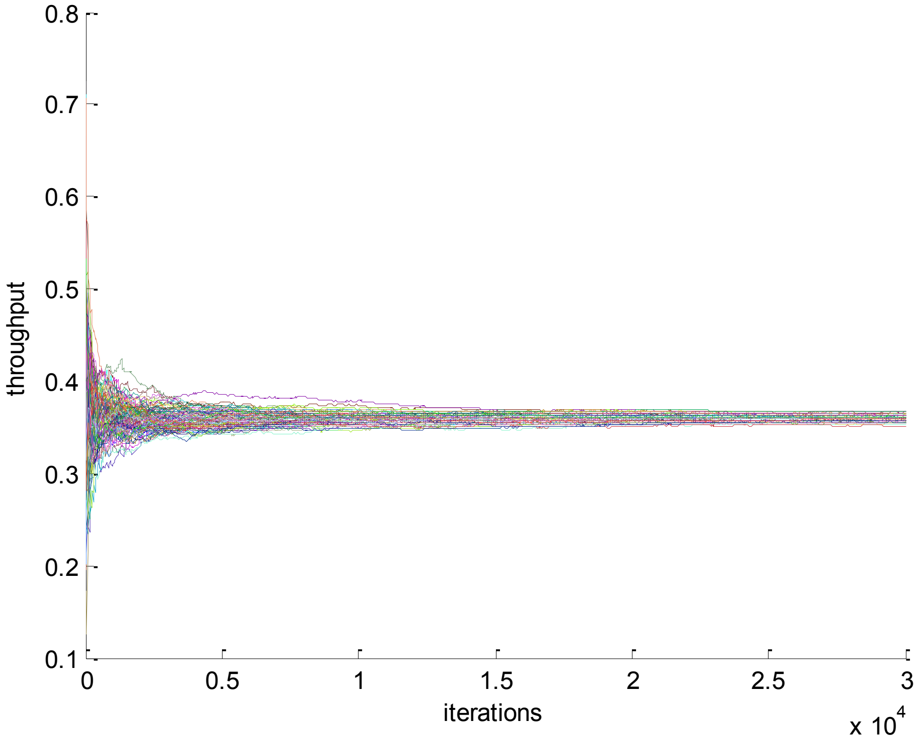


Figure 4.7: Throughput evolution for each simulation run

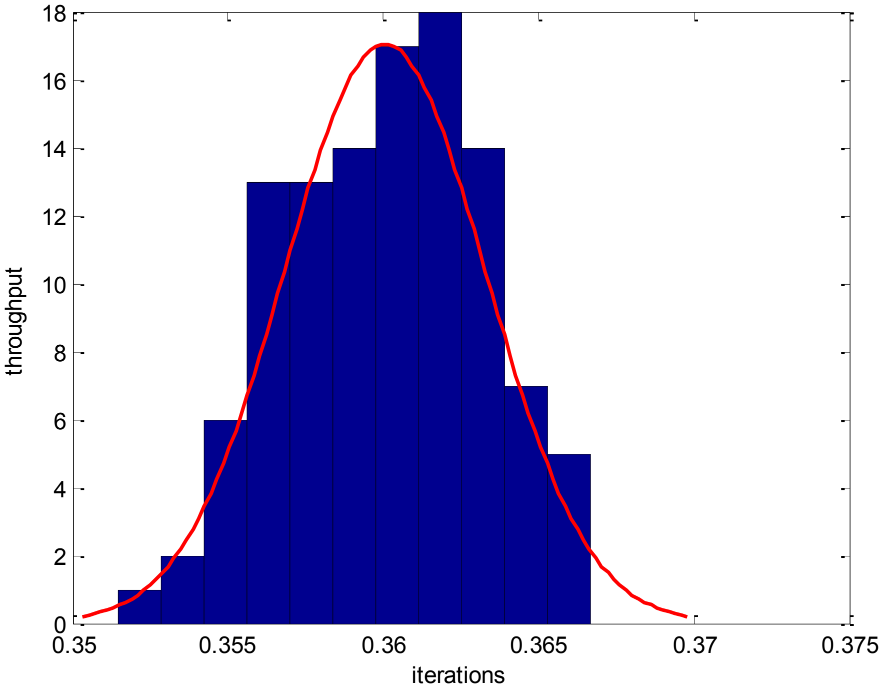


Figure 4.8: Distribution of the throughput values obtained after 30000 events

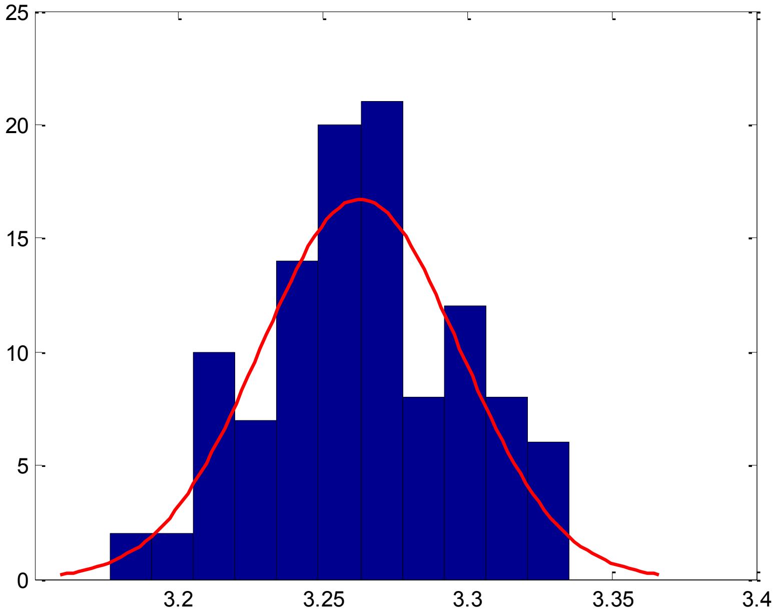


Figure 4.9: Distribution of the average buffer values obtained after 30000 events

			Lower Bound of C.I. (α=0.05)	*Upper Bound of C.I. (α=0.05)*	*Sample Mean of Simulation results (A)*	*Analytical result (B)*	**100 ·** $\left\|\frac{A-B}{B}\right\|$
		Throughput	0.35944	0.36067	0.3601	0.3602	**0.02%**
		Buffer level	3.25630	3.26936	3.26283	3.26002	**0.09%**
State Probabilities	***Machine 1***	***Working – Full Speed***	0.123572	0.124437	0.12400	0.12429	**0.23%**
		Working – Reduced Speed	0.24669	0.248081	0.24732	0.24702	**0.12%**
		Failed	0.338508	0.340901	0.33970	0.34016	**0.14%**
		Blocked	0.287978	0.289964	0.28897	0.28854	**0.15%**
	Machine 2	***Working – Full Speed***	0.420365	0.421828	0.42110	0.42123	**0.03%**
		Failed	0.360042	0.361856	0.36095	0.36045	**0.14%**
		Starved	0.216897	0.219012	0.21795	0.21832	**0.17%**

Table 4.6: Summary of the results

4.6 Numerical results

In order to check the accuracy of the analytical model developed in this work, several tests have been conducted on systems having different layouts. The represented layouts may not represent existing systems, but they are just used to check that the analytical model converges and can provide reasonably accurate results, in comparison with the simulation. With R simulation runs, the percentage error of the analytical method A in the estimation of the total throughput E_{ij}^{Tot}, for the generic buffer B_{ij}, versus simulation is estimated by using the following equations:

$$\varepsilon_{TH} = \frac{E_{ij}^{tot,A} - \hat{E}_{ij}^{tot,sim}}{\hat{E}_{ij}^{tot,sim}} \cdot 100, \tag{4.64}$$

where $\hat{E}_{ij}^{tot,sim}$ is computed as

$$\hat{E}_{ij}^{tot,sim} = \frac{\sum_{r=1}^{R} E_{ij}^{tot,sim}}{R} \tag{4.65}$$

Moreover, for each buffer B_{ij} the percentage error in the evaluation of the average level $\bar{X}_{ij}$ is calculated with the following equation:

$$\varepsilon_{BL} = \frac{\bar{X}_{ij}{}^{A} - \widehat{\bar{X}}_{ij}{}^{sim}}{\widehat{\bar{X}}_{ij}{}^{sim}} \cdot 100 \tag{4.66}$$

Where $\widehat{\bar{X}}_{ij}{}^{sim}$ is computed as

$$\widehat{\bar{X}}_{ij}{}^{sim} = \frac{\sum_{r=1}^{R} \bar{X}_{ij,r}{}^{sim}}{R} \tag{4.66}$$

The percentage error related to the overall *WIP* is obtained similarly.

4.6.1 System 1

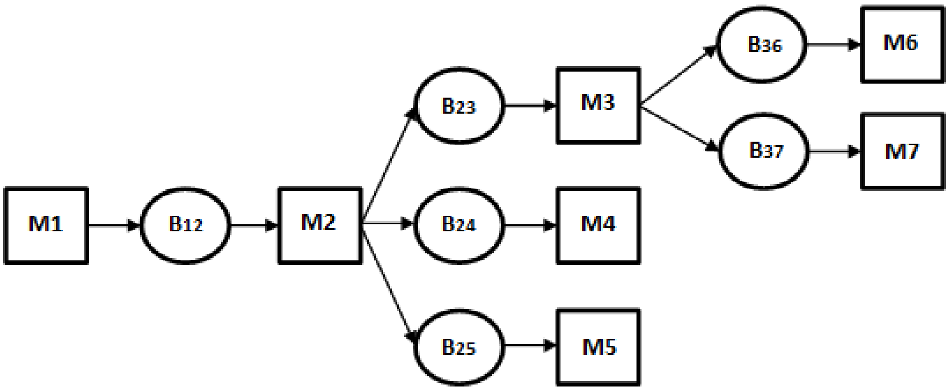


Figure 4.10: Layout of System 1

The system that we consider now is composed by 7 machines, where machines M_2 and M_3 are separators and $M_6 \in OUT_T$. The flow is made of 5 different elements, whose input concentration is defined by the vector $\boldsymbol{c}_{12}$:

$$\boldsymbol{c}_{12} = \begin{bmatrix} 0.0413 \\ 0.1866 \\ 0.0381 \\ 0.084 \\ 0.65 \end{bmatrix}$$

The elements $e = 1,2,3$ are considered as target. Regarding the separators, separation matrices $\boldsymbol{Q}_2$ and $\boldsymbol{Q}_3$ and routing vectors $\boldsymbol{r}_2$ and $\boldsymbol{r}_3$ are characterized as follows:

$$\boldsymbol{Q}_2 = \begin{bmatrix} 0.50 & 0.40 & 0.10 \\ 0.50 & 0.35 & 0.15 \\ 0.80 & 0.12 & 0.08 \\ 0.67 & 0.16 & 0.17 \\ 0.05 & 0.40 & 0.55 \end{bmatrix} \quad \boldsymbol{Q}_3 = \begin{bmatrix} 0.90 & 0.10 \\ 0.85 & 0.15 \\ 0.92 & 0.08 \\ 0.67 & 0.33 \\ 0.05 & 0.95 \end{bmatrix}$$

$$\boldsymbol{r}_2 = [3 \quad 4 \quad 5] \quad \boldsymbol{r}_3 = [6 \quad 7]$$

The other input parameters are shown in Table 4.7 and Table 4.8.

	$\boldsymbol{B}_{12}$	$\boldsymbol{B}_{23}$	$\boldsymbol{B}_{24}$	$\boldsymbol{B}_{25}$	$\boldsymbol{B}_{36}$	$\boldsymbol{B}_{37}$
Capacity	6	6	5	4	5	5

Table 4.7: Buffer Data related to System 1

	$\boldsymbol{M_1}$	$\boldsymbol{M_2}$	$\boldsymbol{M_3}$	$\boldsymbol{M_4}$	$\boldsymbol{M_5}$	$\boldsymbol{M_6}$	$\boldsymbol{M_7}$
p_1	0.03	0.015	0.01	0.001	0.001	0.01	0.001
r_1	0.101	0.21	0.09	0.08	0.08	0.06	0.03
p_2	0.004	0.005	0.03	0.007	0.007	0.001	0.03
r_2	0.05	0.05	0.15	0.04	0.04	0.03	0.15
μ (case 1)	1.1	1.5	0.9	0.7	0.7	0.1	0.4
μ (case 2)	1.3	1.1	1.2	1.3	0.9	0.7	1
μ (case 3)	1.4	1.2	1	0.9	0.9	0.5	0.5

Table 4.8: Machine Data related to System 1

As it can be seen from Table 4.9, Table 4.10 and Table 4.11, the analytical method provides excellent results when compared to the simulated ones for what concerns the average throughputs, while the estimations of the average buffer levels are less accurate, especially for buffer B_{37}, which is often empty, but this is a common drawback of the analytical methods based on the decomposition techniques. Also other buffers show quite high relative errors, but except in two cases they're always below 10%. However, in every case for every buffer except B_{37} the error is below 15%. This is reflected by the relative errors on the estimates of the overall *WIP*, which just exceed 7% in the worst case.

	$\boldsymbol{B_{12}}$	$\boldsymbol{B_{23}}$	$\boldsymbol{B_{24}}$	$\boldsymbol{B_{25}}$	$\boldsymbol{B_{36}}$	$\boldsymbol{B_{37}}$
$E_{ij}^{Tot,A}$	0.5036	0.1174	0.1812	0.2050	0.0832	0.0342
$\hat{E}_{ij}^{tot,sim}$	0.5024	0.1171	0.1808	0.2044	0.0830	0.0341
95% C.I.	± 0.001	± 0.0003	± 0.0003	± 0.0004	± 0.0002	± 0.0001
ε_{TH}	0.24	0.26	0.22	0.29	0.24	0.29
$\bar{X}_{ij}{}^{A}$	3.5333	4.3177	0.2072	0.2111	4.9380	0.0141
$\widehat{\bar{X}_{ij}}^{sim}$	4.1001	4.9081	0.2005	0.2127	4.8158	0.0102
95% C.I.	± 0.0166	± 0.0226	± 0.0058	± 0.0042	± 0.0084	± 0.0010
ε_{BL}	-13.83	-12.03	3.37	-0.76	2.54	37.90
WIP^A	13.2214	WIP^{sim}	14.2475	ε_{WIP}	-7.20	

Table 4.9: Results of Case 1

	$\boldsymbol{B_{12}}$	$\boldsymbol{B_{23}}$	$\boldsymbol{B_{24}}$	$\boldsymbol{B_{25}}$	$\boldsymbol{B_{36}}$	$\boldsymbol{B_{37}}$
$E_{ij}^{Tot,A}$	0.7445	0.1736	0.2679	0.3030	0.1230	0.0505
$\hat{E}_{ij}^{tot,sim}$	0.7434	0.1734	0.2675	0.3025	0.1228	0.0505
95% C.I.	±0.0021	±0.0005	±0.0007	±0.0008	±0.0004	±0.0001
ε_{TH}	0.15	0.12	0.15	0.17	0.16	-0.03
$\bar{X}_{ij}{}^{A}$	4.1045	0.1104	0.1525	0.2307	0.1092	0.0102
$\widehat{\bar{X}_{ij}}^{sim}$	4.0261	0.1201	0.1543	0.2343	0.1070	0.0080
95% C.I.	±0.0120	±0.0028	±0.0033	±0.0061	±0.0033	±0.0006
ε_{BL}	1.95	-8.04	-1.16	-1.53	2.16	27.02
WIP^{A}	4.7175	WIP^{sim}	4.6496	ε_{WIP}	1.46	

Table 4.10: Results of Case 2

	$\boldsymbol{B_{12}}$	$\boldsymbol{B_{23}}$	$\boldsymbol{B_{24}}$	$\boldsymbol{B_{25}}$	$\boldsymbol{B_{36}}$	$\boldsymbol{B_{37}}$
$E_{ij}^{Tot,A}$	0.7908	0.1844	0.2846	0.3218	0.1307	0.0537
$\hat{E}_{ij}^{tot,sim}$	0.7893	0.1841	0.2840	0.3212	0.1305	0.0536
95% C.I.	±0.0024	±0.0005	±0.0009	±0.001	±0.0004	±0.0001
ε_{TH}	0.18	0.17	0.20	0.18	0.19	0.13
$\bar{X}_{ij}{}^{A}$	4.0465	0.1691	0.2621	0.2554	0.1946	0.0254
$\widehat{\bar{X}_{ij}}^{sim}$	4.0392	0.1868	0.2569	0.2570	0.1883	0.0198
95% C.I.	±0.0134	±0.0047	±0.0071	±0.0070	±0.0053	±0.0011
ε_{BL}	0.18	-9.49	2.04	-0.61	3.32	27.96
WIP^{A}	4.9532	WIP^{sim}	4.948	ε_{WIP}	0.10	

Table 4.11: Results of Case 3

Regarding the characteristics of the material flow, in particular the values of recovery and grade related to the final product, we see no difference between the simulation and the analytical method. This was expected, since they are unrelated to the system state when rework loops aren't taken into account. In the table below, the results related to the output stations are shown.

	$\boldsymbol{M_4}$	$\boldsymbol{M_5}$	$\boldsymbol{M_6}$	$\boldsymbol{M_7}$
G_i^A	0.2401	0.0864	0.7620	0.2723
G_i^{sim}	0.2401	0.0864	0.7620	0.2723
ε_{TH}	0	0	0	0
R_i^A	0.3248	0.1322	0.4734	0.0696
R_i^{sim}	0.3248	0.1322	0.4734	0.0696
ε_{BL}	0	0	0	0

Table 4.12: Recovery and Grade performances obtained System 1

4.6.2 System 2

This system is composed by 11 machines, where machines M_3 and M_4 are separators and M_{10} is a merge station.

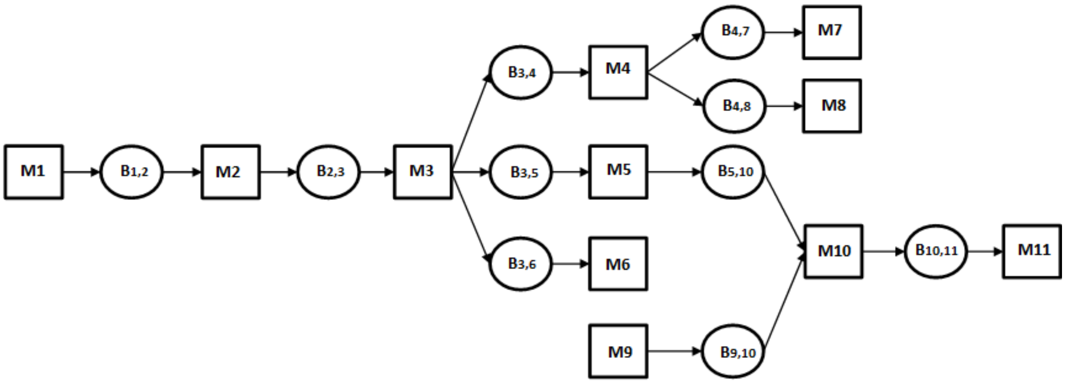


Figure 4.11: Layout of System 2

The flow is made of 5 different elements, whose input concentrations are

$$\boldsymbol{c}_{12} = \begin{bmatrix} 0.0413 \\ 0.1866 \\ 0.0381 \\ 0.084 \\ 0.65 \end{bmatrix} \quad \boldsymbol{c}_{9,10} = \begin{bmatrix} 0.5 \\ 0.3 \\ 0.2 \\ 0 \\ 0 \end{bmatrix}$$

As before, the first 3 elements are considered as target. Separation matrices $\boldsymbol{Q}_3$ and $\boldsymbol{Q}_4$ and routing vectors $\boldsymbol{r}_3$ and $\boldsymbol{r}_4$ are characterized as follows:

$$\boldsymbol{Q}_3 = \begin{bmatrix} 0.50 & 0.40 & 0.10 \\ 0.50 & 0.35 & 0.15 \\ 0.80 & 0.12 & 0.08 \\ 0.67 & 0.16 & 0.17 \\ 0.05 & 0.40 & 0.55 \end{bmatrix} \quad \boldsymbol{Q}_4 = \begin{bmatrix} 0.90 & 0.10 \\ 0.85 & 0.15 \\ 0.92 & 0.08 \\ 0.67 & 0.33 \\ 0.05 & 0.95 \end{bmatrix}$$

$$\boldsymbol{r}_2 = [4 \quad 5 \quad 6] \quad \boldsymbol{r}_3 = [7 \quad 8]$$

Regarding the merge station, it takes 70% of its input material from $B_{5,10}$ and 30% from $B_{9,10}$.

The other input parameters are shown in Table 4.7 and Table 4.8.

	$\boldsymbol{M_1}$	$\boldsymbol{M_2}$	$\boldsymbol{M_3}$	$\boldsymbol{M_4}$	$\boldsymbol{M_5}$	$\boldsymbol{M_6}$	$\boldsymbol{M_7}$	$\boldsymbol{M_8}$	$\boldsymbol{M_9}$	$\boldsymbol{M_{10}}$	$\boldsymbol{M_{11}}$
p_1	0.03	0.03	0.015	0.01	0.001	0.001	0.01	0.001	0.001	0.01	0.001
r_1	0.101	0.101	0.21	0.09	0.08	0.08	0.06	0.03	0.03	0.03	0.03
μ	1.2	1.1	1.5	0.9	0.7	0.7	0.1	0.4	0.4	1.3	1.1

Table 4.13: Machine Data related to System 2

	$\boldsymbol{B_{12}}$	$\boldsymbol{B_{23}}$	$\boldsymbol{B_{34}}$	$\boldsymbol{B_{35}}$	$\boldsymbol{B_{36}}$	$\boldsymbol{B_{47}}$	$\boldsymbol{B_{48}}$	$\boldsymbol{B_{5,10}}$	$\boldsymbol{B_{9,10}}$	$\boldsymbol{B_{10,11}}$
Capacity	5	6	6	5	4	5	5	5	4	5

Table 4.14: Buffer Data related to System 2

Even with a longer line, the error on the estimated throughput is almost negligible, being around 0.2% for all the buffers in the system. Buffer levels show larger errors than in the cases considered for System 1, but the error related the *WIP* is still below 10%.

Finally, in Table 4.17 the quality performances related to the output stations in the system are shown.

	$\boldsymbol{B_{12}}$	$\boldsymbol{B_{23}}$	$\boldsymbol{B_{34}}$	$\boldsymbol{B_{35}}$	$\boldsymbol{B_{36}}$
$E_{ij}^{Tot,A}$	0.5184	0.5184	0.1209	0.1865	0.2110
$\hat{E}_{ij}^{tot,sim}$	0.5175	0.5175	0.1207	0.1862	0.2106
95% C.I.	±0.0012	±0.0013	±0.0003	±0.0005	±0.0005
ε_{TH}	0.18	0.18	0.21	0.16	0.20
	$\boldsymbol{B_{47}}$	$\boldsymbol{B_{48}}$	$\boldsymbol{B_{5,10}}$	$\boldsymbol{B_{9,10}}$	$\boldsymbol{B_{10,11}}$
$E_{ij}^{Tot,A}$	0.0857	0.0352	0.1865	0.0799	0.2664
$\hat{E}_{ij}^{tot,sim}$	0.0855	0.0352	0.1862	0.0798	0.2660
95% C.I.	±0.0002	±0.0001	±0.0005	±0.0002	±0.0006
ε_{TH}	0.26	0.14	0.16	0.10	0.15

Table 4.15: Throughputs of System 2

	$\boldsymbol{B_{12}}$	$\boldsymbol{B_{23}}$	$\boldsymbol{B_{34}}$	$\boldsymbol{B_{35}}$	$\boldsymbol{B_{36}}$
$\bar{X}_{ij}{}^{A}$	4.1767	2.5565	4.3872	0.1602	0.0134
$\widehat{\bar{X}_{\imath\jmath}}{}^{sim}$	4.1294	3.9198	5.1830	0.1824	0.0123
95% C.I.	±0.0104	±0.0228	±0.0220	±0.0082	±0.0009
ε_{BL}	1.13	-34.78	-15.35	-12.19	8.59
	$\boldsymbol{B_{47}}$	$\boldsymbol{B_{48}}$	$\boldsymbol{B_{5,10}}$	$\boldsymbol{B_{9,10}}$	$\boldsymbol{B_{10,11}}$
$\bar{X}_{ij}{}^{A}$	4.97125	0.0047	0.2476	3.9805	0.1413
$\widehat{\bar{X}_{\imath\jmath}}{}^{sim}$	4.8960	0.0034	0.3289	3.9837	0.0727
95% C.I.	±0.0073	±0.0010	±0.0088	±0.0022	±0.0032
ε_{BL}	1.54	38.77	-24.72	-0.08	94.28
WIP^{A}	20.64	WIP^{sim}	22.71	ε_{WIP}	-9.12

Table 4.16: Buffer Levels and WIP of System 2

	$\boldsymbol{M_6}$	$\boldsymbol{M_7}$	$\boldsymbol{M_8}$	$\boldsymbol{M_{11}}$
$\boldsymbol{R_i}$	0.0840	0.2991	0.0443	0.5726
$\boldsymbol{G_i}$	0.0864	0.7620	0.2722	0.4681

Table 4.17: Recovery and Grade at the output stations of System 2