Journal of Manufacturing Systems 69 (2023) 103–118

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys

Continuous-flow simulation of manufacturing systems with assembly/disassembly machines, multiple loops and general layout

Salvatore Scrivano [a,*], Barış Tan [b], Tullio Tolio [a]

[a] Department of Mechanical Engineering, Politecnico di Milano, Milan, Italy
[b] College of Administrative Sciences and Economics, College of Engineering, Koç University, Rumeli Feneri Yolu, Istanbul, Turkey

ARTICLE INFO

Keywords:
Modelling manufacturing systems
Simulation
Manufacturing system analysis
Performance evaluation
Continuous flow

ABSTRACT

Performance evaluation methods are important to design and control manufacturing systems. Approximate analytical methods are fast, but they may be limited by the restrictive assumptions on the system. On the contrary, simulation has not specific limitations in its applicability, but the time to model and analyse a manufacturing system can increase as the level of detail addressed by the model increases. The main contribution of this study is presenting a computationally efficient methodology to simulate single-part continuous-flow manufacturing systems with assembly/disassembly machines, multiple loops, general layout and general inter-event time distributions. By using graph theory, a new method is presented to identify the machines causing slowdown, blocking and starvation in a general layout and determine the time before the occurrence of a state transition for each machine and the time before the fulfilment or depletion of each buffer. By advancing the time clock to the next event-time accordingly, the number of discrete events needed to be simulated is decreased compared to a discrete-event simulation with discrete flow of parts. As a result, the proposed method is on average 15 times faster than DES methods in the analysis of discrete-flow systems, and 110 times faster on average in the analysis of continuous-flow systems. The low computational time of the proposed method allows to simulate systems under general assumptions and in a very short time.

1. Introduction

In the manufacturing industry, using fast and reliable methods for the evaluation of production systems is fundamental for supporting decision making. The performance of existing systems are evaluated by analysing the data collected in the production plant. However, the production performance of systems during the design phase, during the reconfiguration phase and as a result of changes in the control policies of the system can be evaluated by using performance evaluation tools such as simulation and analytical methods [1].

During the design phase, several configurations must be evaluated before the selection and the implementation of the final solution. For example, selecting the appropriate machine for a manufacturing process involves evaluating a multitude of potential solutions that vary in terms of cost, flexibility, and production capacity. Similarly, the allocation of the buffer capacity between production stages directly affects the overall performance of the manufacturing system, and it can pose a challenging problem due to the space limitations usually faced by production facilities. In addition, in the operational phase of a manufacturing system, the vast set of improvements that can be introduced in a plant must be evaluated such that the limited resources of a company in terms of time and investment can be addressed to the implementation of the most profitable actions. Evaluating the alternatives by using simulation requires fast and accurate methods.

This work presents a fast and accurate simulation method for the analysis of manufacturing systems and the evaluation of the production performance, such as the production rate (throughput), the level of the WIP, the lead time of the system, and the portion of up time, downtime and idle time of each machine.

Regardless of whether the material flow in the analysed manufacturing system is composed of discrete parts or a continuous stream, in the proposed simulation method it is represented as a continuous flow. The time is advanced in the simulation between discrete events affecting the flow rate of material in the system. Every continuous change of the state of the system, such as the buffer level, is analytically formalized in a set of equations which are solved at each event time in the simulation. This approach decreases the number of discrete events needed to be simulated compared to a discrete event simulation when the flow of each discrete part is modelled separately.

The proposed simulation method introduces a novel methodology based on graph theory, which can simultaneously identify any cause

[*] Corresponding author.
E-mail address: salvatore.scrivano@polimi.it (S. Scrivano).

https://doi.org/10.1016/j.jmsy.2023.05.028
Received 20 February 2023; Received in revised form 5 May 2023; Accepted 29 May 2023
Available online 17 June 2023

of slowdown, blocking, or starvation for any machine in the system, at any event time during the simulation. This approach enables the determination of effective production rates of all machines in the system in a single step, whenever an event occurs in the simulation.

The manufacturing systems analysed by the continuous-flow simulation method have a general layout, with a linear structure in which the material sequentially flows through all the machines and buffers, or with a non-linear structure because of the presence of assembly/disassembly machines that combine/divide the material flows from/to different portions of the system. Stochastic events occurring in the manufacturing system, such as machine failures and repairs, have general inter-event time distributions. As a result, the proposed method is more general and the computational effort is significantly lower compared to the ones given in the literature.

The rest of the paper is organized as follows. In Section 2, a review of the approaches proposed in the literature for the evaluation of the production performance of manufacturing systems is given. Section 3 describes the main dynamics of the systems under analysis and defines the general notation of the system parameters. In Section 4, the continuous-flow simulation method is defined. In Section 5, the methodology presented in Section 4 is applied to a specific production system to explain how it can be used to analyse a given system. The numerical results and the computational performance of the continuous-flow simulation for four different production layouts are shown in Section 6. In Section 7 an industrial case from the manufacturing industry is presented. In the industrial case, the continuous-flow simulation has been exploited to analyse and optimize the reconfiguration of a production line. Finally, conclusions and future developments are given in Section 8.

2. Literature review

This section gives a review of the two main modelling approaches available in the literature for the evaluation of the production performance of manufacturing systems, which are analytical methods and simulation methods.

2.1. Analytical methods

Among the several alternatives for the evaluation of production systems, approximate analytical methods are proposed as fast and intuitive approaches [2]. The work in [3] provides a complete classification of analytical methods for the performance evaluation of manufacturing systems.

Analytical methods can be classified into discrete-flow methods and continuous-flow methods [4]. In discrete-flow methods, the movement of discrete parts is represented. The work in [5] presents an evaluation model of a discrete-flow transfer line with two unreliable machines and finite storage buffer. When the analysed manufacturing system has finite buffers and more than two machines, decomposition techniques have been developed to estimate manufacturing systems performance [6,7].

Analytical models based on a continuous approximation of the discrete flow of parts have been extensively employed for the analysis of systems with asynchronous machines characterized by deterministic processing times and stochastic failures and repairs. [8–10] are some of the earliest works for the evaluation of unreliable two-station systems with a finite intermediate buffer. The work in [11] analyses continuous-flow two-machine lines with multiple failure modes, while [12,13] generalize the approach to multiple-up and multiple-down machines. In [14,15], continuous-flow analytical models with threshold-based control policy are presented for the analysis of two-stage lines. If one or more machines in the system have stochastic processing times, the work in [16] provides a Markov Chain approximation of stations with general processing time distributions, to be used in continuous-flow models.

Decomposition methods based on continuous-flow analytical models have been developed to evaluate asynchronous serial lines with deterministic processing times. Some of the decomposition methods proposed in the literature are the work in [17] for machines with identical processing times, in [18] for machines with different processing times, in [19] for multiple-up and multiple-down machines, in [20] for systems with merge of material flows, and in [21] for multiple assembly/disassembly machines.

The mathematical complexity that may arise in the analytical modelling of complex real system often requires significant simplification of the problem. These simplifications may limit the applicability of analytical models to analyse a given manufacturing system [22].

2.2. Simulation methods

When no analytical models are available in the literature to analyse the system of interest, simulation techniques are used. Simulation can be utilized for any manufacturing system where different entities move through the system [23].

One of the biggest challenge in the definition of a simulation model is selecting the level of abstraction, that is the level of detail to include in a model to ensure accuracy in its application [24]. As opposed to emulation, simulation does not mimic the entire behaviour of a reference system, but it captures though an abstract model only the behaviours of interest.

Among the simulation techniques, discrete event simulation (DES) usually operates at a low abstraction level (that is a high level of detail), and it is used for the simulation of a given system on operation level. Instead, system dynamics (SD) generally operates at a high abstraction level and it is used to simulate strategic issues [25]. The work in [26] explores the application of DES and SD as decision support systems in logistics and supply chain management.

2.2.1. Simulation of discrete-flow systems

DES is the most popular approach among simulation methods for manufacturing systems [27]. The work in [28] provides a classification of the literature on the use of DES for manufacturing system design and operation problems. In [29], a comprehensive review of DES publications with a particular focus on applications in manufacturing is given.

Some examples of application of DES for manufacturing systems are the work in [30] to design a production line against multiple-user criteria, and the work in [31] to evaluate the production efficiency and reliability of manufacturing systems with human operators or industrial robots.

The employment of DES with discrete flows of entities may face significant limitations because of the complexity and the computational time to simulate it [23]. In DES models with discrete flow of parts, each unitary increase and decrease of the buffer levels is modelled as a discrete event, although it may not affect the material flow in the system [32]. By increasing the simulation time or the size of the simulated system, the number of events in the simulation can dramatically grow up causing long computational times of the simulation.

If DES must be used as an evaluation method in optimization problems such as the buffer capacity allocation problem, the number of system configurations that have to be evaluated can be too high to be simulated in acceptable computational times. Mathematical programming approaches have been developed in the literature for the simulation and optimization of discrete material flow production systems with linear layout, as the works in [33–35].

2.2.2. Simulation of continuous-flow system

The issue of long computational times of DES with discrete entities becomes more critical when the system to be simulated is a continuous-flow system. Continuous material flows can be approximated with discrete flows of entities in DES approaches. The smaller the amount of material represented by a discrete entity, the closer the dynamics of the discrete-flow system are to the dynamics of the continuous-flow system. However, by reducing the amount of material represented by a discrete entity, the number of events in the simulated system increases, and the computational time becomes longer.

Among the simulation techniques, SD is considered an appropriate approach in analysing continuous-flow systems [26]. In SD models, individual entities are not modelled specifically; rather, they are represented as a continuous quantity in a stock [36]. The ordinary differential equations that represent continuous changes of the stocks over time are approximated by calculations performed at fixed time step, called Delta Time (DT).

The first system simulated with the SD approach was a simple inventory control system [37]. The work in [38] investigates the opportunities in the employment of system dynamics in manufacturing system modelling.

SD models have been used in the literature to analyse production lines with constant work in process policy [39], pull production systems with uncertain demand and kanban control policy [40], and manufacturing systems with unexpected problems and disturbances [41].

A key factor in SD models is the definition of the time interval DT between calculations. Indeed, as DT gets smaller, the overall solution of the equations describing the continuous change of the system state becomes more accurate [42]. On the other hand, a small DT increases the number of calculations performed during a simulation run, resulting in longer computational time.

An alternative approach to the time discretization in continuous systems is quantization [43], that is the partitioning of the state space while keeping the time continuous. The main idea of quantization is the definition of the next time at which the variable describing the system state is going to cross the boundaries of the current partition [44]. The formalism defined in Discrete Event System Specifications (DEVS) [43, 45] has shown to exactly represent and simulate continuous systems as discrete event models by means of quantization [46,47].

The work of [48] has proposed a simulation method of manufacturing systems based on the continuous approximation of the material flow in the system. Similarly to quantization, the main idea beyond the simulation of the continuous flow is the definition of the earliest time at which one of the variables describing the system state reaches a threshold value that causes a change in the rate of the material flow. The systems analysed in [48] are serial lines with unreliable machines characterized by stochastic times to failure and times to repair and with finite intermediate buffers. The events that change the rate of the material flow are machine failures and repairs, and buffer fulfilment and depletion. Between two events, the continuous changes of the buffer levels are represented as linear functions depending on the constant production rates of the upstream and downstream machines. The overall trajectory of a buffer level over the simulation time results a piecewise linear segment.

An equivalence between the approach proposed in [48] and the formalism defined in the Generalized Discrete Event Specifications (GDEVS) for modelling dynamic systems [49,50] can be noticed. In GDEVS, the input, output and state trajectories representing a dynamic system are approximated with piecewise polynomial function. The next event in a simulation model with GDEVS is the earliest event that change the gradients or the intersects of the polynomial function of the input trajectory or the state trajectory.

In the works of [32,51], a mathematical programming approach is presented to simultaneously simulate and optimize continuous-flow production systems with two stages and an intermediate buffer.

Regardless of the discrete-event formalism used to simulate continuous systems, the definition of the next event in the simulation of continuous-flow systems becomes complicated and time consuming when the system layout is not a serial line. In a serial line, a flow interruption in a machine can propagate to the upstream portion, which become blocked, and to the downstream portion of the line, which become starved. In non-serial systems, the flow interruption propagation can branch off in more than two directions. Consequently, the identification of all the machines anywhere in the system that cause blocking, starvation or slowdown of the material flow is a hard problem. This problem needs to be solved to define the effective production rate of each machine and the next event in the simulation (i.e. buffer fulfilment, depletion, machine failure, repair, etc.).

In the work [52], a continuous-flow simulation model for assembly and disassembly systems is provided. The complex issue of defining the effective production rate of each machine in the system at any event time is sequentially solved machine by machine through a new algorithm. However, by applying the algorithm in systems with multiple loops, the continuous-flow simulation can get trapped in the execution of an endless sequence of simultaneous events.

2.3. Progress beyond the current state of the art

Starting from the works in [32,48,51,52], in this paper a new and fast discrete-event continuous-flow simulation method is presented. A new methodology based on graph theory is introduced in the simulation method, that extends the results given in [52] to analyse systems with a general layout, assembly/disassembly machines and multiple loops.

Unlike SD approaches, the accuracy of the continuous-flow simulation method presented in this work is not affected by the approximation of the continuous changes of the system variables at fixed discrete steps [26].

Differently to DES methods, in the proposed continuous-flow simulation method each unitary increase or decrease of the level of any buffer in the system does not generate a discrete event in the simulation. As a main consequence, the continuous-flow simulation method proposed in this paper provides a significant improvement in analysis speed for both discrete-flow and continuous-flow systems, with respect to DES approaches with discrete entities.

3. General notation and dynamics of the system

In this section, the general notation of the continuous-flow simulation and the main dynamics of the manufacturing systems under analysis are introduced. All the notations referring to variables or parameters which do not depend on the event time of the simulation are represented with curly brackets, and all the variables in the continuous-flow simulation whose values depend on the event time are represented with round brackets.

The manufacturing system of interest is a general network with M unreliable machines and B finite buffers. Fig. 1 provides the graphical representation of a general network composed by four machines and four buffers. This example will be analysed in Section 5 to explain the methodology presented in this paper in detail.

3.1. Machine parameters

Each machine in the reference system is represented by the notation $\mathrm{M}\{m\}$, with $m = 1, \ldots, M$. The flow of material in the system is continuous, and the behaviour of each machine in isolation can be described by the set of states $S\{m\} = \left\{S_1\{m\}, \ldots, S_{I\{m\}}\{m\}\right\}$, with size $I\{m\}$. At a general event time t_k in the simulation, the current state of a machine $\mathrm{M}\{m\}$ is represented by the column vector $s(m, t_k)$, with size $I\{m\}$. Each entry $s_i(m, t_k)$ in the vector $s(m, t_k)$ is defined as:

$$s_i(m, t_k) = \begin{cases} 1, & \text{if } \mathrm{M}\{m\} \text{ is in the state } S_i\{m\} \text{ at the time } t_k \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

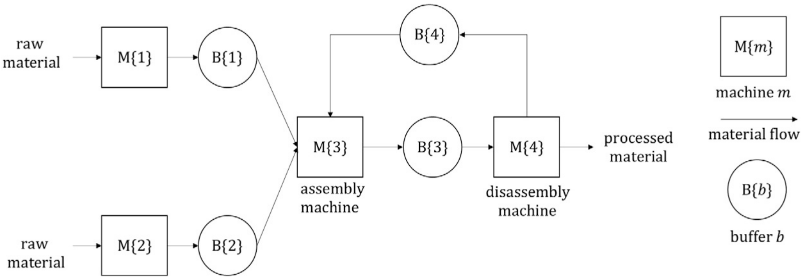


Fig. 1. Graphical representation of a manufacturing system with loops and assembly/disassembly machines.

Since machine M$\{m\}$ can be in only one state at time, it results that only one entry in the vector $s(m,t_k)$ is equal to 1:

$$\sum_{i=1}^{I\{m\}} s_i(m,t_k) = 1, \qquad \forall m, \forall t_k. \tag{2}$$

When the simulation starts at time t_0, the current state $s(m,t_0)$ of each machine M$\{m\}$ is given as the initial state.

The time to transition between two states $S_i\{m\}$ and $S_j\{m\}$ is stochastic with a general probability distribution $D_{i,j}\{m\}$. Times to transition are assumed to be independent between couples of states, that is, the occurrence of a transition from the state $S_i\{m\}$ to a state $S_j\{m\}$ does not affect the remaining time before the occurrence of the transition from $S_i\{m\}$ to another state $S_l\{m\}$. However, the sequence of times generated from the distribution $D_{i,j}\{m\}$ can also be correlated.

In each state $S_i\{m\} \in S\{m\}$, the machine M$\{m\}$ processes material with a nominal production rate $\mu_i\{m\}$. A state $S_i\{m\}$ is defined operational if $\mu_i\{m\} > 0$, while it is defined as a down state if $\mu_i\{m\} = 0$. The row vector of the nominal production rates of all the states $S\{m\}$ of the machine M$\{m\}$ is defined as $\mu\{m\} = \left[\mu_i\{m\}\right]$, with $i = 1,\dots,I\{m\}$. Therefore, the nominal production rate of the state of M$\{m\}$ at time t_k is defined by the matrix product $\mu\{m\} \cdot s(m,t_k)$.

In this work, a transition from a state $S_i\{m\}$ to any state $S_j\{m\}$ is assumed to be operation-dependent if $S_i\{m\}$ is an operational state, while it is assumed to be time-dependent if $S_i\{m\}$ is a down state. An operation-dependent transition from a state $S_i\{m\}$ to a state $S_j\{m\}$ occurs according to its distribution $D_{i,j}\{m\}$ if the machine M$\{m\}$ is not slowed down, blocked or starved by other machines. If M$\{m\}$ is slowed down, the remaining time before the occurrence of an operation-dependent transition is exhausted slowly, proportionally to the ratio between the effective production rate and the nominal rate. Instead, if the machine is blocked or starved, no operation-dependent transition can occur during the whole blocking or starvation period. On the other hand, time dependent transitions do not depend on the operational mode of the machine.

At any event time t_k, the effective production rate of a machine M$\{m\}$ is defined by the variable $\mu(m,t_k)$. The effective production rate $\mu(m,t_k)$ may differ from the nominal production rate $\mu\{m\} \cdot s(m,t_k)$ of M$\{m\}$ at t_k, if at t_k M$\{m\}$ is slowed down, blocked or starved by other machines in the system. The identification of the machines affecting, at time t_k, the production rate of any another machine in the system and the computation of the effective production rate are critical steps in the continuous-flow simulation and they are explained in Section 4.4. In Section 3.3, the definition of the slowdown, blocking and starvation phenomena in a continuous-flow system are given.

3.2. Buffer parameters

Each buffer of the system is represented by the notation B$\{b\}$, with $b = 1,\dots,B$. Each buffer B$\{b\}$ can have a finite or infinite capacity, defined by an upper boundary $N\{b\}$ and by a lower boundary $L\{b\}$, with $N\{b\} > L\{b\}$. If $L\{b\} < 0$, it means that the backlog of orders related to the material stored in the buffer B$\{b\}$ is allowed until the maximum value $|L\{b\}|$, beyond which a stockout occurs.

At any event time t_k, the current level of a buffer B$\{b\}$ is defined by the continuous variable $x(b,t_k)$, such that $L\{b\} \le x(b,t_k) \le N\{b\}$. When the simulation starts at time t_0, the current level $x(b,t_0)$ of each buffer B$\{b\}$ is given as its initial value.

Each buffer B$\{b\}$ is fed by only a single upstream machine, and it feeds only a single downstream machine. On the other hand, each machine can feed more than one buffer and it can be fed by more than one buffer. Machines fed by multiple buffers are defined as assembly machines, while machines feeding multiple buffers are defined as disassembly machines. In a disassembly machine, the flow of material is divided into different material types, one for each immediately downstream buffer. In an assembly machine, the different material types from the immediately upstream buffers are combined in a single flow of material. In this work it, is assumed that each buffer in the system can hold only one type of material.

No split and merge machines are considered in this work. It means that a machine cannot selectively process material from one of its upstream buffers or to one of its downstream buffers. In the proposed model, for each unit of material processed by a machine, it is assumed that one unit of material is received from each upstream buffer and one unit of material is fed into each downstream buffer.

3.3. Dynamics of the system

The layout of the system is formalized through the flow matrix F, composed by B rows and M columns representing respectively the buffers and the machines of the system. Each entry $F_{b,m}$ in the matrix F is defined as:

$$F_{b,m} = \begin{cases} 1, & \text{if the machine M}\{m\}\text{ feeds the buffer B}\{b\} \\ -1, & \text{if the buffer B}\{b\}\text{ feeds the machine M}\{m\} \\ 0, & \text{otherwise,} \end{cases} \qquad \forall b, \forall m. \tag{3}$$

Considering two machines M$\{m\}$ and M$\{m'\}$, M$\{m\}$ is an immediately upstream machine of M$\{m'\}$ and M$\{m'\}$ is an immediately downstream machine of M$\{m\}$, if a buffer B$\{b\}$ exists in the system such that $F_{b,m} = 1$ and $F_{b,m'} = -1$. In a subsystem composed by a buffer B$\{b\}$ and its upstream and downstream machines M$\{m\}$ and M$\{m'\}$, the following dynamics summarize the behaviour of the subsystem.

If the machine M$\{m\}$ is in a state $S_i\{m\}$ and M$\{m'\}$ is in a state $S_j\{m'\}$ such that M$\{m\}$ processes material faster than M$\{m'\}$, that is $\mu_i\{m\} > \mu_j\{m'\}$, after a certain time period the level of the buffer B$\{b\}$ reaches the upper boundary $N\{b\}$, and the machine M$\{m\}$ is forced to process material at the same production rate of M$\{m'\}$. In this condition, if the production rate $\mu_j\{m'\}$ of M$\{m'\}$ is higher than zero, the upstream machine M$\{m\}$ is slowed down at the same rate. Instead, if $\mu_j\{m'\} = 0$, the upstream machine M$\{m\}$ is blocked and it does not process material. The subsystem stays in this state until an event changing the production rate of one of the two machines occurs.

Similarly, if the machine M$\{m\}$ in the state $S_i\{m\}$ is slower than M$\{m'\}$ in the state $S_j\{m'\}$, after a certain time period the buffer level of B$\{b\}$ reaches the lower boundary $L\{b\}$ and the machine M$\{m'\}$ is forced to process material at the same production rate of M$\{m\}$. If the production rate $\mu_i\{m\}$ is higher than zero, the downstream machine M$\{m'\}$ is slowed down at the same production rate, while if $\mu_i\{m\} = 0$, the downstream machine is starved and it does not process material, until another event occurs.

Extending the analysis to a serial line with M machines and $M-1$ buffers, it is possible to state that a machine M$\{m'\}$ is slowed down or starved by an upstream machine M$\{m\}$ if all the buffers between M$\{m\}$ and M$\{m'\}$ are empty and if the production rate of M$\{m\}$ is the lowest among the nominal production rates of all the machines from M$\{m\}$ to M$\{m'\}$. Similarly, a machine M$\{m'\}$ is slowed down or blocked by a downstream machine M$\{m''\}$ if all the buffers between M$\{m'\}$ and M$\{m''\}$ are full and if the production rate of M$\{m''\}$ is the lowest among the nominal production rates of all the machines from M$\{m'\}$ to M$\{m''\}$.

However, when the layout of the system is not a serial line, the identification of the machine causing the slowdown, blocking or starvation of other machines in the system is not so simple. Consider, for instance, an assembly machine that combines the flows of material from several upstream machines. If one of these upstream machines fails, after a certain time period the intermediate buffer between the failed machine and the assembly machine becomes empty, and the assembly machine cannot process material from any upstream buffer. Consequently, all the other upstream machines become blocked, while all the downstream machines become starved. Therefore, in a system with a general layout, a flow interruption or slowdown can propagate across any branch in the whole system through a non-serial path of machines.

The identification of the machine causing the slowdown, blocking or starvation of any machine in a general system, must be carried out by simultaneously analysing every single branch of the material flow within the system. For large systems, this analysis can be computationally demanding and it can dramatically increase the time to run the simulation. In Section 4.4, a new method to quickly perform this analysis at each event time in the simulation is provided.

4. Continuous-flow simulation

The continuous-flow simulation for a system with general layout consists of two main steps which are iteratively performed. The first step is the definition of the advancement rule of the clock time in the simulation. The second step is the update of all the system variables after the clock time advancement. All the mathematical computations performed in the following sections are defined in the extended real number system $\bar{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$.

4.1. Definition of the simulation time advancement

The first step of the continuous-flow simulation is the definition of the next event time t_{k+1} related to a state transition of a machine or to a change of the material flow in one or more machines in the system. The next event time t_{k+1} is equal to:

$$t_{k+1} = t_k + \Delta t_k \tag{4}$$

where Δt_k is the time period before the occurrence of the next event in the system. After the identification at time t_k of the next event that will occur, the clock time is immediately advanced at the time t_{k+1}.

The next event occurring at time t_{k+1} can be of three different types: a state transition of a machine in the system, the fulfilment or depletion of a buffer in the system, and the end of the simulation. The time period Δt_k corresponds to the minimum time before the occurrence of any event among these three types.

Therefore, for the computation of Δt_k it is necessary to define:

- the remaining machine time RM(m, t_k) of each machine in the system, that is the time before the occurrence of a state transition for each machine M$\{m\}$;
- the remaining buffer time RB(b, t_k) of each buffer, that is the time before the fulfilment or depletion of each buffer B$\{b\}$;
- the remaining simulation time $T - t_k$, that is the time before the end of the simulation at time T.

According to this, the time period Δt_k is formalized as follows:

$$\Delta t_k = \min\{\mathrm{RM}(1,t_k), \dots, \mathrm{RM}(M,t_k), \mathrm{RB}(1,t_k), \dots, \mathrm{RB}(B,t_k), T - t_k\}, \tag{5}$$

with RM(m, t_k) and RB(b, t_k) defined in the following sections.

4.2. Remaining machine time

For each machine M$\{m\}$, the remaining machine time RM(m, t_k) that will take before the occurrence of the next event related to a change of the machine state must be defined. Assuming that at time t_k the machine M$\{m\}$ is in the state $S_i\{m\}$, for the definition of RM(m, t_k), it is necessary to know the time period between t_k and the occurrence of a transition from the state $S_i\{m\}$ to each other state $S_j\{m\}$ with $i \neq j$. Indeed RM(m, t_k) will correspond to the minimum among all these time periods.

According to this, a time to transition matrix $\tau(m, t_k)$ is introduced such that each entry $\tau_{i,j}(m, t_k)$ defines the time period between t_k and the occurrence of a transition from the state $S_i\{m\}$ to the state $S_j\{m\}$ of the machine M$\{m\}$. Each entry $\tau_{i,j}(m, t_k)$ is defined under the assumption that at t_k the effective production rate $\mu(m, t_k)$ is equal to the nominal production rate $\mu_i\{m\}$. The time to transition matrix $\tau(m, t_k)$ is a square matrix with size $I\{m\} \times I\{m\}$, with $I\{m\}$ the number of states of M$\{m\}$ as defined in Section 3.1. When the simulation starts at t_0, the value of each entry $\tau_{i,j}(m, t_k)$ with $i \neq j$ is initialized with random variables simulated according to the probability distribution $D_{i,j}\{m\}$ of the time to transition from $S_i\{m\}$ to $S_j\{m\}$. Each entry $\tau_{i,i}(m, t_k)$ is set equal to $+\infty$ since it does not represent any change of state of the machine M$\{m\}$.

If at time t_k the machine M$\{m\}$ is in the state $S_i\{m\}$, the times to transition related to the changes of state that can occur from $S_i\{m\}$ are all the entries in the ith row of the matrix $\tau(m, t_k)$. Therefore, the remaining machine time RM(m, t_k) is the minimum value among $\tau_{i,1}(m, t_k), \dots, \tau_{i,I\{m\}}(m, t_k)$, divided by the capacity saturation ratio $c(m, t_k)$ of M$\{m\}$ at time t_k. The capacity saturation ratio of a machine M$\{m\}$ considers any possible slowdown, starvation or blocking of M$\{m\}$ at time t_k. If at time t_k the machine M$\{m\}$ is not slowed down, blocked or starved by other machines, $c(m, t_k) = 1$. Otherwise, $c(m, t_k)$ is equal to the ratio between the effective production rate and the nominal production rate of M$\{m\}$ at t_k. That is:

$$c(m, t_k) = \begin{cases} 1, & \text{if } \mu(m, t_k) = \mu\{m\} \cdot s(m, t_k) \\ \frac{\mu(m,t_k)}{\mu\{m\} \cdot s(m,t_k)}, & \text{otherwise.} \end{cases} \tag{6}$$

Consequently, the remaining machine time RM(m, t_k) is computed as:

$$\mathrm{RM}(m, t_k) = \frac{\min\{\tau_{i,1}(m,t_k), \dots, \tau_{i,I\{m\}}(m,t_k)\}}{c(m, t_k)}, \qquad \text{with } i : s_i(m, t_k) = 1. \tag{7}$$

Each entry of $\tau(m, t_k)$ has been defined under the assumption $\mu(m, t_k) = \mu\{m\} \cdot s(m, t_k)$. Accordingly, when the machine is slowed down, the remaining time before the occurrence of the operation dependent transition from $S_i\{m\}$ to $S_j\{m\}$ is exhausted more slowly. The transition will occur after the time period $\frac{\tau_{i,j}(m,t_k)}{c(m,t_k)}$ with $c(m, t_k) < 1$, if no other events occur in the meanwhile. If at time t_k the machine is blocked or starved, that is if $c(m, t_k) = 0$, the remaining machine time RM(m, t_k) is equal to $+\infty$. In this case, there will be no state transitions of the machine M$\{m\}$ until an event that will change its effective production rate occurs in the simulation.

Eqs. (6) and (7) also address the case of time dependent transitions, which are not affected by the effective production rate of the machine.

According to Section 3.3, only the transitions from down states are assumed to be time-dependent. Since in a down state $\mu(m,t_k) = \mu\{m\} \cdot s(m,t_k) = 0$, the capacity saturation ratio $c(m,t_k)$ is equal to 1 in Eq. (6), such that $c(m,t_k)$ does not affect the computation of $\mathrm{RM}(m,t_k)$ in Eq. (7).

4.3. Remaining buffer time

The remaining buffer time $\mathrm{RB}(b,t_k)$ of a buffer $\mathrm{B}\{b\}$ is defined as the time period before the fulfilment or depletion of $\mathrm{B}\{b\}$. To define $\mathrm{RB}(b,t_k)$, the vector $\beta(b,t_k)$ of size 1×2 is introduced such that the entry $\beta_1(b,t_k)$ defines the time to fulfil $\mathrm{B}\{b\}$, while the entry $\beta_2(b,t_k)$ defines the time to deplete $\mathrm{B}\{b\}$. The values of $\beta_1(b,t_k)$ and $\beta_2(b,t_k)$ depend on the current buffer level $x(b,t_k)$ and on the effective production rates of the immediately upstream and downstream machines $\mathrm{M}\{m\}$ and $\mathrm{M}\{m'\}$ such that:

$$\beta_1(b,t_k) = \begin{cases} \frac{N\{b\}-x(b,t_k)}{\mu(m,t_k)-\mu(m',t_k)}, & \text{if } \mu(m,t_k) > \mu(m',t_k) \\ +\infty, & \text{otherwise} \end{cases} \tag{8}$$

$$\beta_2(b,t_k) = \begin{cases} \frac{x(b,t_k)-L\{b\}}{\mu(m',t_k)-\mu(m,t_k)}, & \text{if } \mu(m,t_k) < \mu(m',t_k) \\ +\infty, & \text{otherwise} \end{cases} \tag{9}$$

with m and m' such that $F_{b,m} = 1$ and $F_{b,m'} = -1$.

In Eq. (8), the time to fulfil the buffer $\beta_1(m,t_k)$ is set equal to $+\infty$ if at t_k the upstream machine $\mathrm{M}\{m\}$ is not faster than the downstream machine $\mathrm{M}\{m'\}$. Similarly, the time to deplete the buffer $\beta_2(m,t_k)$ is set equal to $+\infty$ in Eq. (9) if at t_k the downstream machine $\mathrm{M}\{m'\}$ is not faster than the upstream machine $\mathrm{M}\{m\}$.

The remaining buffer time $\mathrm{RB}(b,t_k)$ is equal to the minimum value between the time to fulfil the buffer and the time to deplete the buffer:

$$\mathrm{RB}(b,t_k) = \min\{\beta_1(b,t_k), \beta_2(b,t_k)\}. \tag{10}$$

Since the remaining machine times and the remaining buffer times depend on the effective production rates at time t_k of the machines in the system, the computation of $\mu(m,t_k)$ for each machine $\mathrm{M}\{m\}$ must be defined in order to finally solve Eq. (5) of the time period Δt_k and Eq. (4) of the next event time t_{k+1}.

4.4. Propagation analysis of flow interruptions and slowdowns

As explained in Section 3.3, the identification of the machines causing the slowdown, blocking or starvation of any other machine in the system and then the computation of the effective production rate of each machine at each event time t_k are complicated tasks when systems have a general layout, and they can increase the time to run the continuous-flow simulation significantly. In this section, a new methodology based on the graph theory is presented to address this problem.

In order to identify the machines causing the slowdown, blocking or starvation in a general layout, the adjacency matrix $V(t_k)$ and the reachability matrix $W(t_k)$ are introduced in the continuous-flow simulation method. By means of $V(t_k)$ and $W(t_k)$, all the machines causing slowdown, blocking or starvation of the material flow anywhere in the system are simultaneously identified in few simple steps and then the computation of the effective production rates of all the machines is quickly performed.

Definition 1 (*Adjacency Matrix*)**.** At each event time t_k, the adjacency matrix $V(t_k)$ is a square Boolean matrix with size $M \times M$, where M is the number of machines in the system. Each entry $V_{m,m'}(t_k)$ in $V(t_k)$ is equal to 1 if at time t_k the effective production rate of the machine $\mathrm{M}\{m\}$ can be affected by the state of the immediately upstream/downstream machine $\mathrm{M}\{m'\}$, otherwise it is equal to 0. Therefore, each entry $V_{m,m'}(t_k)$ is defined as follows:

$$V_{m,m'}(t_k) = \begin{cases} 1, & \text{if } \exists b : \left(F_{b,m} = 1 \wedge F_{b,m'} = -1 \wedge x(b,t_k) = N\{b\}\right) \\ 1, & \text{if } \exists b : \left(F_{b,m} = -1 \wedge F_{b,m'} = 1 \wedge x(b,t_k) = L\{b\}\right) \\ 0, & \text{otherwise}, \end{cases} \quad \forall m, \forall m'. \tag{11}$$

According to Eq. (11), an entry $V_{m,m'}(t_k)$ is equal to 1 in two different cases. The first case occurs if the machine $\mathrm{M}\{m'\}$ is an immediately downstream machine of $\mathrm{M}\{m\}$, that is $\exists b : F_{b,m} = 1 \wedge F_{b,m'} = -1$, and if at time t_k the intermediate buffer $\mathrm{B}\{b\}$ is full, that is $x(b,t_k) = N\{b\}$. In this case $\mathrm{M}\{m\}$ can be potentially blocked or slowed down by $\mathrm{M}\{m'\}$. The second case occurs if the machine $\mathrm{M}\{m'\}$ is an immediately upstream machine of $\mathrm{M}\{m\}$, that is $\exists b : F_{b,m} = -1 \wedge F_{b,m'} = 1$, and if at time t_k the buffer $\mathrm{B}\{b\}$ is empty, that is $x(b,t_k) = L\{b\}$. In this case $\mathrm{M}\{m\}$ can be potentially starved or slowed down because of $\mathrm{M}\{m'\}$.

Definition 2 (*Reachability Matrix*)**.** At each event time t_k, the reachability matrix $W(t_k)$ is a square boolean matrix with size $M \times M$, where M is the number of machines in the system. Each entry $W_{m,m'}(t_k)$ in $W(t_k)$ is equal to 1 if at time t_k the effective production rate of the machine $\mathrm{M}\{m\}$ can be affected by the state of the machine $\mathrm{M}\{m'\}$ located anywhere in the system, and 0 otherwise.

To compute the reachability matrix $W(t_k)$, it is necessary to introduce and define in this work the system parameter M^*, that is the maximum length of propagation of a flow interruption or slowdown in the system.

Definition 3 (*Maximum Length Of Propagation*)**.** The maximum length M^* of propagation of a flow interruption or slowdown in a manufacturing system is defined as the length of longest sequence of adjacent and distinct machines, through which a flow interruption or slowdown can potentially propagate in the system. M^* is constant over the simulation time, and for any system with M machines the inequality $M^* \leq M-1$ is always verified.

Considering a serial line with M machines, a flow interruption or slowdown caused by the first machine $\mathrm{M}\{1\}$ can propagate at most to the last machine $\mathrm{M}\{M\}$ through all the other $M-2$ machines, such that $M^* = M-1$. Instead, when the machines are not serially arranged in the system, a sequence of M distinct and adjacent machines does not exist, and so a flow interruption/slowdown cannot sequentially propagate to $M-1$ machines, that is $M^* < M-1$. Therefore, it is proved that the inequality $M^* \leq M-1$ in Definition 3 is always verified for any system.

In this paper, M^* is computed by using the results from the graph theory. First of all, a graph G is defined with M nodes representing the M machines in the manufacturing system. In the graph G, an edge is added between each couple of nodes (m,m'), if in the manufacturing system a buffer $\mathrm{B}\{b\}$ exists between the machines $\mathrm{M}\{m\}$ and $\mathrm{M}\{m'\}$. For each couple of nodes (m,m'), all the possible paths connecting m and m' are found through the Depth First Search algorithm presented in [53]. The length of the longest path between m and m' is defined as $l_{m,m'}$. The maximum length of propagation M^* is equal to the maximum value among all the lengths $l_{m,m'}$:

$$M^* = \max_{m,m'} \left\{ l_{m,m'} : m = 1,\ldots,M \wedge m' = 1,\ldots,M \right\}. \tag{12}$$

At this point, the reachability matrix $W(t_k)$ is given in Lemma 1 as a function of $V(t_k)$ and M^*. Since Lemma 1 makes use of the boolean sum, the boolean product and the power of boolean matrices, the definitions of these boolean operations according to the work in [54] are reported in Appendix A.

Lemma 1. *Given at time t_k, the adjacency matrix $V(t_k)$ of a manufacturing system with M machines and with maximum length of propagation M^*, the reachability matrix $W(t_k)$ is computed as follows:*

$$W(t_k) = \left(I + V(t_k)\right)^{M^*} \tag{13}$$

where I is the identity matrix of size $M \times M$.

The proof of Eq. (13) in Lemma 1 is given in Appendix B by using the results from the graph theory. Indeed, the problem of defining if a machine M$\{m\}$ can be affected by the propagation of a flow interruption/slowdown from a machine M$\{m'\}$ is equivalent to the problem of defining if in a graph G a node m is reachable from a node m'.

Once $W(t_k)$ is calculated, the effective production rate $\mu(m,t_k)$ of each machine M$\{m\}$ can be computed as the minimum value among the nominal production rates at time t_k of all the machines M$\{m'\}$ which can potentially affect M$\{m\}$, that is:

$$\mu(m,t_k) = \min_{m'} \left\{ \mu\{m'\} \cdot s(m',t_k) \,:\, m' = 1,\dots,M \wedge W_{m,m'}(t_k) = 1 \right\}, \quad \forall m. \tag{14}$$

By computing $W(t_k)$ according to Eq. (13), it is possible to notice that the diagonal entries of $W(t_k)$ are all equal to 1. Therefore, in Eq. (14), the effective production rate of each machine M$\{m\}$ at time t_k depends also on the current state at t_k of the machine M$\{m\}$ itself.

With the definition of the effective production rates, all the variables required for the computation of the next event time t_{k+1} have been provided.

4.5. System state update

After the definition of the inter-event time Δt_k, the clock time of the simulation can be instantaneously advanced at the time $t_{k+1} = t_k + \Delta t_k$. All the variables describing the whole system must now be updated.

The new state $s(m,t_{k+1})$ of each machine M$\{m\}$ at t_{k+1} is $s(m,t_{k+1}) = s(m,t_k)$ if the event at time t_{k+1} is not a state transition of the machine M$\{m\}$, while $s_j(m,t_{k+1}) = 1$ and $s_i(m,t_{k+1}) = 0\ \forall i \neq j$ if the event at time t_{k+1} is a transition from a state $S_i\{m\}$ to the state $S_j\{m\}$ of M$\{m\}$.

The new level $x(b,t_{k+1})$ of each buffer B$\{b\}$ in the system is defined as the level of the buffer B$\{b\}$ at time t_k, plus the difference between the quantities of material processed by the upstream machine M$\{m\}$ and by the downstream machine M$\{m'\}$ in the time period Δt_k, that is:

$$x(b,t_{k+1}) = x(b,t_k) + \left(\mu(m,t_k) - \mu(m',t_k)\right) \cdot \Delta t_k \tag{15}$$

with m and m' such that $F_{b,m} = 1$ and $F_{b,m'} = -1$.

The matrices $V(t_{k+1})$ and $W(t_{k+1})$ and the effective production rate $\mu(m,t_{k+1})$ of each machine are updated as defined in the previous section.

The last variable that must be updated at time $t_k + 1$ is the time to transition matrix $\tau(m,t_{k+1})$ of each machine M$\{m\}$. Only the entries of $\tau(m,t_{k+1})$ in the row i corresponding to the state $S_i\{m\}$ of M$\{m\}$ at t_k must be computed, and each entry $\tau_{i,j}(m,t_{k+1})$ is computed as the difference between the time to transition $\tau_{i,j}(m,t_k)$ and the product between the inter-event time Δt_k and the capacity saturation ratio $c(m,t_k)$. Instead, all the other rows in $\tau(m,t_{k+1})$ are equal to the corresponding rows in $\tau(m,t_k)$. Therefore, each entry in the matrix $\tau(m,t_{k+1})$ is updated as follows:

$$\tau_{i,j}(m,t_{k+1}) = \begin{cases} \tau_{i,j}(m,t_k) - \Delta t_k \cdot c(m,t_k), & \text{if } s_i(m,t_k) = 1 \\ \tau_{i,j}(m,t_k), & \text{otherwise,} \end{cases} \quad \forall i, \forall j. \tag{16}$$

The update of the time to transition matrix $\tau(m,t_k+1)$ performed in Eq. (16) can also be implemented through the following matrix notation:

$$\tau(m,t_{k+1}) = \tau(m,t_k) - \Delta t_k \cdot c(m,t_k) \cdot \left[s(m,t_k),\dots,s(m,t_k)\right] \tag{17}$$

with the matrix $\left[s(m,t_k),\dots,s(m,t_k)\right]$ obtained by horizontally concatenating the state vector $s(m,t_k)$ with itself for a number of times equal to the number of states of the machine M$\{m\}$, that is $I\{m\}$ times.

If the event at time t_{k+1} is a transition of the machine M$\{m\}$ from the state $S_i\{m\}$ to a state $S_j\{m\}$, the corresponding entry $\tau_{i,j}(m,t_{k+1})$ results equal to 0 in Eq. (16). In this case, a new value of $\tau_{i,j}(m,t_{k+1})$ is randomly generated according to the probability distribution $D_{i,j}\{m\}$ of the time to transition from $S_i\{m\}$ to $S_j\{m\}$.

At this point, all the variables describing the system have been updated at time t_{k+1}. The clock time advancement described in Section 4.1 and the update of the system variables described in this section are then iteratively performed for $t_{k+2},\dots,t_K$ until the end of the simulation at time $t_K = T$.

4.6. Estimation of the system performance

The output of the continuous-flow simulation model is the event log with the state of the whole system described at any event time t_k. By means of the event log of a single simulation run, the main aggregated performance of a manufacturing system, such as the system throughput, the average buffer levels and the machine state probabilities can be easily computed. In the following, the formulas to compute the machine throughput, the average buffer level and the time percentages of full and empty buffer are provided as example.

Given the simulation time T of a simulation run with K events, the average throughput $th(m)$ of a machine M$\{m\}$ is

$$th(m) = \frac{\sum_{k=0}^{K-1} \mu(m,t_k) \cdot \Delta t_k}{T}, \tag{18}$$

the average buffer level $\bar{x}(b)$ of a buffer B$\{b\}$ is

$$\bar{x}(b) = \frac{\sum_{k=0}^{K-1} \left(x(b,t_k) + x(b,t_{k+1})\right) \cdot \Delta t_k}{2 \cdot T}, \tag{19}$$

the time percentage during which the buffer level of B$\{b\}$ is empty, that is the buffer level is equal to the lower boundary $L\{b\}$, is:

$$P\left(L\{b\}\right) = \frac{\sum_k \Delta t_k}{T} \cdot 100\%, \quad \text{with } k : \left(x(b,t_k) = L\{b\} \wedge x(b,t_{k+1}) = L\{b\}\right) \tag{20}$$

and the time percentage during which the buffer level of B$\{b\}$ is full is:

$$P\left(N\{b\}\right) = \frac{\sum_k \Delta t_k}{T} \cdot 100\%, \quad \text{with } k : \left(x(b,t_k) = N\{b\} \wedge x(b,t_{k+1}) = N\{b\}\right). \tag{21}$$

5. Analysis of a specific system with the continuous-flow simulation

In this section, the specific manufacturing system with general layout provided in Fig. 1 is analysed to explain in detail the computation of the effective production rates of the machines, which is the most critical step of the continuous-flow simulation. In this example, the simulation time T is assumed to be equal to 10^5 time units. In Section 5.1, the clock time is assumed to have been advanced until the 8th event in the simulation at time $t_k = t_8 = 129.7$, and the numerical values of the variables describing the system at this time are provided. In Section 5.2 the computation of the effective production rates is performed. In Appendix C, the computation of the next event time $t_{k+1} = t_9$ and the update of the variables describing the system are given.

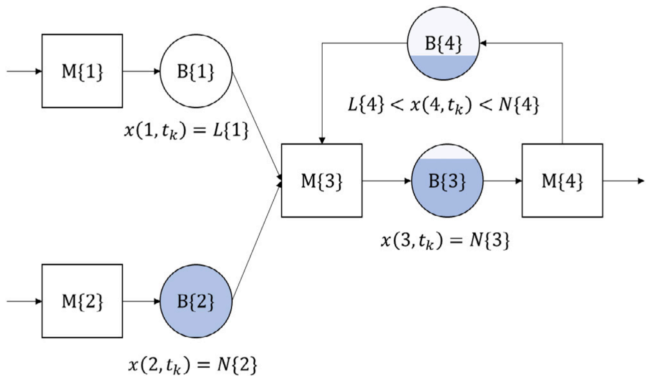


Fig. 2. Example of the buffer levels at time $t_k = t_8$ in a general network.

5.1. System variables

The following sets of states $S\{1\},\dots,S\{4\}$ and row vectors of nominal production rates $\mu\{1\},\dots,\mu\{4\}$ are assumed to represent the behaviour in isolation of the corresponding machines M{1}, …, M{4} in the system:

$$\begin{aligned}
&S\{1\} = \{S_1\{1\}, S_2\{1\}\}, && \mu\{1\} = [\mu_1\{1\}, \mu_2\{1\}] = \begin{bmatrix}1.3 & 0\end{bmatrix}\\
&S\{2\} = \{S_1\{2\}, S_2\{2\}\}, && \mu\{2\} = [\mu_1\{2\}, \mu_2\{2\}] = \begin{bmatrix}1.7 & 0\end{bmatrix}\\
&S\{3\} = \{S_1\{3\}, S_2\{3\}, S_3\{3\}\}, && \mu\{3\} = [\mu_1\{3\}, \mu_2\{3\}, \mu_3\{3\}] = \begin{bmatrix}2.1 & 0 & 0\end{bmatrix}\\
&S\{4\} = \{S_1\{4\}, S_2\{4\}\}, && \mu\{4\} = [\mu_1\{4\}, \mu_2\{4\}] = \begin{bmatrix}1.9 & 0\end{bmatrix}.
\end{aligned}$$

Therefore, in this example machine M{3} has a single up state and two down states, while all the other machines have a single up and a single down state. The values provided for each nominal production rate are expressed in material unit per time unit.

At time t_8 in the simulation, the machines are assumed in the following states:

$$s(1,t_8) = \begin{bmatrix}1\\0\end{bmatrix}, \quad s(2,t_8) = \begin{bmatrix}1\\0\end{bmatrix}, \quad s(3,t_8) = \begin{bmatrix}1\\0\\0\end{bmatrix}, \quad s(4,t_8) = \begin{bmatrix}0\\1\end{bmatrix}$$

that is machines M{1}, M{2} and M{3} are respectively in the up states $S_1\{1\}$, $S_1\{2\}$ and $S_1\{3\}$, while M{4} is in the down state $S_2\{4\}$.

For reason of simplicity, the lower boundary of each buffer in the system as been assumed equal to 0, while the upper boundary and the current level at time t_8 have been set as follows:

$$\begin{array}{llll}
L\{1\} = 0, & L\{2\} = 0, & L\{3\} = 0, & L\{4\} = 0.\\
N\{1\} = 8, & N\{2\} = 10, & N\{3\} = 15, & N\{4\} = 15.\\
x(1,t_8) = 0, & x(2,t_8) = 10, & x(3,t_8) = 12, & x(4,t_8) = 5.
\end{array}$$

So at time t_8, the buffer B{1} is empty, the buffer B{2} is full, and the buffers B{3} and B{4} are partially full. Fig. 2 provides a graphical representation of the above buffer levels at time t_8.

The layout of the above system is formalized in the flow matrix F, which has size $B \times M = 4 \times 4$. Eq. (3) defines that each entry $F_{b,m}$ in the matrix F, with $b = 1,\dots,B$ and $m = 1,\dots,M$, is equal to 1 if the machine M{m} feeds the buffer B{b}, is equal to −1 if the buffer B{b} feeds the machine M{m}, and 0 if no direct flow of material exists between M{m} and B{b}. According to this definition, the flow matrix F of this specific case results:

$$F = \begin{bmatrix}1 & 0 & -1 & 0\\ 0 & 1 & -1 & 0\\ 0 & 0 & 1 & -1\\ 0 & 0 & -1 & 1\end{bmatrix}.$$

5.2. Computation of the effective production rates

By knowing the current states of the machines and the current levels of the buffers, the effective production rates $\mu(1,t_8),\dots,\mu(4,t_8)$ of the machines at time t_8 can be computed through the new set of equations defined in Section 4.4. First of all, the machines which can potentially affect at time t_8 the effective production rate of their immediately upstream or downstream machines are identified through the adjacency matrix $V(t_8)$. According to the definition in Eq. (11), the matrix $V(t_8)$ of this example has size $M \times M = 4 \times 4$, and it results:

$$V(t_8) = \begin{bmatrix}0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\\ 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}.$$

The entry $V_{2,3}(t_8)$ in this example is equal to 1, since the buffer B{2} is full and so the machine M{2} can be blocked or slowed down by the machine M{3}. Also the entry $V_{3,1}(t_8)$ is equal to 1, since the buffer B{1} is empty and so the machine M{3} can be starved or slowed down by the machine M{1}.

However, the matrix $V(t_8)$ indicates only the local limitations of a machine caused by its immediately upstream and downstream machines, without considering the limitations caused by more distant machines. For this reason, the reachability matrix $W(t_8)$ is computed. $W(t_8)$ identifies which machines can be potentially affected at time t_8 by any other machine anywhere in the system.

To compute $W(t_8)$ according to Eq. (13), the maximum length of propagation of a flow interruption/slowdown M^* must be defined. Since M^* does not depend on the event time t_k, it is defined only once before running the simulation. In the specific system analysed in this section, it is possible to state that $M^* = 2$ even without applying the Depth First Search algorithm used in Section 4.4. Indeed, the longest propagation paths in the system of Fig. 2 are M{1} → M{3} → M{4} and the opposite one (with length 2), M{1} → M{3} → M{2} and the opposite one (with length 2), M{2} → M{3} → M{4} and the opposite one (with length 2). Instead, all the propagation paths starting from M{3} have length equal to 1.

Therefore, according to Eq. (13), $W(t_8)$ is defined as:

$$W(t_8) = \left(\mathrm{I} + V(t_8)\right)^2 = \mathrm{I} \vee V(t_8) \vee V(t_8)^2$$

where

$$\begin{aligned}
V(t_8)^2 &= V(t_8) \wedge V(t_8)\\
&= \begin{bmatrix}
\bigvee_{m=1}^{4}\left(V_{1,m}(t_8) \wedge V_{m,1}(t_8)\right) & \dots & \bigvee_{m=1}^{4}\left(V_{1,m}(t_8) \wedge V_{m,4}(t_8)\right)\\
\vdots & \ddots & \vdots\\
\bigvee_{m=1}^{4}\left(V_{4,m}(t_8) \wedge V_{m,1}(t_8)\right) & \dots & \bigvee_{m=1}^{4}\left(V_{4,m}(t_8) \wedge V_{m,4}(t_8)\right)
\end{bmatrix},
\end{aligned}$$

and it results:

$$\begin{aligned}
W(t_8) &= \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \vee \begin{bmatrix}0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\\ 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix} \vee \begin{bmatrix}0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}\\
&= \begin{bmatrix}1 & 0 & 0 & 0\\ 1 & 1 & 1 & 0\\ 1 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}.
\end{aligned}$$

Looking at the matrix $W(t_8)$, since the entries $W_{2,1}(t_8)$ and $W_{2,3}(t_8)$ are equal to 1, the effective production rate $\mu(2,t_8)$ of the machine M{2} can be affected by the machine M{1} or by the machine M{3}. Similarly, the effective production rate $\mu(3,t_8)$ of the machine M{3} can be affected by the machine M{1}, since $W_{3,1}(t_8) = 1$. Moreover, each entry in the diagonal of $W(t_8)$, that is $W_{1,1}(t_8),\dots,W_{4,4}(t_8)$, is equal to 1 since the effective production rate of each machine depends also on its own state at time t_8.

According to Eq. (14), the effective production rate of each machine results:

$$\begin{aligned}
\mu(1,t_8) &= \mu\{1\}\cdot s(1,t_8) = 1.3,\\
\mu(2,t_8) &= \min\big\{\mu\{1\}\cdot s(1,t_8), \mu\{2\}\cdot s(2,t_8), \mu\{3\}\cdot s(3,t_8)\big\}\\
&= \min\big\{1.3, 1.7, 2.1\big\} = 1.3,\\
\mu(3,t_8) &= \min\big\{\mu\{1\}\cdot s(1,t_8), \mu\{3\}\cdot s(3,t_8)\big\} = \min\big\{1.3, 2.1\big\} = 1.3,\\
\mu(4,t_8) &= \mu\{4\}\cdot s(4,t_8) = 0.
\end{aligned}$$

Therefore, the machines M$\{2\}$ and M$\{3\}$ are slowed down at the same production rate of the machine M$\{1\}$. For each machine, the capacity saturation ratio is computed according to Eq. (6) and it results:

$$c(1,t_8) = 1, \quad c(2,t_8) = \frac{\mu(2,t_8)}{\mu\{2\}\cdot s(2,t_8)} = \frac{1.3}{1.7} = 0.76,$$
$$c(3,t_8) = \frac{\mu(3,t_8)}{\mu\{3\}\cdot s(3,t_8)} = \frac{1.3}{2.1} = 0.62, \quad c(4,t_8) = 1.$$

At this point, the next event time in the simulation can be defined. The numerical computation for this example is provided in Appendix C.

6. Numerical results

In the numerical experiments, the continuous-flow simulation model presented in this work has been used to evaluate the production performance of 8 cases related to 4 different system layouts: serial lines; assembly and disassembly lines; single loop lines; multiple loop lines with assembly and disassembly machines. Each system layout has been simulated with both a small number of machines (5) and a large number of machines (50) in the system, and with simulation time of 10^5 time units. For each case, 5 replicates of the simulation have been run.

The simulations of all the tested cases have been carried out using a computer with 2.5 GHz Intel core i7 processor with 8 GB RAM. The continuous-flow simulation has been implemented in a MATLAB function (R2021b release). The results of the continuous-flow simulation have been compared with the results of DES models with discrete flow of parts, which is the most employed simulation method for the analysis of manufacturing systems. The DES models have been developed in the industrial grade MATLAB/Simulink software package (R2021b release).

In the following, the generation of the tested cases is explained.

6.1. Generation of the tested cases

The computational performance of the continuous-flow simulation have been tested over 8 different cases, which consider 4 different layouts of a manufacturing system. The tested cases are summarized in Table 1, in which the columns M and B refer respectively to the number of machines and buffers in the system.

In the tested cases, each machine in the system has a single up state and a single down state. The production rate of each machine in the up state is deterministic and it has been randomly generated from the continuous uniform distribution $U(1,3)$ $[\frac{\text{material}}{\text{time}}]$. The time to failure and the time to repair are exponentially distributed, with failure rate randomly generated from the distribution $U(0.005, 0.02)$ and repair rate randomly generated from $U(0.05, 0.2)$ $[\frac{1}{\text{time}}]$. Therefore, the efficiency of a machine in isolation can vary from 60% to 97.5%, while the throughput of the machines in isolation varies from 0.6 to 2.925 $[\frac{\text{material}}{\text{time}}]$.

In the serial line, the material flows sequentially through all the M machines of the system, which are partially decoupled by $B = M - 1$ finite buffers. The tested cases for the serial line layout are Case 1 and Case 2 in Table 1. In the closed single loop systems the number of machines M is equal to number of buffers B, and the material flows sequentially from the first machine M$\{1\}$ to the last machine M$\{M\}$ through $M - 1$ buffers, and then it flows back from the last machine M$\{M\}$ to the first machine M$\{1\}$ through the last buffer B$\{B\}$. The tested cases for the single loop layout are Case 3 and Case 4 in Table 1.

Differently from the serial lines and the closed single loops, in a general system with assembly and disassembly machines, the way in which the machines and the buffers are arranged is not uniquely defined for any system. In this work, the continuous-flow model has been tested in the analysis of the small assembly/disassembly system shown in Fig. 3(a), and in the analysis of the large assembly/disassembly system shown in Fig. 3(c) (Cases 5 and 6 in Table 1). The system in Fig. 3(a) is composed by 5 machines, with one assembly/disassembly machine M$\{3\}$, and 4 finite buffers.

The assembly/disassembly system in Fig. 3(c) is composed at the first stage by 16 machines and their own downstream buffers. The second stage is composed by 8 assembly machines which combine the flows of material from couples of buffers of the first stage. Each machine at the second stage has a downstream dedicated buffer. The flows of material from the buffers at the second stage are then combined by the assembly machine M$\{25\}$ at the third stage. The combined flow goes through the buffer B$\{25\}$ and then is divided by the disassembly machine M$\{26\}$ in 8 different flows of material. These flows of material go through 8 dedicated buffers and then they are processed by 8 disassembly machines, which divide each flow in two separated flows of materials. Each one of the 16 flows of material goes through a dedicated buffer and it is finally processed by a machine at the last stage of the system.

In the serial lines, the closed single loops and the assembly/disassembly systems the capacity of each buffer has been randomly generated from the discrete uniform distribution $Ud(2,10)$.

The multiple loop systems tested in this work (Case 7 and 8 in Table 1) are represented in Fig. 3(b) and Fig. 3(d), and they are composed by two internal loops and by an external loop connecting the two internal loops. Each internal loop of the system in Fig. 3(b) is composed by an assembly machine, a disassembly machine and two buffers. In the multiple loop system of Fig. 3(d), each internal loop is composed by 24 machines, among which an assembly machine and a disassembly machine, and 24 buffers.

The capacities of all the buffers in Fig. 3(b) and of the buffers in the internal loops of Fig. 3(d) have been randomly generated from the same discrete uniform distribution $Ud(2,10)$. The buffers B$\{25\}$, B$\{26\}$, B$\{51\}$ and B$\{52\}$ of the external loop of the system in Fig. 3(d) have larger capacity, with values randomly generated from the discrete uniform distribution $Ud(20,40)$.

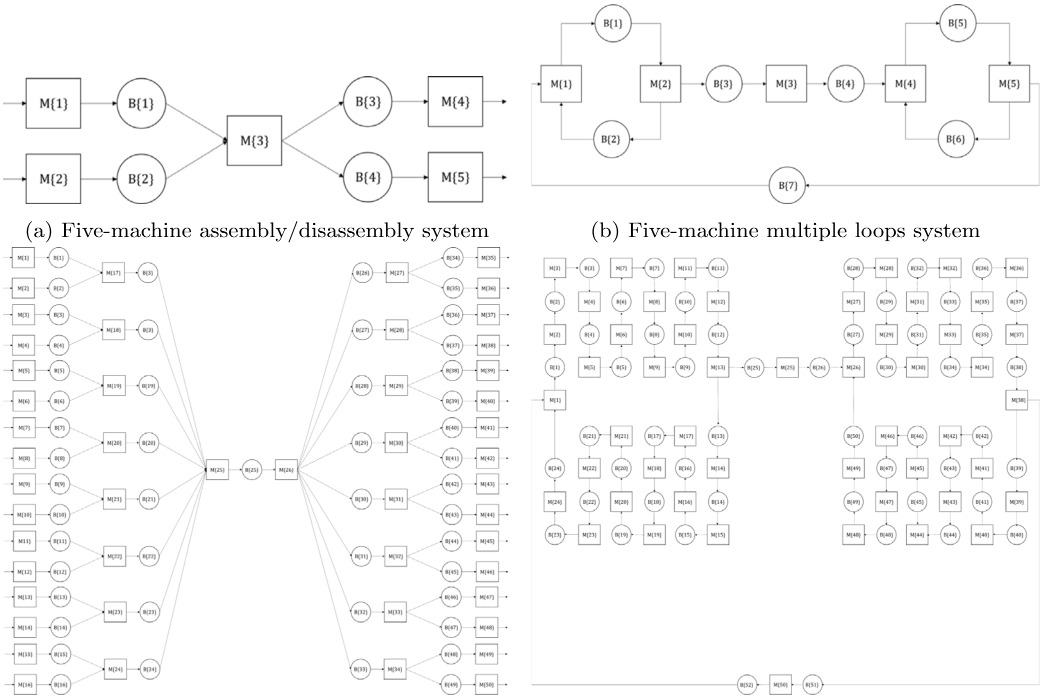
In any system with one or more closed loops, the quantity of material flowing in each loop is constant over the time and it is called population of the loop. For the tested cases with single or multiple loops, the population of each loop has been randomly generated such that it saturates from 60% to 80% the overall capacity of the buffers in the loop.

6.2. Computational time and accuracy

The results of the continuous-flow simulation (CF simulation) in the analysis of the tested cases have been compared with the results of a discrete-flow simulation (DCF simulation) that approximates a continuous material flow. To approximate a continuous flow of material with a discrete flow, each discrete part processed in the DCF simulation represents 0.1 units of continuous material.

In Table 1, the results for the tested cases described in the previous section are given. The column Comp Time of Table 1 provides the computational time of the two simulation methods. As shown in the table, the CF simulation is from 57 to 249 times faster than the DCF simulation. This result is related to the number of events generated in a run by the two simulation methods. Indeed, the number of events generated by the CF simulation is on average 150 times lower than the number of events generated in the DCF simulation.

The columns $\Delta th\%$ and $\Delta\bar{x}\%$ show the accuracy of the continuous-flow simulation in the evaluation of the throughput and the average buffer levels of the system. The percentage difference between the CF



(a) Five-machine assembly/disassembly system

(b) Five-machine multiple loops system

(c) Fifty-machine assembly/disassembly system

(d) Fifty-machine multiple loops system

Fig. 3. Layouts of the tested cases.

Table 1
Performance of the simulation methods in the analysis of continuous-flow systems.

Layout	Case	M	B	Comp time [s]		Events		$\Delta th\%$	$\Delta \bar{x}\%$
				CF	DCF	CF	DCF		
Serial line	1	5	4	3	185	$1.4{\cdot}10^4$	$2.2{\cdot}10^6$	0.64	0.82
	2	50	49	44	2558	$1.4{\cdot}10^5$	$2.1{\cdot}10^7$	0.72	0.74
Assembly/ Disassembly	3	5	4	2	202	$1.5{\cdot}10^4$	$2.2{\cdot}10^6$	0.02	0.59
	4	50	49	25	2452	$1.1{\cdot}10^5$	$1.6{\cdot}10^7$	0.55	0.83
Single loop	5	5	5	2	204	$1.4{\cdot}10^4$	$3.5{\cdot}10^6$	0.17	1.42
	6	50	50	46	2631	$1.4{\cdot}10^5$	$1.8{\cdot}10^7$	0.56	0.66
Multiple loops	7	5	7	2	453	$1.2{\cdot}10^4$	$9.5{\cdot}10^5$	0.04	0.06
	8	50	52	45	9425	$1.4{\cdot}10^5$	$1.5{\cdot}10^7$	0.68	0.84

simulation and the DCF simulation in the throughput analysis is on average 0.42%, while the percentage difference in the buffer level analysis is on average 0.74%. These results show the high accuracy of the proposed continuous-flow simulation in the analysis of manufacturing systems and in the evaluation of the main production performance. More precisely, the small difference between the CF simulation and the DCF simulation in the evaluation of the production performance is much more likely to be ascribed to the approximation introduced by the DCF simulation in discretizing the continuous flow of material.

Fig. 4 shows the computational time of the continuous-flow simulation for the four types of layout tested in the experimental campaign, as the number of machines in the system changes. In all these experiments, the simulation time has been defined equal to 10^5 time units. From these results, the computational time increases almost linearly as the number of machines in the system increases, and the slope of the curve depends on the system layout. The explanation of these patterns is related to the computation of the reachability matrix $W(t_k)$ in Eq. (13), which is the most time consuming computation in the whole method. Indeed, the time to compute the power of a matrix depends on the matrix size, which in Eq. (13) is $M \times M$ with M the number of machines in the system, and on the power, which in Eq. (13) is the parameter M^* that depends on the system layout. Instead, the effects of all the other parameters of the system, such as the number of buffers or the number of states of each machine, are negligible since they do not affect Eq. (13).

The continuous-flow simulation presented in this work has also been tested in the analysis of discrete-flow systems. The tested layouts are the same system layouts presented in Section 6.1. The production performance of the discrete-flow system has been evaluated by means of a discrete-flow simulation model (DF simulation). Table 2 shows the computational time and the accuracy of the CF simulation in the analysis of the discrete-flow system, with respect to the performance of the DF simulation. From the results, the number of events generated by the CF simulation in a simulation run is on average an order of magnitude lower than the number of events generated by the DF simulation. Consequently, the CF simulation is from 6 to 21 times faster than the DF simulation.

The differences between the performance estimation of the CF simulation and the DF simulation define the accuracy of the CF simulation in the analysis of a discrete-flow system. Column $\Delta th\%$ in Table 2 shows that the continuous approximation of a discrete material flow does not have a big impact on the estimation of the throughput of a discrete-flow system, since the average percentage error on the throughput is 1.45%, with a maximum error of 2.96%. On the other hand, the average percentage error on the average buffer level is 5.13%, with maximum error of 13.83% (column $\Delta \bar{x}\%$ in Table 2). Nevertheless, this error on

Table 2
Performance of the simulation methods in the analysis of discrete-flow systems.

Layout	Case	M	B	Comp time [s]		Events		$\Delta th\%$	$\Delta \bar{x}\%$
				CF	DF	CF	DF		
Serial line	1	5	4	3	23	$1.4{\cdot}10^4$	$2.1{\cdot}10^5$	0.89	5.85
	2	50	49	44	276	$1.4{\cdot}10^5$	$2.0{\cdot}10^6$	2.39	4.55
Assembly/ Disassembly	3	5	4	2	24	$1.5{\cdot}10^4$	$2.1{\cdot}10^5$	0.81	5.15
	4	50	49	25	266	$1.1{\cdot}10^5$	$1.5{\cdot}10^6$	2.96	5.57
Single loop	5	5	5	2	27	$1.4{\cdot}10^4$	$3.5{\cdot}10^5$	0.92	13.83
	6	50	50	46	318	$1.4{\cdot}10^5$	$1.8{\cdot}10^6$	1.23	3.29
Multiple loops	7	5	7	2	39	$1.2{\cdot}10^4$	$9.4{\cdot}10^4$	0.59	0.07
	8	50	52	45	399	$1.4{\cdot}10^5$	$1.5{\cdot}10^6$	1.80	2.73

Table 3
Machines parameters of the industrial case.

Station	Machine	Cycle time [t.u.]	MTTF [t.u.]	MTTR [t.u.]
ST1	A	5.06	1834.41	19.82
	B		259.02	11.04
	C		2234.97	37.72
ST2	D	5.75	147.10	8.32
	E		324.31	8.64
	F		519.50	7.07
	G		851.90	3.76
ST3	H	3.85	274.39	8.08
	I		9183.66	32.10
ST4	R	3.85	4371.43	2.39

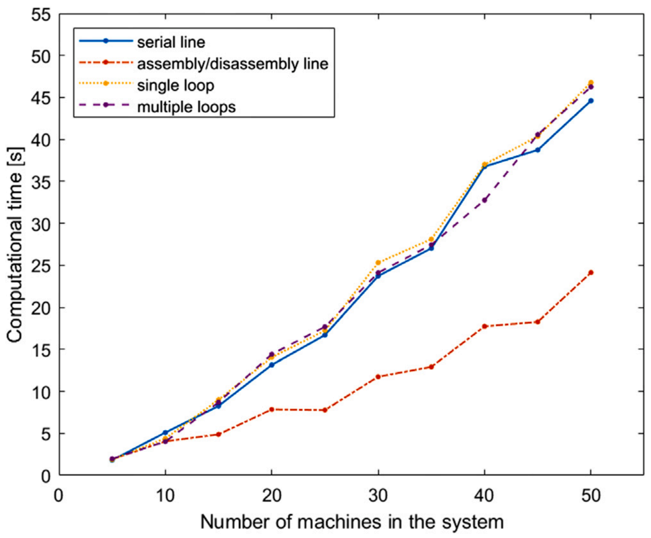


Fig. 4. The computational time of the continuous-flow simulation for different layouts as the number of machines in the system changes.

the average buffer level is in line with the errors of the continuous-flow analytical models proposed in the literature, and it can be reduced by replacing the formula in Eq. (19) with a formula discretizing the level of the buffer in the computation of the average buffer level, such as the formula proposed in [15].

According to these results, the continuous-flow simulation allows to analyse discrete-flow systems with general layout in much shorter time than the DES with discrete flow of parts, without significantly affecting the accuracy in the performance estimation.

7. Analysis of an industrial case

The proposed simulation method has been applied in an industrial case from the manufacturing industry, with the aim to analyse and optimize the production performance of a manufacturing system during the reconfiguration phase. The system under analysis is a highly automated production line of an Italian manufacturer of drawers for kitchens. In particular, the analysed production line performs the assembly of the drawer sides. The manufacturing company wants to automate the loading operation of parts on pallets at the beginning of the assembly line. The main solution that the company wants to evaluate is the exploitation of the unloading robot at the end of the line also to perform the loading operation of parts on pallets at the first machine. In this way, the operator that is currently dedicated to the manual loading of parts on pallets could be fully employed in other tasks, such as fixing minor failures of the machines in the line.

7.1. “As is” configuration of the production line

The production line is composed by four stations ST1, ST2, ST3 and ST4, which are partially decoupled by three inter-operational buffers B$\{1\}$, B$\{2\}$, B$\{3\}$ with capacities equal to 4, 4 and 7 respectively. In the first two stations of the line, components are assembled to the main body of the drawer side. In station ST3 additional components are welded to the main body and an automated visual check with other final operations are performed. In station ST4 the unloading robot R automatically unloads the drawer sides from the pallets. In the current configuration of the system, at the beginning of the line the main body of the drawer sides is manually loaded on a pallet by a dedicated operator, such that the line is almost never short of parts to process.

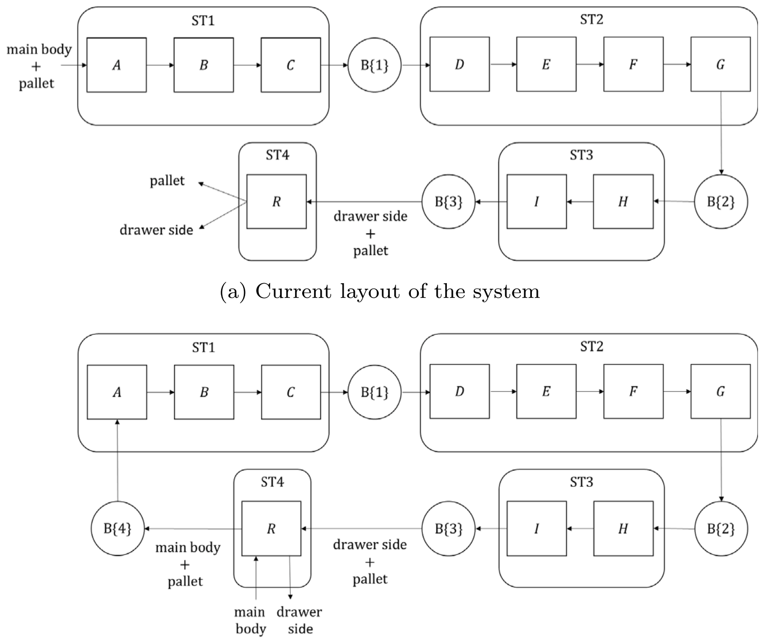
Stations ST1, ST2 and ST3 are composed by several machines, which are labelled for confidentiality reasons from A to I plus the unloading robot R. Machines in the same station have the same cycle time and they are serially arranged. Fig. 5(a) shows a graphical representation of the current layout of the production system.

Table 3 shows the cycle time, the mean time to failure (MTTF) and the mean time to repair (MTTR) provided by the company for each machine of the production line. For confidentiality reasons, the numerical values of these parameters have been anonymized and they have reported with the general unit of measurement “time units” (t.u.).

The blocking rule applied to each machine in the line is the Blocking After Service (BAS). According to the BAS rule, a machine is defined blocked when it cannot unload a processed part since the immediately downstream portion of the line is full.

From the data provided by the company, at the moment the production line is characterized by a steady state throughput equal to 0.156 $[\frac{\text{drawer sides}}{\text{t.u.}}]$. In order to apply the continuous-flow simulation method in the evaluation of the reconfiguration actions on the production line, the method has been validated by evaluating the throughput of the current configuration.

For all the machines $A, \dots, I$ and the robot R in the real system, the machines M$\{A\}, \dots,$ M$\{I\}$ and M$\{R\}$ are defined in the simulation model. Each machine M$\{m\}$ among the above machines has been defined with a single up state $S_1\{m\}$ and a single down state $S_2\{m\}$. In the up state, each machine processes parts with nominal production rate $\mu\{S_1\{m\}\}$ defined as the reciprocal of the machine cycle time given in Table 3. Since for each machine of the system only the values



(a) Current layout of the system

(b) New layout of the system

Fig. 5. Reconfiguration of a production system for the assembly of drawer sides.

of the MTTF and the MTTR have been provided by the company, in the simulation model the time to failure from the state $S_1\{m\}$ to the state $S_2\{m\}$ and the time to repair from $S_2\{m\}$ to $S_1\{m\}$ have been assumed exponentially distributed, with failure rate $\frac{1}{\mathrm{MTTF}}$ and repair rate $\frac{1}{\mathrm{MTTR}}$.

When the BAS rule is used in a production system, the working position of each machine behaves like a small buffer of capacity equal to 1 part. For this reason, in the continuous-flow simulation model the buffers $\mathrm{B}\{A\}, \mathrm{B}\{B\}, \mathrm{B}\{D\}, \mathrm{B}\{E\}, \mathrm{B}\{F\}$ and $\mathrm{B}\{H\}$ with capacity equal to 1 have been added downstream of the corresponding machines, in order to model the working positions of these machines. Therefore, each buffer $\mathrm{B}\{b\}$ among the above buffers have lower boundary $L\{b\} = 0$ and upper boundary $N\{b\} = 1$. Similarly, in the continuous-flow simulation model the capacities of the buffers $\mathrm{B}\{1\}, \mathrm{B}\{2\}$ and $\mathrm{B}\{3\}$ have been increased by 1 unit in order to model the working position of their upstream machines $\mathrm{M}\{C\}, \mathrm{M}\{G\}$ and $\mathrm{M}\{I\}$. Therefore, buffers $\mathrm{B}\{1\}, \mathrm{B}\{2\}$ and $\mathrm{B}\{3\}$ have lower boundaries $L\{1\} = 0, L\{2\} = 0$ and $L\{3\} = 0$, and upper boundaries $N\{1\} = 5, N\{2\} = 5$ and $N\{3\} = 8$.

By evaluating the production line with the continuous-flow simulation method presented in this paper, the steady state throughput of the system results 0.159 $\frac{\text{drawer sides}}{\text{t.u.}}$. The system has been simulated over 5 replicates, each one with simulation time equal to 10^6 t.u. The percentage error on the throughput estimation with respect to the value provided by the company is 1.92%. This error is in line with the level of accuracy shown by the model in Section 6.2 in the analysis of discrete-flow systems. Therefore, the model can be considered accurate in the evaluation of the production line under analysis, and it can be exploited for the optimization of the system performance during the reconfiguration phase.

7.2. Reconfiguration of the production line

By exploiting the unloading robot R to load the main body of the drawer side on the pallet at the beginning of the production line, the layout of the system would change as shown in Fig. 5(b). Therefore, the inflow of the main body of the drawer slide in the production line would occur at station ST4 rather than at station ST1.

After the reconfiguration of the system, as soon as a finished drawer side arrives to station ST4, the robot R works as follows: first of all R unloads the drawer side from the pallet; then R loads a new main body on the empty pallet. From the tests carried out by the company, the robot R can be set up to perform the unloading–loading cycle in 5.36 [t.u.] without affecting its reliability.

In the new system configuration, a buffer B{4} to store the main bodies loaded on the pallets will be designed between stations ST4 and ST1. Because of space constraints in the plant, the overall capacity of all the buffers in the system cannot be higher than 20 parts. Considering that the current capacities of buffers B{1}, B{2} and B{3} are 4, 4 and 7 respectively, buffer B{4} can have a maximum capacity of 5 parts. After the reconfiguration, the company would agree to a reallocation of the total buffer capacity among the buffers in the systems, as long as the reallocation does not requires large movements of the stations in the plant.

As shown in Fig. 5(b), after the reconfiguration the layout of the production line will be a closed single loop system, in which the number of pallets in the system is constant over the time and it represents the population of the system. The population directly affect the performance of the closed loop system. Indeed, if too few pallets are placed in the system, the machines will be starved for most of the time. On the other hand, if too many pallets are placed in the system, the machine will be blocked for most of the time. Consequently, the number of pallets in the system is one of the parameter that must be defined during the system reconfiguration.

The proposed continuous-flow simulation method has been applied for the optimization of the throughput of the system with respect to the number of pallets in the production line. In the continuous-flow simulation model representing the reconfigured system, the same machines and buffers defined in the model of Section 7.1 are introduced. Because of the unloading–loading cycle performed by the robot R, the up state $S_1\{R\}$ of the machine $\mathrm{M}\{R\}$ has nominal production rate $\mu\{S_1\{R\}\} = \frac{1}{5.36}[\frac{\text{parts}}{\text{t.u.}}]$. Furthermore, the buffer B{4} is added in the model. The lower boundary of B{4} is $L\{4\} = 0$, while the upper boundary takes in account also the working position of the robot R, that is $N\{4\} = 5 + 1 = 6$.

Considering the working positions of the 10 machines in the system and the total capacity of all the buffers, the maximum number of pallets that can be placed in the system must be lower than 30. For this reason,

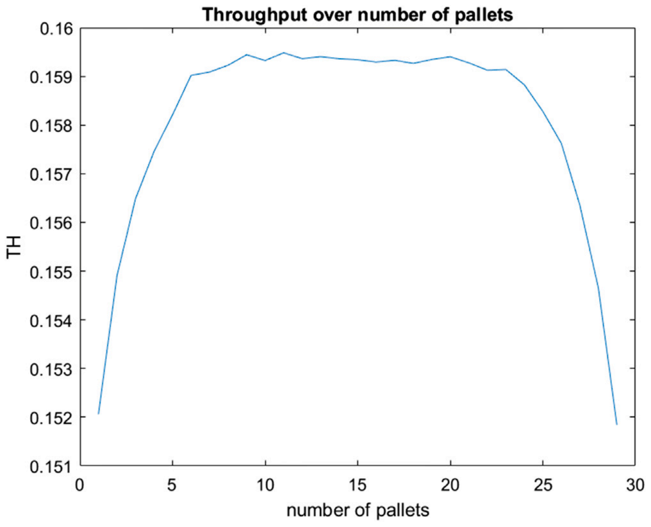


Fig. 6. Throughput of the reconfigured production line as the number of pallets changes.

the population of the system has been varied from 1 to 29 pallets, and the throughput of the system for each value of the population has been evaluated through the continuous-flow simulation.

The curve shown in Fig. 6 has been obtained by simulating 10 replicates for each value of the population, with simulation time equal to 10^6 [t.u] for each replicate. Fig. 6 is in line with the findings on the effect of the number of pallets on the throughput in closed-loop system, e.g., [55].

The computational time of the continuous-flow simulation for the entire curve is 1.04 h, while the time to simulate the same curve by means of a DES model with a discrete flow of parts is 4.88 h. Both simulation approaches have been run with the hardware and software conditions described in Section 6.

Based on this analysis, the number of pallets to place in the system in order to optimize the throughput is a value in the range $[10, 20]$.

7.3. Reallocation of the buffer capacity

The throughput of the reconfigured production line can be further increased by reallocating the total buffer capacity among the buffers of the system. For this reason, different combinations of buffer capacities have been analysed with the continuous-flow simulation. In particular, the capacities of the buffer $B\{1\}, B\{2\}, B\{3\}$ and $B\{4\}$ have been varied between 4 and 8 in order to avoid reallocation alternatives which require large movements of the stations in the plant. In each reallocation alternative, the total buffer capacity in the system cannot change.

Fig. 7 provides the throughput of the different alternatives of buffer capacity reallocation. The axes in Fig. 7 show the capacities of the buffers $B\{1\}, B\{2\}$ and $B\{3\}$. For each combination of values of the capacities of $B\{1\}, B\{2\}$ and $B\{3\}$, the capacity of the buffer $B\{4\}$ is straightforwardly defined as the difference between the total capacity 20 and the sum of the capacities of $B\{1\}, B\{2\}$ and $B\{3\}$. Each sphere in Fig. 7 represents a reallocation alternative and the colour of the sphere represents the throughput of the system for that combination of buffer capacities. For each reallocation alternative, 10 replicates with simulation time equal to 10^6 [t.u.] have been simulated. The overall computational time of the continuous-flow simulation to evaluate the 35 reallocation alternatives is 1.29 h, while the time to simulate the alternatives by means of a DES model with a discrete flow of parts is 6.38 h.

As shown in the figure, the throughput of the system increases as the capacity of $B\{1\}$ increases, and the optimal throughput is reached with the combination $[8, 4, 4, 4]$ of the buffer capacities of $B\{1\}, B\{2\}, B\{3\}$ and $B\{4\}$.

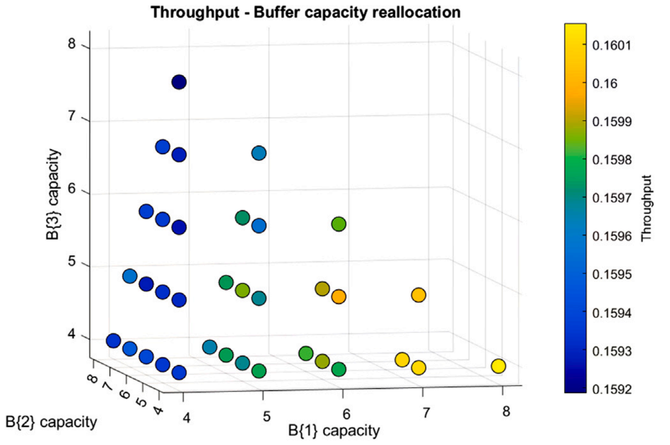


Fig. 7. Throughput of the system by reallocating the buffer capacities.

8. Conclusions and future developments

In this paper, a continuous-flow simulation method to evaluate the performance of manufacturing systems with single part-type production, general layout, assembly and disassembly machines, single or multiple loops and general inter-event time distributions has been presented.

The numerical results provided in the paper have shown that the usage of the continuous-flow simulation for the performance evaluation of manufacturing systems brings to a significant reduction of the number of discrete events generated in a simulation run. As main consequence, in the tested cases the continuous-flow simulation is on average 15 times faster than DES in the evaluation of discrete-flow systems, and 110 times faster in the evaluation of continuous-flow systems.

The applicability of the proposed method has been shown in an industrial case from the manufacturing system. In this industrial case, the continuous-flow simulation has been exploited during the reconfiguration phase of a production line of drawer sides. Several alternatives of reallocation of the total buffer capacity have been evaluated with the proposed method, and the optimal solution has been identified.

The approach proposed in this paper can be extended in various discussions: currently, the continuous-flow simulation technique is inadequate for analysing multi-product manufacturing systems due to two primary concerns. Firstly, when buffers are shared between multiple products in the system, the proposed model of the buffer level does not consider the single quantities of different products that a shared buffer is accommodating. Secondly and of greater significance, the proposed method lacks a general formalization of the various dispatching policies for the selection of the next product type for processing, particularly in systems that involve flexible machines handling multiple products.

As future research, the continuous-flow simulation method will be generalized for the evaluation of multi-product systems with split and merge of material flows and with specific dispatching policies. To reach these results, a general framework for the formalization of the most common policy managing the material flow in the manufacturing industry will be studied.

The definition of a control policy requires often the optimization of one or more control parameters. Starting from the continuous-flow simulation, two different optimization approaches will be studied.

One approach will be the formalization of the analytical equations of the continuous-flow simulation as a linear programming problem, in which the control variables of the problem will be the parameters of the control policy and the optimization function will be the production performance of interest in the system.

When the dynamics of the system cannot be formalized as linear constraints of an optimization problem, an alternative solution that will be addressed is the integration in the continuous-flow simulation of the infinitesimal perturbation analysis, for the gradient estimation of the performance measures with respect to small variations of the control parameters. Both the linear programming and the infinitesimal perturbation analysis will provide fast tools for the joint simulation–optimization of manufacturing systems.

One of the next future developments of this work will be the definition of the joint probability distribution of the buffer levels from the event log of the continuous-flow simulation. This analysis will support the definition of many control policies of the material flow or of the energy consumption, which are based on the joint conditions of several resources of the manufacturing system under analysis.

As a summary, the proposed work provides an efficient performance evaluation method to analyse single part-type continuous-flow manufacturing systems with general layout and general inter-event time distributions. This method allows very fast simulation of systems under general assumptions and enables determining the parameters used in design and control of manufacturing systems efficiently.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors received no financial support for the research, authorship, and/or publication of this article.

Appendix A. Boolean matrix operations

In the following, the Boolean operations used for the computation of the reachability matrix $W(t_k)$ in Eq. (13) are defined.

Given two square boolean matrices O and P of size $M \times M$, each entry $Q_{i,j}$ of the boolean sum $Q = O \vee P$ is defined as

$$Q_{i,j} = O_{i,j} \vee P_{i,j}, \qquad \forall i, \forall j,$$

each entry $Q_{i,j}$ of the boolean product $Q = O \wedge P$ as

$$Q_{i,j} = \bigvee_{m=1}^{M} (O_{i,m} \wedge P_{m,j}) = (O_{i,1} \wedge P_{1,j}) \vee \dots \vee (O_{i,M} \wedge P_{M,j}), \qquad \forall i, \forall j,$$

and the nth power of a boolean matrix as

$$O^n = O^{n-1} \wedge O, \qquad \text{with } n = 2, 3, \dots \text{ and } O^1 = O.$$

Appendix B. Proof of Lemma 1

In the following, the proof of Lemma 1 in Section 4.4 is given.

Proof. Consider a manufacturing system with M machines and with maximum length of propagation of a flow interruption/slowdown equal to $M^* \leq M - 1$. At time t_k the adjacency matrix $V(t_k)$ of the system is defined according to Definition 1 in Section 4.4, while the reachability matrix $W(t_k)$ is unknown. A directed graph G with M nodes is defined such that a directed edge between each couple of nodes (m, m') exists in G if $V_{m,m'}(t_k) = 1$. According to the graph theory, $V(t_k)$ is the adjacency matrix of the graph G, that defines whether pairs of nodes are adjacent or not in G, and M^* is the maximum distance between two nodes in G. The transitive closure of G is a square boolean matrix of size $M \times M$, that defines the reachability relation of G, that is the ability to reach or not a node m from a node m' through a sequence of adjacent nodes. Since the definition of the transitive closure of the graph G is equivalent to the definition of the reachability matrix $W(t_k)$ of the manufacturing system (Definition 3 in Section 4.4), $W(t_k)$ is the transitive closure of G. According to the work in [56], the transitive closure $W(t_k)$ of the graph G can be computed through the following boolean operations:

$$W(t_k) = \left(\mathrm{I} + V(t_k)\right)^M = \mathrm{I} \vee V(t_k) \vee V(t_k)^2 \vee \dots \vee V(t_k)^M. \tag{B.1}$$

In the work presented in [57] it has been demonstrated that, given a graph G with adjacency matrix $V(t_k)$, the entry (m, m') of the matrix $V(t_k)^n$ is 1 if in G it is possible to proceed from the node m to the node m' in exactly n steps, and 0 otherwise. Since the maximum distance between two nodes in G is $M^* \leq M-1$, no path of length $n > M^*$ exists between any two nodes in G. Consequently, any matrix $V(t_k)^n$ with $n > M^*$ is a zero matrix which does not affect the computation of $W(t_k)$ and it can be omitted from Eq. (B.1). Therefore, by replacing M with M^* in Eq. (B.1), Eq. (13) is obtained and Lemma 1 is demonstrated. □

The replacement of M with M^* in the computation of $W(t_k)$ does not affect the final result of $W(t_k)$, but it speeds up the time to run the continuous-flow simulation by reducing the computational time of Eq. (13).

Appendix C. Definition of the next event time in a specific system

In the following, the computation of the next event time t_{k+1} and the update of the system variables are provided for the numerical example presented in Section 5.

For the definition of the next event time $t_{k+1} = t_9$, the inter-event time Δt_8 must be computed. As defined in Eq. (5), Δt_8 depends on the remaining machine time of each machine and on the remaining buffer time of each buffer.

For each machine in the system of Fig. 2, the following time to transition matrices represent the remaining time before a transition between any couple of states:

$$\tau(1,t_8) = \begin{bmatrix} +\infty & 250.2 \\ 29.4 & +\infty \end{bmatrix}, \quad \tau(2,t_8) = \begin{bmatrix} +\infty & 144.2 \\ 17.7 & +\infty \end{bmatrix},$$

$$\tau(3,t_8) = \begin{bmatrix} +\infty & 121.5 & 78 \\ 24.21 & +\infty & 38.4 \\ 7.19 & 83.7 & +\infty \end{bmatrix}, \quad \tau(4,t_8) = \begin{bmatrix} +\infty & 57.6 \\ 31.1 & +\infty \end{bmatrix}.$$

For instance, the entry $\tau_{1,2}(3,t_8)$ defines that the remaining time before the occurrence of a transition from the state $S_1\{3\}$ to the state $S_2\{3\}$ of the machine M$\{3\}$ is equal to 121.5 time units. This value is valid under the assumptions that the machine M$\{3\}$ at time t_8 is in the state $S_1\{3\}$, the effective production rate $\mu(3,t_8)$ is equal to the nominal production rate $\mu_1\{3\}$, and that no other events occur in the meanwhile.

According to the current states $s(1,t_8), \dots, s(4,t_8)$ and to Eq. (7), the remaining machine time of each machine is computed as follows:

$$\mathrm{RM}(1,t_8) = \frac{\min\left\{\tau_{1,1}(1,t_8), \tau_{1,2}(1,t_8)\right\}}{c(1,t_8)} = \frac{\min\left\{+\infty, 250.2\right\}}{1} = 250.2,$$

$$\mathrm{RM}(2,t_8) = \frac{\min\left\{\tau_{1,1}(2,t_8), \tau_{1,2}(2,t_8)\right\}}{c(2,t_8)} = \frac{\min\left\{+\infty, 144.2\right\}}{0.76} = 189.74,$$

$$\mathrm{RM}(3,t_8) = \frac{\min\left\{\tau_{1,1}(3,t_8), \tau_{1,2}(3,t_8), \tau_{1,3}(3,t_8)\right\}}{c(3,t_8)} = \frac{\min\left\{+\infty, 121.5, 78\right\}}{0.62} = 195.97,$$

$$\mathrm{RM}(4,t_8) = \frac{\min\left\{\tau_{2,1}(2,t_8), \tau_{2,2}(2,t_8)\right\}}{c(4,t_8)} = \frac{\min\left\{31.1, +\infty\right\}}{1} = 31.1.$$

Therefore, the next event among the machines is the repair of the machine M$\{4\}$.

For the definition of the next event among the buffers, the times to fulfil and the times to deplete the buffers must be computed. Considering now the current levels $x(1,t_8), \dots, x(4,t_8)$ of the buffers, the times to fulfil the buffers $\beta_1(1,t_8),\dots,\beta_1(4,t_8)$ are computed according to Eq. (8) as follows:

$$\beta_1(1,t_8)=+\infty,$$
$$\beta_1(2,t_8)=+\infty,$$
$$\beta_1(3,t_8)=\frac{N\{3\}-x(3,t_8)}{\mu(3,t_8)-\mu(4,t_8)}=\frac{15-12}{1.3-0}=2.31,$$
$$\beta_1(4,t_8)=+\infty.$$

At time t_8 in the simulation, the time to fulfil the buffer B$\{1\}$ is set equal to $+\infty$ since the upstream and downstream machines M$\{1\}$ and M$\{2\}$ have the same production rate, and so the level of the buffer is stationary. Also the times to fulfil the buffers B$\{2\}$ and B$\{4\}$ are set equal to $+\infty$. Indeed, the buffer B$\{2\}$ is already full and so the next event in the simulation cannot be its fulfilment. Instead, considering B$\{4\}$ its level is decreasing because its downstream machine M$\{3\}$ is faster than the upstream machine M$\{4\}$, and so the buffer cannot become full.

Similarly, the times to deplete the buffers $\beta_2(1,t_8),\dots,\beta_2(4,t_8)$ are computed according to Eq. (9) as follows:

$$\beta_2(1,t_8)=+\infty,$$
$$\beta_2(2,t_8)=+\infty,$$
$$\beta_2(3,t_8)=+\infty,$$
$$\beta_2(4,t_8)=\frac{x(4,t_8)-L\{4\}}{\mu(3,t_8)-\mu(4,t_8)}=\frac{5-0}{1.3-0}=3.84.$$

The times to deplete the buffers B$\{1\}$, B$\{2\}$ and B$\{3\}$ are all set equal to $+\infty$, since the buffer B$\{1\}$ is already empty, the level of B$\{2\}$ is stationary because the upstream and downstream machines have the same production rate, and the level of B$\{3\}$ is increasing because the upstream machine is faster than the downstream machine.

At this point, the remaining buffer time of each buffer is computed according to Eq. (10).

$$\mathrm{RB}(1,t_8)=\min\left\{\beta_1(1,t_8),\beta_2(1,t_8)\right\}=\min\left\{+\infty,+\infty\right\}=+\infty,$$
$$\mathrm{RB}(2,t_8)=\min\left\{\beta_1(2,t_8),\beta_2(2,t_8)\right\}=\min\left\{+\infty,+\infty\right\}=+\infty,$$
$$\mathrm{RB}(2,t_8)=\min\left\{\beta_1(3,t_8),\beta_2(3,t_8)\right\}=\min\left\{2.31,+\infty\right\}=2.31,$$
$$\mathrm{RB}(2,t_8)=\min\left\{\beta_1(4,t_8),\beta_2(4,t_8)\right\}=\min\left\{+\infty,3.84\right\}=3.84.$$

The inter-event time Δt_8 is now computed according to Eq. (5):

$$\begin{aligned}\Delta t_8&=\min\{\mathrm{RM}(1,t_8),\dots,\mathrm{RM}(4,t_8),\mathrm{RB}(1,t_8),\dots,\mathrm{RB}(4,t_8),T-t_8\}=\\&=\min\{250.2,189.74,195.97,31.1,+\infty,+\infty,2.31,3.84,10^5-129.7\}\\&=2.31.\end{aligned}$$

Therefore, the next event occurring in the system is the fulfilment of the buffer B$\{3\}$, since the remaining buffer time RB$(3,t_8)$ has the lowest value among the remaining machine times, the remaining buffer times and the remaining simulation time. The fulfilment of the buffer B$\{3\}$ occurs at time t_9:

$$t_9=t_8+\Delta t_8=129.7+2.31=132.01.$$

The clock time is instantaneously advanced from the event time t_8 to event time t_9, and all the variables describing the system are updated. Since the event at time t_9 is not related to a state transition of any machine, the current state of each machine does not change from t_8 to t_9, that is $s(1,t_8)=s(1,t_9),\dots,s(4,t_8)=s(4,t_9)$.

The current level of each buffer is updated according to Eq. (15):

$$x(1,t_9)=x(1,t_8)+\left(\mu(1,t_8)-\mu(3,t_8)\right)\cdot\Delta t_8=0+(1.3-1.3)\cdot 2.31=0,$$
$$x(2,t_9)=x(2,t_8)+\left(\mu(2,t_8)-\mu(3,t_8)\right)\cdot\Delta t_8=10+(1.3-1.3)\cdot 2.31=10,$$
$$x(3,t_9)=x(3,t_8)+\left(\mu(3,t_8)-\mu(4,t_8)\right)\cdot\Delta t_8=12+(1.3-0)\cdot 2.31=15,$$
$$x(4,t_9)=x(4,t_8)+\left(\mu(4,t_8)-\mu(3,t_8)\right)\cdot\Delta t_8=5+(0-1.3)\cdot 2.31=2.$$

The steps to update the effective production rate of each machine and then the capacity saturation ratio are the same steps which have been performed at time t_8 in Section 5.2. For this reason, they are omitted now.

The last set of variables to update are the time to transition matrices $\tau(1,t_9),\dots,\tau(4,t_9)$. As defined in (16), in each time to transition matrix only the row corresponding to the state of the machine at time t_8 must be updated. All the other rows in the matrix do not change from t_8 to t_9. By applying the matrix notation provided in Eq. (17), the following time to transition matrices are obtained:

$$\begin{aligned}\tau(1,t_9)&=\tau(1,t_8)-\Delta t_8\cdot c(1,t_8)\cdot\left[s(1,t_8),s(1,t_8)\right]=\\&=\begin{bmatrix}+\infty & 250.2\\ 29.4 & +\infty\end{bmatrix}-2.31\cdot\begin{bmatrix}1 & 1\\ 0 & 0\end{bmatrix}=\begin{bmatrix}+\infty & 247.89\\ 29.4 & +\infty\end{bmatrix},\end{aligned}$$

$$\begin{aligned}\tau(2,t_9)&=\tau(2,t_8)-\Delta t_8\cdot c(2,t_8)\cdot\left[s(2,t_8),s(2,t_8)\right]=\\&=\begin{bmatrix}+\infty & 144.2\\ 17.7 & +\infty\end{bmatrix}-1.76\cdot\begin{bmatrix}1 & 1\\ 0 & 0\end{bmatrix}=\begin{bmatrix}+\infty & 142.44\\ 17.7 & +\infty\end{bmatrix},\end{aligned}$$

$$\begin{aligned}\tau(3,t_9)&=\tau(3,t_8)-\Delta t_8\cdot c(3,t_8)\cdot\left[s(3,t_8),s(3,t_8),s(3,t_8)\right]=\\&=\begin{bmatrix}+\infty & 121.5 & 78\\ 24.21 & +\infty & 38.4\\ 7.19 & 83.7 & +\infty\end{bmatrix}-1.43\cdot\begin{bmatrix}1 & 1 & 1\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}\\&=\begin{bmatrix}+\infty & 120.07 & 76.57\\ 24.21 & +\infty & 38.4\\ 7.19 & 83.7 & +\infty\end{bmatrix},\end{aligned}$$

$$\begin{aligned}\tau(4,t_9)&=\tau(4,t_8)-\Delta t_8\cdot c(4,t_8)\cdot\left[s(4,t_8),s(4,t_8)\right]=\\&=\begin{bmatrix}+\infty & 57.6\\ 31.1 & +\infty\end{bmatrix}-2.31\cdot\begin{bmatrix}0 & 0\\ 1 & 1\end{bmatrix}=\begin{bmatrix}+\infty & 57.6\\ 28.79 & +\infty\end{bmatrix}.\end{aligned}$$

If the event at time t_9 were a state transition, just like a transition from the state $S_1\{1\}$ to the state $S_2\{1\}$ of M$\{1\}$, the corresponding entry $\tau_{1,2}(1,t_9)$ would have resulted equal to 0. In this case, a new value $\tau_{1,2}(1,t_9)$ would have been randomly generated according to the distribution $D_{1,2}\{1\}$ of the time to transition from $S_1\{1\}$ to $S_2\{1\}$.

At this point, the definition of the next event time and the update of the system variables can be iteratively performed for $t_{10},t_{11},\dots$ until the end of the simulation.

References

[1] Adane TF, Bianchi MF, Archenti A, Nicolescu M. Application of system dynamics for analysis of performance of manufacturing systems. J Manuf Syst 2019;53:212–33. http://dx.doi.org/10.1016/j.jmsy.2019.10.004.

[2] Colledani M, Tolio T. A decomposition method to support the configuration / reconfiguration of production systems. CIRP Ann - Manuf Technol 2005;54:441–4. http://dx.doi.org/10.1016/S0007-8506(07)60140-1.

[3] Papadopoulos CT, Li J, O'Kelly ME. A classification and review of timed Markov models of manufacturing systems. Comput Ind Eng 2019;128. http://dx.doi.org/10.1016/j.cie.2018.12.019.

[4] De Koster M, Wijngaard J. Continuous vs. discrete models for production lines with blocking. Queueing Netw Blocking 1989;175–92.

[5] Gershwin SB, Berman O. Analysis of transfer lines consisting of two unreliable machines with random processing times and finite storage buffers. AIIE Trans 1981;13:2–11. http://dx.doi.org/10.1080/05695558108974530.

[6] Gershwin SB. An efficient decomposition method for the approximate evaluation of tandem queues with finite storage space and blocking. Oper Res 1987;35(2):291–305. http://dx.doi.org/10.1287/opre.35.2.291.

[7] Dallery Y, David R, Xie X-L. An efficient algorithm for analysis of transfer lines with unreliable machines and finite buffers. IIE Trans 1988;20(3):280–3. http://dx.doi.org/10.1080/07408178808966181.

[8] Wijngaard J. The effect of interstage buffer storage on the output of two unreliable production units in series, with different production rates. AIIE Trans 1979;11:42–7. http://dx.doi.org/10.1080/05695557908974399.

[9] Yeralan S, Franck WE, Quasem MA. A continuous materials flow production line model with station breakdown. European J Oper Res 1986;27:289–300. http://dx.doi.org/10.1016/0377-2217(86)90326-7.

[10] Mitra D. Stochastic theory of a fluid model of producers and consumers coupled by a buffer. Adv Appl Probab 1988;20:646–76. http://dx.doi.org/10.2307/1427040.

[11] Tolio T, Matta A, Gershwin SB. Analysis of two-machine lines with multiple failure modes. IIE Trans 2002;34:51–62. http://dx.doi.org/10.1080/07408170208928849.

[12] Tan B, Gershwin SB. Analysis of a general Markovian two-stage continuous-flow production system with a finite buffer. Int J Prod Econ 2009;120:327–39. http://dx.doi.org/10.1016/j.ijpe.2008.05.022.

[13] Tan B, Gershwin SB. Modelling and analysis of Markovian continuous flow systems with a finite buffer. Ann Oper Res 2011;182:5–30. http://dx.doi.org/10.1007/s10479-009-0612-6.

[14] Tolio TA, Ratti A. Performance evaluation of two-machine lines with generalized thresholds. Int J Prod Res 2018;56(1–2):926–49. http://dx.doi.org/10.1080/00207543.2017.1420922.

[15] Magnanini MC, Tolio TA. Performance evaluation of asynchronous two-stage manufacturing lines fabricating discrete parts. CIRP J Manuf Sci Technol 2021;33:488–505. http://dx.doi.org/10.1016/j.cirpj.2021.04.002.

[16] Scrivano S, Tolio T. A Markov chain model for the performance evaluation of manufacturing lines with general processing times. Proc CIRP 2021;103:20–5. http://dx.doi.org/10.1016/j.procir.2021.10.002, 9th CIRP Global Web Conference – Sustainable, resilient, and agile manufacturing and service operations : Lessons from COVID-19.

[17] Bihan HL, Dallery Y. A robust decomposition method for the analysis of production lines with unreliable machines and finite buffers. Ann Oper Res 2000;93:265–97. http://dx.doi.org/10.1023/a:1018996428429.

[18] Levantesi R, Matta A, Tolio T. Performance evaluation of continuous production lines with machines having different processing times and multiple failure modes. Perform Eval 2003;51:247–68. http://dx.doi.org/10.1016/S0166-5316(02)00098-6.

[19] Colledani M, Gershwin SB. A decomposition method for approximate evaluation of continuous flow multi-stage lines with general Markovian machines. Ann Oper Res 2013;209:5–40. http://dx.doi.org/10.1007/s10479-011-0961-9.

[20] Helber S, Jusić H. A new decomposition approach for non-cyclic continuous material flow lines with a merging flow of material. Ann Oper Res 2004;125:117–39. http://dx.doi.org/10.1023/B:ANOR.0000011188.11019.33.

[21] Bai Y, Zhang L. Recursive decomposition/aggregation algorithms for performance metrics calculation in multi-level assembly/disassembly production systems with exponential reliability machines. Int J Prod Res 2023;1–26. http://dx.doi.org/10.1080/00207543.2023.2166622.

[22] Colledani M, Ekvall M, Lundholm T, Moriggi P, Polato A, Tolio T. Analytical methods to support continuous improvements at scania. Int J Prod Res 2010;48(7):1913–45. http://dx.doi.org/10.1080/00207540802538039.

[23] Oyarbide A, Baines T, Kay J, Ladbrook J. Manufacturing systems modelling using system dynamics: forming a dedicated modelling tool. J Adv Manuf Syst 2003;2(01):71–87. http://dx.doi.org/10.1142/S0219686703000228.

[24] Vasudevan K, Devikar A. Selecting simualtion abstraction levels in simulation models of complex manufacturing systems. In: Proceedings of the 2011 winter simulation conference. 2011, p. 2268–77. http://dx.doi.org/10.1109/WSC.2011.6147938.

[25] Jovanoski B, Nove Minovski R, Lichtenegger G, Voessner S. Managing strategy and production through hybrid simulation. Ind Manag Data Syst 2013;113(8):1110–32. http://dx.doi.org/10.1108/IMDS-09-2012-0342.

[26] Tako AA, Robinson S. The application of discrete event simulation and system dynamics in the logistics and supply chain context. Decis Support Syst 2012;52(4):802–15. http://dx.doi.org/10.1016/j.dss.2011.11.015.

[27] Banks J, Carson JS, Nelson BL, Nicol D. Discrete-event system simulation. 5th ed. Prentice Hall; 2010, http://dx.doi.org/10.2307/1268124.

[28] Smith JS. Survey on the use of simulation for manufacturing system design and operation. J Manuf Syst 2003;22(2):157–71. http://dx.doi.org/10.1016/S0278-6125(03)90013-6.

[29] Negahban A, Smith JS. Simulation for manufacturing system design and operation: Literature review and analysis. J Manuf Syst 2014;33(2):241–61. http://dx.doi.org/10.1016/j.jmsy.2013.12.007.

[30] Michalos G, Fysikopoulos A, Makris S, Mourtzis D, Chryssolouris G. Multi criteria assembly line design and configuration–an automotive case study. CIRP J Manuf Sci Technol 2015;9:69–87. http://dx.doi.org/10.1016/j.cirpj.2015.01.002.

[31] Kampa A, Gołda G, Paprocka I. Discrete event simulation method as a tool for improvement of manufacturing systems. Computers 2017;6(1):10. http://dx.doi.org/10.3390/computers6010010.

[32] Tan B. Mathematical programming representations of the dynamics of continuous-flow production systems. IIE Trans 2015;47:173–89. http://dx.doi.org/10.1080/0740817X.2014.892232.

[33] Helber S, Schimmelpfeng K, Stolletz R, Lagershausen S. Using linear programming to analyze and optimize stochastic flow lines. Ann Oper Res 2011;182:193–211. http://dx.doi.org/10.1007/s10479-010-0692-3.

[34] Alfieri A, Matta A. Mathematical programming formulations for approximate simulation of multistage production systems. European J Oper Res 2012;219(3):773–83. http://dx.doi.org/10.1016/j.ejor.2011.12.044.

[35] Alfieri A, Matta A. Mathematical programming representation of pull controlled single-product serial manufacturing systems. J Intell Manuf 2012;23:23–35. http://dx.doi.org/10.1007/s10845-009-0371-x.

[36] Brailsford SC, Hilton NA. A comparison of discrete event simulation and system dynamics for modelling health care systems. In: Proceedings of the 26th meeting of the ORAHS working group 2000. Glasgow Caledonian University; 2001.

[37] Forrester JW. The beginning of system dynamics. The McKinsey Q 1995;(4):4–5.

[38] Baines T, Harrison D. An opportunity for system dynamics in manufacturing system modelling. Prod Plan Control 1999;10(6):542–52. http://dx.doi.org/10.1080/095372899232830.

[39] Huang M, Ip W, Yung KL, Wang X, Wang D. Simulation study using system dynamics for a CONWIP-controlled lamp supply chain. Int J Adv Manuf Technol 2007;32:184–93. http://dx.doi.org/10.1007/s00170-005-0324-2.

[40] Lanka K, Gopal P. System dynamics modelling of fixed and dynamic kanban controlled production systems: a supply chain perspective. J Model Manag 2023;18(1):17–35. http://dx.doi.org/10.1108/JM2-06-2020-0168.

[41] Stadnicka D, Litwin P. Value stream mapping and system dynamics integration for manufacturing line modelling and analysis. Int J Prod Econ 2019;208:400–11. http://dx.doi.org/10.1016/j.ijpe.2018.12.011.

[42] Duggan J. System dynamics modeling with R. Cham: Springer International Publishing; 2016, p. 1–24. http://dx.doi.org/10.1007/978-3-319-34043-2_1.

[43] Zeigler BP, Kim TG, Praehofer H. Theory of modeling and simulation. Academic Press; 2000.

[44] Zeigler BP. DEVS theory of quantized systems. In: Advanced simulation technology thrust. DARPA contract; 1998.

[45] Van Tendeloo Y, Vangheluwe H. Discrete event system specification modeling and simulation. In: 2018 Winter simulation conference. IEEE; 2018, p. 162–76. http://dx.doi.org/10.1109/WSC.2018.8632372.

[46] Kofman E, Junco S. Quantized-state systems: a DEVS approach for continuous system simulation. Trans Soc Model Simul Int 2001;18(3):123–32.

[47] Fernández J, Kofman E. A stand-alone quantized state system solver for continuous system simulation. Simulation 2014;90(7):782–99. http://dx.doi.org/10.1177/0037549714536255.

[48] Kouikoglou VS, Phillis YA. Discrete event modeling and optimization of unreliable production lines with random rates. IEEE Trans Robot Autom 1994;10:153–9. http://dx.doi.org/10.1109/70.282540.

[49] Giambiasi N, Escude B, Ghosh S. GDEVS: A generalized discrete event specification for accurate modeling of dynamic systems. In: Proceedings 5th international symposium on autonomous decentralized systems. IEEE; 2001, p. 464–9. http://dx.doi.org/10.1109/ISADS.2001.917452.

[50] Giambiasi N, Carmona JC. Generalized discrete event abstraction of continuous systems: Gdevs formalism. Simul Model Pract Theory 2006;14(1):47–70. http://dx.doi.org/10.1016/j.simpat.2005.02.009.

[51] Hosseini B, Tan B. Simulation and optimization of continuous-flow production systems with a finite buffer by using mathematical programming. IISE Trans 2017;49:255–67. http://dx.doi.org/10.1080/0740817X.2016.1217103.

[52] Kouikoglou VS. An efficient discrete event model of assembly/disassembly production networks. Int J Prod Res 2002;40(17):4485–503. http://dx.doi.org/10.1080/00207540210155846.

[53] Jungnickel D. Graphs, networks and algorithms, vol. 3. Heidelberg, Germany: Springer Berlin; 2005.

[54] Warshall S. A theorem on boolean matrices. J ACM 1962;9:11–2. http://dx.doi.org/10.1145/321105.321107.

[55] Gershwin SB, Werner LM. An approximate analytical method for evaluating the performance of closed-loop flow systems with unreliable machines and finite buffers. Int J Prod Res 2007;45(14):3085–111. http://dx.doi.org/10.1080/00207540500385980.

[56] Fischer MJ, Meyer AR. Boolean matrix multiplication and transitive closure. In: 12th Annual symposium on switching and automata theory. 1971, p. 129–31. http://dx.doi.org/10.1109/SWAT.1971.4.

[57] Prosser RT. Applications of boolean matrices to the analysis of flow diagrams. In: Proceedings of the eastern joint computer conference. 1959, p. 133–8.