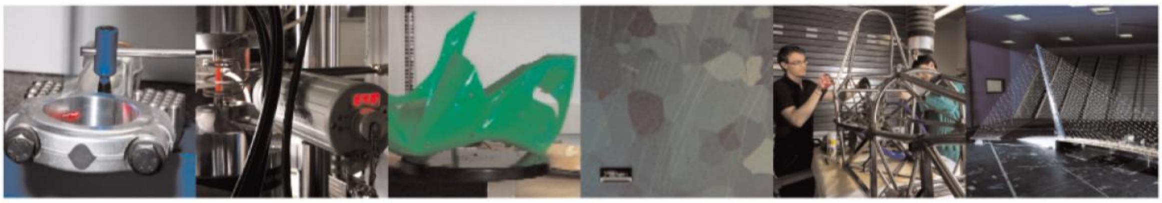
POLITECNICO MILANO 1863

DIPARTIMENTO DI MECCANICA



Energy-Efficient Control in Multi-Stage Production Lines with Parallel Machine Workstations and Production Constraints

Alberto Loffredo, Nicla Frigerio, Ettore Lanzarone, Andrea Matta

This is an Accepted Manuscript of an article published by Taylor & Francis in IISE Transactions on 13 Jan 2023, available online: https://doi.org/10.1080/24725854.2023.2168321.

DIPARTIMENTO DI MECCANICA ■ POLITECNICO DI MILANO
via G. La Masa, 1 ■ 20156 Milano ■ EMAIL (PEC): pecmecc@cert.polimi.it
http://www.mecc.polimi.it
Rev. 0

Energy-Efficient Control in Multi-Stage Production Lines with Parallel Machine Workstations and Production Constraints

Alberto Loffredo [a], Nicla Frigerio [a], Ettore Lanzarone [b,c] and Andrea Matta [a]

[a] Department of Mechanical Engineering, Politecnico di Milano, Milan, Italy
[b] Department of Management, Information and Production Engineering, University of Bergamo, Italy
[c] Institute for Applied Mathematics and Information Technologies (IMATI), National Research Council of Italy (CNR), Milan, Italy.

Abstract

Nowadays, the growing interest in industry for enhancing manufacturing processes sustainability is a major trend. One of the most supported strategies to increase the energy-efficiency of manufacturing activities is the control of machine state towards the optimum trade-off between production rate and energy demand. This method is referred to as energy-efficient control and it triggers machines in a standby state with low power request. In this paper, multi-stage production lines composed of identical parallel machine workstations are the systems of interest, and the energy-efficient control policies make use of buffer level information. Each machine can be switched off instantaneously and switched on with a stochastic startup time. Problem objective is to minimize the energy demand while assuring production constraints. The paper proposes a novel approach to solve the problem at hand. An exact model for two-stage system is formulated using a Markov Decision Process to be solved with a linear programming methodology. A novel technique, namely the Backward-Recursive approach, is used to address systems with more than two stages. Numerical experiments confirm the effectiveness of the proposed approach.

Keywords: Energy-Efficiency Control; Parallel Machines; Manufacturing Automation; Markov Decision Process; Linear Programming.

1 Introduction

The industrial sector is accounted for 42% of global energy consumption, mostly caused by manufacturing activities (Center, 2020), and therefore manufacturing companies are significantly increasing their interest in the energy-efficiency topic. In a manufacturing system, energy-efficiency can be addressed at different hierarchical levels of the automation pyramid (Can et al., 2019): global supply chain, facility, production line, and machine tool (*machine* from now on in the paper) which accounts for approximately 50% of the total amount of electricity utilized in manufacturing (Hu et al., 2017). In this work, the machine level is analyzed in order to assess the resulting benefits for the entire production line. A major energy consumption is caused by machines kept in ready-for-process conditions even during idle periods, i.e. when the part flow is interrupted, and the machine is not operating on parts but still has a remarkable energy request. This policy to manage machine operations is known as *Always On* (AOn) policy. On the other hand, energy-efficient control (EEC) policies can be used to decrease machine energy consumption during idle periods. The key idea behind EEC is that the machine can be switched off when the production is not needed because the part flow is interrupted so that machine power request is reduced and, then, it can be switched on again when the production is about to resume. Resume the service often requires a delay due to the machine startup procedure. A proper EEC policy should decrease energy consumption while assuring target levels for different production performance indicators (e.g. system throughput and workstation availability levels). EEC complexity increases when policies are applied at multiple machines simultaneously.

Multi-stage manufacturing lines with parallel machine workstations are very common. As particular example, machining is one of the key processes in the automotive industry but not one of the fastest and therefore machining operations are often performed with parallel machine workstations managed with AOn policy. Figure 1 represents the conceptual layout of an automated line producing cylinder heads for automotive (Loffredo et al., 2022). This example consists of twenty-one operations including four machining operations (Op.125, 375, 390, and 525) performed by parallel machine workstations. Also, Op.500 and 515 are by far the slowest processes of the entire line leading to frequent blocking for Op.125, 375, and 390 and to frequent starvation for Op.525. As a consequence, the machining production stages are almost never used at maximum capacity, and some resources could be switched off for energy saving without undermining the overall system production rate.

The extensive use of multi-stage production lines made of workstations with parallel identical machines that are commonly managed with the AOn policy leads to great potential for increasing the energy-efficiency of these systems. Therefore, being able to reduce the environmental impact of systems with this layout can strongly improve industrial processes' sustainability. This goal can be achieved with EEC and, for this reason, this work focuses

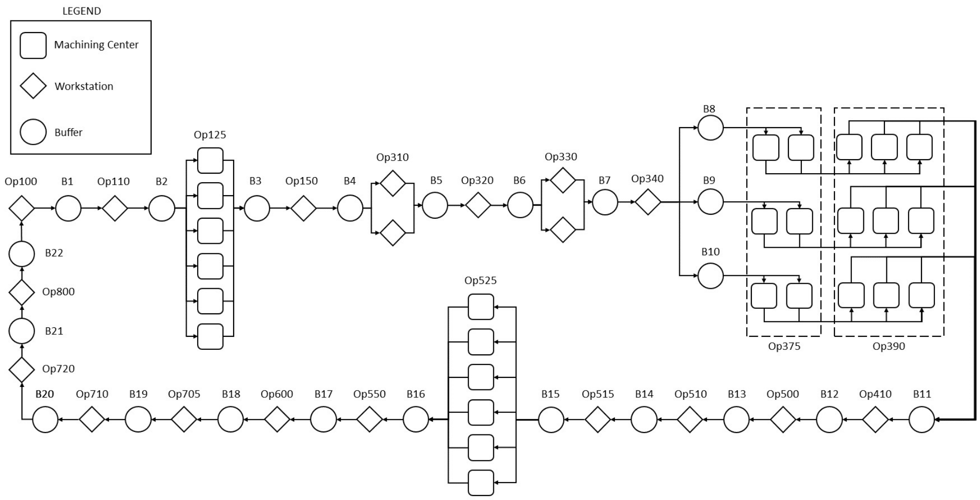


Figure 1: Multi-stage production line with parallel machines (Loffredo et al., 2022).

on the EEC of multi-stage production lines made of workstations with parallel identical machines. The goal is to obtain an EEC policy using buffer level information to reduce the overall energy consumption of the production system while reaching desired target levels on general system performance such as production rate, machine availability, or others.

1.1 *Related Literature*

Machine energy consumption can be seen as the sum of a *Base Load* energy independent of the process and required to maintain ready-for-process machine conditions, and a *Load Dependent* energy required to operate on parts (Dahmus and Gutowski, 2004). The principal techniques to reduce the *Base Load* energy through machine state control are the EEC and the energy-efficient scheduling (EES).

EES is connected with the production activities scheduling, i.e. the detailed plan for the use of the machines to perform a set of production activities called “jobs”. Most of the time scheduling plan is often defined before its actual realization on the shop floor since all the relevant data are assumed known and deterministic (Anghinolfi et al., 2021). The goal of EES becomes the trade-off between energy cost and a main classic scheduling objective, such as total makespan or jobs’ tardiness (Wang et al., 2018). The objective is to have a production schedule leading to the optimal energy demand profile: jobs are allocated to machines minimizing the number of non-productive periods and, consequently, decreasing the plant energy consumption. Furthermore, the EES approach was also exploited to improve the energy-efficiency of parallel machine workstations (Wang et al., 2018; Anghinolfi et al., 2021; Heydar et al., 2022) and a complete EES literature review can be found in (Gahm et al., 2016). On the other hand, this work deals with EEC. EEC provides policies to be applied in real-time during production progress, without deterministic information on the next part arrival to the machine. Thus, EEC and EES have two completely different approaches to deal with the energy-efficient problem and are characterized by different research streams.

Literature for EEC in manufacturing is becoming wide and is expanding in recent years. A complete and recent literature review on this topic can be found in (Renna and Materi, 2021). The first level of analysis where EEC can be implemented is the single machine and, consequently, the single-buffer-single-machine layout has been extensively studied in this field. The first work on this theme is (Mouzon et al., 2007), with several switch off dispatching rules for a non-bottleneck machine in a production system. Subsequently, in (Frigerio and Matta, 2015) the authors studied analytically an EEC policy for a single machine, developing also an adaptive EEC policy based on machine learning techniques that is able to adapt the control for varying system parameters (Frigerio et al., 2021). The subsequent level of analysis is represented by the EEC of an entire production line, modeled as a series of single machines and single finite buffers. A first approach relates to the concept of temporal opportunity window (OW): this is defined as the longest possible downtime of a station that does not result in permanent production loss at the end-of-line station (Chang et al., 2012); thus, during these OWs it is possible to implement EEC on the machine, reduce energy consumption and not jeopardize the production rate. In (Sun and Li, 2012) an analytical algorithm is presented to estimate temporal opportunity windows to implement EEC action on machines in a line. The OW approach was also used in (Chang et al., 2012), indicating different real-time switching on/off policies for machines to be applied during OW caused by random failures in a production system. However, in all the works dealing with the OW concept, the startup time for machines is never considered and this is not aligned with most manufacturing equipment where a startup transitory is needed to resume the service and therefore causing a production loss. A different method was proposed in (Jia et al., 2016) where the authors used work-in-process information to develop effective EEC policies for the production line; however, their model was limited by the assumption of machines following the Bernoulli reliability model, machines with constant and identical cycle time, and the possibility to control only some machines. Most recently, in (Zhang et al., 2019) the authors developed a Gaussian mixture model to predict machines idle periods duration and, consequently, to be able to implement EEC actions during the predicted idle periods. Finally, a recent work (Cui et al., 2021) proposed an optimal EEC method for the whole production line using buffer level information to reduce energy consumption while only slightly decreasing productivity. However, the authors only consider machines that are characterized by fixed and identical processing time. As for the literature, only (Loffredo et al., 2021) addressed the EEC for the parallel machine configuration by providing a model for the EEC of a single stand-alone workstation composed of an upstream buffer and multiple identical parallel machines. As extension, the provided model is applied individually on more workstations pertaining to the same production system (Loffredo et al., 2022). Differently from this work, the EEC is executed independently at multiple workstations considering the workstation state and without considering the overall system state in each control action.

1.2 *Contribution*

In literature, models leading to EEC policies for multi-stage production lines with parallel machine workstations are not present. This paper focuses on the EEC of multi-stage production lines composed of identical parallel machine workstations, a widely used layout in manufacturing. A novel approach is proposed to minimize the system energy demand while assuring quite general production constraints. In particular, this work provides the following contributions. At first, a novel model, referred to as *2S-Model*, is proposed. The 2S-Model leads to an exact solution for the EEC of a two-stage system. The problem is formalized as a Discrete Time Markov Decision Process (DT-MDP) and a linear programming (LP) approach is used to solve it. The second contribution is represented by a novel technique, referred to as *Backward-Recursive approach*, that is proposed to deal with long production lines. Starting from the last machines and progressing backward, two-stage subsystems are considered and the solution to the problem is created recursively. The third contribution is that the proposed formulation also includes general production constraints, e.g. target system throughput, and target workstations availability, to be met. In this way, the resulting EEC policy will lead to the optimum trade-off between energy demand and the required performance indicators. Numerical experiments on realistic manufacturing system configurations are performed to assess the effectiveness of the 2S-Model and to show the efficiency of the Backward-Recursive approach in solving large problems.

The remainder of the paper is organized as follows. Section 2 describes the system under investigation. Problem formulation for two-stage lines (i.e., the 2S-Model) is in section 3 and section 4 formulates the Backward-Recursive approach for more than two stages. Numerical analysis is presented in sections 5 and 6 respectively for two-stage lines and lines with more than two stages. Section 7 concludes the work and discusses further developments.

2 Problem Description

Let us consider a production line composed of m stages as the system to be controlled, where each stage is composed of a buffer of finite capacity and a workstation with identical parallel machines. The scope of the EEC is to find a policy leading to the optimum trade-off between performance indicators and energy demand in the production system under control. Specifically, if the target is just the minimization of the system energy consumption, the *unconstrained* EEC problem is addressed. On the other hand, if there are some production constraints to be satisfied (e.g. one or more performance indicators that must be higher/lower/equal than a certain target) the problem is referred to as *constrained* EEC problem.

2.1 *System Description*

The system layout is represented in Figure 2. Stages are denoted as S_i with $i \in \{1, \dots, m\}$, buffers as B_i with $i \in \{1, \dots, m\}$, and M_{ij} with $j \in \{1, \dots, c_i\}$ indicates machine j of the c_i parallel machines in the i-th stage of the system. Furthermore, for each stage i, machines M_{ij} with $j \in \{1, \dots, c_i\}$ are identical and work in parallel and buffer B_i has a finite holding capacity $0 < K_i < \infty$. The system is characterized by different stochastic processes, i.e. machine processing time, machine startup time, and the arrival of parts to S_1. These processes are assumed to be Poisson processes, independent of each other and stationary. Specifically, parts arrive at S_1 following a Poisson process with rate λ, and this process stops when this buffer is full. Furthermore, machines of S_i are characterized by exponentially distributed startup times with rate δ_i. Moreover, each machine M_{ij} has processing times exponentially distributed with service rate μ_i. Machines are unreliable, i.e. they can be subject to failures. The stochasticity provoked by machine failures is modeled by embedding them into machine processing times: in this sense, the overall processing time (considering also service interruptions caused by failures) is a stochastic variable. Thus, in the proposed model a stochastic distribution, i.e. the exponential distribution, is used to represent this stochastic variable. The assumption of having Poisson processes representing the aforementioned stochastic processes is considered realistic for many industrial cases of multi-stage production lines: examples in literature with this same assumption can be found in (Govil and Fu, 1999; Loffredo et al., 2021, 2022; Zhao and Li, 2013). Furthermore, because it naturally gives rise to Markov processes, the exponential distribution has been widely used in the literature (Dallery and Gershwin, 1992) and, from (Inman, 1999), it is possible to state that even if the exponential assumption is violated, Markov model's results are relatively insensitive to this violation, i.e it could still provide accurate estimates of the real system's average performance at the steady state. Therefore, this assumption is considered reasonable and applicable to the system under study without undermining the system model and consequent results.

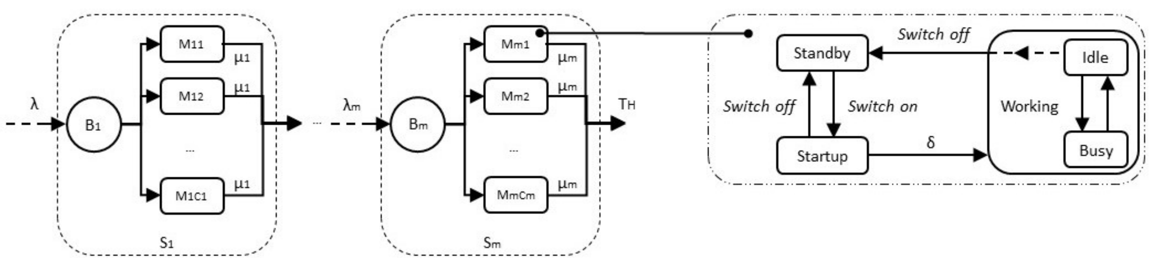


Figure 2: Layout of the production line under analysis with machine state model.

Machines M_{ij} are starved if they can process parts but B_i is empty while are blocked if they can process parts but B_{i+1} is full. As an exception, machines M_{mj} of the last stage S_m cannot be blocked since there is an infinite capacity buffer downstream the system: processed parts leave the system immediately after the process completion, leading to system throughput T_H. In addition, all the machines in the line work on a single part type, and first come first served rule is applied. Finally, machines cannot be switched off while operating on items, i.e. part processing cannot be interrupted by the control.

2.2 *Machine Energetic State Model*

Machines M_{ij} can be controlled for energy saving purposes. It is assumed that each machine is *busy* while working on parts and *idle* when it is in ready-for-process conditions but it is not operating on parts. *Busy* and *idle* are two sub-states composing the machine *working* state. When the machine is blocked it is also *busy*, since the part is still inside it, and the machine cannot be switched off. The machine is productive only when *busy*, i.e. characterized by an exponentially distributed service rate μ_i. When in the *idle* state, the machine can be switched off going instantaneously into the *standby* state: a lower power request state where only emergency services are active so that the machine cannot process parts and the service is interrupted. From the *standby* state, the machine must execute a startup procedure to resume the service, so that the *startup* state is visited before the *working* state. It is assumed that the startup procedure can be interrupted by the control, but only to switch off the device and set the device from *startup* to *standby* state. Machine energetic state model is as in Figure 2. Therefore, machine M_{ij} is characterized by the following state set $\Theta = \{w, sb, su, id, b\}$, respectively: *working*, *standby*, *startup*, *idle* and *busy*.

It is assumed that machines of a certain stage i are characterized by the same power consumption in the different states: they require a constant non-negative amount of power depending on the energetic state they are in. During the *standby* state most of machine modules are deactivated and only emergency services are active: the power requested in this state is assumed to be almost null ($w_{sb} \simeq 0$). While *idle*, the machine has all its modules activated but is not carrying out any process: the power consumption is therefore assumed to be higher than the standby one ($w_{id} > w_{sb} \simeq 0$) but not excessively high. During the *startup* phase all the procedures required to make the machine suitable for processing are executed, so that quality and tolerance requirements can be met. The amount of power required in this case is assumed to be greater than other not-productive states because the machine is actually performing procedures even if it is not processing parts ($w_{su} > w_{id} > w_{sb} \simeq 0$). Finally, the highest power request occurs during the part-processing, i.e. in *busy* state, since this represents the moment of maximum device effort required to properly operate on items: consequently, it is assumed that $w_b > w_{su} > w_{id} > w_{sb} \simeq 0$. It must be noted that the power request while in the working state is a weighted average of w_b and w_{id} depending on the amount of time the machine spends as busy or idle.

3 The Exact 2S-Model

Let us consider a production line consistent with the description of section 2 and $m = 2$ stages, S_i with $i \in \{1, 2\}$. This section is focused on the model to identify an EEC policy for this layout, namely the 2S-Model. The problem is formalized as a Continuous Time MDP (CT-MDP) and then converted into a DT-MDP with the uniformization technique (Lippman, 1975). Then, an LP approach can be used to solve the DT-MDP problem and the exact solution can be identified (Puterman, 2014). Sections from 3.1 to 3.5 describe the MDP. Section 3.6 introduces the LP formulation which allows enriching the MDP formulation by including general production constraints in the problem. Section 3.7 provides a descriptive example of a control policy for the EEC problem at hand.

3.1 *Decision Epochs*

The time horizon is divided into periods $k = \{1, 2, \dots\}$ of variable length, according to the occurrence of event $y_k \in \mathbb{Y}$. The event y_k happens at the end of period k and the decision epochs k correspond to instances of the event y_k. The event set can be defined as $\mathbb{Y} = \{A_1, A_2 \equiv D_1, D_2, E_1, E_2\}$ where event A_i indicates a part arrival to stage i, event D_i a part departure from stage i due to process completion, and event E_i a startup completion for one of the machines at stage i. Trivially, in this model $A_2 \equiv D_1$; furthermore, departures cannot happen when the respective stage is empty, arrivals cannot happen when the correspondent buffer is full, and a startup completion occurs only when at least one machine in the stage is in *startup* state.

3.2 *State Space and Action Space*

The system state is defined as $\mathbf{s} \in \mathbb{S}$, where $\mathbb{S}$ is the discrete state space representing all possible system states, denoted by the ordered vector $\mathbf{s} = \{n_1, n_2, x_1, x_2\}$: integer variable $x_i \in \{0, 1, 2, \dots, c_i\}$ represents the number of machines in working state in S_i and the number of parts in stage S_i is represented with the integer variable $n_i \in \{0, 1, 2, \dots, K_i + c_i\}$.

A certain machine M_{ij} has a service rate equal to $\mu_i > 0$ when busy or 0 otherwise. S_i has, at any given time, capacity equal to $K_i + x_i$, i.e. only working machines can hold parts, and therefore S_i can hold up to $K_i + c_i$ parts at full capacity. State set $\mathbb{S}$ is finite since the number L of possible system states is finite and it is given by all the possible combinations of n_1, n_2, x_1 and x_2. At the beginning of period k, $\mathbf{s}$ is referred to as $\mathbf{s}_k = \{n_{1,k}, n_{2,k}, x_{1,k}, x_{2,k}\}$.

The control action $\mathbf{a} = [a_1, a_2]$ is applied to control the numbers of machines to be in working state x_1 and x_2. Hence, $\mathbf{a}$ determines the switching on/off of machines in S_1 and S_2. The allowable action space is $\mathbb{A}_s(y)$ depending on system state s and event occurrence y and it represents the set of actions that can be chosen in $\mathbf{s}$, i.e. the allowable values $\mathbf{a}$ can assume when a control action is executed. $\mathbb{A}_s(y)$ is determined by part processing that cannot be interrupted by the control, i.e. it is not allowed to choose a control action $\mathbf{a}$ imposing a switch off of a *busy* machine. At the same time, $\mathbb{A}_s(y)$ is also determined by trivial boundaries: (i) if all machines in S_i are already working or executing startup, i.e. switched on, it is not possible to switch on any additional machine in S_i, and, (ii) if all machines are already in standby state, i.e. switched off, it is not allowed to switch off any additional machine in S_i. At the end of the period k, after the event y_k is observed, $x_{1,k}$ and $x_{2,k}$ can be controlled with the action $\mathbf{a}_k = [a_{1,k}, a_{2,k}]$. The optimal policy $\pi^* : \mathbb{S} \times \mathbb{Y} \to \mathbb{A}_s(y)$ maps the optimal action $\mathbf{a}_k^*(\mathbf{s}_k, y_k)$ given system state $\mathbf{s}_k$ and occurrence of event y_k.

3.3 *System Dynamics and Uniformization*

System dynamics is assumed to be stationary and it is represented by functional $\mathcal{Z} : \mathbb{S}$ x $\mathbb{Y}$ x $\mathbb{A}_s(y) \to \mathbb{S}$. Given system state $\mathbf{s}_k$, event y_k and control action $\mathbf{a}_k$, the next system state $\mathbf{s}_{k+1} = \{n_{1,k+1}, n_{2,k+1}, x_{1,k+1}, x_{2,k+1}\}$ is defined as follows:

$$\mathbf{s}_{k+1} = \begin{cases} \{\min[n_{1,k}+1, K_1 + x_{1,k}], n_{2,k}, \min[a_{1,k}, x_{1,k}], \min[a_{2,k}, x_{2,k}]\} \quad if\ y_k = A_1 \\ \{\max[n_{1,k}-1, 0], \min[n_{2,k}+1, K_2 + x_{2,k}], \min[a_{1,k}, x_{1,k}], \min[a_{2,k}, x_{2,k}]\} \quad if\ y_k = A_2 \equiv D_1 \\ \{n_{1,k}, \max[n_{2,k}-1, 0], \min[a_{1,k}, x_{1,k}], \min[a_{2,k}, x_{2,k}]\} \quad if\ y_k = D_2 \\ \{n_{1,k}, n_{2,k}, \min[a_{1,k}, x_{1,k}+1], \min[a_{2,k}, x_{2,k}]\} \quad if\ y_k = E_1 \\ \{n_{1,k}, n_{2,k} \min[a_{1,k}, x_{1,k}], \min[a_{2,k}, x_{2,k}+1]\} \quad if\ y_k = E_2 \end{cases} \tag{1}$$

The number of parts $n_{i,k}$ decreases with the occurrence of departures and increases with arrivals up to the holding capacity of S_i, including parts held in buffer and in working machines, i.e., $K_i + x_{i,k}$. The number of working machines changes according to the control. When $a_{i,k} \leq x_{i,k}$, $x_{i,k} - a_{i,k}$ machines are switched off and immediately enter the standby state, thus the number of working machines $x_{i,k+1} = \min[a_{i,k}, x_{i,k}]$. Whereas, when $a_{i,k} > x_{i,k}$, $a_{i,k} - x_{i,k}$ machines enter in startup to resume the service, and the number of working machines does not change until a startup completion occurs increasing the number of working machines by one unit. In order to fully understand system dynamics, the implicit effect of the control must be expressed. Indeed, the control $a_{i,k}$ also determines the number of machines in standby and startup states in the next period, respectively $su_{i,k+1}$ and $sb_{i,k+1}$. When $a_{i,k} > x_{i,k}$, a switch on command is applied so as $su_{i,k+1} = \max[0, a_{i,k} - x_{i,k}]$. As a consequence, machines in standby are not working nor in startup state: $sb_{i,k+1} = c_i - x_{i,k} - su_{i,k+1}$. In addition, among the working machines, some are actually busy whilst others might be starving of raw parts (i.e., idle). The number of busy machines in S_i is $bu_{i,k} = \min[x_{i,k}, n_{i,k}]$ and, consequently, the number of idle machines is $id_{i,k} = x_{i,k} - bu_{i,k}$ and the overall service rate of S_i is defined as $\mu_{i,k}^{tot} = bu_{i,k}\mu_i$. Finally, the number of parts in buffer B_i is $nb_i = \min[0, n_{i,k} - bu_{i,k}]$. At each period of time, $su_{i,k}$, $sb_{i,k}$, and nb_i are tracked to compute the payoff function (section 3.4). As an illustrative example, let us assume to observe a station with $c_i = 6$ so that $n_{i,k} = 4$ and $x_{i,k} = 2$. $n_i = 4$ indicates that four parts are in S_i where two parts are processed by the $x_{i,k} = 2$ busy machines and two parts wait in the buffer (i.e., $bu_{i,k} = 2, id_{i,k} = 0$, $nb_i = 2$). The control action is $a_{i,k} = 5$ so that a switch on command is issued. Consequently, working machines keep processing parts ($x_{i,k+1} = 2$), three machines enter in startup ($su_{i,k+1} = 3$) and one machine is in standby $sb_{i,k} = 1$. Lastly, if $su_{i,k} > su_{i,k+1}$ the startup is actually interrupted on some machines that are switched off and go into the standby state.

The MDP transition probabilities $p(s_k, s_{k+1}, y_k, \mathbf{a_k})$ at given event y_k are:

$$p(\mathbf{s}_k, \mathbf{s}_{k+1}, y_k = A_1, \mathbf{a_k}) = \begin{cases} 0 \quad if \;\; n_{1,k} = K_1 + x_{1,k} \\ \\ \lambda \quad otherwise \end{cases} \tag{2}$$

$$p(\mathbf{s}_k, \mathbf{s}_{k+1}, y_k = A_2 \equiv D_1, \mathbf{a_k}) = \begin{cases} 0 \quad if \;\; n_{1,k} = 0 \wedge n_{2,k} = K_2 + x_{2,k} \\ \\ \mu_{1,k}^{tot} \;\; otherwise \end{cases} \tag{3}$$

$$p(\mathbf{s}_k, \mathbf{s}_{k+1}, y_k = D_2, \mathbf{a_k}) = \begin{cases} \mu_{2,k}^{tot} \quad if \;\; n_{2,k} > 0 \\ \\ 0 \quad otherwise \end{cases} \tag{4}$$

$$p(\mathbf{s}_k, \mathbf{s}_{k+1}, y_k = E_i, \mathbf{a_k}) = \begin{cases} 0 & if \;\; x_{i,k} = c_i \\ \\ \delta_i & otherwise \end{cases} \tag{5}$$

in this way, the system is described by a continuous time Markov chain. With the uniformization technique, it is possible to convert the continuous time Markov chain into a discrete time Markov chain using a uniform transition rate ν defined as follows (Lippman, 1975): $\nu = \lambda + c_1(\delta_1 + \mu_1) + c_2(\delta_2 + \mu_2)$. Finally, defining a discount factor $0 < \xi < 1$ and $\eta = \xi + \nu$ it is possible to define the transition probabilities for the infinite horizon discounted cost scenario as $\widetilde{p}(s_k, s_{k+1}, y_k, \mathbf{a_k}) = \frac{1}{\eta} p(s_k, s_{k+1}, y_k, \mathbf{a_k})$.

3.4 *Payoff Function*

The payoff function for each production stage S_i consists of four non-negative and finite elements, respectively the working, startup, standby, and holding powers. The working power is the one requested for machines M_{ij} while in working state and it is function of $w_{i,b}$, $w_{i,id}$ and the number of busy and idle machines, respectively bu_i and id_i; thus, the working power for the whole stage S_i is equal to $(bu_i w_{i,b} + id_i w_{i,id})$. Similarly, the startup power is the one required for machines M_{ij} while in startup state and it depends on $w_{i,su}$ and on the number of machines in startup state su_i; thus, the startup power for the whole stage S_i is equal to $w_{i,su} su_i$. The third element is the standby power, representing the request of power by M_{ij} during the standby state; this is directly related to $w_{i,sb}$ and the number of machines in standby state sb_i; therefore, the standby power for the whole stage S_i is $w_{i,sb} sb_i$. Finally, the last item to be considered is the holding power. It is assumed, indeed, that a power request $w_{i,h}$ is required to hold a part in stage S_i. This represents a penalty imposed to the system for maintaining parts in the buffer B_i and not processing them: consequently, as $w_{i,h}$ increases, the control is prone to be more productive. Hence, the holding power for the whole stage S_i is equal to $w_{i,h} bn_i$.

3.5 *Optimality Equation*

All the DT-MDP elements are identified and it is possible to define Bellman's optimality equation for the infinite horizon discounted cost. It must be noticed that: (i) the discount factor ξ and belongs to the interval $0 < \xi < 1$, (ii) the state space $\mathbb{S}$ is discrete and finite, and, (iii) system dynamics and payoff function are stationary, i.e. independent from the period k considered. When these three conditions are verified, there always exists an optimal stationary policy π^* for any MDP evaluated with the infinite horizon discounted cost criterion (Puterman, 2014). This means that also for the considered DT-MDP, there is an optimal policy leading to the solution of Bellman's optimality equation defining our problem and this policy is stationary: it does not change over time and is independent of the period k considered. The control action $\mathbf{a}_k$ depending on the optimal policy π^* and affecting the Bellman's equation is independent of the period k. For ease of notation, $\mathbf{s}_k$, $\mathbf{s}_{k+1}$, y_k and $\mathbf{a_k}$ become $\mathbf{s}$, $\mathbf{s}'$, y and $\mathbf{a}$, and the optimality equation can be written as:

$$V^*(\mathbf{s}) = \min_{\mathbf{a} \in \mathbb{A}_s(y)} \left[g(\mathbf{s}) + \sum_{\mathbf{s}' \in \mathbb{S}} \widetilde{p}(\mathbf{s}, \mathbf{s}', y, \mathbf{a})(V^*(\mathbf{s}') + \eta a_c(\mathbf{s})) \right] \tag{6}$$

where $g(\mathbf{s})$ represents the state cost, $a_c(\mathbf{s})$ the action cost and ϕ is the energy cost:

$$g(\mathbf{s}) = \frac{\phi}{\eta} \sum_{i=1}^{m=2} \left((bu_i w_{i,b} + id_i w_{i,id}) + w_{i,h} bn_i + w_{i,sb} sb_i \right) \tag{7}$$

$$a_c(\mathbf{s}) = \frac{\phi}{\eta} \sum_{i=1}^{m=2} w_{i,su} su_i \tag{8}$$

It must be noticed that both $g(\mathbf{s})$ and $a_c(\mathbf{s})$ are also time-dependent (e.g. how much time a machine is in working state influences the working power). However, the time horizon is taken into account by means of η that introduces the infinite horizon discounted cost criteria in the DT-MDP considered. In this way, the time-dependency is taken into account in the problem. Furthermore, the solution of Equation (6) represents the minimum expected energy cost that the system, starting from state $\mathbf{s}$, will incur when the optimal control action $\mathbf{a}^*$ is applied. The optimal policy π^* maps the optimal action $\mathbf{a}^*$ to be implemented. It is noteworthy that π^* is based only on n_1 and n_2 and event y because $\mathbf{a}^*$ directly controls x_1 and x_2. In this way, the optimal EEC policy for a two-stage production line with parallel machine workstations can be obtained and the unconstrained EEC problem can be solved.

3.6 *LP Formulation with Production Constraints*

LP formulation can be used for determining the optimal control policy for a DT-MDP; with this approach, the LP solution is equivalent to that of the DT-MDP (Puterman, 2014). Examples of this MDP to LP formulation can be found in (Nadar et al., 2016; Hosseini and Tan, 2019); this choice is motivated by the requirement of inserting constraints in the solution provided by the MDP. With this technique, this becomes feasible and it is possible to obtain a constrained optimal solution for a problem modeled with an MDP. Furthermore, it must be noted that: (i) both the state and action costs are bounded (i.e. $|g(\mathbf{s})| \leq Z < \infty \wedge |a_c(\mathbf{s})| \leq Z < \infty$ for all $\mathbf{s} \in \mathbb{S}$ and $\mathbf{a} \in \mathbb{A}_s$), (ii) the state space $\mathbb{S}$ is discrete and, (iii) the allowable action space $\mathbb{A}_s$ is finite for each possible system state $\mathbf{s} \in \mathbb{S}$. Under these three conditions, for an LP formulation for a DT-MDP problem evaluated with the infinite horizon discounted cost criterion, there always exists an optimal solution corresponding to optimal stationary and deterministic policy (Puterman, 2014). This means that, being the policy deterministic, in each state the action choice is performed with certainty. Let us define $\alpha(\mathbf{s})$, satisfying $\sum_{\mathbf{s}\in\mathbb{S}} \alpha(\mathbf{s}) = 1$, as the initial probability distribution over $\mathbb{S}$, and $\beta(\mathbf{s})$ as the total discounted probability that the system occupies state $\mathbf{s}$ given a certain $\alpha(\mathbf{s})$. The decision variable for the LP model is $\beta(\mathbf{s})$ and the objective is the minimization of the infinite horizon discounted cost that can be found solving the following LP problem:

$$min \sum_{\mathbf{s}\in\mathbb{S}} \sum_{\mathbf{a}\in\mathbb{A}_s} \beta(\mathbf{s})(g(\mathbf{s}) + a_c(\mathbf{s})) \tag{9}$$

$$s.t. \qquad \beta(s) - \sum_{\mathbf{s}\in\mathbb{S}} \sum_{\mathbf{a}\in\mathbb{A}_s} \sum_{y\in\mathbb{Y}} \widetilde{p}(\mathbf{s}, \mathbf{s}_{+1}, y, \mathbf{a})\beta(\mathbf{s}) = \alpha(\mathbf{s}) \tag{10}$$

$$0 \leq \beta(\mathbf{s}) \leq 1 \quad \forall \mathbf{s} \in \mathbb{S} \tag{11}$$

$$g(\beta(\mathbf{s})) \geq G^* \tag{12}$$

where Equation (9) is the objective function, Equation (10) is a structural property to be ensured in an MDP to LP formulation (proof in Puterman (2014)), and Equation (11) represents the boundary conditions, i.e. $\beta(\mathbf{s}) \in [0, 1]$. Equation (12) defines a general production constraint to be respected while guaranteeing the minimization of the energy cost. Equation (12) is the novelty introduced compared to a classical MDP formulation and it enables the solution of constrained EEC problems. Equation (12) indicates that a certain performance indicator $g(\beta(\mathbf{s}))$, depending on system probabilities $\beta(\mathbf{s})$ of being in a certain state $\mathbf{s}$, must be higher or equal than a specific target G^*. Function $g(\beta(\mathbf{s}))$ can represent many key performance indicators. System throughput is the most common and valuable example of a target (T_H^{target}) to be met at least, so that Equation (12) becomes:

$$\sum_{\mathbf{s}\in\mathbb{S}} \beta(\mathbf{s})\mu_{i,k}^{tot} \geq T_H^{target} \quad for \;\; i = 1,2 \qquad (13).$$

similarly a minimum availability target A_V^{target} for a certain workstation i becomes:

$$\sum_{\mathbf{s}\in\mathbb{S}} \beta(\mathbf{s})\frac{x_i}{c_i} \geq A_V^{target} \qquad (14)$$

and a maximum WIP (wip^{target}) constraint becomes:

$$\sum_{\mathbf{s}\in\mathbb{S}} \beta(\mathbf{s})(n_1 + n_2) \leq wip^{target} \qquad (15)$$

Define $\beta^*(\mathbf{s})$ as the optimal solution of problem (13-16); from $\beta^*(\mathbf{s})$ the associated optimal EEC policy π^* can be derived, since π^* and $\beta^*(\mathbf{s})$ are directly connected (proof in Puterman (2014)). Moreover, $\beta^*(\mathbf{s})$ and π^* do not depend on the initial state distribution $\alpha(\mathbf{s})$ (proof in Puterman (2014)), which can therefore be arbitrarily selected as long as it stands that $\sum_{\mathbf{s}\in\mathbb{S}} \alpha(\mathbf{s}) = 1$. π^* leads to the optimal solution of the constrained EEC problem for a two-stage production line. It must be also noted that, especially in presence of strict constraints, the optimal solution of the presented LP formulation might lead to an optimal policy π^* equal to the AOn policy.

3.7 *Policy Illustrative Example*

An optimal EEC policy π^* maps the optimal actions $\mathbf{a}^*$ to be implemented in the system, i.e. for each possible state $\mathbf{s}$, π^* indicates the corresponding optimal number of machines in S_1 and S_2 that should be in working state: $[a_1^*, a_2^*]$; moreover, π^* can be based only on n_1 and n_2 because $\mathbf{a}^*$ directly controls x_1 and x_2 (section 3.5). To better clarify how π^* works, let us assume to have $K_1 + c_1 = 4$, $K_2 + c_2 = 3$ and $c_1 = c_2 = 2$. π^* indicates that, for instance, if the system is in state $\mathbf{s} = \{4, 2, 1, 1\}$ and the optimal action for $[n_1, n_2] = [4, 2]$ is $\mathbf{a}^* = [2, 1]$, then one additional machine is switched on in S_1 and none in S_2 leading to $su_1' = 1$; when the startup on this machine will be completed, the state will become $\mathbf{s'} = \{4, 2, 2, 1\}$. In order to give a benchmark, the AOn policy would indicate to maintain always 2 machines switched on in both stages: $\mathbf{a} = [2, 2]$, for any state possible state $\mathbf{s}$, i.e. any value of $[n_1, n_2]$.

4 The Approximate Model for Long Production Lines

Let us consider a production line consistent with section 2 and $m > 2$ stages, S_i with $i \in \{1, \dots, m\}$. As easily understandable, the problem size grows drastically with m and, consequently, an exact analytical solution cannot be identified for large values of m. This section describes an approximated solving approach for $m > 2$, namely the *Backward-Recursive* approach. The main idea is (i) to break down the original problem into a series of two-stage sub-systems (couple $[S_i; S_{i+1}]$), i.e. a series of sub-problems solvable in an exact way, (ii) to solve the last sub-problem (couple $[S_{m-1}; S_m]$) so that local optimal policy π_i^* with $i = m$ is found, (iii) to proceed backward towards the first sub-problem (couple $[S_1; S_2]$) solving recursively sub-problems. The locally exact sub-problem solutions are combined to approximately identify a unique EEC policy for the entire production line under analysis. An extended version of the 2S-Model is required to comply with this approach; indeed, three additional issues must be addressed: the blocking condition, the policy-separation assumption, and the policy-uniqueness constraint.

4.1 *The Extended-2S*

The 2S-Model is extended to model generally two consecutive stages of a production line, i.e. the couple $[S_{i-1}; S_i]$. First of all, if $i < m$ stage S_i might be blocked and the model must be extended considering the *Blocking Condition*. Secondly, as for the creation of sub-problems in the proposed approach, each stage S_i with $i \in \{2, ..., m-1\}$ is included in two sub-problems (couples $[S_{i-1}; S_i]$ and $[S_i; S_{i+1}]$). Nevertheless, the obtained policy to be applied in stage S_i must be consistent in mapping actions to states for S_i, and a *Policy-Uniqueness* constraint is added. Lastly, the proposed approach requires the EEC policy of a stage to be independent of the other stages resulting in a simplification of the EEC policy applied. Thus, the *Policy-Separation* assumption is introduced.

(i) The Blocking Condition: for the generic couple $[S_{i-1}; S_i]$ with $i < m$, stage S_i might block because buffer B_{i+1} is full. Assuming that sub-problem $[S_i; S_{i+1}]$ has been solved previously, the policy π_{i+1}^* to be applied to $[S_i; S_{i+1}]$ is known. The steady-state probability $P_{bl,i}$ of having the buffer B_{i+1} full and S_i blocked can be computed with a Markov chain representing the behaviour of couple $[S_i; S_{i+1}]$. At the same time, the probability of having B_{i+1} not-full and S_i not-blocked will be $1 - P_{bl,i}$. Hence, the blocking of S_i can be represented with a Bernoulli variable and the Extended-2S is modified as follows: a Bernoulli state variable bl representing the blocking of S_i is included in system state $\mathbf{s} = \{n_{i-1}, n_i, x_{i-1}, x_i, bl\}$, where $bl = 1$ if S_i is blocked and 0 otherwise. All the remainder of the Extended-2S is consistent with section 3. The formulation of the problem to be solved is still described by Equations (9-12) except for system dynamics (Equations 10) which now includes the blocking event. Indeed, when the system may have $bl_k = 1$ with a probability $P_{bl,i}$ and $bl_k = 0$ with $1 - P_{bl,i}$, in compliance with the Bernoulli distribution. Lastly, in a blocked stage, departures cannot occur.

(ii) The Policy-Separation Assumption: this simplifies the EEC policy applied so as it is assumed that actions a_i^* only depend on stage S_i, i.e. $a_i^*(n_i)$, for all stages. Thus, policy π_i^* for couple $[S_{i-1}, S_i]$ maps actions $a_{i-1}^*(n_{i-1})$ and $a_i^*(n_i)$ independently. To better clarify, as an example, let us consider the couple $[S_{i-1}, S_i]$ and assume $c_{i-1} = c_i = 4$ and $K_{i-1} = K_i = 5$. In a specific moment, let us assume that $n_i = 3$, i.e. there are 3 parts in S_i, and $x_i = 1$, i.e. there is 1 working machine in S_i. If π^* indicates that, for instance, the associated optimal action for $n_i = 3$ is ${a_i}^* = 2$, i.e. there should be 2 working machines in S_i, then one additional machine is switched on in S_i, and this is independent of the n_{i-1} and x_{i-1} values; on the other hand, ${a_{i-1}}^*$ for S_{i-1} does not depend on n_i and x_i.

(iii) The Policy-Uniqueness Constraint: let us consider three consecutive stages S_{i-i}, S_i and S_{i+1} of the production line forming two couples: $[S_{i-1}; S_i]$ and $[S_i; S_{i+1}]$. Let us define π_{i+1}^* as the solution obtained from solving the sub-problem associated to $[S_i; S_{i+1}]$ and, similarly, π_i^* for $[S_{i-i}; S_i]$. While considering the whole line, decisions regarding stage S_i must be unique, thus, policy π_i^* and policy π_{i+1}^* must be consistent in mapping actions to states for S_i. The following *Policy-Uniqueness* constraint is added to the LP problem: $a_i^* = \widetilde{a}_i^*$. In this case $\widetilde{a}_i^*$ is the action on stage S_i selected by π_{i+1}^*, and, similarly a_i^* is the action on S_i selected by π_i^*.

4.2 *The Backward-Recursive Approach*

The proposed Backward-Recursive approach is represented in Figure 3 and fully described in Algorithm 1. The algorithm starts breaking down the original system into a series of two-stage sub-systems. Couples $[S_{i-1}; S_i]$ with $i \in \{2, \ldots, m\}$ are created (*STEP 1*) and each couple represents a sub-problem that can be solved in an exact way. Starting from the end of the line ($i = m$), the algorithm solves sub-problem $[S_{i-1}; S_i]$ with the Extended-2S model (*STEP 2*). Since $P_{bl,m} = 0$ and stage S_m does not require the *Policy-Uniqueness* constraint, the sub-problem is solvable and policy π^*_m is found. Therefore, actions $\tilde{a} = \tilde{a}^*_{i-1}$ can be extracted. Sub-system Markov chain is created in *STEP 3* and probability $\tilde{p} = P_{bl,m-1}$ is computed. Moving backward, the following sub-system is solved. Therefore *STEP 4* recursively imposes $i = i - 1$ until $i = 2$ and the general couple $[S_{i-1}; S_i]$ is solved with the Extended-2S model. The boundary conditions $P_{bl,i} = \tilde{p}$ and $\tilde{a}^*_i = \tilde{a}$ are known. Also, the sub-system Markov chain is created and probability $\tilde{p} = P_{bl,i-1}$ is computed. Lastly, *STEP 5* combines all the optimal local EEC policies π^*_i with $i \in \{2, \ldots, m\}$ in the unique approximate policy Π^* that represents system control policy to reduce the energy consumption for the entire line.

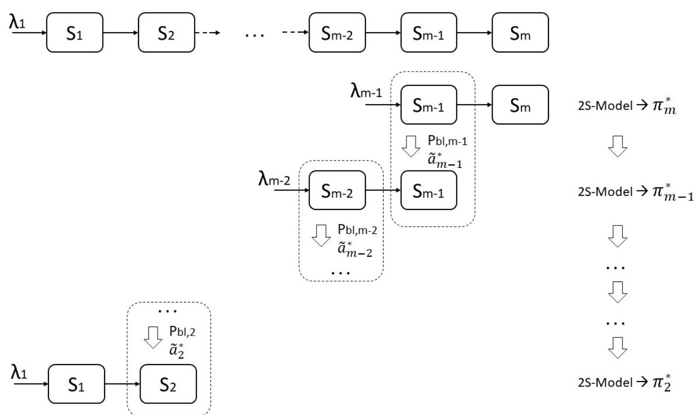


Figure 3: Illustration of the "Backward-Recursive" approach.

5 Numerical Experiments with Two-Stage Lines

A numerical analysis is carried out to show the performance of the 2S-Model. Section 5.1 assesses the computation time as the number of possible system states enhances. Model effectiveness is studied in section 5.2 and in section 5.3 a sensitivity analysis is performed. In all the experiments the energy cost is $\phi = 1$, the discount factor is $\xi = 0.80$, and arrival rate is $\lambda = 0.04$. The model is implemented with *Matlab R2020a* and *ILOG CPLEX 12.10* and results are obtained with 4.90GHz i7 Intel Core and 16GB RAM. The computation times correspond to the overall experiments duration with both softwares.

Algorithm 1 The Backward-Recursive approach.

STEP 1: Couple the m stages: $[S_{i-1}; S_i]$ with $i \in \{2, \ldots, m\}$

STEP 2: Solve sub-problem $[S_{m-1}; S_m]$

Do $P_{bl,m} = 0$; $i = m$

Relax Policy-Uniqueness Constraint and solve the sub-system with the Extended-2S model

Extract $\tilde{a} = \widetilde{a}^*_{m-1}$ from obtained π^*_m

STEP 3: Evaluate sub-system $[S_{m-1}; S_m]$ *under* π^*_m

Create the Markov chain of $[S_{m-1}; S_m]$

Compute $\tilde{p} = P_{bl,m-1}$

STEP 4: Recursively solve sub-system $[S_{i-1}; S_i]$

while $i \geq 2$ **do**

Do $i = i - 1$; $P_{bl,i} = \tilde{p}$; $\widetilde{a}^*_i = \tilde{a}$

Solve sub-system $[S_{i-1}; S_i]$ with the Extended-2S model

Extract $\tilde{a} = \widetilde{a}^*_{i-1}$ from obtained π^*_i

Create the Markov chain of $[S_{i-1}; S_i]$

Compute $\tilde{p} = P_{bl,i-1}$

end while

STEP 5: Combine obtained policies π^*_i with $i \in \{2, \ldots, m\}$ and find Π^*

5.1 *Computation Time Analysis*

The problem size, i.e. the number of system states L, directly impacts on the computation time to reach a solution. Given by all the possible combinations of state variables (n_1, n_2, x_1 and x_2), L can be computed as: $L = (c_1 + K_1 + 1)(c_2 + K_2 + 1)(c_1 + 1)(c_2 + 1)$. Without loss of generality, we considered $K = K_1 = K_2$ and $c = c_1 = c_2$. Then, 13 scenarios varying the buffer capacity K and the number of machines c are sampled, so that different problem sizes are represented. Each scenario is replicated (10 replications) and the computation time is extracted with a 95% confidence level on the mean value. Figure 4 shows the results of this analysis as confirmation that the computation time grows significantly when L increases. Also, time variability is really low, as shown in Figure 4, and this confirms the effectiveness of the 2S-Model in a limited amount of time. In detail, until L is lower or equal than 2000, the solution can be reached in less than 3 minutes while if L is higher than 5500, the 2S-Model starts taking more than 1 hour to provide a solution. Since there is a high number of possible system configurations leading to less than 5500 possible system states the 2S-Model nearly always leads to an exact solution in a short-medium amount of time.

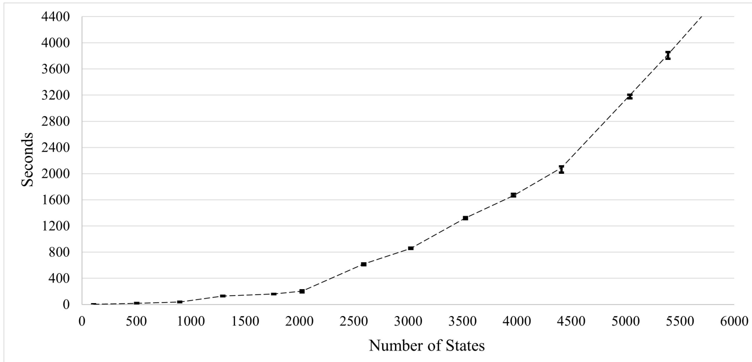


Figure 4: Time duration of the experiments vs number of system states.

5.2 *2S-Model Effectiveness*

Firstly, we introduce a novel indicator named the *Power-Request Configuration Ratio* or *PCR*. It compares the power requested in EEC-related states (i.e., *startup* and *standby*) with that consumed in the states representing machine common behavior (i.e., *busy* and *idle*) under the AOn policy; thus, it indicated machine EEC potential. We define: $PCR_i = (w_{i,b} + w_{i,id})/(w_{i,sb} + w_{i,su})$. High PCR indicates major saving potential, while low PCR indicates minor potential because machine power during the working states is similar to that in EEC-related states. We consider 2-stage lines with equal buffer capacity $K = K_1 = K_2$, constant holding power $w_h = w_{1,h} = w_{1,h}$, and stages composed by machines with equal power requests and equal startup rate $\delta = \delta_1 = \delta_2$. The choice to assume identical K, δ, w_h, and machine power consumption in both stages does not lead to any loss of generality. A 2^k factorial design with center points (Montgomery, 2017) with nine factors at two levels (Table 1) is used to generate 516 different experiments. The center points are considered to test the linearity effect of factors on the resulting energy saving. Factors are: the buffer capacity K, the numbers of machines c_1 and c_2, the startup rate δ, the holding cost w_h, the system configuration (i.e., balanced or not balanced), the station saturation level ρ in isolation, the PCR, and the throughput target (i.e., constrained or unconstrained problem). In addition, a center point is added for each numerical factor, generating 4 additional experiments (Table 1). In the center points, the value of each numerical factor corresponds to the "center" value between the low and the high level in the 2^k factorial design (e.g. center for K is 4 since the low level is 2 and the high is 6). However, "Constraint" and "Balanced" factors cannot have a "center" since they can only be equal to "Yes" or "No".

Table 1: Factors and levels for the 2^k factorial design along with the 4 center points values.

2^k **factorial Design**									
Factor	K	δ	c_1	c_2	ρ	PCR	w_h	Constraint	Balanced
Low Level	2	0.02	2	2	0.3	1.2	0.5 (kW)	Yes	Yes
High Level	6	0.1	6	6	0.9	4	10 (kW)	No	No
Center Points									
Factor	K	δ	c_1	c_2	ρ	PCR	w_h	Constraint	Balanced
Exp. 1 - Values	4	0.06	4	4	0.6	2.6	5.25	Yes	Yes
Exp. 2 - Values	4	0.06	4	4	0.6	2.6	5.25	Yes	No
Exp. 3 - Values	4	0.06	4	4	0.6	2.6	5.25	No	Yes
Exp. 4 - Values	4	0.06	4	4	0.6	2.6	5.25	No	No

Hence, the number of additional experiments is equal to 4: an experiment is generated for each possible combination of values for the two categorical factors, while the values of the numerical factors are maintained fixed and equal to their respective "center" values. The saturation ρ_i of stage i is computed as follows: $\rho_i = \lambda/(c_i\mu_i)$. Therefore, μ_i is computed for each experiment given λ and the factors c_1, c_2 and ρ. Machine power requests in experiments with $PCR = 1.2$ are: $[w_{i,sb}, w_{i,su}, w_{i,id}, w_{i,b}] = [0, 9.5, 1.5, 10]$ kW. Similarly for $PCR = 4$: $[w_{i,sb}, w_{i,su}, w_{i,id}, w_{i,b}] = [0, 6.25, 5, 20]$ kW. The system configuration can be: (i) balanced, imposing $\rho_1 = \rho_2$, or (ii) not-balanced, imposing $\rho_1 = 0.9\rho_2$, i.e., stage S_2 is the bottleneck. Lastly, experiments including a throughput constraint where the target level T_H^{target} to be satisfied is imposed as equal to the 90% of the maximum throughput achievable by the most saturated stage in the line.

The designed experiments represent a variety of configurations of manufacturing systems. At first, in this analysis, the 2S-Model is applied to all the 516 manufacturing systems generated, leading to a suitable EEC policy for each case. Subsequently, for each case, it is computed the percentage of energy saving when the respective EEC policy is applied in comparison to the same configuration but with the AOn policy applied. In all the 516 analyzed cases, the application of the 2S-Model is able to offer an appropriate and effective EEC policy, and, for this reason, the model effectiveness is verified. Figure 5 shows the percentages of energy saving obtained when the identified policy π^* differs from the AOn policy. Indeed, in 95 cases, the optimal policy is actually keeping the machines always ready for process parts (idle state) due to specific conditions such as a high saturation level and a strict throughput constraint. As a result, the machines are required to work at full capacity and, for this reason, a switch off is not advantageous. On the other hand, in more than 80% of the analyzed cases (421), switching off/on the machines leads to energy saving.

The maximum saving is equal to 33.68%. The highest savings are achieved, as expected, in cases where saturation is low ($\rho = 0.3$) so that few parts are in the system and machines are frequently starving. The main effect plots for the resulting percentage energy saving are obtained and represented in Figure 6 along with the Kruskal-Wallis test results to assess the significance of each factor. The high p-values for c_1 and c_2 indicate that these factors are not significant in this analysis. Moreover, the main effect plot shows that higher savings can be achieved when: the system is unbalanced, there is not any throughput constraint, the startup rate δ is high (i.e., the startup time is low), the saturation level is low, the PCR is high, the holding power w_h is low, and buffer capacity K is high. Lastly, Figure 6 shows how the center points do not deny the linear effect on the results.

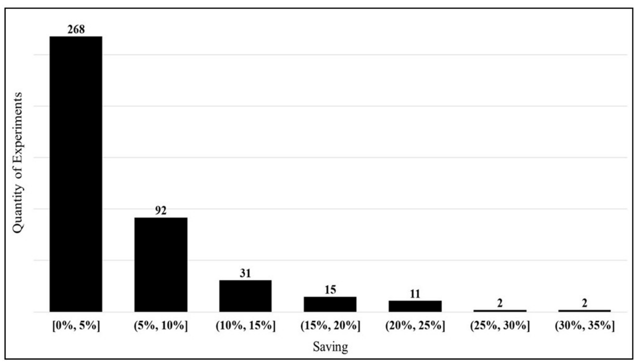


Figure 5: 2^k factorial analysis: achieved energy saving when π^* differs from the AOn policy.

Factor	P-Value
Balanced System	0.01
Constraint	0.00
δ	0.00
K	0.00
c_1	0.52
c_2	0.28
ρ	0.00
$PCCR$	0.00
w_h	0.00

Figure 6: 2^k factorial analysis: main effects plot with Kruskal-Wallis test p-values.

5.3 *Sensitivity Analysis*

A sensitivity analysis is performed to detail the effect of the significant factors: w_h, PCR, δ and K. A set of experiments is designed starting from four main configurations (Table 2), namely: *Best, Medium - 1, Medium - 2,* and *Worst.* The *Worst* configuration is the experiment from the 2^k factorial design of section 5.2 that obtains the lowest savings (Figure 6) and, similarly, the *Best* configuration is the experiment obtaining the highest savings. Configurations Medium - 1 and *Medium - 2* represent intermediate configurations. For each configuration, the sensitivity analysis varies one significant factor at a time. Figure 7 shows the percentage of energy savings obtained for the evaluated cases.

Table 2: Base configurations for the sensitivity analysis.

Configuration	Balanced System	Constraint	δ	ρ	PCR	w_h	K
Best	No	No	0.1	0.3	4	0.5 (kW)	6
Medium - 2	No	No	0.05	0.7	2.5	1 (kW)	4
Medium - 1	Yes	No	0.03	0.8	2	4 (kW)	4
Worst	Yes	Yes	0.02	0.9	1.2	10 (kW)	6

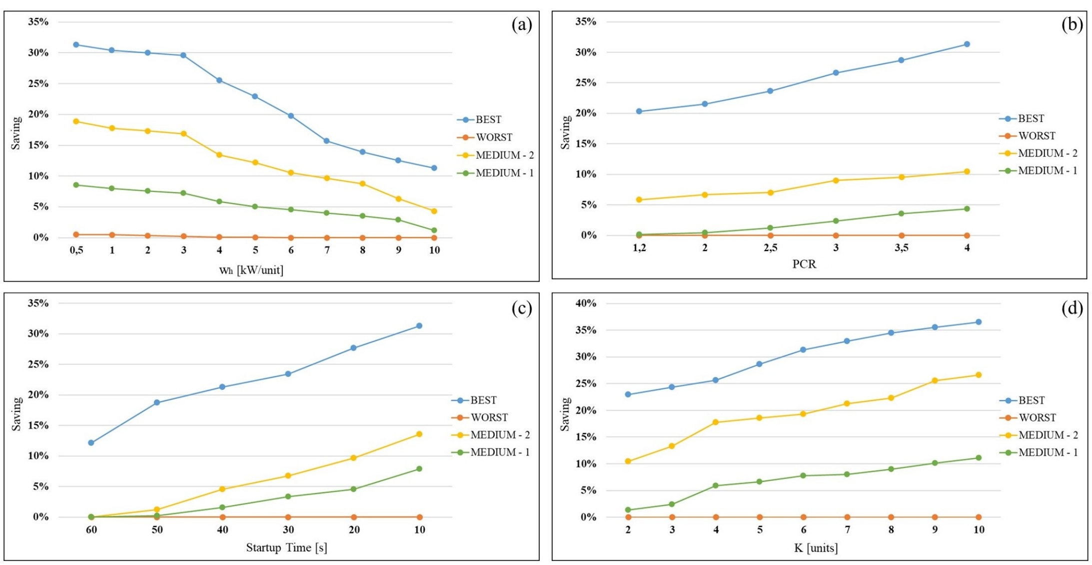


Figure 7: Sensitivity analysis when w_h (a), PCR (b), startup time (c) and K vary.

For the *Worst* configuration, the obtained savings are always null (i.e., 0%) because the optimal policy π^* is actually the *AOn*. For all the other configurations, the sensitivity analysis shows what follows. When w_h increases the saving decreases: to avoid high holding power consumption, parts are processed and not maintained in the buffer, leading to a reduced number of switching off actions and reduced savings. Similarly, the saving decreases when K increases, since maintaining a growing number of parts in the buffers leads to higher holding power consumption: to avoid that, the number of switching off actions is reduced and, consequently, also the saving. Furthermore, the saving increases also when PCR increases: high PCR indicates major saving potential. Then, increasing δ increases (i.e. decreasing startup time) also leads to better savings: a fast startup leads to a faster switch on action, increasing the possibility of switching on/off one or more machines during the line operation. Finally, as the saturation increases, the saving decreases as well: high saturation leads to really rare idle machine periods.

6 Numerical Experiments with Longer Lines

A numerical analysis is carried out to show the effects of the proposed Backward-Recursive approach for lines with $m > 2$. The Backward-Recursive approach is used to solve a set of problems. Discrete Event Simulation (DES) is used to estimate the actual performance of the obtained solutions. Expected savings obtained with the Backward-Recursive approach are compared with those obtained with DES. The expected throughput loss computed with DES is also compared to the target throughput loss.

6.1 *Design of Experiments*

The experimental campaign is focused on two system layouts: a medium line composed of five-stages and a long line with fifteen-stages. For the sake of simplicity, the production lines are composed of stages of two types: A (or ws_A) and B (or ws_B) with parameters as in Table 3. Parameters for ws_A and ws_B are selected considering insights from section 5 as well as values to have studied systems aligned with realistic cases in manufacturing. As confirmed by its parameters, ws_A represents a type of workstation very fast from a processing time point of view (low saturation level along with high δ) and characterized by high power requests (high PCR); this combination of parameters makes ws_A very suitable for EEC application, since, according to section 5, high savings are expected for this workstation. On the contrary, ws_B represents a slow workstation (high saturation level along with low δ) characterized by low consumption (low PCR); this means that, according to section 5, EEC application on ws_B should not lead to great savings. Hence, ws_A and ws_B represent two opposite types of workstations from an EEC point of view and therefore are considered worthy of interest for this analysis.

Table 3: Parametes of ws_A and ws_B.

Stage	K	c	δ	ρ	PCR	w_h
A (ws_A)	5	2	0.05	0.75	2.5	1 (kW)
B (ws_B)	5	2	0.02	0.9	1.2	3 (kW)

In all the experiments the energy cost is $\phi = 1$, the discount factor is $\xi = 0.80$, and the arrival rate is $\lambda = 0.04$. Also, a throughput constraint is imposed such that the expected throughput loss of the system is at most 3% with respect to the same configuration but applying the AOn policy. Both for medium and long lines, four different cases are studied: (case i) a balanced case with only B-type stages, and three unbalanced cases where the slowest stages (B-type) are at the beginning of the line (case ii), in the central part of the line (case iii), and at the end of the line (case iv). These configurations are selected according to the results of section 5: case (i) should lead to lower savings due to the balanced system analyzed while cases (ii-iii-iv) should produce higher savings. DES models are developed in *Matlab* environment. Both energy saving and throughput loss are extracted with a 95% confidence level on the mean value: thus, simulations are replicated 10 times. The simulation ends after producing 5000 parts, a value ensuring short width for the confidence interval on results, and simulation warm-up is the production of 1000 parts: this represents an overestimation, for computational-accuracy reasons, of the transient period identified with the Welch method (Welch, 1983).

6.2 *Results*

Tables 4 and 5 show the experimental results for analyzed configurations with, respectively, five and fifteen stages. The expected throughput TH and the expected energy consumed per produced part EN when the AOn policy is applied are computed with DES and reported as a reference. In all evaluated cases, the DES throughput loss is higher than the target: the approximation introduced by the Backward-Recursive approach might underestimate the blocking probability so that the actual throughput is lower than expected. As for the five-stages, this difference is below 1% both for throughput loss and savings and this confirms the effectiveness of the proposed model in solving medium lines. On the other hand, in the fifteen-stages cases, the differences are higher (up to 2.5%) since the approximation introduced increases with the number of stages: the model does not lose effectiveness for long lines but leads to near-optimal solutions. In all the analyzed cases, the application of the Backward-Recursive approach is able to significantly reduce energy consumption while slightly decreasing the system throughput, in compliance with the imposed production constraint (except for the aforementioned approximations). In particular, the achieved savings for the five-stage cases are included in a range going from about 3.50 to about 5% while for the fifteen-stage cases the savings increase drastically (from about 12 to about 15.50%). Hence, applying EEC to more workstations in a production line leads to higher benefits in terms of environmental impact. Finally, for both configurations the balanced system, as expected, is characterized by lower energy saving than the unbalanced systems.

Table 4: Results of five-stage production line experiments.

Stages Sequence	$TH(AOn)$ [part/min]	$EN(AOn)$ [kJ/part]	Savings	Savings DES	Th.loss Target	Th.loss DES
i: *B-B-B-B-B*	2.08 ± 0.01	1.46 ± 0.01	3.40%	3.52 ± 0.08%	3%	3.11 ± 0.05%
ii: *B-B-A-A-A*	2.17 ± 0.02	2.29 ± 0.02	4.43%	4.57 ± 0.07%	3%	3.23 ± 0.07%
iii: *A-A-B-A-A*	2.20 ± 0.02	2.34 ± 0.01	4.68%	4.92 ± 0.11%	3%	3.42 ± 0.08%
iv: *A-A-A-B-B*	2.16 ± 0.03	2.23 ± 0.01	4.99%	5.11 ± 0.10%	3%	3.29 ± 0.06%

Table 5: Results of fifteen-stage production line experiments.

Stages Sequence	$TH(AOn)$ [part/min]	$EN(AOn)$ [kJ/part]	Savings	Savings DES	Th.loss Target	Th.loss DES
i: *B-B-B-B-B-B-B-B-B-B-B-B-B-B-B*	1.67 ± 0.02	9.80 ± 0.03	9.80%	11.91 ± 0.09%	3%	3.96 ± 0.09%
ii: *B-B-A-B-B-A-A-A-A-A-A-A-A-A-A*	1.86 ± 0.01	12.74 ± 0.04	13.32%	15.39 ± 0.11%	3%	4.15 ± 0.11%
iii: *A-A-A-A-A-B-B-A-B-B-A-A-A-A-A*	1.94 ± 0.01	14.01 ± 0.05	12.40%	14.34 ± 0.23%	3%	4.43 ± 0.08%
iv: *A-A-A-A-A-A-A-A-A-A-B-B-A-B-B*	1.84 ± 0.01	12.10 ± 0.04	13.60%	15.59 ± 0.22%	3%	4.89 ± 0.10%

Two main insights can be extracted from numerical results. First of all, the obtained EEC policies are *threshold based* so that the switch off/on of each machine is triggered at two specific buffer level values, defined as n_{ij}^{off} and n_{ij}^{on}, to respectively switch off and on M_{ij} in S_i. Numerical evidence shows that the control policy might be simplified without affecting the results. Threshold values for the analyzed cases are reported in Tables 6 and 7 showing that the number of active machines increases as parts accumulate in buffers. As an example, let us focus on stage S_1 of case (i) for the five-stage layout: the EEC policy indicates that in S_1, when $n_1 = 1$ the first machine must be in working state, and the same occurs for the second machine when $n_1 = 2$: S_1 has $n_{11}^{on} = 1$ and $n_{12}^{on} = 2$; for the same

Table 6: Thresholds obtained for the five-stage configuration experiments.

Case	**i**	**ii**	**iii**	**iv**
Stage	$(n_{i1}^{off}, n_{i1}^{on})$ $(n_{i2}^{off}, n_{i2}^{on})$	$(n_{i1}^{off}, n_{i1}^{on})$ $(n_{i2}^{off}, n_{i2}^{on})$	$(n_{i1}^{off}, n_{i1}^{on})$ $(n_{i2}^{off}, n_{i2}^{on})$	$(n_{i1}^{off}, n_{i1}^{on})$ $(n_{i2}^{off}, n_{i2}^{on})$
S_1	$(0,1)$ $(1,2)$	$(-,0)$ $(0,1)$	$(0,1)$ $(2,4)$	$(0,1)$ $(1,2)$
S_2	$(0,1)$ $(1,2)$	$(-,0)$ $(0,1)$	$(0,1)$ $(1,3)$	$(0,3)$ $(2,4)$
S_3	$(0,1)$ $(1,2)$	$(0,1)$ $(1,3)$	$(-,0)$ $(2,4)$	$(0,1)$ $(0,1)$
S_4	$(0,1)$ $(1,2)$	$(0,2)$ $(2,4)$	$(0,1)$ $(1,2)$	$(-,0)$ $(1,2)$
S_5	$(0,1)$ $(1,2)$	$(0,2)$ $(3,4)$	$(0,1)$ $(1,2)$	$(0,1)$ $(2,3)$

Table 7: Thresholds obtained for the fifteen-stage configuration experiments.

Case	**i**	**ii**	**iii**	**iv**
Stage	$(n_{i1}^{off}, n_{i1}^{on})$ $(n_{i2}^{off}, n_{i2}^{on})$	$(n_{i1}^{off}, n_{i1}^{on})$ $(n_{i2}^{off}, n_{i2}^{on})$	$(n_{i1}^{off}, n_{i1}^{on})$ $(n_{i2}^{off}, n_{i2}^{on})$	$(n_{i1}^{off}, n_{i1}^{on})$ $(n_{i2}^{off}, n_{i2}^{on})$
S_1	$(0,1)$ $(1,2)$	$(-,0)$ $(-,0)$	$(0,1)$ $(1,3)$	$(0,1)$ $(1,2)$
S_2	$(0,1)$ $(1,2)$	$(-,0)$ $(-,0)$	$(0,1)$ $(1,2)$	$(0,1)$ $(1,2)$
S_3	$(0,1)$ $(1,2)$	$(-,0)$ $(-,0)$	$(0,1)$ $(1,2)$	$(0,1)$ $(2,3)$
S_4	$(0,1)$ $(1,2)$	$(-,0)$ $(-,0)$	$(0,1)$ $(1,2)$	$(0,1)$ $(2,3)$
S_5	$(0,1)$ $(1,2)$	$(0,1)$ $(0,2)$	$(0,1)$ $(1,3)$	$(0,1)$ $(2,3)$
S_6	$(0,1)$ $(1,2)$	$(0,1)$ $(0,2)$	$(0,1)$ $(2,3)$	$(0,1)$ $(0,2)$
S_7	$(0,1)$ $(1,2)$	$(0,1)$ $(1,3)$	$(0,1)$ $(2,3)$	$(0,1)$ $(0,2)$
S_8	$(0,1)$ $(1,2)$	$(0,2)$ $(1,3)$	$(0,1)$ $(1,2)$	$(0,1)$ $(1,2)$
S_9	$(0,1)$ $(1,2)$	$(0,1)$ $(2,3)$	$(0,1)$ $(1,2)$	$(0,1)$ $(1,2)$
S_{10}	$(0,1)$ $(1,2)$	$(0,1)$ $(2,3)$	$(0,1)$ $(1,2)$	$(0,1)$ $(1,2)$
S_{11}	$(0,1)$ $(1,2)$	$(0,1)$ $(2,3)$	$(0,1)$ $(1,2)$	$(0,1)$ $(2,3)$
S_{12}	$(0,1)$ $(1,2)$	$(0,1)$ $(1,3)$	$(0,1)$ $(1,2)$	$(0,1)$ $(1,2)$
S_{13}	$(0,1)$ $(1,2)$	$(0,2)$ $(1,3)$	$(0,1)$ $(1,2)$	$(0,1)$ $(2,3)$
S_{14}	$(0,1)$ $(1,2)$	$(0,2)$ $(1,3)$	$(0,1)$ $(1,2)$	$(0,1)$ $(3,4)$
S_{15}	$(0,1)$ $(1,2)$	$(0,1)$ $(1,2)$	$(0,1)$ $(1,2)$	$(0,1)$ $(1,2)$

reason, S_1 has $n_{11}^{off} = 0$ and $n_{12}^{off} = 1$, since the second machine will be switched off when $n_1 = 1$ and the first when $n_1 = 0$. As expected, in the unbalanced systems A-type stages are less saturated than B-type ones, and, consequently, they are frequently idle and might be switched off to save energy. Finally, $(n_j^{off}, n_j^{on}) = (-, 0)$ indicates to apply AOn policy on that machine; this often occurs for the B-type stages in the unbalanced cases: being highly-saturated, these machines are almost never idle. Lastly, regarding long production lines as the cases with fifteen-stages, future work could regard the use of Π^* as a starting point for a calibration process of the policy parameters; afterward, a suitable and modified EEC policy Π^*_{suit} can be identified and applied in the system, leading to a resulting energy saving and throughput level closer to the target.

7 Discussion and Conclusions

In this work, it is proposed a novel method to address the problem of controlling multi-stage production lines composed of parallel machine workstations with EEC policies. The approach minimizes energy consumption and includes desired target levels on the system performance indicators as problem constraints. The proposed method, exact for two-stage lines with Markovian processes, allows setting different and general production constraints on many key performance indicators leading to the optimum trade-off between energy demand and system performance. Numerical results show that the proposed approach is effective for all evaluated cases and that the solution obtained does not lose effectiveness for long lines, leading to near-optimal solutions. The effectiveness of the proposed approach might be directly translated into a successful application in a real-world industrial case where, starting from the actual system parameters, the model could be applied to lead to enhanced industrial processes sustainability without jeopardizing system performance indicators. Results obtained highlight that higher savings can be achieved with workstations characterized by: (i) low saturation, (ii) low holding power consumption, (iii) high PCR, (iv) short startup time, and (v) high buffer capacity. Thus, from the point of view of practitioners, a preemptive analysis of the line parameters is useful to assess the saving potential of the EEC. Furthermore, in unbalanced systems less saturated stages are frequently idle: the operation of these stages is more affected by EEC since the reduction of their idle period also means a reduction of their environmental impact; on the other hand, more saturated stages, i.e. bottlenecks, are rarely switched off since they are almost never idle. This last piece of information is also a useful insight to be applied in practice, to understand which are the workstations with more saving potential and where to apply a proper EEC policy.

Limitations of this work are related to the assumptions regarding the system under control, especially the exclusive presence of Markovian processes, and the approximations introduced for long lines. Future efforts will be devoted to extend the proposed approach to represent long lines and systems with more general assumptions. In this case, the proposed approach could be used as starting point for a local search aiming at tuning the control and improving the solution. Lastly, a further development might be the use of machine learning techniques in the model to adapt the control to non-stationary system dynamics.

Data availability statement: The authors confirm that the data supporting the findings of this study are available within the article its supplementary materials.

References

Anghinolfi, D., M. Paolucci, and R. Ronco (2021). A bi-objective heuristic approach for green identical parallel machine scheduling. *European Journal of Operational Research 289*(2), 416–434.

Can, A., G. Thiele, J. Krüger, J. Fisch, and C. Klemm (2019). A practical approach to reduce energy consumption in a serial production environment by shutting down subsystems of a machine tool. *Procedia Manufacturing 33*, 343–350.

Center, B. P. (2020). Annual energy outlook 2020. *Energy Information Administration, Washington, DC 12*, 1672–1679.

Chang, Q., G. Xiao, S. Biller, and L. Li (2012). Energy saving opportunity analysis of automotive serial production systems (march 2012). *IEEE Transactions on Automation Science and Engineering 10*(2), 334–342.

Cui, P.-H., J.-Q. Wang, Y. Li, and F.-Y. Yan (2021). Energy-efficient control in serial production lines: Modeling, analysis and improvement. *Journal of Manufacturing Systems 60*, 11–21.

Dahmus, J. B. and T. G. Gutowski (2004). An environmental analysis of machining. In *ASME Int. Mech. Eng. Congress and Exposition*, Volume 47136, pp. 643–652.

Dallery, Y. and S. B. Gershwin (1992). Manufacturing flow line systems: a review of models and analytical results. *Queueing Systems 12*(1), 3–94.

Frigerio, N., C. F. Cornaggia, and A. Matta (2021). An adaptive policy for on-line energy-efficient control of machine tools under throughput constraint. *Journal of Cleaner Production 287*, 125367.

Frigerio, N. and A. Matta (2015). Analysis on energy efficient switching of machine tool with stochastic arrivals and buffer information. *IEEE Transactions on Automation Science and Engineering 13*(1), 238–246.

Gahm, C., F. Denz, M. Dirr, and A. Tuma (2016). Energy-efficient scheduling in manufacturing companies: A review and research framework. *European Journal of Operational Research 248*(3), 744–757.

Govil, M. K. and M. C. Fu (1999). Queueing theory in manufacturing: A survey. *Journal of Manufacturing Systems 18*(3), 214–240.

Heydar, M., E. Mardaneh, and R. Loxton (2022). Approximate dynamic programming for an energy-efficient parallel machine scheduling problem. *European Journal of Operational Research 302*(1), 363–380.

Hosseini, B. and B. Tan (2019). Modelling and analysis of a cooperative production network. *International Journal of Production Research 57*(21), 6665–6686.

Hu, L., C. Peng, S. Evans, T. Peng, Y. Liu, R. Tang, and A. Tiwari (2017). Minimising the machining energy consumption of a machine tool by sequencing the features of a part. *Energy 121*, 292–305.

Inman, R. R. (1999). Empirical evaluation of exponential and independence assumptions in queueing models of manufacturing systems. *Production and Operations Management 8*(4), 409–432.

Jia, Z., L. Zhang, J. Arinez, and G. Xiao (2016). Performance analysis for serial production lines with bernoulli machines and real-time wip-based machine switch-on/off control. *International Journal of Production Research 54*(21), 6285–6301.

Lippman, S. A. (1975). Applying a new device in the optimization of exponential queuing systems. *Operations Research 23*(4), 687–710.

Loffredo, A., N. Frigerio, E. Lanzarone, M. Ghassempouri, and A. Matta (2022). Energy-efficient control of parallel and identical machines: Impact on the overall production system. *Procedia CIRP 105*, 739–744.

Loffredo, A., N. Frigerio, E. Lanzarone, and A. Matta (2021). Energy-efficient control policy for parallel and identical machines with availability constraint. *IEEE Robotics and Automation Letters 6*(3), 5713–5719.

Montgomery, D. C. (2017). *Design and analysis of experiments*. John Wiley & Sons.

Mouzon, G., M. B. Yildirim, and J. Twomey (2007). Operational methods for minimization of energy consumption of manufacturing equipment. *International Journal of production research 45*(18-19), 4247–4271.

Nadar, E., M. Akan, and A. Scheller-Wolf (2016). Experimental results indicating lattice-dependent policies may be optimal for general assemble-to-order systems. *Production and Operations Management 25*(4), 647–661.

Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Renna, P. and S. Materi (2021). A literature review of energy efficiency and sustainability in manufacturing systems. *Applied Sciences 11*(16), 7366.

Sun, Z. and L. Li (2012). Opportunity estimation for real-time energy control of sustainable manufacturing systems. *IEEE Transactions on Automation Science and Engineering 10*(1), 38–44.

Wang, S., X. Wang, J. Yu, S. Ma, and M. Liu (2018). Bi-objective identical parallel machine scheduling to minimize total energy consumption and makespan. *Journal of Cleaner Production 193*, 424–440.

Welch, P. D. (1983). The statistical analysis of simulation results. *The Computer Performance Modeling Handbook 22*, 268–328.

Zhang, Y., Z. Sun, R. Qin, and H. Xiong (2019). Idle duration prediction for manufacturing system using a gaussian mixture model integrated neural network for energy efficiency improvement. *IEEE Transactions on Automation Science and Engineering 18*(1), 47–55.

Zhao, C. and J. Li (2013). Analysis of multiproduct manufacturing systems with homogeneous exponential machines. *IEEE Transactions on Automation Science and Engineering 11*(3), 828–838.