POLITECNICO
MILANO 1863

School of Industrial Engineering
Department of Mechanical Engineering
Doctoral Programme in Mechanical Engineering

Reinforcement Learning for Energy-Efficient Control of Production Systems

Supervisor: **Prof. Andrea Matta**
Tutor: **Prof. Francesco Braghin**

Doctoral Dissertation of:
Alberto Loffredo
Student number: **938443**

2023 - Cycle XXXV

Abstract

Nowadays, energy efficiency has become a critical issue in industry, with rising energy costs, environmental concerns, and stricter regulations. In particular, manufacturing accounts for a considerable portion of global energy consumption, and reducing energy usage in this sector can improve production systems sustainability. In this context, energy-efficient control (EEC) of manufacturing equipment has emerged as a promising solution. EEC aims to minimize the energy consumption while maintaining production targets. However, traditional EEC methods face several barriers and limitations, including the stochasticity of manufacturing systems, the limited control actions available, and the unknown system dynamics.

Recent research has demonstrated the applicability and potential of Reinforcement Learning (RL) to successfully control production systems. RL is a type of Machine Learning that enables agents to learn from their environment by interacting with it, providing an alternative method to handle incomplete or uncertain information. RL algorithms are indeed adaptive, designed to deal with the system dynamics during the learning process and to adjust their strategies accordingly over time. RL adaptability and effectiveness in addressing control problems that entail high levels of stochasticity make it a suitable candidate for effectively applying EEC policies to manufacturing equipment.

The research goal of this thesis is to develop novel RL-based models that overcome the actual EEC barriers and are capable of dealing with the EEC task for one, more or all the machines in a manufacturing line. The system of interest are workstations composed of several machines in parallel, a widely used layout to obtain a balanced production system in terms of workstations workload. Despite their widespread use and high energy consumption, there is a lack of research that focuses on the potential for energy savings for this system type. Additionally, there has been no exploration in literature of the potential for RL to apply EEC in manufacturing.

The proposed and innovative RL-based models apply EEC to single parallel machine workstations and multi-stage production lines with parallel machine workstations. These models are adaptive and general, enabling them to identify suitable EEC policies for various performance indicators while maintaining production constraints. Numerical results confirm model benefits when applied to a real line from the automotive sector. Further experiments confirm effectiveness and generality of the approach. The results of this research will contribute to the development of more efficient and sustainable production systems, enabling manufacturers to reduce their energy consumption and increase their competitiveness.

Keywords: Artificial Intelligence; Reinforcement Learning; Sustainability; Energy-Efficient Control; Manufacturing Systems; Energy Efficiency.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over the past decade, energy-efficiency has emerged as a critical area of research within the field of production systems, alongside traditional areas of focus such as productivity improvement and quality enhancement. The increasing demand for "eco-green" tools is driven by several factors as financial gains, improved reputation, compliance with regulations, and addressing global challenges, which can lead to long-term viability. From a financial point of view, indeed, sustainability initiatives can lead to cost savings, improved efficiency, and reduced risk for companies; for example, implementing energy-efficient practices can reduce energy costs and investing in renewable energy can hedge against future fuel price increases. Furthermore, companies that prioritize sustainability often see an improvement in their brand and customer loyalty. Compliance and legal requirements are also a concern, as many governments and industry groups have implemented regulations and standards for sustainability, and companies that do not comply may face penalties or fines. Lastly, companies that prioritize sustainability are better equipped to navigate the challenges of the future and to adapt to an ever-changing business environment, enhancing their long-term viability.

The goal of this thesis is to contribute to the research area of industrial systems planning, control and optimization by developing new models and methods for the effective energy-efficient control of production systems.

1.1 Background

In recent years, energy-efficiency topic is drastically increasing its relevance in manufacturing industries management. The industrial sector is indeed responsible for almost the 40% of global energy consumption, and this is mostly due to manufacturing (Figure 1.1-a). Among the industrial sector, indeed, energy-intensive manufacturing, which includes fields such as food, chemicals, refining, iron and steel, is the largest contributor, accounting for almost the 52% of total industrial energy consumption. This is followed by not energy-intensive manufacturing (metal-based durables, and other manufacturing) and not-manufacturing activities (e.g. agriculture and mining), with a share of around 35% and 13% respectively (Figure 1.1-b). As a result, the industrial sector also contributes significantly to global CO_2 emissions related to energy consumption, and this impact is expected to further rise in the upcoming years (Figure 1.2).

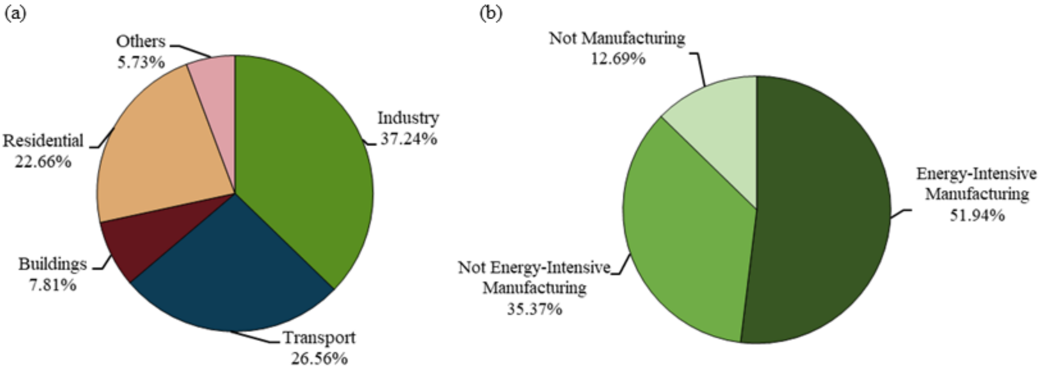


Figure 1.1: Percentage of global energy consumption by sector (a) and detail of how the share of the industrial field is composed (b) [1].

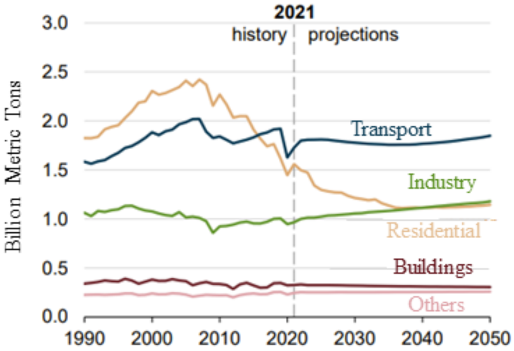


Figure 1.2: CO_2 energy-related emissions by sector [1].

Given this scenario, companies have been increasingly implementing energy-efficient measures to reduce their energy costs, improve their reputation, comply with regulations and increase their long-term viability, also considering the recent major trend of rising electricity prices [1]. However, they face several challenges when implementing these practices. One of these is the high initial investment required for energy-efficient equipment and technologies. Additionally, industrial corporations may face resistance from employees or other stakeholders who are not familiar with or do not support the changes. Furthermore, many companies lack the necessary knowledge and expertise to implement and maintain energy-efficient measures. Lastly, there is a scarcity of standardization and compatibility among energy-efficient technologies which can make it difficult to integrate them into existing systems. Despite these challenges, enterprises are recognizing the importance of sustainability and are taking steps to overcome these obstacles in order to reduce their energy consumption and emissions.

1.2 Energy-Efficiency Measures in Manufacturing

In a manufacturing system, energy-efficiency can be addressed at different hierarchical levels of the automation pyramid [2] (Figure 1.3): (i) global supply chain, (ii) facility, (iii) production line, and (iv) machine tool (“machine" from now on). This work places its focus on the machine and production line levels.

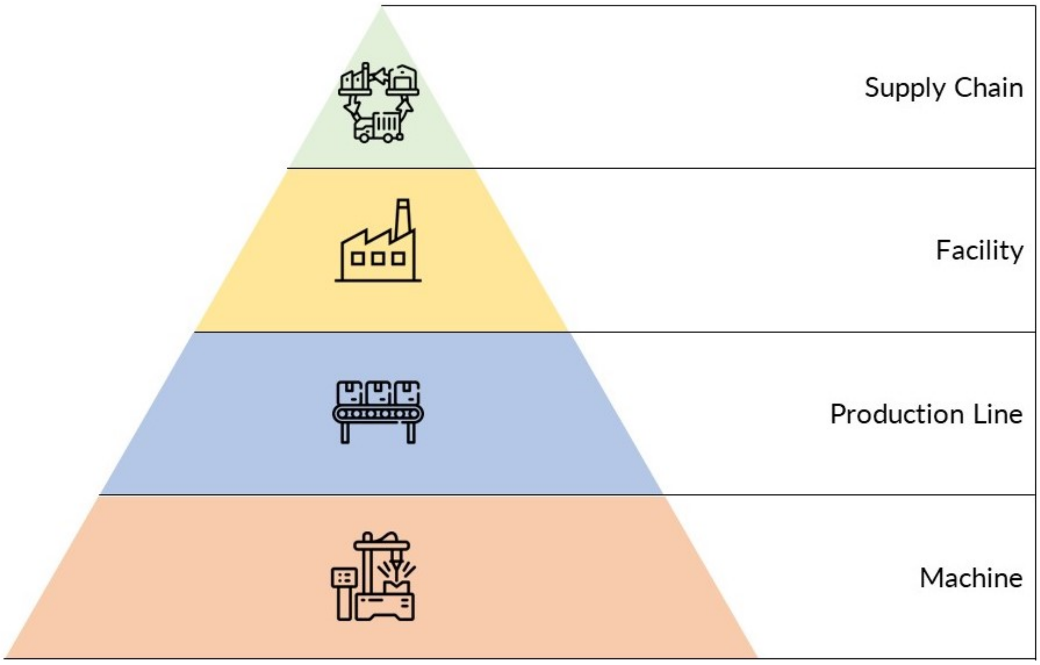


Figure 1.3: Different hierarchical levels of the automation pyramid.

Machines are responsible for approximately 50% of the total electricity consumed in an average factory [3]. Their energy consumption can be seen as the sum of a *Base Load* and a *Load Dependent* portions [4]. The former is independent from the process and is required to maintain the machine in ready-for-process conditions; the latter is requested to operate on parts and to execute the main machine process.

Before a machine is integrated and operating into the production line, two main strategies can be implemented to reduce its environmental impact: the focus on the machine design and the proper calibration of the process parameters. The goal of the first approach is to minimize the power demands of machine components and this can be accomplished by utilizing more efficient technologies such as latest-generation motor drives, advanced material handling methods, precision clamping systems, and optimized hydraulic and pneumatic systems. On the other hand, the second approach is based on identifying proper process parameters (e.g. spindle rotating or cutting speed) allowing to minimize the *Load Dependent* energy; however, this strategy also requires carefully balancing between machine cycle time and power consumption.

Nevertheless, both of the above methods are well-suited for designing new machines for a production line or re-calibrating a production process, but are not as effective for applying on already existing and operational machines. Indeed, when companies have already made significant investments in their existing and in-use machines in the respective production lines, it is important to investigate alternative strategies for reducing the environmental impact of these, while still preserving the investment made for buying and installing them. In this situation, two main approaches can be identified: energy-efficient scheduling (EES) and energy-efficient control (EEC) that deal with the same problem but from distinct levels. EES is based on the production scheduling, i.e. defining a detailed plan to assign “jobs" (production activities) to machines in the shop floor. Most of the time, the production scheduling is defined beforehand, assuming certain knowledge about all production data and dynamics [5]. EES aims at reducing the energy consumption through the production plan: the goal is the minimization of machine non-productive periods by effectively allocating different jobs to machines. Furthermore, job-allocation is also aimed at reducing the shop floor power peak consumption according to electricity prices: working loads are shifted from higher price periods to cheaper ones [5]. A complete EES literature review can be found in [6].

On the other hand, EEC is based on implementing real-time actions during the production process, without prior knowledge of the upcoming jobs to be processed by the machine to reach production targets. EEC, based on the uselessness of machine idle periods, is the focus of this research work. A machine is idle when simultaneously switched on but not operating on parts, i.e. it provokes a significant energy consumption to maintain the ready-for-process conditions (*Base Load* energy) but is not productive. Hence, the goal of EEC is to provide policies for switching off machines as well as its components when idle and switching them on again only when production must resume. A proper EEC policy controls machine states towards the optimum trade-off between system production rate and energy demand.

1.3 Barriers for EEC Implementation

Energy efficiency is a key factor in achieving sustainability, particularly in the manufacturing industry, where the control of production systems is a complex task that involves the coordination of multiple processes and resources. One of the main challenges in production systems control is to enhance energy efficiency while maintaining high throughput and quality. To this end, the EEC approach represents a potential solution but its implementation is not without barriers and limitations:

1. One of the primary challenges in applying EEC regards the high stochasticity inherent to manufacturing systems, which can result in unpredictable outcomes. Manufacturing lines indeed frequently undergo operational changes such as redesigning part of the entire system, recalibrating production parameters and processes, or experiencing degradation processes that result in non-stationary system parameters.

2. The limited set of possible control measures further complicates the application of the EEC approach: with only on/off switching of idle machines as control actions, it becomes challenging to reach high and significant savings.

3. The production system dynamics are often unknown: this requires learning it before being able to optimize the EEC solution. Indeed, the existing EEC methods often assume complete knowledge of system dynamics and parameters, such as machine processing times and failure data. In reality, such assumptions are commonly unrealistic, resulting in limitations in the accuracy, effectiveness, and generality of the related EEC models and results.

To overcome these limitations, alternative methods that can handle incomplete or uncertain information are required. The incorporation of machine learning algorithms and data-driven models for optimizing the EEC is a promising avenue for future research, enabling more accurate, effective, and generalizable EEC models to be developed.

1.4 Research Goal

This thesis investigates the application of *Reinforcement Learning* (RL) algorithms to overcome the limitations of the existing EEC methods. RL is a type of

Machine Learning that enables agents to learn from their environment by interacting with it, providing an alternative method to handle incomplete or uncertain information. Recent research demonstrated the applicability and potential of RL to perform optimal control actions for complex systems. RL algorithms are indeed adaptive: they are designed to adapt to the system dynamics during the learning process and adjust their strategies accordingly over time. Its adaptability and effectiveness in addressing control problems that entail high levels of stochasticity make RL a suitable candidate for effectively applying EEC policies to manufacturing equipment.

The research goal is to develop novel RL-based models that are capable of dealing with the EEC task for one, more or all the machines in a manufacturing line. These models overcome the EEC barriers related to the stochasticity of manufacturing systems, the limited control actions available, and the unknown system dynamics. RL-based models indeed do not rely on complete knowledge of system dynamics and, therefore, are not limited by the assumption of having full knowledge about system dynamics and parameters. RL also ensures high flexibility and ease of use of models and control policies. This leads to great accuracy, effectiveness, and generality of the proposed models.

The results of this research will contribute to the development of more efficient and sustainable production systems, enabling manufacturers to reduce their energy consumption and increase their competitiveness. The focus is placed on the EEC of workstations composed by several machines in parallel; this is a widely used layout to obtain a balanced production system in terms of workstations workload. An example of single parallel machine workstation layout is visible in Figure 1.4 while Figure 1.5 shows a multi-stage production line with parallel machine workstations.

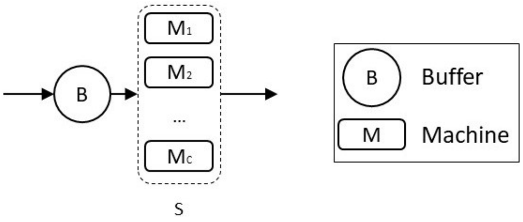


Figure 1.4: Layout of a single parallel machine workstation.

After conducting a thorough literature analysis in Chapter 2, it has become evident that there is a significant gap in the exploration of the EEC approach for parallel machine layouts, both for individual workstations and for the overall production line. Despite their widespread use and high energy consumption, there is a lack of research that focuses on the potential for energy savings for this system type. Additionally, there has been no exploration of the potential

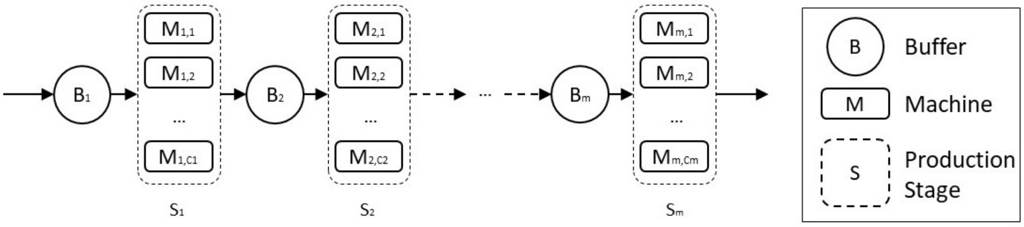


Figure 1.5: Layout of a multi-stage production line with parallel machine workstations.

for RL to apply EEC in manufacturing. Therefore, the primary focus of this thesis is to address these research gaps and investigate the potential for RL to improve energy efficieny of manufacturing equipment.

The proposed and innovative RL-based models are able to apply the switch off/on control simultaneously to one or more machines in parallel in order to reduce the overall energy consumption while not jeopardizing the system productivity. It must be also noted that: if the environment to be controlled (i) satisfies the Markov property, i.e. the probability distribution of future states and the reward function depend only on the present state and not on the past, and (ii) is fully observable, i.e. system dynamics is fully known with steady and defined transition probabilities, the RL task is called a Markov Decision Process (MDP). In other words, MDPs are a mathematical form of the RL problem for which precise theoretical statements can be made [7].

The following sub-sequential objectives to be achieved are identified for this thesis:

1. The formulation of an MDP-based model leading to EEC policies for a single parallel machine workstation.

2. The creation of an MDP-based model for the EEC of multi-stage production lines with parallel machine workstations.

3. The development of an adaptive and general RL framework-based novel model for the EEC of a single parallel machine workstation.

4. The formulation of an adaptive and general RL framework-based novel model to identify suitable EEC policies for multi-stage production lines with parallel machine workstations.

It is worth noting that all models are also applicable when workstations consisting of a single machine are present, as it is a specific sub-case of the parallel-machines category. Furthermore, the models consider production constraints by reducing energy consumption while maintaining target levels for various performance indicators such as system throughput or machine availability.

In summary, the proposed models provide new methods and tools using Reinforcement Learning methodology for reducing the environmental impact of manufacturing processes without compromising production and without relying on complete knowledge of system dynamics.

1.5 Thesis Structure

The thesis is organized in seven chapters as follows:

- **Chapter 2** provides an overview on the state of the research related to EEC and RL for production control applications. The chapter also highlights the main contributions of this work.
- **Chapter 3** offers a detailed explanation on the MDP-based model for the EEC of a single parallel-machines manufacturing workstation. Furthermore, most of the proposed models have been tested in a real industrial case from the automotive sector. The industrial case is presented in Chapter 3 but it has been used as real-world use-case in Chapters 3 , 5, and 6.
- In **Chapter 4** the second novel model introduced by this work is presented: the MDP-based model for the EEC of the entire production line.
- **Chapter 5** details the development of the first adaptive and general RL framework-based model for the EEC of a single workstation made of parallel machines.
- **Chapter 6** outlines a complete description of the last and most advanced novel model presented by this research, i.e. the adaptive and general RL framework-based for the EEC of the overall production system.
- Lastly, **Chapter 7** closes this work with the respective conclusion and further developments given the results illustrated in Chapters 3, 4, 5, and 6.

Chapters 3, 4, 5, and 6 address the EEC problem with increasing complexity of the system to control and/or overcoming the assumption of having complete knowledge of system dynamics. In this sense, Chapter 3 is the starting point: a *simple* system is controlled (single workstation) while relying on the assumption of fully observable system. In Chapter 4, the assumption on known system dynamics is maintained, but the system under control increases in size (whole production line). Chapter 5, instead, prevails the system observability limitation and provides a model able to control the single workstation

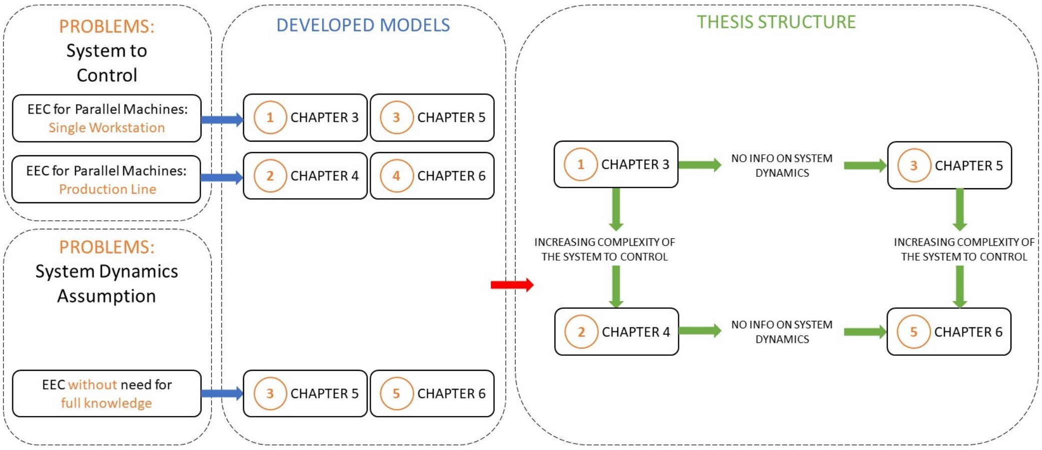

PROBLEMS:
System to Control
EEC for Parallel Machines: Single Workstation
EEC for Parallel Machines: Production Line
PROBLEMS:
System Dynamics Assumption
EEC without need for full knowledge
DEVELOPED MODELS
1 CHAPTER 3
3 CHAPTER 5
2 CHAPTER 4
4 CHAPTER 6
3 CHAPTER 5
5 CHAPTER 6
THESIS STRUCTURE
1 CHAPTER 3
NO INFO ON SYSTEM DYNAMICS
3 CHAPTER 5
INCREASING COMPLEXITY OF THE SYSTEM TO CONTROL
INCREASING COMPLEXITY OF THE SYSTEM TO CONTROL
2 CHAPTER 4
NO INFO ON SYSTEM DYNAMICS
5 CHAPTER 6

Figure 1.6: Thesis work framework and proposed models characteristics.

but without relying on complete knowledge of system dynamics. Lastly, Chapter 6 closes the circle: a complex system (production line) is controlled even when there is no information on system dynamics and parameters. Complete overview on the presented work framework is in Figure 1.6.

Chapter 2

Literature Review

In recent years, there has been a growing interest in academia and industry for developing energy-efficient control (EEC) strategies for production systems. On the other hand, Reinforcement learning (RL) has emerged as a promising technique for solving complex control problems in various domains, including production control. The scope of this Chapter is to present a comprehensive literature review on EEC and RL for production control, focusing on their applications, challenges, and potential benefits. The review also highlights the main issues and limitations of existing EEC, such as high computational complexity, the need for large amounts of data, and the absence of models for the control of relevant manufacturing system layouts. Lastly, the Chapter discusses the contribution of this thesis to the field of EEC by integrating the RL framework in the control tasks.

The Chapter structure is: Section 2.1 provides an overview of existing literature regarding the EEC approach while Section 2.2 presents the key features of RL along the recent literature for its use in production control. Lastly, Section 2.3 closes the literature review illustrating how the thesis contribution fits in the current state of the art.

2.1 Energy-Efficient Control in Production Systems

Currently, the worldwide energy consumption is greatly affected by manufacturing systems. One of the main issues is the idle time of machines. A machine is said to be idle when it is simultaneously switched on but not operating on parts, i.e. it provokes a significant energy consumption to maintain the ready-for-process conditions but is not productive. Therefore, reducing idle time is important. In managing one or more manufacturing workstations, control policies are used to dictate actions under specific system conditions to achieve target goals related to system performance indicators, such as system throughput and system availability level. The goal of EEC is to provide policies for switching off machines when idle and switching them on again only when production must resume. A proper EEC policy controls machine states towards the optimum trade-off between system production rate and energy demand. The EEC takes advantage of the switch on/off approach and is in contrast to the so-called *Always-On* (*AOn*) policy, where the machine is always switched on during idle periods, consuming unnecessary energy while not producing parts. However, the *AOn* policy is the most optimal policy in terms of production readiness, as it avoids the switching-off of resources and results in the highest production rate but leads to a high level of energy consumption. Despite its drawbacks, *AOn* remains a common practice in manufacturing.

Research in the field of EEC for manufacturing systems is expanding in recent years: a complete and recent literature review on this topic can be found in [8]. The issue of EEC is approached from two separate perspectives in research: (i) controlling a single workstation (see Section 2.1.1), and (ii) controlling the overall production system (see Section 2.1.2). The total number of publications in the EEC field along the years is visible in Figure 2.1, with a detail overview of the controlled system layout in Figure 2.2 ([8, 9]). The main works in this field are explained with more detail in Sections 2.1.1 and 2.1.2.

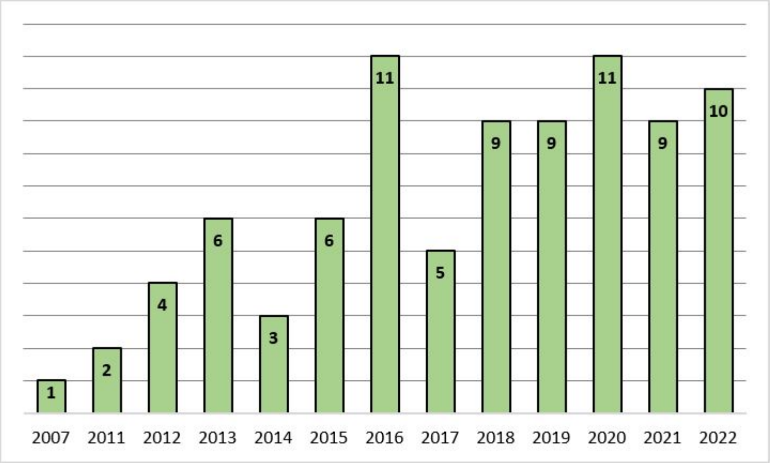


Figure 2.1: Number of EEC papers published for each year.

The existing EEC literature always focuses on production systems modeled as a series of single workstations [10], where each of them is composed of a single buffer followed by a single machine (Figure 2.3). EEC complexity increases when policies are applied at multiple workstations simultaneously. As a result, the literature on the EEC of the overall production line is more

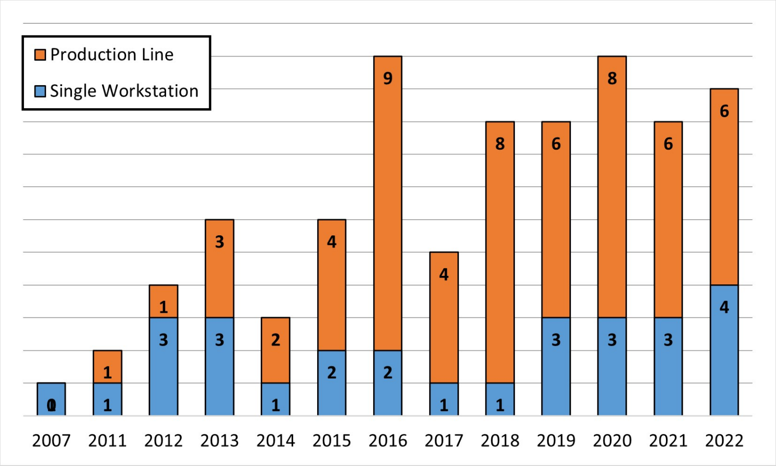


Figure 2.2: Detail of the EEC papers published for each year.

recent and builds upon the research on single workstations EEC.

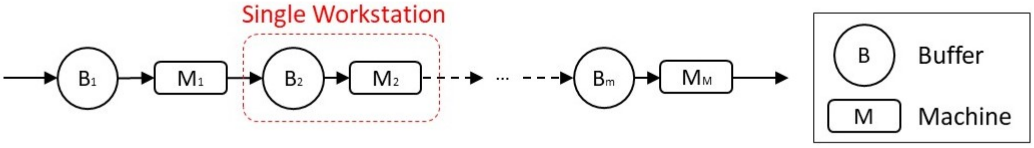


Figure 2.3: Production Line Layout.

2.1.1 Energy-Efficient Control of Single Workstations

The first level of analysis considers EEC for the simplest possible manufacturing system configuration, i.e. the single-buffer-single-machine layout. The first work on this theme can be found in [11], with several switch off dispatching policies to reduce the energy consumption of manufacturing equipment that is not fully utilized, i.e. non-bottleneck machines, in a production system. The authors utilized optimization techniques to forecast the upcoming arrival of parts, taking into account the non-stationary nature of the process. Subsequently, in [12], the authors modeled the single machine as an M/G/1 queue and studied the optimal switch on policy when: (a) the machine is switched off, (b) upstream buffer is initially empty, and (c) the service must be resumed when a certain number of parts enters in the buffer; however, this policy is valid only as far as the M/G/1 model stands. Similarly, in [13], it is possible to find an analysis on the trade-off between energy consumption and waiting time of parts in the buffer when the machine is again modeled as an M/G/1 queue. The machine is shutting down when the queue becomes empty and it is switched on again only when a certain number of parts N is in the buffer. Several closed-form formulas are derived to demonstrate the trade-offs between energy consumption and mean productivity by changing the switch-on threshold. It is shown that larger N reduces the energy consumption, but there may exist N > 1 that minimizes the mean productivity drop. Again, all their results are valid exclusively for the M/G/1 model.

A more general approach has been proposed in [14], where the authors developed an MDP model for a single machine and single product-type manufacturing system, with the objective of minimizing the sum of production cost and energy consumption cost. Specific cases have been employed to analyze the structural characters of the optimal control policy, and its general structure and main parameters are obtained. However, this work was limited by the assumption of having only two possible states for the machine (switched on and switched off) and without taking into account the start-up time of the resource. The absence of startup time is not aligned with most manufacturing equipment where a startup transitory is needed to resume the service and therefore causing a production loss. This limitation has been overcame in [15], where an MDP model for the EEC of a single machine was developed, including startup state of the machine and considering a transitory time to resume the machine service. This model also considered part admission control to the buffer to avoid part discard due to full queue. However, the results were limited to the consideration of only exponentially distributed processing, startup, and interarrival times.

Regarding analytical approaches, Frigerio and Matta studied analytically an EEC policy for a single machine under the assumption of stochastic arrivals, constant startup time and no information from the buffer in front of the machine [16]. Subsequently, they extended their analysis modelling explicitly the start-up time as dependent on the time period the machine stays in a standby state [17, 18] and adding buffer level information in their model [19]. Moreover, the same authors developed a model to provide EEC policies considering the switch on/off of only some components inside the machine, to have multiple sleeping states of the single machine [18]. Later, they also developed an adaptive EEC policy for the single machine based on machine learning techniques to predict idle periods [20] and that is able to adapt the control for varying system parameters eventually self-calibrating in real-time the control policy [21].

Also, the authors of [22] contributed with the implementation of a fuzzy controller used to switch on/off the single machine with stochastic arrivals but this method has been successfully tested only on specific real systems and not validated for general cases.

All the cited works, except for [21], were limited by the use of models requiring full information about system dynamics. This assumption is often unrealistic in real-world applications and can result in limitations in the accuracy, effectiveness, and generality of the related EEC models and results. Another strong limitation of existing EEC literature for the single workstation is that, despite the different approaches, the positive outcomes, and substantial efforts, there are no models taking into account the EEC of a single workstation composed of a single buffer followed by multiple parallel machines. This configuration is widely used in industry to have balanced manufacturing systems in terms of workstations workload. Also, this workstation architecture operates differently, as parts are stored in a common upstream buffer and can be processed by any of the parallel machines. Therefore, the system behavior is not as straightforward as that of a single machine case, and the EEC perspective differs significantly. During periods of low production with few parts in the buffer, more machines in the workstation may be switched off off to conserve energy. Consequently, the EEC focuses on determining the optimal number of machines to switch on or off at any given time to improve energy efficiency in such scenarios.

Using queueing theory, the parallel machine system under investigation can be modeled as a multi-server queue with a common upstream buffer of finite capacity, for instance as an M/M/c/K queue where the parallel machines are represented by the multiple servers. Queueing theory literature for controlling this type of system is wide, even though the none of them is related to manufacturing field. Control policies for M/M/c systems are indeed widely developed for server farms and data-centers. Gandhi et al. [23], [24] introduced a *staggered-setup* policy for an M/M/c queue with exponential startup time. This policy imposes a number of switched on servers equal to the number of items waiting to be processed and servers to be switched off as soon as they become idle. The same model was analyzed by Mitrani [25], where a certain number of servers are switched on when the number of elements in the system exceeds a certain threshold; then, they are switched off when the number of parts becomes lower than another threshold. Another approach was used by Xu and Tian [26] with (e, d)-control policies for M/M/c systems: when d servers are idle, e servers are switched off. Lastly, an extensive analysis for M/M/c queues in server farms and data-centers was performed by Maccio and Down [27], [28], [29], [30]. In [27] and [29] the authors extracted structural properties pertaining to the optimal policy to be applied for controlling M/M/c queues. Then, in [28] and [29], they analyzed system performances when applying two specific policies satisfying those structural properties. In their most recent work on this topic [30], policies for an M/M/c configuration are studied under the asymptotic regime where the number of servers approaches infinity while the load per server remains constant. However, all the mentioned works on server farms and data-centers are characterized by assumptions not properly aligned with manufacturing field, as part processing and startup procedure that can be interrupted and finite buffer capacities. Thus, all the extracted properties and policies might be not suitable for manufacturing systems.

2.1.2 Energy-Efficient Control of Production Lines

The second level of analysis considers the EEC for the entire production line. The system under control is herein modeled as a series of single workstations with a single-buffer-single-machine architecture. An important milestone in this field was reported in [31], where the performance of the overall production system when single workstations are subject to local EEC is analyzed, and it has been proved that the local control of machines without taking into account interactions among workstations is a sub-optimal control. This has been confirmed by several numerical cases in [32]. This paved the way for achieving optimal solution with the joint control of more (or all) workstations in a system.

A first work in this field was developed in [33], investigating energy consumption reduction in Bernoulli serial lines with buffer stripping, i.e., with the buffers that must be depleted at the end of each shift. This buffer depletion activity is widely used in production systems with perishable products to avoid quality deterioration. Their results were promising but limited to the Bernoulli reliability model and to having machines not requiring a startup time: these assumptions are not aligned with most manufacturing equipment.

Another approach in this research stream 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 [34]. During these OWs it is possible to implement EEC on the machine, reduce energy consumption and not jeopardize the production rate. In [35, 36] a method is presented to estimate the temporal opportunity windows and assess when to implement EEC actions on machines in a line. The OW method was also used in [34], indicating different real-time switching on/off policies for machines to be applied during OW caused by random failures in a production system. The OW approach was also applied by Brundage et al. in an integrated production line with HVAC system [37]. The same authors also developed the concept of energy profit bottleneck, i.e. a machine that, when its switched off time is reduced, leads to the highest overall profit increase [38]. An energy OW control methodology to identify when to switch on the energy profit bottleneck is used in order to reduce the overall energy consumption while maintaining throughput. In [39], a method to predict OW based on both energy saving and preventive maintenance requirements was developed. The same authors also developed a data-driven model to predict energy saving opportunities and their impact on production [40]. However, in all the works dealing with the OW concept, the startup time for machines is never considered. An exception in this sense can be found in [41], where the OW concept has been applied to *sensible* manufacturing systems, i.e. where resources are connected and equipped with diverse sensors in physical space and also interconnected in cyberspace where they can sense the information of each other and actively decide themselves operation modes considering some specified objectives. However, despite considering machines startup time, this work was limited by the requirement of having cyber and physical interconnection among all the line machines and absence of human operators in the line, that is rare in nowadays manufacturing systems.

Outside of the OW-field, a significant contribution was provided by [42]: the authors developed a novel model leading to EEC policies for the entire production line using work-in-process information; 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 [43], 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 [44] 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.

Different methodologies to achieve the EEC of a production line can be found in: (i) [45] with the use of a dynamic adaptive fuzzy reasoning petri net, (ii) [46] using an event-based analysis of production dynamics and an analytical approach to quantitatively predict the system level production loss resulted from an energy saving control event, and (iii) [47] with an adaptive control strategy to switch off/on the machines of a production line under pull control policy.

A detailed classification of all the EEC cited references for single workstations and production lines can be found in Table 2.1. Nevertheless, two main limitations arise for the existing EEC works for the overall production system, and they have been already outlined also for the EEC of single workstations.

The first limitation is that, in all the mentioned works, a production line where one or more parallel machine workstations are present is never considered as the system where to apply EEC. This is considered a main limitation since multi-stage production lines with parallel machine workstations are very common in manufacturing. As particular example, machining is one of the key processes in the automotive industry but not one of the fastest [48] and therefore machining operations are often performed with parallel machine

Table 2.1: Detailed classification of the mentioned EEC references. Legend: Ref. = Reference; WS = Workstation; Prod. = Production; En. Cons. = Energy Consumption; NC = Not Considered; DP = Dynamic Programming; Stoch. = Stochastic; Deter. = Deterministic; TH = Throughput.

			Machine States						Processes				Optimization/Control Problem		
Ref.	EEC of	Buffer	Standby	Idle	Startup	Busy	Closedown	Failed	Proc. Time	Arrival Time	Startup Time	Optim. Method	Obj. Function	Constraint	Sys. Dynamics
[11]	Single WS	NC	X	X	X	X	X		Deter.	Stoch.	NC	Analytical	En. Cons	No	Fully Known
[13]	Single WS	Infinite	X	X	X	X	X		Stoch.	Stoch.	NC	Analytical	En. Cons	TH	Fully Known
[12]	Single WS	Infinite	X	X	X	X	X		Stoch.	Stoch.	Stoch.	Analytical	En. Cons	No	Fully Known
[14]	Single WS	Infinite	X			X			Stoch.	Stoch.	NC	DP	En. + Prod. Cost	No	Fully Known
[15]	Single WS	Finite	X	X	X	X			Stoch.	Stoch.	Stoch.	DP	En. Cons	No	Fully Known
[16]	Single WS	NC	X	X	X	X			Stoch.	Stoch.	Deter.	Analytical	En. Cons	No	Fully Known
[17]	Single WS	NC	X	X	X	X			Stoch.	Stoch.	Stoch.	Analytical	En. Cons	No	Fully Known
[19]	Single WS	Finite	X	X	X	X			Stoch.	Stoch.	Stoch.	Analytical	En. Cons	No	Fully Known
[20]	Single WS	NC	X	X	X	X			Stoch.	Stoch.	Stoch.	Analytical	En. Cons	TH	Fully Known
[18]	Single WS	Finite	X	X	X	X			Stoch.	Stoch.	Stoch.	Analytical	En. Cons	TH	Fully Known
[21]	Single WS	Finite	X	X	X	X			Stoch.	Stoch.	Stoch.	Analytical	En. Cons	TH	Not-Fully Known
[22]	Single WS	Finite	X	X	X	X	X	X	Deter.	Stoch.	Deter.	Simulation	En. Cons	No	Fully Known
[31]	Prod. Line	Finite	X	X	X	X			Stoch.	Stoch.	Deter.	Analytical	En. Cons.	TH	Fully Known
[32]	Prod. Line	Finite	X	X	X	X			Stoch.	Stoch.	Deter.	Simulation	En. Cons.	TH	Fully Known
[33]	Prod. Line	Finite	X			X		X	Stoch.	Stoch.	NC	Analytical	En. Cons.	TH	Fully Known
[34]	Prod. Line	Finite	X	X		X		X	Stoch.	Stoch.	NC	Analytical	En. Cons.	No	Fully Known
[35]	Prod. Line	Finite	X			X		X	Stoch.	Stoch.	NC	Analytical	En. Cons.	No	Fully Known
[36]	Prod. Line	Finite	X			X		X	Stoch.	Stoch.	NC	Analytical	En. Cons.	No	Fully Known
[37]	Prod. Line	Finite	X	X		X		X	Deter.	Stoch.	NC	Analytical	En. Cons.	No	Fully Known
[38]	Prod. Line	Finite	X	X		X		X	Deter.	Stoch.	NC	Analytical	En. Bottleneck	No	Fully Known
[39]	Prod. Line	Finite	X	X		X		X	Deter.	Deter.	NC	Analytical	En. Cons.	No	Fully Known
[40]	Prod. Line	Finite	X	X		X		X	Deter.	Deter.	NC	Analytical	En. + Maint. Cost	No	Fully Known
[41]	Prod. Line	Finite	X	X	X	X	X	X	Stoch.	Stoch.	Stoch.	Analytical	En. Cons.	No	Fully Known
[42]	Prod. Line	Finite	X	X	X	X	X	X	Deter.	Stoch.	Deter.	Analytical	En. Cons.	No	Fully Known
[43]	Prod. Line	Finite	X	X	X	X			Deter.	Deter.	Deter.	Analytical	En. Cons.	No	Fully Known
[44]	Prod. Line	Finite	X	X	X	X			Deter.	Stoch.	Deter.	Analytical	En. Cons.	TH	Fully Known
[45]	Prod. Line	Finite	X	X		X		X	Deter.	Deter.	NC	Simulation	En. Cons.	No	Fully Known
[46]	Prod. Line	Finite	X	X		X		X	Deter.	Stoch.	NC	Analytical	En. + Prod. Cost	No	Fully Known
[47]	Prod. Line	Finite	X	X	X	X			Stoch.	Stoch.	Stoch.	Simulation	En. Cons.	No	Fully Known

workstations managed with *AOn* policy. As a consequence, the machining production stages might not be always used at maximum capacity, and some resources could be switched off for energy saving without undermining the overall system production rate. Hence, 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. This can be achieved with EEC. Therefore, being able to reduce the environmental impact of systems with this layout can strongly improve industrial processes' sustainability.

The second main limitation of the existing literature for the EEC of production lines is that all models rely on complete knowledge of system dynamics and parameters, such as machine processing times and failure data and they struggle to offer effective solutions with incomplete information.

2.2 Reinforcement Learning and its Use for Production Control

Recent research in the area of Machine Learning demonstrated the applicability and potential of RL to perform optimal control actions for complex systems, as in the EEC task for one or more machines in a manufacturing line. RL is one of the three categories of machine learning algorithms along with supervised and unsupervised learning [49]. It is a powerful tool for solving complex and dynamic decision-making and optimal control problems [50] in a wide range of complex and dynamic systems, from robotics and control systems to game playing and decision-making under uncertainty. Furthermore, RL is seen as a valuable tool in the development of self-optimizing key techniques for many aspects of Industry 4.0, e.g., predictive maintenance, real-time diagnostics, management and control of manufacturing activities and processes [51, 52].

An overview on the key features of RL is presented in Section 2.2.1 while the main literature related to RL applications for production control is in Section 2.2.2.

2.2.1 Overview of Reinforcement Learning Framework

RL algorithms are characterized by two main elements: agent and environment, where the agent is the active component and is subject to a learning process through direct interaction with the environment. RL, indeed, enables the agent to learn how to behave in an environment by performing certain actions and observing the rewards or penalties associated with those actions. The agent continuously interacts with its environment to optimize its behavior and achieve a specific goal, making RL a useful technique for solving control optimization problems. This involves recognizing the best action in each state to optimize an objective function, such as the total discounted reward over a given time horizon. A basic overview of the RL framework is explained in the following and is extracted from [7]. Everything begins with the agent observing the environment state $s_t \in \mathbb{S}$, and selecting an action $a_t \in \mathbb{A}$. $\mathbb{S}$ represents the state space, i.e. the set of possible agent observations of the environment: it must provide information on the current production state so that the agent may choose actions optimally solving the control problem; on the other hand, $\mathbb{A}$ is the action space: the set of possible actions the agent can implement. After a_t is performed, the environment responds with the resulting state s_{t+1} while the agent is rewarded with a reward r_{t+1}. Afterwards, the next iteration can start (complete overview in Figure 2.4). The agent's goal is to maximize the long-term, cumulative reward by optimizing the action-selection policy, i.e. by learning an optimal control policy to apply to the environment (in the EEC framework, this indicates the learning of an optimal EEC policy).

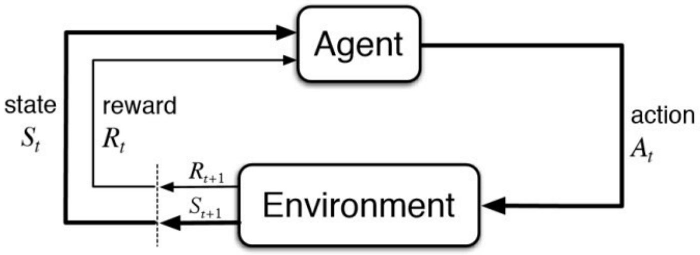


Figure 2.4: Overview of RL-Framework [7].

Two main classes of RL algorithms can be identified: based on *Policy Gradient* [53] methods and on *Q-Learning* [54] approach.

In policy gradient methods, the goal is directly to learn a policy that maps states to actions. This approach directly updates the policy to implement in order to maximize the expected reward. On the other hand, Q-learning learns the policy *indirectly*. The goal is indeed to learn the optimal action-value function (so-called Q-function): it estimates the expected cumulative reward of taking a specific action in a specific state and following an optimal policy thereafter. The optimal policy is then derived from the learned Q-function by selecting the action that maximizes the expected cumulative reward in each state. In this sense, the optimal policy is derived indirectly from the Q-function, rather than being learned directly as in policy gradient methods. Both policy gradient methods and Q-learning use a Neural Network (NN) as a function approximator to manage the agent learning procedure but with significant differences explained thereafter.

It is also recalled that a Markov Decision Process (MDP) is a mathematical form of the RL problem for which precise theoretical statements can be made [7]. With an MDP, the associated control policy remains steady over time: it is assumed that the agent has a complete knowledge of the environment's dynamics, leading to a high confidence level in the learning process and consequent control policy identification. On the other hand, when the Markov property is not satisfied and the environment is not fully observable since system dynamics varies over time, the RL associated control policy is said to be adaptive: it self-adjusts over time according to the varying system dynamics to be understood during the learning process.

Policy Gradient Methods

With policy gradient methods, the NN is used to directly optimize the action-selection, i.e. the policy to be applied. The NN can learn to predict how valuable are the possible future actions for the RL target via reward estimation (complete overview in Figure 2.5). The agent observes the current state of the environment s_t and this is the NN input. The output of the NN is a probability distribution over the possible actions that the agent can take from that state. Each action is associated with a probability value: actions with higher probability are more likely to lead to a higher expected reward and the sum of the probabilities is 1. The agent selects the action a_t that, starting from s_t, has the highest probability to lead to the highest estimated associated reward. Once a_t is executed, the agent is rewarded with the real reward r_{t+1} associated to a_t. Also, after a_t is executed, the environment provides the new state s_{t+1} and the next iteration can start. The NN utilizes a function consisting of coefficients, or weights to approximate the mathematical relationship between inputs (states) and output (the probability for each possible action to maximize the reward). The NN learning procedure exploits the real reward received to improve the network weights, by iteratively adjusting them along gradients that reduce the error between the estimated and the real reward [55]. The feedback provided to the agent is used to improve its estimation of the probability distribution. In each interaction the agent updates the optimal action-selection, i.e. the optimal policy to implement, based on the computed gradient values: this leads to the “policy gradient" name of the method [53]. After multiple interactions with the environment and consequent multiple policy updates, the agent's estimation of the future rewards slowly improves, which enables the selection of more optimal actions in future interactions, i.e. the selection of the optimal policy to apply.

One of the main challenges in policy gradient methods is the choice of step size or learning rate, which determines how much the policy is updated at each iteration. Policy updates indeed depend on the estimated gradients of the expected rewards. If the step size is too large, the estimated gradients may be noisy, and the updates may overshoot the optimal policy parameters. Conversely, if the step size is too small, the updates may not be significant enough to improve the policy effectively. To overcome these issues, two common types of effective RL agents are used: the TRPO agent [56] and the PPO agent [57].

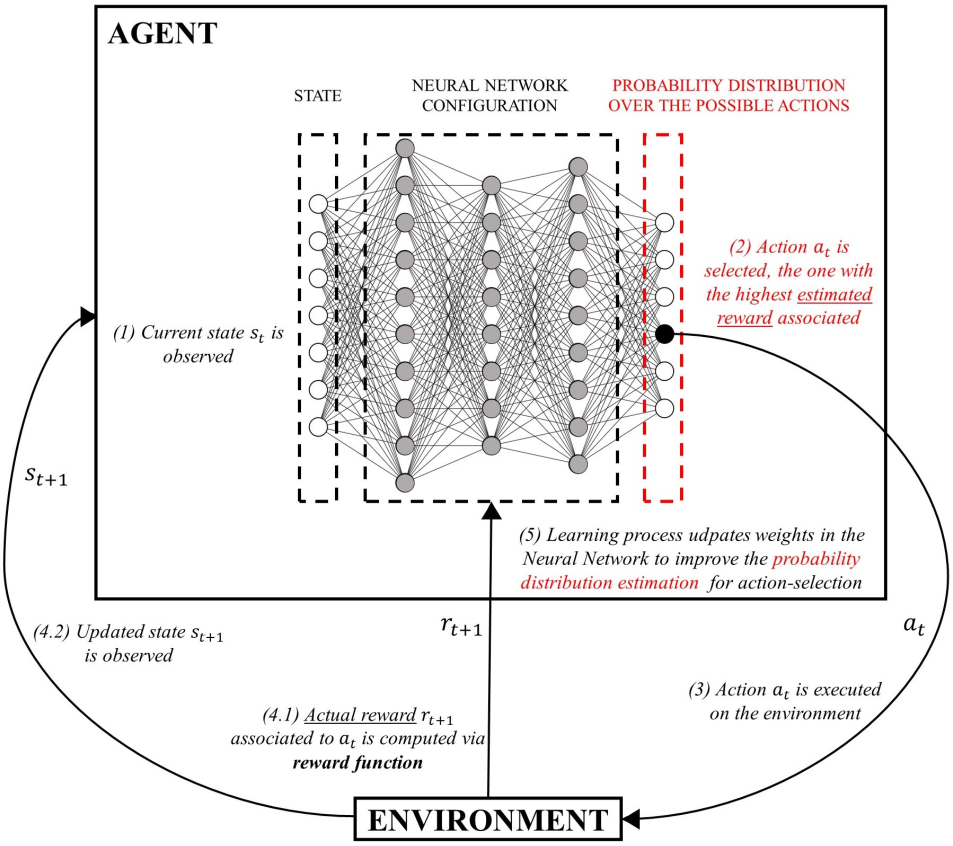

AGENT
STATE
NEURAL NETWORK CONFIGURATION
PROBABILITY DISTRIBUTION OVER THE POSSIBLE ACTIONS
(2) Action a_t is selected, the one with the highest estimated reward associated
(1) Current state s_t is observed
s_{t+1}
(5) Learning process udpates weights in the Neural Network to improve the probability distribution estimation for action-selection
(4.2) Updated state s_{t+1} is observed
r_{t+1}
a_t
(3) Action a_t is executed on the environment
(4.1) Actual reward r_{t+1} associated to a_t is computed via reward function
ENVIRONMENT

Figure 2.5: Detailed scheme of the RL-Framework for Policy gradient methods.

The TRPO agent is based on the *Trust Region Policy Optimization* algorithm. It addresses the sensitivity of policy gradient methods to the choice of step size by constraining the policy updates to a trust region, which ensures that the policy changes are not too large and that the performance of the new policy is not significantly worse than the old policy [56]. The trust region is a region around the current policy that contains policies that are sufficiently close in terms of expected rewards. The policy updates are then performed by finding the policy that maximizes the expected rewards within the trust region. TRPO can prevent the policy updates from being too large and unstable but has some limitations, such as computational complexity and difficulty in handling high-dimensional action space [56].

The PPO agent exploits the *Proximal Policy Optimization* method and has some of the TRPO-benefits, but it is much simpler to implement and more general [57]. With PPO, the policy updates are made by computing the gradients of a surrogate objective function, which approximates the expected rewards. The surrogate objective function is constructed to ensure that the policy updates are not too large, which makes the learning more stable and efficient [57]. While TRPO is still a powerful algorithm, PPO has been shown to achieve better performance and sample efficiency in many benchmark tasks. However, the choice between TRPO and PPO ultimately depends on the specific problem and requirements of the application [57].

Q-Learning Methods

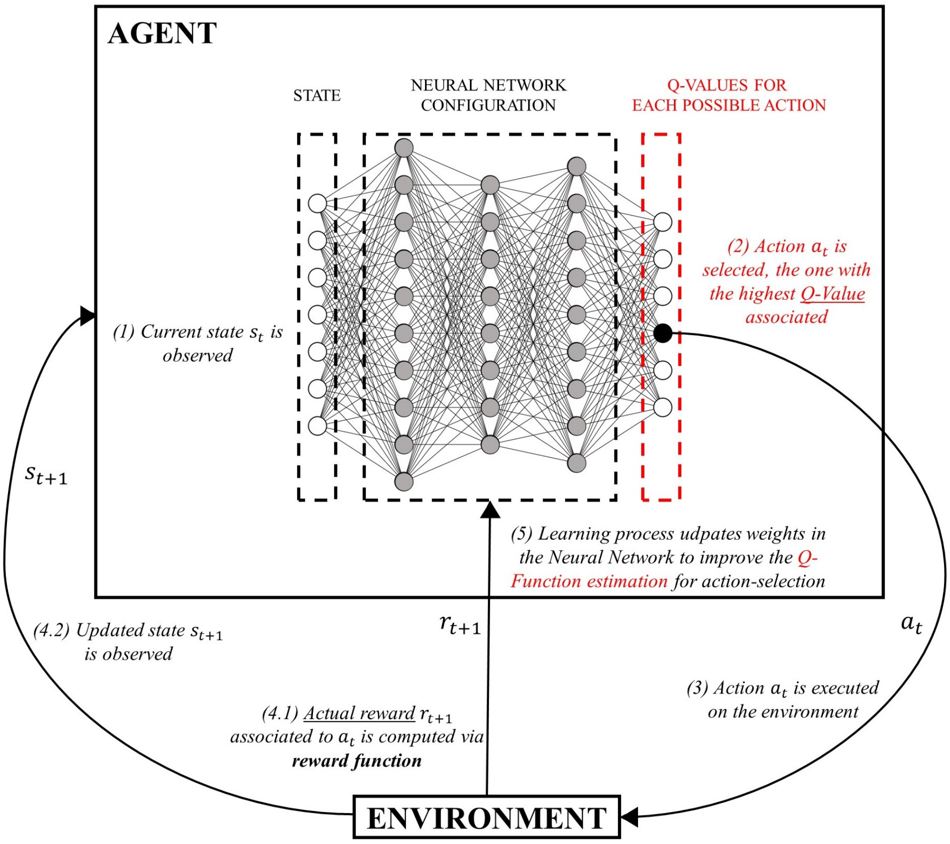

AGENT
STATE
NEURAL NETWORK CONFIGURATION
Q-VALUES FOR EACH POSSIBLE ACTION
(1) Current state s_t is observed
(2) Action a_t is selected, the one with the highest Q-Value associated
(5) Learning process udpates weights in the Neural Network to improve the Q-Function estimation for action-selection
s_{t+1}
r_{t+1}
a_t
(4.2) Updated state s_{t+1} is observed
(4.1) Actual reward r_{t+1} associated to a_t is computed via reward function
(3) Action a_t is executed on the environment
ENVIRONMENT

Figure 2.6: Detailed scheme of the RL-Framework for Q-learning (DQN) methods.

The most used agent from Q-learning methods is the DQN [58] that exploits a *Deep Q-network* approach. In a DQN agent, the role of the NN is to approximate the optimal Q-function that maps states and actions to expected cumulative rewards (complete overview in Figure 2.6). The NN input is again the current state of the environment s_t observed by the agent. The NN output is, instead, a Q-value for each possible action that can be taken in that state. The Q-values represent a collection of estimates of the Q-function. The agent selects the action a_t that, starting from s_t, has the highest associated Q-value. Then, a_t is executed, the agent is rewarded with the real reward r_{t+1}, and the new state s_{t+1} is observed. The NN is used to estimate the Q-function, therefore its weights approximate the mathematical relationship between states (input) and Q-values (output). Being based on a value-function, Q-learning methods and therefore also DQN are also called “value-based" methods [7].

The NN learning procedure exploits the real reward received to improve the network weights and the Q-function estimates. After multiple interactions with the environment, the agent's estimation of Q-values slowly improves, until the agent learns the optimal Q-function. Subsequently, the learned Q-function can be utilized to select the action that maximizes the expected cumulative reward in each state, ultimately obtaining the optimal policy. DQN is effective because it combines the power of RL with the ability of deep neural networks to approximate complex functions. It can handle high-dimensional input spaces and can generalize well to unseen states.

Overview on the Methods

The main characteristics of Q-learning and Policy Gradient methods are reported in Tables 2.3 and 2.4. The choice between the two ultimately depends on the specific task and the trade-offs between convergence speed, stability, and performance [7]. Q-Learning is generally considered more suitable for problems with discrete action spaces and a small number of states, while Policy Gradient methods are better suited for continuous action spaces and problems with large state spaces. Q-Learning is also generally more sample-efficient, meaning it requires fewer samples to converge to an optimal policy, while Policy Gradient methods are generally more computationally efficient, as they do not require the computation of a value function. However, it is important to note that both methods have their own advantages and disadvantages, and the choice between them depends on the specific problem and the desired performance criteria. Ultimately, it is recommended to try out both methods on a given problem and compare their performance to determine which one is more suitable for the specific application [7].

Table 2.3: Main characteristics of Policy Gradient methods.

Method	**Policy Gradient**
Goal	Maximize the expected reward
What it learns	The policy to implement
How to extract the Policy	It is directly learned
NN input	Current environment state
NN output	A probability for each action: actions with higher probabilities are more likely to result in a higher expected reward
NN update method	Gradient-based
NN update rule	Follows the gradient of the expected reward
Action Space	Can handle continuous action spaces
Sample Efficiency	Less sample efficient: it requires more observations
Convergence properties	(1) Lower probability to reach global optimum (2) It can converge to a suboptimum (might be faster)

Table 2.4: Main characteristics of Q-learning methods.

Method	**Q-learning**
Goal	Maximize the expected reward
What it learns	The optimal Q-function
How to extract the Policy	From the Q-function: selecting the action that maximizes it.
NN input	Current environment state
NN output	A Q-value for each action: actions with higher Q-values are more likely to result in a higher expected reward
NN update method	Value-based
NN update rule	Follows the difference between real and expected Q-values that are extracted from real reward
Action Space	Can handle only discrete action spaces
Sample Efficiency	More sample efficient: it requires less observations
Convergence properties	Higher probability to reach global optimum (might be slower)

2.2.2 Reinforcement Learning for Production Control

Literature offers plenty of successful RL applications in the most varied fields. Examples can be found in the optimal allocation of computer resources [59], techniques for collision-avoidance in dense airspace [60], and, in great number, in robotics (a literature survey for RL in robotics is provided by [61]). However, RL still remains under-exploited in the industrial area, especially with respect to other machine learning techniques [62].

Recently, [63] developed an extensive literature review of RL applied in production systems. In [63], the authors identified seven main categories of production disciplines where RL has been applied. In particular, three of these categories are related to production planning and control models [64] for operational decision-making: production scheduling, production dispatching, and plant-internal logistic. Production control is the application of production planning and control techniques [65] in the domain of operational decisions.

Production scheduling refers to the detailed planning of the production process in order to optimize system performance over time [66]. This concerns the management of workstations usage and products allocation, with a focus on long-term planning rather than individual product path control. The overarching goal is to schedule the production of parts over time, taking into account a range of production system factors such as dynamics, material flow, and supply chain considerations. Optimal scheduling requires a comprehensive understanding of these factors. The scheduling of production processes can be highly complex due to uncertainties surrounding customized products, shut-downs, and other factors. However, 89% of RL implementations in benchmarked studies were found to improve scheduling performance, resulting in lower total tardiness, higher profits, and other problem-specific objectives [63]. For instance, in [67] it was proposed a DQN approach that utilized local information to schedule job shops in semiconductor manufacturing, resulting in superior performance and reduced makespans and flow times. Similarly, [68] and [69] adopted self-learning approaches to optimize job-shop scheduling, achieving local and global production objectives and adapting to prevent rescheduling costs. Other studies have demonstrated the benefits of deep RL-based rescheduling, such as [70], that minimized completion times for randomly incoming orders, and [71], who achieved faster and more efficient rescheduling compared to heuristics. RL has also shown promise in other industries, such as semiconductor manufacturing [72], batch processing scheduling [73], chemical scheduling to deal with fluctuating prices, shifting demands, and stoppages [74], and paint job scheduling to minimize costs of color changeovers within the automotive industry [75].

Production dispatching deals with assigning products to machines within the factory according to where they can be processed [76]. The problem is a combination of routing and sequencing, and due to its complexity it is approached by priority rules, i.e. heuristics [77]. Hence, dispatching deals with item-routing depending on where the part can be operated. Production dispatching must perform control actions at a higher level of detail than production scheduling since it is referred to the path-control of each single product within the plant. At the same time, it is based on a lower level of information regarding the system, i.e. ignoring supply chain, and addresses the control problem on a shorter time-horizon [78]. Dispatching is also strongly correlated with customization: individual product configurations must be taken under consideration along with several technical and logistic constraints [76]. According to [79], in managing production control related to transportation as dispatching, a control system that is both flexible and stable is required, which can handle unexpected changes and disruptions. This is in contrast to mathematical optimization techniques that are used to develop long-term planning schedules. In [80], indeed, the authors utilized a single RL-agent DQN to meet the requirements of wafer fabrication dispatching. The DQN was able to achieve predefined work-in-progress (WIP) targets while satisfying strict time constraints better than competitive heuristics. In [81] and [82], an adaptive RL-based production control system for a complex semiconductor job shop has been proposed: it maximized machines utilization and reduced lead and throughput times in partially known environments compared to multiple general heuristics that struggle with incomplete system information. [76] introduced an RL-based dispatching control system to meet flexible objectives in wafer processing, resulting in higher flexibility and fewer delays. The results achieved were comparable to multiple heuristics. Lastly, in [83] the authors proposed a deep RL algorithm as an alternative to conventional Kanban or CONWIP pull production controls. The algorithm balanced conflicting throughput and WIP level targets to optimize local and global production indices simultaneously. By making dynamic adjustments, the algorithm reduced WIP levels without affecting the total throughput.

Effective RL algorithms can be found also for plant-internal logistic tasks, i.e. optimal routing of autonomous mobile robots and automated guided vehicles within the plant. In [84] the authors developed an intra-logistics dispatching solution that utilised autonomous mobile robots (AMRs) to handle real-time production requirements. The AMRs negotiated with each other and virtually raised bids for orders, resulting in improved performance. Simulation was used in [85] with a self-regulating modular production system where an RL agent optimized machine usage based on job information and station status, leading to reduced lead times. In [86] it was implemented a mixed rule dispatching approach for automated guided vehicles (AGVs) that determined the dispatching rule based on the observed state, reducing makespan and delay ratio. The authors of [87] investigated pick-and-place of items from a conveyor belt into baskets using PPO, which achieved a remarkable success rate of 48%.

A detailed classification of all the RL for production control cited references can be found in Table 2.5. Albeit the many successful applications, the actions of RL agents are not fully explainable [77] yet. Moreover, despite RL's proven effectiveness for production planning and control problems, no RL-based models for implementing EEC on manufacturing systems can be found in the literature.

Table 2.5: Detailed classification of the mentioned RL in production control references.

	Category				
Reference	Scheduling	Dispatching	Plant-internal Logistic	Application	Algorithm
[67]	X			Job-Shop Production	DQN
[68]	x			Job-Shop Production	PPO
[69]	x			Job-Shop Production	PPO
[70]	x			Job-Shop Production	DQN
[71]	x			Flow Line Manufacturing	Policy Gradient
[72]	x			Semiconductor	PPO
[74]	x			Chemical Manufacturing	Policy Gradient
[75]	x			Paint-Job Manufacturing	DQN
[80]		x		Wafer Fabrication	DQN
[81]		x		Chemical Manufacturing	DQN
[82]		x		Wafer Fabrication	TRPO
[76]		x		Wafer Fabrication	DQN
[83]		X		WIP Bounding	DQN
[84]			X	AMR Dispatching	TRPO
[85]			X	AGV Scheduling	DQN
[86]			X	AGV Scheduling	DQN
[87]			X	Pick-and-place Robot	PPO

2.3 Research Gaps and Contribution

2.3.1 Research Gaps

Energy efficiency is a key factor for achieving sustainability in manufacturing. The EEC approach offers solutions to minimize the environmental impact of manufacturing equipment, but its implementation faces barriers and limitations. Despite the significant progress made in the field, there indeed remain several notable research gaps that limit the ability of manufacturing systems to achieve optimal energy efficiency. Firstly, a main issue is that current EEC methods assume complete knowledge of system dynamics and parameters, which is often unrealistic. Another strong limitation is related to the manufacturing system layout to be controlled. The application of Machine Learning techniques to forecast and study manufacturing behavior is scarce, thereby impeding the potential for significant improvements in system efficiency and productivity. Secondly, the lack of works utilizing Reinforcement Learning for optimal control of manufacturing systems is a significant research gap that inhibits the development of effective real-time control strategies. Moreover, a scarcity of works addressing the joint EEC problem with other aspects such as maintenance hinders the development of a comprehensive understanding of the interplay between these factors and the resulting impact on system performance. Lastly, parallel machines configuration is indeed widely used in manufacturing but literature does not address the EEC approach for this type of system. Indeed, models to identify EEC policies for single parallel machine workstations or for multi-stage production lines with parallel machine workstations are not present. Addressing these research gaps has the potential to significantly advance the field of EEC for manufacturing equipment.

2.3.2 Thesis Contribution

This thesis aims at overcoming research limitations and providing new ideas and results that have the potential to significantly advance the understanding of EEC for production systems. The innovative methodologies and original findings presented in this work are expected to generate new avenues for future research and shape the direction of the field. Therefore, the thesis presents two main contributions:

1. The creation of a novel research stream, focused on the EEC of manufacturing systems where parallel machine workstations are present. The objective is to propose innovative models for the EEC of single workstations with parallel machines (presented in Chapters 3 and 5) and novel models for the EEC of multi-stage production lines with parallel machine workstations (presented in Chapters 4 and 6).

2. The exploitation of RL techniques to effectively apply EEC to the manufacturing equipment to be controlled without relying on full information about system dynamics. In this way the proposed models, being based on the RL framework, are also adaptive: they can adapt to system dynamics over time without requiring prior and complete knowledge about it. Therefore, they self-adjust during the learning process. In detail, models presented in Chapters 3 and 4 are "yet" MDP-based: although they assume system fully observability, these models nevertheless represent an initial attempt for applying RL for the EEC of manufacturing systems with parallel machines and pave the way for cutting-edge models. As such, they represent a useful analytical benchmark against which more evolved models can be evaluated. Chapters 5 and 6 then present these more advanced RL-based models, which are both general and adaptive

in nature, i.e. they do not require complete information about system dynamics, and which bring this Thesis to a close.

It is pertinent to highlight that all the developed models can be effectively employed in scenarios where workstations with a single machine are present, which is essentially a sub-case of the broader parallel-machines category. Notably, all models incorporate production constraints by optimizing energy consumption while maintaining predefined benchmarks for performance metrics such as system throughput and machine availability.

Chapter 3

MDP-based Model for EEC of Single Workstations

This chapter introduces a novel model to identify EEC policies for a single manufacturing workstation with parallel machines and an upstream buffer of finite capacity. The proposed model reduces energy consumption while ensuring workstation availability. The model assumes fully observable environment that also satisfies the Markov property. Dynamic programming methodology is used to address the problem through MDP. The model is validated through numerical experiments and applied to a real workstation in the automotive sector. Also, proper EEC policies are identified for more parallel machines workstations that are part of the same production line. The effect that these energy-efficient actions have on the overall production system in terms of throughput and energy consumption is therefore studied.

In Section 3.1 a complete formulation of the proposed model is reported. Section 3.2 provides a numerical validation of the model. Section 3.3 presents an overview on the reference industrial case, i.e. the automotive production line. Section 3.4 shows the resulting benefits of the model when applied to the single workstation from the industrial case along with sensitivity analysis of results. Section 3.5 treats the impact that applying EEC on the single workstations has on the overall production system. Conclusions are in section 3.6.

3.1 System Assumptions and Model Formulation

A workstation composed by an upstream buffer of finite capacity and multiple identical parallel machines working on a single part type is considered as the system to be controlled (Figure 3.1). This section is focused on the model to identify an EEC policy for this layout. 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.

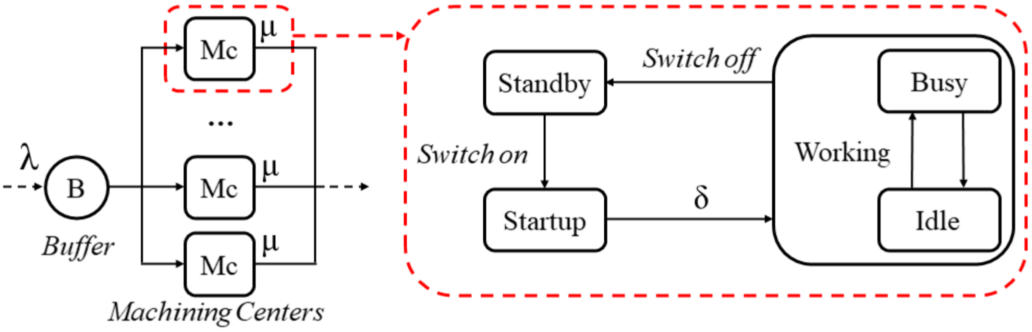


Figure 3.1: The identical parallel machines system under analysis along with the machine state model. Parts arrive to the buffer following a stochastic process with rate λ. Machines have stochastic processing and startup times with rates μ and δ.

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. The proposed model is therefore composed of two modules.

First Module - Sections from 3.1.1 to 3.1.5 describe the first module, addressing the *unconstrained* EEC issue. The problem is formalized as a Continuous Time (CT) MDP and then converted into a DT-MDP with the uniformization technique. Dynamic programming methodology is used to solve the problem. The identified solution of the MDP problem corresponds to the optimal EEC policy π^* minimizing the expected total discounted energy cost over an infinite horizon. The goal of the first module is to identify the optimal EEC policy based on buffer level information.

Second Module - In section 3.1.6 an availability constraint is introduced to address the *constrained* EEC problem. Let us define u_{π^*} as the availability level reached by the system when optimal EEC policy found out with the first module is implemented. If $u_{\pi^*} < u_{target}$, i.e. π^* does not guarantee the achievement of the target level on system availability u_{target}, it is not suitable to be applied to the system under investigation. Starting from π^*, the second module iteratively modifies the policy to be applied until the desired availability level is reached. In this way, a new and suitable EEC policy π_{suit} is obtained. π_{suit} does not guarantee the global optimum in terms of energy saving, but it improves system sustainability while assuring a target availability level.

A complete framework on how to use the model to solve unconstrained and constrained EEC problems is presented in Figure 3.2.

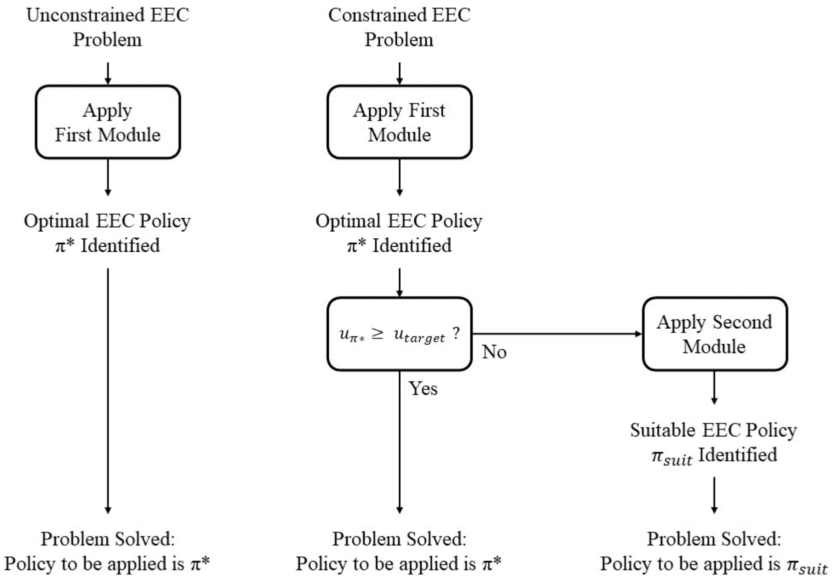

Unconstrained EEC Problem
Apply First Module
Optimal EEC Policy π* Identified
Problem Solved: Policy to be applied is π*
Constrained EEC Problem
Apply First Module
Optimal EEC Policy π* Identified
$u_{\pi*} \geq u_{target}$?
No
Yes
Problem Solved: Policy to be applied is π*
Apply Second Module
Suitable EEC Policy π_{suit} Identified
Problem Solved: Policy to be applied is π_{suit}

Figure 3.2: Framework highlighting how to use the proposed model to address unconstrained and constrained EEC problems.

3.1.1 System Description and Assumptions

The following assumptions are introduced to be aligned with the manufacturing system under investigation (Figure 3.1) and are assumed to be valid in this chapter. The system is composed by c identical parallel machines $M_i, i \in \{1, \dots, c\}$ and one shared upstream buffer with finite capacity $0 < K < \infty$. First come first served rule is applied. There is an infinite capacity buffer downstream the system. Hence, machines cannot be blocked and processed parts leave the system immediately after the process completion. This assumption simplifies the control problem since the blocking effect is not taken into account.

The system is also characterized by different stochastic processes, i.e. machines processing time, machines startup time, and the arrival of parts to the buffer. These processes are assumed to be Poisson processes, independent of each other and stationary. Thus, machines are characterized by exponentially distributed startup time with rate δ and processing times exponentially distributed with rate μ; parts arrive to the buffer following a Poisson process with rate λ. It must be noted that 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. Machines work on a single part type and each machine can be switched off instantaneously. Part arrival process stops when the buffer is full. Part processing and the startup procedure cannot be interrupted by the control.

It must be noted that the assumption of having Poisson processes representing the involved stochastic processes is considered realistic for many industrial cases: examples in literature with this same assumption can be found in [88, 89, 90]. Furthermore, because it naturally gives rise to Markov processes, the exponential distribution has been widely used in the literature [91] and, from [92], 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. even if a model incorrectly assumes that processing times or interarrival times are exponentially distributed, 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.

Each machine is *busy* while working on parts with stochastic processing times with rate μ; on the other hand, the machine is *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. However, the machine is productive only when *busy*. When in *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: a stochastic startup time with rate δ is required to resume inactive machine components. Therefore, each machine is characterized by the following state set $\Theta = \{w, sb, su, id, b\}$, respectively: *working*, *standby*, *startup*, *idle* and *busy*. A detailed overview of the active components for all the machine states or sub-states is reported in Table 3.1 according to ISO-14955-1 standard [93].

All machine states require a constant non-negative amount of power w_s, depending on the visited state or sub-state $s = \{w, sb, su, id, b\}$. In particular, since in *standby* only emergency services are enabled, the power requested in this state is assumed to be almost null ($w_{sb} \simeq 0$). To be in *idle* state, instead, machine has all its modules activated to maintain the ready-for-process conditions but is not executing any actual procedure: the power consumption is high though not the highest ($w_{id} > w_{sb} \simeq 0$). Machine starts executing startup procedures during the respective *startup* phase. Indeed, specific operations are required to make the machine suitable for processing so that quality and tolerance requirements can be met. Examples of these operations are the

Table 3.1: Active components for the machine states or sub-states according to ISO-14955-1 standard. Sub-states idle and busy characterize the working state.

Operative State or Sub-state	Emergency Services	Mains and Mach. Control	Peripheral Units	Processing Unit	Motion Unit	Axes Unit
Standby	ON	OFF	OFF	OFF	OFF	NOT MOVING
Startup	ON	ON	ON	ON - NO MACHINING	ON	MOVING
Idle	ON	ON	ON	HOLD	HOLD	NOT MOVING
Busy	ON	ON	ON	ON MACHINING	ON	MOVING

activation phase of the spindle, the reaching of working temperature for the coolant pump, the starting of the fans and so on. Even if the machine is not processing parts, it is anyway carrying out procedures and for this reason is is assumed that: $w_{su} > w_{id} > w_{sb} \simeq 0$. Finally, it is assumed that the highest power is requested in *busy* state, since the machine has all its modules actives and is also executing part-processing procedure: $w_b > w_{su} > w_{id} > w_{sb} \simeq 0$. 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.

3.1.2 Decision Epochs

The time horizon is divided into periods $k = \{1, 2, \dots\}$ of variable length, according to the occurrence of event $y \in \mathbb{Y}$. Define the event set $\mathbb{Y} = \{A, D, E\}$: part arrival to the buffer $(y = A)$, part departure due to process completion $(y = D)$, and startup completion $(y = E)$. Trivially, departures cannot happen when the system is empty, arrivals cannot happen when the system is full and startup completion occurs only when at least one machine is in *startup* state. The event y_k happens at the end of period k and the decision epochs k correspond to instances of the event y_k.

3.1.3 State Space and Action Space

The system state $\mathbf{s}_k \in \mathbb{S}$, at the beginning of period k, is denoted by the ordered triple $\mathbf{s}_k = \{n_k, x_k, z_k\}$. The number of parts in the upstream buffer is represented with the integer variable $n \in \{0, 1, 2, \dots, K\}$. The number of enabled machines is represented with the integer variable $x \in \{0, 1, 2, \dots, c\}$. The enabled machines are the switched on machines, i.e. that are in *working* or *startup* state. Lastly, $z \in \{0, 1, 2, \dots, c\}$ counts how many machines are in *startup* state. Consequently, the number of machines in *working* state is $(x - z) \in \{0, 1, 2, \dots, c\}$. Each machine has a service rate equal to $\mu > 0$ (when in *working* state) or 0 (when in *startup* or *standby* state). For this reason, the service rate of the overall workstation is $\mu_{tot} = (x-z)\mu \in \{0, \mu, 2\mu, ..., c\mu\}$.

The control action a is applied to control x: the number of machines to be enabled. Consequently, also the number of not-enabled machines, equal to $(c-x)$, is controlled. The not-enabled machines are the switched off machines, i.e. that are in *standby* state. The set of feasible action is $\mathbb{A}(\mathbf{s}, y)$, depending on system state s and event occurrence y. $\mathbb{A}(\mathbf{s}, y)$ is determined by model assumptions and trivial boundaries (e.g. when $x = 0$ it is not possible to switch off any machine and when $x = c$ it is not possible to switch on any machine). At the end of the period k, after the event y_k is observed, x_k can be controlled with the action a_k. The optimal policy π^* maps the optimal action $a_k^*(\mathbf{s}_k, y_k)$ from system state $\mathbf{s}_k$ and event y_k.

3.1.4 System Dynamics and Uniformization

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

$$\mathbf{s}_{k+1} = \begin{cases} \{\min[n_k+1, K], a_k, z_k + \max[0, a_k - x_k]\} & if \;\; y_k = A \\ \{\max[n_k-1, 0], a_k, z_k + \max[0, a_k - x_k]\} & if \;\; y_k = D \\ \{n_k, a_k, \max[z_k + \max[0, a_k - x_k] - 1, 0]\} & if \;\; y_k = E \end{cases} \tag{3.1}$$

To fully understand system dynamics and how z_{k+1} is computed, the implicit effect of the control must be expressed. x and z are indeed strictly dependent: when $a_k > x_k$, a switch on command is applied so and the number of machines in *startup* state z_k is incremented by a quantity equal to $a_k - x_k$ because the machine goes in *startup* state after the switch on transition. In this case: $z_{k+1} = z_k + a_k - x_k$. On the other hand, when $a_k \leq x_k$, machine is switched off, x_{k+1} decreases but z_k does not vary, since the machine immediately goes in *standby* state. In this case: $z_{k+1} = z_k$. Merging the two cases, $z_{k+1} = z_k + \max[0, a_k - x_k]$. Of course, if $y_k = E$ the previous quantity must be decreased by one and, hence, $z_{k+1} = \max[z_k + \max[0, a_k - x_k] - 1, 0]$.

According to the event y_k, the transition probabilities $p(\mathbf{s}_k, \mathbf{s}_{k+1}, y_k, a_k)$ for the MDP problem are:

$$p(\mathbf{s}_k, \mathbf{s}_{k+1}, y_k = A, a_k) = \begin{cases} 0 & if \;\; n_k = K \\ \lambda & otherwise \end{cases} \tag{3.2}$$

$$p(\mathbf{s}_k, \mathbf{s}_{k+1}, y_k = D, a_k) = \begin{cases} 0 \quad if \;\; n_k = 0 \\ (x_k - z_k)\mu \quad otherwise \end{cases} \tag{3.3}$$

$$p(\mathbf{s}_k, \mathbf{s}_{k+1}, y_k = E, a_k) = \begin{cases} 0 \quad if \;\; z_k = 0 \\ \delta \quad otherwise \end{cases} \tag{3.4}$$

In this way, the system is described by a Continuous Time Markov Chain (CTMC). With the uniformization technique [94], it is possible to convert the CTMC into a Discrete Time Markov Chain (DTMC) using a uniform transition rate ν defined as follows:

$$\nu = \lambda + c(\delta + \mu) \tag{3.5}$$

ν represents the exponential parameter that yields the minimum expected transition duration. The transition probabilities become $p'(\mathbf{s}_k, \mathbf{s}_{k+1}, y_k, a_k) = \frac{1}{\nu}p(\mathbf{s}_k, \mathbf{s}_{k+1}, y_k, a_k)$.

3.1.5 Payoff Function and Optimality Equation

The payoff function consists of three elements: the machine power cost per unit time, the holding cost and the startup cost. Also, the energy cost σ must be considered. The machine power request in working state is $w_w(\mathbf{s}, w_i, w_b)$. Defining $m_b = [\min[(x - z), n]$ as the number of *busy* machines and $m_i = (x - z) - m_b$ as the number of *idle* machines, $w_w(\mathbf{s}, w_i, w_b)$ can be defined as $w_w(\mathbf{s}, w_i, w_b) = m_b w_b + m_i w_i$. Thus, the machine power cost per unit time is $c_w(\mathbf{s}, w_i, w_b) = \sigma w_w(\mathbf{s}, w_i, w_b)$. A non-negative constant power consumption w_h is associated to the buffer level n: this value determines a linear holding cost $c_h(n) = \sigma w_h n$. $c_h(n)$ is used to represent a penalty imposed to the system for maintaining parts in the buffer and not processing them; in this way, as w_h increases, the control is prone to be more productive. Being the power consumption in *standby* state w_{sb} close to zero (by assumption), the correspondent standby cost c_{sb} is considered null. The startup cost is non-negative, constant and represented by $c_{su} = \sigma z w_{su}$.

Defining a discount factor $0 < \rho < 1$ and $\eta = \rho + \nu$ it is possible to define the transition probabilities for the infinite horizon discounted cost scenario as $\widetilde{p}(\mathbf{s}_k, \mathbf{s}_{k+1}, y_k, a_k) = \frac{\nu}{\eta}p'(\mathbf{s}_k, \mathbf{s}_{k+1}, y_k, a_k)$. Thereby, the Bellman's optimality equation for the infinite horizon discounted cost can be expressed as:

$$V^*(\mathbf{s}_k) = \min_{a_k}\Big[\, g_k(\mathbf{s}_k) + \sum_{\mathbf{s}_{k+1} \in \mathbb{S}} \widetilde{p}(\mathbf{s}_k, \mathbf{s}_{k+1}, y_k, a_k)(V^*(\mathbf{s}_{k+1}) \; + \max[0, (a_k - x_k)]c_{su}))\Big] \tag{3.6}$$

Where $g_k(\mathbf{s}_k)$ is the stage cost:

$$g_k(\mathbf{s}_k) = \frac{c_w(\mathbf{s}, w_i, w_b) + c_h(n)}{\eta} \tag{3.7}$$

The value iteration method [95] can be used to numerically approximate Equation 3.6. This numerical approximation represents the minimum expected cost that the system, starting from state $\mathbf{s}_k$, will incur when the optimal control action a_k^* is applied. Thus, the optimal policy π^* indicates the optimal action $a_k^*(\mathbf{s}_k, y_k)$ to be implemented, i.e. the number of machines to be enabled from system state $\mathbf{s}_k$ and event y_k to minimize the energy consumption. It must be noticed that, being the system dynamics stationary and the cost functions independent from the period k considered, also the obtained policy is independent from the period k considered. As said, π^* is based on s and y; however, system state is represented by $\mathbf{s} = \{n, x, z\}$ and, as stated in section 3.1.3, a determines both x and z. Thus, the only non-controlled parameter in s is the buffer level n. Hence, π^* can be based on buffer level n and event y. In this way, the optimal EEC policy using buffer level information to switch off/on identical parallel machines is obtained.

3.1.6 Availability Constraint

When a policy π is applied to the system, a Markov Chain can be generated: it describes the system behavior when π is imposed and can be used to compute system performance indicators. In particular, the system availability when π is applied ($u_\pi \in [0, 100\%]$) can be extracted. u_π is a continuous variable: $u_\pi = 0\%$ if all the machines are always not-enabled and $u_\pi = 100\%$ if all the machines are always enabled. If u_{π^*} is higher or equal than the target availability level to be guaranteed u_{target}, π^* is suitable for the system under investigation. On the contrary, if $u_{\pi^*} < u_{target}$, π^* must be modified. Hence, an availability constraint: π^* is iteratively modified until u_{target} is satisfied.

New variables are defined: (i) a_n: the number of machines to be enabled, according to a policy π, when the buffer level is n, (ii) n_s: the highest value of n for which a_n is lower than c, and (iii) a_{n_s}: the number of machines to be enabled, when the buffer level is n_s. In each iteration, a_{n_s} is increased. In this way, a modified policy π' is obtained along with a new system availability $u_{\pi'}$. This procedure is reiterated until a suitable EEC policy π_{suit} is obtained. π_{suit} guarantees $u_{\pi_{suit}} > \quad u_{target}$; thus, it reduces the energy consumption of the identical parallel machines, with a switch off/on approach, while satisfying the target level on the system availability. A complete overview on the model proposed in this work is shown in Algorithm 1 and it includes the two modules presented in section 3.1.

Algorithm 1 The proposed MDP-based model for EEC of identical-parallel machines with availability constraint.

```
First Module: Solve Eq. 3.6 with the value iteration method and find $\pi^*$
Impose $\pi^*$ in the system and compute $u_{\pi^*}$
if $u_{\pi^*} \geq u_{target}$ then
    $\pi_{suit} = \pi^*$.
else
    Second Module:
    $\pi = \pi^* \;\&\; u_\pi = u_{\pi^*}$
    while $u_\pi < u_{target}$ do
        From $\pi$, find $n_s = \max[\, n \mid a_n \neq c \,]$
        Impose $a_{n_s} = a_{n_s} + 1$
        New policy $\pi'$ obtained
        Compute availability $u_{\pi'}$
        $\pi = \pi' \;\&\; u_\pi = u_{\pi'}$
    end while
    $\pi_{suit} = \pi$
end if
```

3.2 Model Validation

A numerical analysis is carried out to show the validity of the proposed model for general cases of identical parallel machines manufacturing systems. A full factorial design with 8 factors at 2 levels [96] is used to generate 256 different cases where the proposed model has been applied. In all the systems, the upstream buffer has a finite capacity equal to 10, a target availability level of 90% is required, the discount factor ρ is equal to 0.80, and the energy cost is $\sigma = 1$. For the varying factors, the low levels of machine power requests are: $[w_b, w_i, w_h, w_{su}] = [20, 0.5, 0.5, 5.5]$ kW. The corresponding respective high levels are: $[w_b, w_i, w_h, w_{su}] = [60, 5, 25, 19.5]$. The mean processing, part arrival and startup times have the same low level equal to 10 s and the same high level equal to 60 s. The corresponding rates μ, λ and δ, reported in Table 3.2.

Table 3.2: Factors and levels for the full factorial design.

Factor	c	λ	μ	δ
Low Level	3	0.02	0.02	0.02
High Level	6	0.1	0.1	0.1

Factor	w_b	w_i	w_h	w_{su}
Low Level	20 (kW)	0.5 (kW)	0.5 (kW)	5.5 (kW)
High Level	60 (kW)	5 (kW)	25 (kW)	19.5 (kW)

In all the experiments, the value iteration method is used to numerically approximate optimality Equation 3.6 with *Matlab (Mathworks, Natick, MA, US)* software (10^3 iterations). The mean computational time for one experiment is lower than 2 s. Figure 3.3 shows the resulting energy saving values for the analyzed configurations. A detailed overview of the numerical results for all the experiments of the factorial design is in Table B.1 of Appendix B.

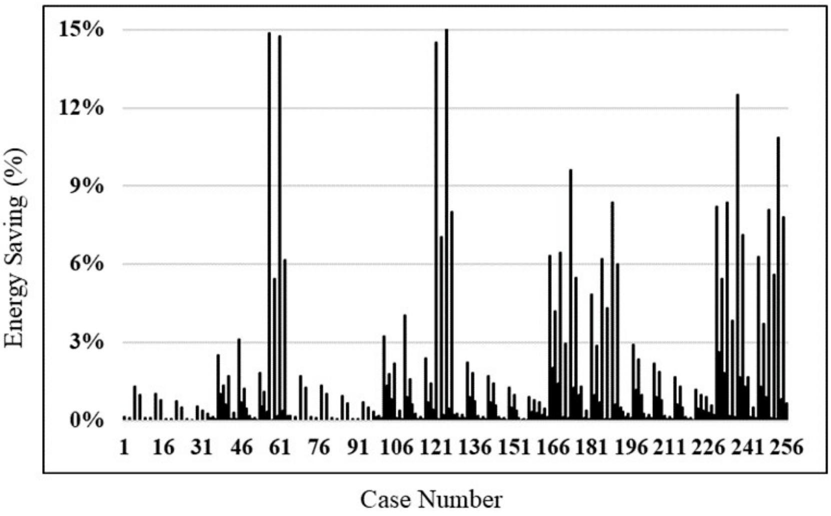


Figure 3.3: Full factorial analysis: achieved energy saving in the different cases.

In 32 cases, due to specific conditions (e.g. fast arrival of parts and/or high machine processing time), the identified policy π_{suit} corresponds to the *AOn* policy. In these cases, the machines are almost never idle and, for this reason, they cannot be switched off for energy saving purposes. In all the other 224 cases, π_{suit} is different from the *AOn* policy and this leads to significant energy saving values, up to 15.60%. The highest savings are achieved with parameters leading to frequent production periods with few parts in the buffer (e.g. slow arrival of parts and/or low machine processing time). In these cases, more machines in the workstation are frequently idle and might be switched off to save energy. However, in all the analyzed situations, the application of the proposed model is able to offer an appropriate and effective EEC policy and, for this reason, the model effectiveness is verified.

3.3 Industrial System Description

A real industrial system is used as reference case study where the proposed MDP model can be applied and its effects can be analyzed. The production line under investigation is a manufacturing system producing cylinder heads in the automotive sector (Figure 3.4). A detailed overview on system parameters is reported in Table A.1 of Appendix A. In this line, parts are carried on pallets and pallets are carried by a conveyor connecting all the workstations. The number of pallets circulating in the system is fixed and constant.

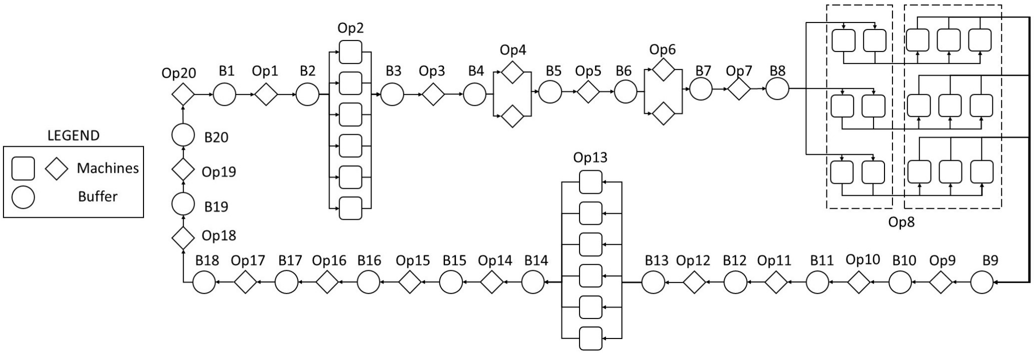


Figure 3.4: Layout of the industrial system under investigation.

The production process consists of 20 total operations performed by automated and semi-automated equipment. In particular, three of these operations are machining operations (Op.$2, 8$, and 13). All the system devices are unreliable and have stochastic processing time. Machines Time To Failure (*TTF*) and Time To Repair (*TTR*) are all stochastic distributed. First come first serve and blocking after service rules are applied. All the buffers have finite capacity. The proposed model is at first applied only on Op.2, without focusing on its interactions with the shop floor and, subsequently the EEC is implemented on both Op.2 and Op.13 to assess the impact of the energy-efficient actions on the overall production line.

3.4 Numerical Experiments

3.4.1 Industrial Case Analysis

In this section, the proposed model is applied to an industrial workstation to assess its benefits when implemented on a real system. The latter is an identical parallel machines workstation performing Op.2 in the manufacturing line described in section 3.3. The energy cost is set as $\sigma = 1$. The workstation is composed by 6 identical parallel machines. The upstream buffer has a finite capacity equal to 10. For this system, a target availability level of 85% is required. The mean processing, part arrival and startup times are, respectively, equal to 83.70, 20 and 30 s. The corresponding rates μ, λ and δ, along with the other system parameters are reported in Table 3.3. These parameters are provided by the company owning the industrial system under study.

The value iteration method [95] is used to numerically approximate optimality Equation 3.6 with *Matlab* software (10^3 iterations). The computational time

Table 3.3: System parameters for the industrial case under investigation.

Parameter	Value	Parameter	Value
c	6	ν	0.34
K	10	w_b	15 (kW)
λ	0.05	w_i	9.30 (kW)
μ	0.01	w_h	0.75 (kW)
δ	0.03	w_{su}	10 (kW)
ρ	0.80	w_{sb}	0 (kW)

is lower than 2 s. The performances of interest are the energy consumption and the availability level of the standalone workstation Op.2.

At first, the model is applied to the system without the availability constraint. The optimal EEC policy π^* is obtained (reported in Table 3.4). π^* indicates, for each possible n, the optimal action a_n to perform, i.e. the number of machines to be enabled in the workstation. The resulting average energy saving is 9.38% and it is computed comparing π^* to the *AOn* policy. However, the system availability (81.94%) is lower than the target (85%) and, for this reason, π^* is not suitable for the system under investigation. Afterwards, the complete version of the model is applied, inserting the availability constraint. Starting from π^*, 2 iterations are required to reach the target availability level required:

1. Iteration 1 $\rightarrow n_s = 5 \rightarrow a_5 = a_5 + 1 = 6 \rightarrow u_{\pi'} = 83.95\% < u_{target}$.
2. Iteration 2, starting from the policy π obtained in iteration 1 $\rightarrow n_s = 3 \rightarrow a_4 = a_4 + 1 = 5 \rightarrow u_{\pi'} = 86.49\% > u_{target}$. The policy to be applied is $\pi_{suit} = \pi'$.

In this way, a suitable policy π_{suit} is obtained (Table 3.4). The new availability level (86.49%) is higher than the target (85%): π_{suit} is suitable for the industrial case. Finally, the resulting average energy saving (8.76% in respect to the *AOn* policy) is lower than the previous case (9.38%), but still represents a significant value.

Table 3.4: EEC policy π^* and π_{suit} for the industrial workstation under investigation.

Policy	n:	0	1	2	3	4	5	$6,\dots,10$	En. Sav. [%]	u_{target}	$u_{\pi'}$
π^*	a_n:	0	1	2	3	4	5	6	9.38%	-	81.94%
π_{suit}	a_n:	0	1	2	3	5	6	6	8.76%	85.00%	86.49%

3.4.2 Sensitivity Analysis

A sensitivity analysis of results is performed to understand into details the effect of u_{target} and w_h (i.e. the target availability and the holding power consumption) on system performances.

Target Availability - In this analysis u_{target} varies while the other system parameters remain unchanged (Table 3.3). Increasing u_{target} leads to increasing system throughput until the maximum possible throughput for the system under study is reached. This value is equal to the arrival rate of parts λ because of flow conservation (Figure 3.5-(a)). Moreover, when system availability increases, machine power cost enhances and energy saving decreases until a null value is reached (Figure 3.5-(b)). System availability equal to 100%, indeed, corresponds to the *AOn* policy condition; thus, the energy saving achieved in this case is null. Finally, the achievement of u_{target} is always guaranteed (Figure 3.5-(c)). A detailed overview of the numerical results is in Table B.2 of Appendix B.

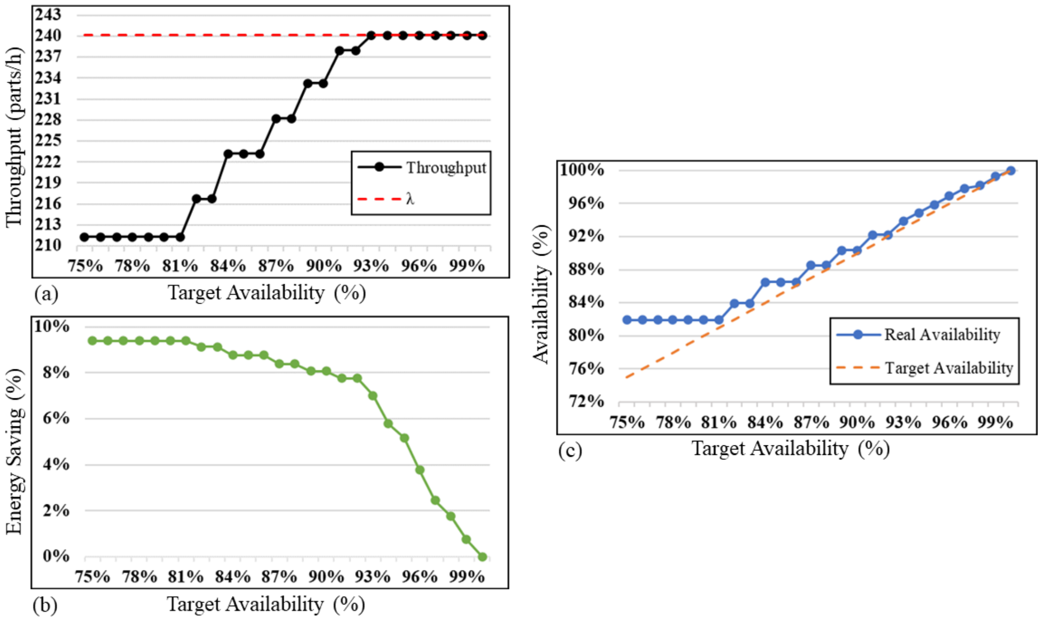


Figure 3.5: Sensitivity analysis for the industrial case: throughput (a), energy saving (b) and system availability (c) achieved when u_{target} varies.

Holding Cost - In this case the holding power w_h varies and the other system parameters remain the same as in Table 3.3. Figure 3.6 shows system performance for this analysis. For low values of w_h, the optimal EEC policy might indicate to keep the system availability low, since maintaining parts in the buffer (and not processing them) has a low total cost in this case. However, the availability constraint imposes the system availability to be above the target level. At the same time, the energy saving slightly decreases because of the increasing holding cost. For high values of w_h, the situation is different. The resulting high holding cost leads to EEC policies imposing high system availability (and high throughput): in this way, parts are processed and not maintained in the buffer. However, high system availability means enhancing machine power cost and decreasing energy saving for high $w_h(n)$ values. Finally, the achievement of u_{target} is always guaranteed. A detailed overview of the numerical results is in Table B.3 of Appendix B.

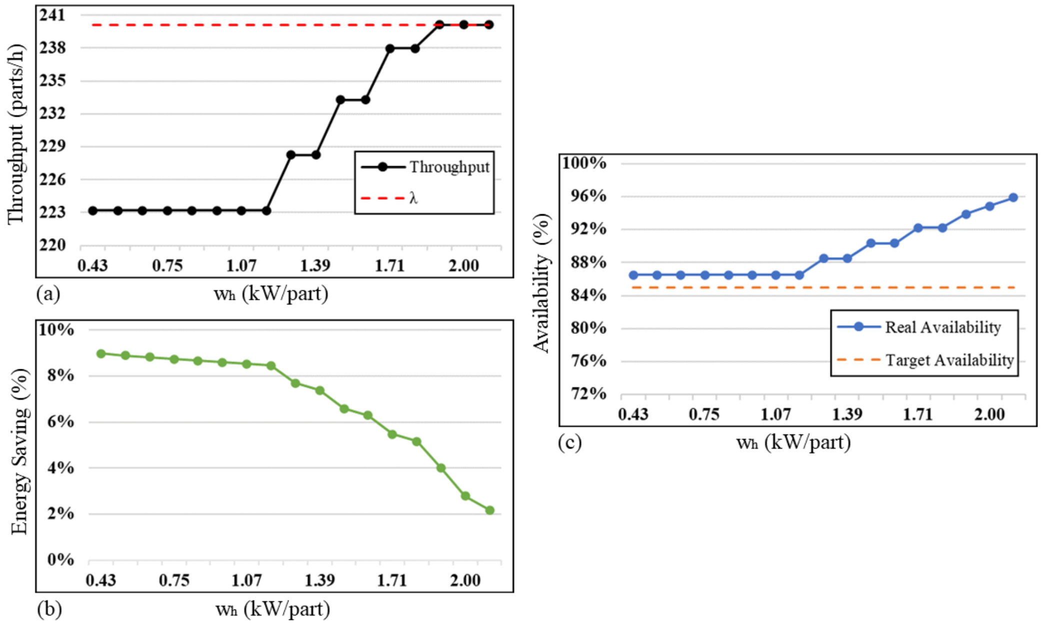


Figure 3.6: Sensitivity analysis for the industrial case: throughput (a), energy saving (b) and system availability (c) achieved when w_h varies.

3.4.3 Assumptions Modification

The policy π_{suit} identified in section 3.4.1 is then applied to Op.2 of the industrial case but with different assumptions, in order to verify the effectiveness of the proposed model also for more general cases. System assumptions on stochastic processes involved are modified in this analysis: machine processing and startup times are imposed following two lognormal distributions equal to, respectively, lognormal of parameters $(4.32, 0.47)$ and lognormal of parameters $(3.28, 0.47)$. The two distributions lead to the same coefficient of variation (0.50) and mean values equal to, respectively, 83.70 and 30 s. The mean values are the same as in section 3.4.1, to compare two cases with aligned parameters, evaluating only the difference in terms of stochastic distribution involved. The lognormal distribution is selected because it is always statistically plausible to model a stochastic time distribution with a lognormal if its coefficient of variation is lower than 1 [97]. Thus, this distribution leads to a system modelling undoubtedly aligned to real manufacturing systems. All the other system parameters remain the same as in Table 3.3.

Discrete event simulation is used for performance evaluation: the number of experiments is equal to 25 with a simulation length of 100 days. The system simulation model is developed in *Arena* environment. Under these conditions, the resulting average energy saving is equal to $7.22 \pm 0.06\%$ in respect to the *AOn* policy; this value is extracted with a confidence level of 95% on its confidence interval. Due to the different stochastic distribution involved, lognormal instead of exponential, the resulting energy saving value is lower than the original case analyzed in section 3.4.1 (8.76%) but still represents a significant value. This confirms the effectiveness of the proposed model also with a modification of the assumptions.

3.5 Impact on the Overall Production System

3.5.1 Scenarios Analyzed

To assess the impact that the EEC on one or more parallel machines workstations has on the overall production system, suitable EEC policies are identified and applied to Op.2 and Op.13 of the industrial case. These workstations are identical and have parameters reported in section 3.4.1 and Table 3.3. In particular, seven scenarios are studied (Table 3.5).

Table 3.5: Scenarios analyzed.

Scenario Number	**EEC Policy on Op.2**	u_{target} **for Op.2**	**EEC Policy on Op.13**	u_{target} **for Op.13**
1	No	-	No	-
2	Yes	85%	No	-
3	Yes	90%	No	-
4	No	-	Yes	85%
5	No	-	Yes	90%
6	Yes	85%	No	85%
7	Yes	90%	No	90%

Scenario 1 represents the base case, i.e. system without any EEC policy implemented, where the *AOn* policy is applied on Op.2 and Op.13. Scenario 1 is used as benchmark to compare the performance of the system as-is with the cases where EEC is implemented on one or more workstations. In Scenarios 2 and 3 only Op.2 is controlled, but varying the target availability level to be guaranteed for that workstation (from 85% to 90%). In these cases, the *AOn* policy is applied to Op.13. The reverse situation is represented by scenarios 4 and 5, where Op.13 is controlled, but with two distinct target availability levels (85% and 90%) and Op.2 is subject to the *AOn* policy. Finally, in scenarios 6 and 7, both workstations are controlled, but with increasing u_{target} in the two situations (still from 85% to 90%).

In all the cases, the *AOn* policy is implemented on all the other system devices. Numerical experiments are conducted with discrete event simulation. The performance of interest in this analysis are the production system througput and the energy consumptions of both workstations Op.2 and Op.13. In all the experiments the energy cost is $\sigma = 1$.

3.5.2 EEC Policies Computation

As reported in section 3.1, to apply the proposed model and obtain a suitable EEC policy for one parallel machines workstation, it is required to define the following inputs: the workstation parameters, a discount factor ρ, a uniform transition rate ν and the target availability level u_{target}. However, in all the scenarios, ρ and ν do not vary: they are fixed and equal, respectively, to 0.80 and 0.25. Moreover, also the workstation parameters for Op.2 and Op.13 do not vary: they are fixed and identical for both of them in each scenario (see Section 3.4.1). Thus, the only varying parameter for both workstations in the different scenarios is u_{target}. Nevertheless, Op.2 and Op.13 are subject to the same two u_{target} values and this leads to two different EEC policies:

- Policy π_{85} when u_{target} is equal to 85%. To be applied to Op.2 in Scenario 2 and 6, and to Op.13 in Scenario 4 and 6.

- Policy π_{90} when u_{target} is equal to 90%. To be applied to Op.2 in Scenario 3 and 7, and to Op.13 in Scenario 5 and 6.

Once all the fixed input parameters are defined, the model can be implemented twice: one for each different u_{target} value. The obtained EEC policies are presented in Table 3.6.

The control is executed using buffer level information: each policy indicates, for each possible upstream buffer level n, the corresponding number of machines in the station that should be switched on: a_n. As an example, π_{85} indicates that whenever the upstream buffer level n is equal to 4, in Op.2 (or Op.13) three machines must be switched on ($a_{n=4} = a_4 = 3$) and three machine must be switched off ($c - a_4 = 3$).

Table 3.6: EEC policies π_{85} and π_{90} for Op.2 and Op.13 in the industrial case under investigation.

Policy	n:	0	1	2	3	4	$5, \dots, 10$
π_{85}	a_n:	0	1	2	3	5	6
π_{90}	a_n:	0	1	2	4	6	6

3.5.3 Simulation Model Parameters

The simulation model is consistent with the system described in sections 3.1.1 and 3.3. Given the presence of stochastic parameters as *TTR* and *TTF* for each system device, also the simulation model is stochastic. Thus, the target parameters of the analysis, i.e. the system throughput and the energy consumption of Op.2 and Op.13, are random variables. Because of that, for each scenario, the experiment is replicated for a fixed number of times, equal to 10, and all the target parameters are extracted with a 90% confidence level on the respective mean value. All the experiments are performed with an effective simulation length of 50 weeks, considering 127 hours per week: the total duration corresponds to one year of production time. The same transient period is imposed for each experiment, equal to 30 days of production period: this represents an overestimation, for computational-accuracy reasons, of the transient period identified with the Welch method ([98]).

3.5.4 Experimental Results

Experimental results of scenarios from 2 to 7 are then compared to scenario 1, to assess the impact at system level of the EEC application on Op.2 and Op.13. Results are reported in Table 3.7 for the throughput loss, and in Table 3.8 for the energy saving. A comprehensive overview is provided in Figure 3.7.

Table 3.7: Impact of the EEC policies application on the industrial system daily throughput, all scenarios are compared to the scenario 1.

Scenario Number	**Throughput [Parts/Day]**	**Throughput Loss in respect to the base case [%]**
1	1445.09 ± 2.12	-
2	1434.30 ± 2.05	0.75 ± 0.16
3	1445.03 ± 0.77	0
4	1430.08 ± 1.04	1.04 ± 0.10
5	1444.94 ± 1.56	0
6	1427.85 ± 2.89	1.19 ± 0.21
7	1446.25 ± 0.85	0

Table 3.8: Impact of the EEC policies application on the daily comprehensive energy consumption of Op.2 and Op.13, all scenarios are compared to scenario 1.

Scenario Number	Energy Consumption for Op.2 and Op.13 [kWh in one day]	Energy Saving in respect to the base case [%]
1	3137.82 ± 1.85	-
2	3009.38 ± 2.67	4.09 ± 0.10
3	3080.81 ± 1.89	1.82 ± 0.08
4	3018.62 ± 2.06	3.80 ± 0.19
5	3098.38 ± 2.97	1.26 ± 0.20
6	2908.82 ± 3.48	7.30 ± 0.21
7	2955.47 ± 3.10	5.81 ± 0.17

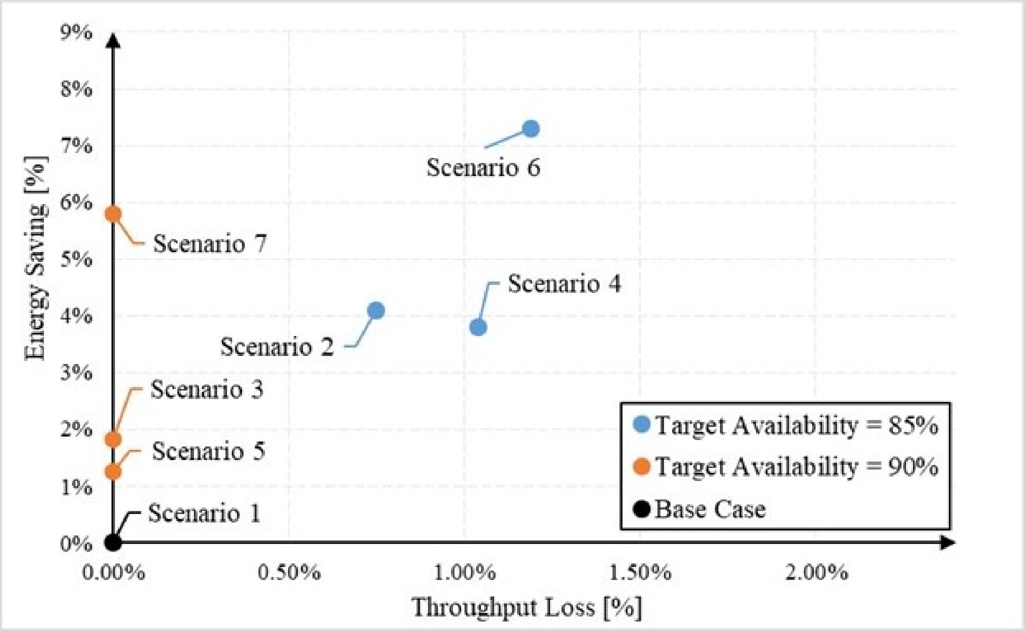


Figure 3.7: Impact of the EEC policies application industrial case under investigation, all the scenarios are compared to the base case (scenario 1).

In all the scenarios, the EEC implementation always significantly reduces the energy consumption. However, in scenarios 2, 4 and 6, where π_{85} is applied to Op.2 and/or Op.13, a slight productivity reduction is present. The u_{target} associated to π_{85} is not very high and, for this reason, the workstations with π_{85} applied are characterized by machines frequently switched off. This leads to a great improvement regarding the process sustainability in respect to the base case but, being the corresponding u_{target} not excessive, there is a throughput loss. On the other hand, the u_{target} associated to π_{90} is higher and the workstations with π_{90} applied are characterized by higher availability, i.e. higher productivity: this leads to a null throughput loss while the energy saving is lower but still significant, as in scenarios 3, 5 and 7. Furthermore, the EEC implementation on two workstations always leads to a major reduction on the energy consumption: scenarios 6 and 7, indeed, are characterized by higher energy saving than the other cases.

Scenario 7 can be identified as the best and preferable option to be realized in the industrial system under study: this configuration is the one with the highest energy saving among the cases with null throughput loss. Indeed, if u_{target} is progressively reduced from scenario 7, both the throughput loss and the energy saving will gradually increase, as in scenario 6. Therefore, although it is possible to achieve larger energy savings, this will also lead to higher productivity drops.

The results confirm that, even applying an EEC policy to only one workstation, the corresponding benefits in terms of environmental impact are relevant. In addition, through a trade-off between system production rate and energy demand, it is possible to select properly the target workstation availability and choose if it is preferable to decrease more the energy consumption, but causing a productivity reduction, or to maintain the same production level and anyway reduce the workstation energy use. Moreover, the more the EEC is executed in the production system, the higher the benefits in terms of environmental impact.

3.5.5 Detailed analysis of Scenarios 6 and 7

Scenarios 6 and 7 are characterized by the highest energy saving values. Hence, it is interesting to understand how the EEC policies modify the operation of Op.2 and Op.13 in these cases. In Figure 3.8, the percentages of time that each controlled workstation spends with a certain number of switched on machines in both scenarios are reported.

Trivially, when the *AOn* policy is applied on Op.2 and Op.13 as in scenario 1, all the 6 machines in the workstations are always switched on. On the other hand, when π_{90} is applied (scenario 7, Figure 3.8-(a)), the amount of time in which all the 6 machines in the workstation are switched on is decreased. Moreover, for a consistent amount of time only 2 or 4 machines are switched on and, in addition, for small periods there is even only 1 or 0 switched on machines. Finally, it never occurs to have exactly 3 or 5 switched on machines, since in π_{90} a_n is never equal to 3 or 5. This different workstation functioning leads to the energy saving observed in Table 3.8, although it has no effects on the system throughput.

A different situation can be observed when π_{85} is implemented (scenario 6, Figure 3.8-(b)). The amount of time spent with 6 switched on machines is further reduced while for long periods the workstations operates with 2, 3 or 5 switched on machines and, for small periods, Op.2 and Op.13 work with only 1 or 0 switched on machines. Finally, being a_n never equal to 4 according to π_{75}, the workstations never has exactly 4 switched on machines. This behavior of the two workstations leads to a higher energy saving than in scenario 7, but also causes a throughput loss.

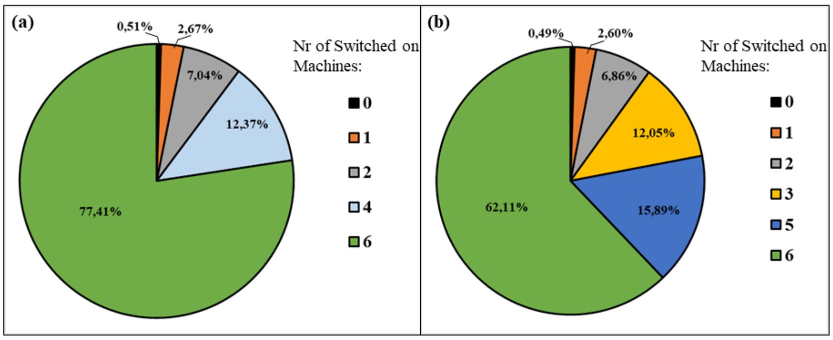


Figure 3.8: Percentages of time in which a specific number of machines are switched on in Op.2 and Op.13 in scenario 7 (a) and scenario 6 (b).

3.6 Conclusions

In this chapter, a novel model is presented: it is used to identify an EEC policy based on buffer level information for identical parallel machines with finite buffer capacity used in manufacturing systems. The proposed model leads to the reduction of the energy consumption; at the same time, it ensures a target level on system availability without large computational efforts. The model is also applicable to a workstation with a single machine, as it is a specific sub-case of the parallel-machines category. Numerical results are presented, showing model benefits when applied to a real industrial case from the automotive sector.

Furthermore, the model has been applied to more identical parallel machines workstation pertaining to the same manufacturing line and, afterwards, it has been studied the impact that this energy-efficiency action has on the entire production line. Numerical results are presented, showing the corresponding benefits when the policy is applied to one or more workstations pertaining to a real industrial system from the automotive sector. The EEC policies always lead to significant reduction in terms of energy consumption (1 to 7.50% of saving). Moreover, with an appropriate selection of the target workstation availability level to be satisfied, it is possible to decrease the energy use without jeopardizing the production rate (0 to 1.50% of throughput loss). Finally, the more workstations are controlled in an energy-efficient way, the higher the impact in terms of energy saving.

Although this model has shown promise and effectiveness, it is restricted to situations where Markovian processes are exclusively present, there is no focus on the interactions between workstations and the surrounding shop floor, and only the availability level is considered as a constraint. A challenging topic regards the creation of a novel model leading to a unique EEC policy for the overall production system, where the control is executed jointly in all the workstations, considering the overall system state in each control action. In this way, also blocking effects on each workstation can be taken into account in the control problem. Additionally, it might be required to include a variety of general production constraints, not just the availability of a single station. These conclusions have been the starting point leading to the analyses and work discussed in Chapter 4.

Chapter 4

MDP-based Model for EEC of Multi-Stage Production Lines

Chapter 3 paved the way for the MDP-based EEC of single manufacturing workstations, highlighting the potential benefits of this approach for such layout. At the same time, the related results have been limited by the focus on the stand-alone workstation without considering its interactions with surrounding the shop floor and by the consideration of a constraint only on station availability. This chapter aims at overcoming these limitations.

The goal is to develop a novel MDP-based model for the EEC of multi-stage production lines, reducing the overall energy consumption while reaching desired target levels on system performance. At first, a novel model, referred to as *2S-Model*, is proposed to get exact solutions for the EEC of a 2-stage production line. Subsequently, a novel technique, referred to as *Backward-Recursive approach*, is proposed to deal with longer production lines. The model can also be applied when workstations with a single machine are present in the line, since they represent a sub-case of the parallel-machines category. Numerical experiments on realistic manufacturing system configurations are performed to assess the effectiveness of the proposed model.

The remainder of the chapter is organized as follows. Section 4.1 describes the system under investigation. Problem formulation for 2-stage lines (i.e., the 2S-Model) is in section 4.2 and section 4.3 formulates the Backward-Recursive approach for more than two stages. Numerical analysis is presented in sections 4.4 and 4.5 respectively for 2-stage lines and longer ones. Section 4.6 concludes the chapter.

4.1 System 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 following assumptions are related to the system to be controlled and are considered valid for the model presented in this chapter. The system layout is represented in Figure 4.1.

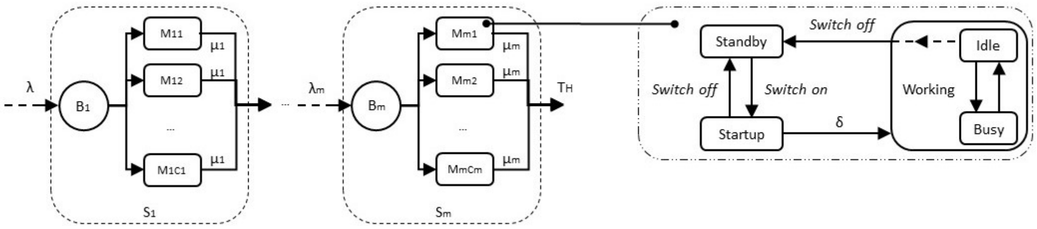


Figure 4.1: Layout of the production line under analysis with machine state model.

Stages are denoted as S_i with $i \in \{1, \ldots, m\}$, buffers as B_i with $i \in \{1, \ldots, m\}$, and M_{ij} with $j \in \{1, \ldots, c_i\}$ 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 $j \in \{1, \ldots, c_i\}$ 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. 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. 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. As illustrated in section 3.1.1 the assumption of having Poisson processes representing the aforementioned stochastic processes is considered reasonable and applicable to the system under study without undermining the system model and consequent results.

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.

Machines M_{ij} can be controlled for energy saving purposes. Each machine has an energetic state model consistent with section 3.1.1 and reported in Figure 4.1, i.e. characterized by the following state set $\Theta = \{w, sb, su, id, b\}$, respectively: *working*, *standby*, *startup*, *idle* and *busy*. It is also assumed that machines of a certain stage i are characterized by the same power consumption in the different states: consequently, it is assumed that $w_b > w_{su} > w_{id} > w_{sb} \simeq 0$ (consistently with section 3.1.1).

4.2 Formulation of the Exact 2S-Model

Let us consider a production line consistent with the description of section 4.1 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. It is assumed that the environment to be controlled satisfies the Markov property and is fully observable. Then, the problem is formalized as a Continuous Time MDP (CT-MDP) and then converted into a DT-MDP with the uniformization technique [94]. Then, a a linear programming (LP) approach can be used to solve the DT-MDP problem and the exact solution can be identified [99]. Sections from 4.2.1 to 4.2.5 describe the MDP. Section 4.2.6 introduces the LP formulation which allows enriching the MDP formulation by including general production constraints in the problem. Section 4.2.7 provides a descriptive example of a control policy for the EEC problem at hand.

4.2.1 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.

4.2.2 State Space and Action Space

The system state is defined as **s** $\in \mathbb{S}$, where $\mathbb{S}$ is the discrete state space representing all possible system states, denoted by the ordered vector **s** $= \{n_1, n_2, x_1, x_2\}$: integer variable $x_i \in \{0, 1, 2, \ldots, c_i\}$ 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 $n_i \in \{0, 1, 2, \ldots, K_i + c_i\}$. 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, **s** is referred to as **s**$_k = \{n_{1,k}, n_{2,k}, x_{1,k}, x_{2,k}\}$.

The control action **a** $= [a_1, a_2]$ is applied to control the numbers of machines to be in working state x_1 and x_2. Hence, **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 state **s**, i.e. the allowable values **a** can assume each time 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 **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 **a**$_k = [a_{1,k}, a_{2,k}]$. The optimal policy $\pi^* : \mathbb{S} \times \mathbb{Y} \rightarrow \mathbb{A}_s(y)$ maps the optimal action **a**$_k^*$(**s**$_k, y_k$) given system state **s**$_k$ and occurrence of event y_k.

4.2.3 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 $\mathbb{A}_s(y) \rightarrow \mathbb{S}$. Given system state **s**$_k$, event y_k and control action **a**$_k$, the next system state **s**$_{k+1} = \{n_{1,k+1}, n_{2,k+1}, x_{1,k+1}, x_{2,k+1}\}$ is defined as follows:

$$\mathbf{s}_{k+1} = \begin{cases} \{\min[n_{1,k}+1, K_1+x_{1,k}], n_{2,k}, \min[a_{1,k}, x_{1,k}], \min[a_{2,k}, x_{2,k}]\} & if\ y_k = A_1 \\ \{\max[n_{1,k}-1, 0], \min[n_{2,k}+1, K_2+x_{2,k}], \min[a_{1,k}, x_{1,k}], \min[a_{2,k}, x_{2,k}]\} & if\ y_k = A_2 \equiv D_1 \\ \{n_{1,k}, \max[n_{2,k}-1, 0], \min[a_{1,k}, x_{1,k}], \min[a_{2,k}, x_{2,k}]\} & if\ y_k = D_2 \\ \{n_{1,k}, n_{2,k}, \min[a_{1,k}, x_{1,k}+1], \min[a_{2,k}, x_{2,k}]\} & if\ y_k = E_1 \\ \{n_{1,k}, n_{2,k} \min[a_{1,k}, x_{1,k}], \min[a_{2,k}, x_{2,k}+1]\} & if\ y_k = E_2 \end{cases} \tag{4.1}$$

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 4.2.4).

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:

$$p(\mathbf{s}_k, \mathbf{s}_{k+1}, y_k = A_1, \mathbf{a_k}) = \begin{cases} 0 \quad if \;\; n_{1,k} = K_1 + x_{1,k} \\ \lambda \quad otherwise \end{cases} \tag{4.2}$$

$$p(\mathbf{s}_k, \mathbf{s}_{k+1}, y_k = A_2 \equiv D_1, \mathbf{a_k}) = \begin{cases} 0 \quad if \;\; n_{1,k} = 0 \;\wedge\; n_{2,k} = K_2 + x_{2,k} \\ \mu_{1,k}^{tot} \quad otherwise \end{cases} \tag{4.3}$$

$$p(\mathbf{s}_k, \mathbf{s}_{k+1}, y_k = D_2, \mathbf{a_k}) = \begin{cases} \mu_{2,k}^{tot} \quad if \;\; n_{2,k} > 0 \\ 0 \quad otherwise \end{cases} \tag{4.4}$$

$$p(\mathbf{s}_k, \mathbf{s}_{k+1}, y_k = E_i, \mathbf{a_k}) = \begin{cases} 0 \quad if \;\; x_{i,k} = c_i \\ \delta_i \quad otherwise \end{cases} \tag{4.5}$$

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 [94]: $\nu = \lambda + c_1(\delta_1 + \mu_1) + c_2(\delta_2 + \mu_2)$. 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 $\widetilde{p}(s_k, s_{k+1}, y_k, \mathbf{a_k}) = \frac{1}{\eta} p(s_k, s_{k+1}, y_k, \mathbf{a_k})$.

4.2.4 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 $w_{i,h} bn_i$.

4.2.5 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 [99]. 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:

$$V^*(\mathbf{s}) = \min_{\mathbf{a} \in \, \mathbb{A}_s(y)} \left[g(\mathbf{s}) + \sum_{\mathbf{s}' \in \, \mathbb{S}} \widetilde{p}(\mathbf{s}, \mathbf{s}', y, \mathbf{a})(V^*(\mathbf{s}') \; + \eta a_c(\mathbf{s})) \right] \tag{4.6}$$

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

$$g(\mathbf{s}) = \frac{\sigma}{\eta} \sum_{i=1}^{m=2} \big((bu_i w_{i,b} + id_i w_{i,id}) + w_{i,h} bn_i + w_{i,sb} sb_i \big) \tag{4.7}$$

$$a_c(\mathbf{s}) = \frac{\sigma}{\eta} \sum_{i=1}^{m=2} w_{i,su} su_i \tag{4.8}$$

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 4.6 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 2-stage production line with parallel machine workstations can be obtained and the unconstrained EEC problem can be solved.

4.2.6 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 [99]. Examples of this MDP to LP formulation can be found in [100, 101, 102]; 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 [99]. 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:

$$min \sum_{\mathbf{s}\in\mathbb{S}} \sum_{\mathbf{a}\in\mathbb{A}_s} \beta(\mathbf{s})(g(\mathbf{s}) + a_c(\mathbf{s})) \tag{4.9}$$

$$s.t. \qquad \beta(s) - \sum_{\mathbf{s}\in\mathbb{S}} \sum_{\mathbf{a}\in\mathbb{A}_s} \sum_{y\in\mathbb{Y}} \widetilde{p}(\mathbf{s}, \mathbf{s}_{+1}, y, \mathbf{a})\beta(\mathbf{s}) = \alpha(\mathbf{s}) \tag{4.10}$$

$$0 \leq \beta(\mathbf{s}) \leq 1 \quad \forall \mathbf{s} \in \mathbb{S} \tag{4.11}$$

$$g(\beta(\mathbf{s})) \geq G^* \tag{4.12}$$

where Equation 4.9 is the objective function, Equation 4.10 is a structural property to be ensured in an MDP to LP formulation (proof in [99]), and Equation 4.11 represents the boundary conditions, i.e. $\beta(\mathbf{s}) \in [0, 1]$. Equation 4.12 defines a general production constraint to be respected while guaranteeing the minimization of the energy cost. Equation 4.12 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 ($T_H^{\textit{target}}$) to be met at least, so that Equation 4.12 becomes:

$$\sum_{\mathbf{s}\in\mathbb{S}} \beta(\mathbf{s})\mu_{i,k}^{tot} \geq T_H^{\textit{target}} \quad for \;\; i = 1,2 \tag{4.13}$$

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

$$\sum_{\mathbf{s}\in\mathbb{S}} \beta(\mathbf{s})\frac{x_i}{c_i} \geq A_V^{\textit{target}} \tag{4.14}$$

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

$$\sum_{\mathbf{s}\in\mathbb{S}} \beta(\mathbf{s})(n_1 + n_2) \leq wip^{\textit{target}} \tag{4.15}$$

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 [99]). Moreover, $\beta^*(\mathbf{s})$ and π^* do not depend on the initial state distribution $\alpha(\mathbf{s})$ (proof in [99]), 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 2-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.

4.2.7 Policy Illustrative Example

An optimal EEC policy π^* maps the optimal actions **a*** to be implemented in the system, i.e. for each possible state **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 **a*** directly controls x_1 and x_2 (section 4.2.5).

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 **s** $= \{4, 2, 1, 1\}$ and the optimal action for $[n_1, n_2] = [4, 2]$ is **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 **s'** $= \{4, 2, 2, 1\}$. On the other hand, if the system is in state **s** $= \{0, 1, 2, 2\}$ and the optimal action for $[n_1, n_2] = [0, 1]$ is **a**$^* = [1, 1]$, then one machine is switched off in both S_1 and S_2 leading to new state **s'** $= \{0, 1, 1, 1\}$. In order to give a benchmark, the *AOn* policy would indicate to maintain always

2 machines switched on in both stages: **a** $= [2, 2]$, for any state possible state **s**, i.e. any value of $[n_1, n_2]$.

4.3 Formulation of the Approximate Model for Long Production Lines

Let us consider a production line consistent with the description of section 4.1 and $m > 2$ stages, S_i with $i \in \{1, \ldots, m\}$. 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 2-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.

An overview of the extended 2S-Model, namely the Extended-2S, is presented in section 4.3.1 while details on the Backward-Recursive approach are in section 4.3.2.

4.3.1 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 **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 4.2. The formulation of the problem to be solved is still described by Equations 4.9-4.12 except for system dynamics (Equations 4.10) 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.

4.3.2 The Backward-Recursive Approach

The Backward-Recursive approach is represented in Figure 4.2 and Algorithm 2. The algorithm starts breaking down the original system into a series of 2-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 $\tilde{a} = \widetilde{a}_{i-1}^*$ 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. *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 $\widetilde{a}_i^* = \tilde{a}$ 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 $i \in \{2, \dots, m\}$ in the unique approximate policy Π^*: the control policy to reduce the system energy consumption.

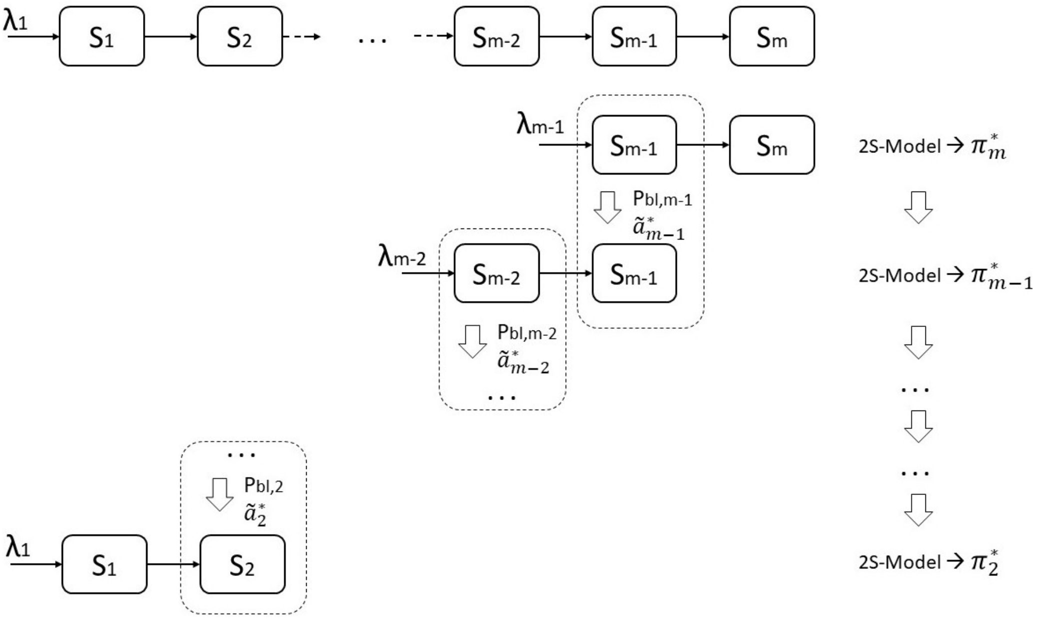


Figure 4.2: Illustration of the "Backward-Recursive" approach.

4.4 Numerical Experiments with 2-stage Lines

A numerical analysis is carried out to show the performance of the 2S-Model. Section 4.4.1 assesses the computation time for the 2S-Model as the number of possible system states enhances. Model effectiveness is studied in section 4.4.2 and, starting from these results, in section 4.4.3 a sensitivity analysis on main problem parameters is performed.

In all the experiments the energy cost is $\sigma = 1$, the discount factor $\xi = 0.80$, and arrival rate $\lambda = 0.04$. The model is implemented with *Matlab R2020a*

Algorithm 2 The Backward-Recursive approach.

```
STEP 1: Couple the $m$ stages: $[S_{i-1}; S_i]$ with $i \in \{2, \dots, m\}$
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 $\pi^*_m$
STEP 3: Evaluate sub-system $[S_{m-1}; S_m]$ under $\pi^*_m$
  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 $\pi^*_i$
  Create the Markov chain of $[S_{i-1}; S_i]$
  Compute $\tilde{p} = P_{bl,i-1}$
end while
STEP 5: Combine obtained policies $\pi^*_i$ with $i \in \{2, \dots, m\}$ and find $\Pi^*$
```

and *ILOG CPLEX Optimization Studio 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.

4.4.1 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 4.3 shows the results of this analysis as confirmation that the computation time grows significantly when L increases. A detailed overview of the numerical results is in Table B.4 of Appendix B.

Time variability is really low, 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; on the other hand, 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 able to provide less than 5500 possible system states the 2S-Model nearly always leads to an exact solution in a short-medium amount of time.

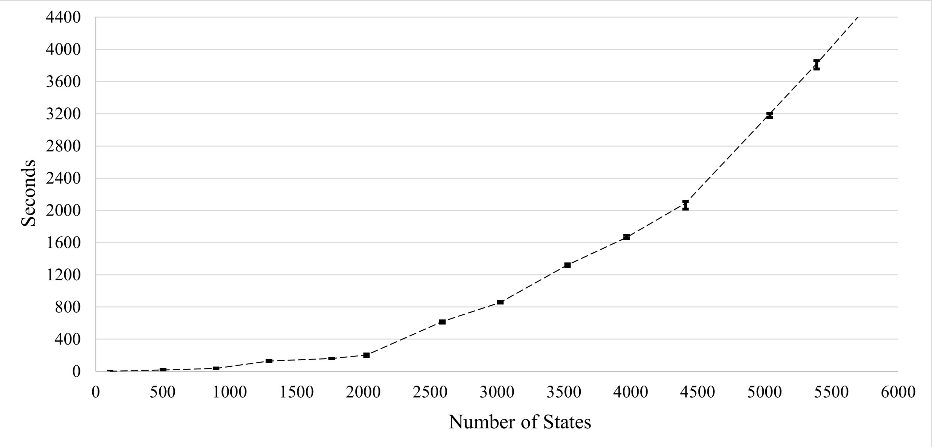


Figure 4.3: Time duration of the experiments vs number of system states.

4.4.2 2S-Model Effectiveness

Firstly, let us 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.

Let us 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 [96] with nine factors at two levels (Table 4.1) 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). The selection of the different levels is based on real-world industrial cases analyzed (see Section 3.3). In addition, a center point is added for each numerical factor, generating 4 additional experiments (Table 4.1).

In the center points, the value of each numerical factor corresponds to the

Table 4.1: Factors and levels for the 2^k factorial design 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

“center” value between the low and the high level in the 2^k factorial design (e.g. K has a center value equal to 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”. 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 (e.g. one replicate for the “Yes/Yes” combination, with all the other value equal to the “center” ones, and so on).

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 equal to the 90% of the maximum throughput achievable by the most saturated stage.

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 4.4 shows the percentages of energy saving obtained when the identified policy π^* differs from the *AOn* policy.

In 95 cases, the optimal policy is actually keeping the machines always

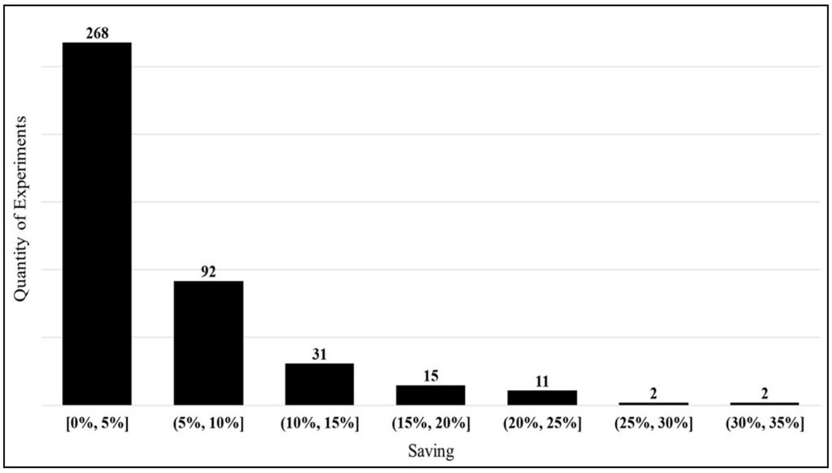


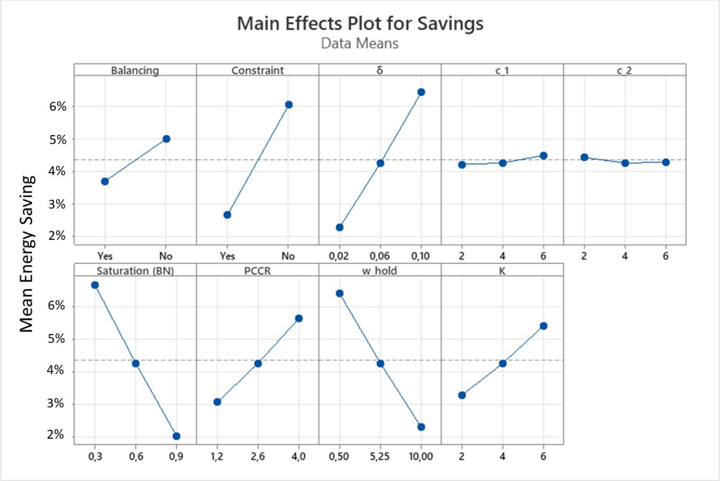
Figure 4.4: 2^k factorial analysis: energy saving when π^* differs from the AOn policy.

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 4.5 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 4.5 shows how the center points do not deny the linear effect on the results. A detailed overview of the numerical results is in Table B.5 of Appendix B.

4.4.3 Sensitivity Analysis

A sensitivity analysis is performed to detail the effect of the following significant factors (as from section 4.4.2): w_h, *PCR*, δ and K. A set of experiments is designed starting from four main configurations (Table 4.2), namely: *Best, Medium - 1, Medium - 2,* and *Worst*. The *Worst* configuration is the experiment from the 2^k factorial design of section 4.4.2 that obtains the lowest savings



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 4.5: 2^k factorial analysis: main effects plot with p-values for each factor.

(Figure 4.5) and, similarly, the *Best* configuration is the experiment obtaining the highest savings. Configurations Medium - 1 and *Medium - 2* are newly created to represent intermediate configurations.

Table 4.2: Base configurations for the sensitivity analysis.

Configuration	Balanced System	Constraint	δ	ρ	*PCR*	w_h	K
Best	No	No	0.10	0.30	4	0.5 (kW)	6
Medium - 2	No	No	0.05	0.70	2.50	1 (kW)	4
Medium - 1	Yes	No	0.03	0.80	2	4 (kW)	4
Worst	Yes	Yes	0.02	0.90	1.2	10 (kW)	6

For each configuration, the sensitivity analysis varies one significant factor at a time. Figure 4.6 shows the percentage of energy savings obtained for the evaluated cases. It must be noticed that for each combination of parameters starting from the *Worst* configuration, the obtained savings are 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 and, consequently, low opportunities to switch off one or more machines. A detailed overview of the numerical results is in Table B.6 of Appendix B.

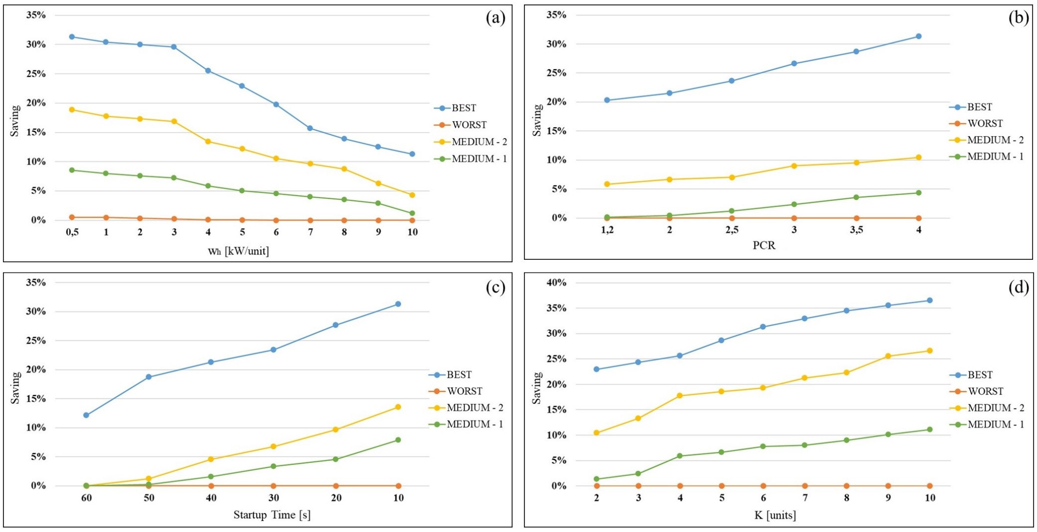


Figure 4.6: Sensitivity analysis when w_h (a), PCR (b), startup time (c), and K vary.

4.5 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. A complete overview of the design of experiments is presented in section 4.5.1 while results are discussed in section 4.5.2.

4.5.1 Design of Experiments

The experimental campaign is focused on two system layouts: a medium line composed of 5-stages and a long line with 15-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 4.3.

Table 4.3: Parametes of ws_A and ws_B.

Stage	K	c	δ	ρ	*PCR*	w_h
A (ws_A)	5	2	0.05	0.75	2.50	1 (kW)
B (ws_B)	5	2	0.02	0.90	1.20	3 (kW)

A-type stage is composed by 2 parallel machines with $w_{sb} = 0$, $w_{su} = 7.5$ kW, $w_{id} = 7$ kW $w_b = 12$ kW (i.e., $PCR = 2.5$) and the process rate is $\mu_a = 0.029$ given by the saturation level $\rho_A = 0.75$. B-type stage is composed by 2 parallel machines with $w_{sb} = 0$, $w_{su} = 9.5$ kW, $w_{id} = 1.5$ kW $w_b = 10$ kW (i.e., $PCR = 1.2$) and the process rate is $\mu_b = 0.022$ given by the saturation level $\rho_B = 0.9$. Parameters for ws_A and ws_B are selected considering insights from section 4.4 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 4.4, 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 4.4, 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.

In all the experiments the energy cost is $\sigma = 1$, 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 4.4: case (i) should lead to lower savings due to the balanced system analyzed while cases (ii-iii-iv) should produce higher savings. Simulation 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 [98].

4.5.2 Results

Tables 4.4 and 4.5 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 5-stages, this difference is below 1% both for thr

oughput loss and savings and this confirms the effectiveness of the proposed model in solving medium lines. On the other hand, in the 15-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 5-stage cases are included in a range going from about 3.50 to about 5% while for the 15-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 4.4: Results of 5-stage production line experiments.

Case	Stages Sequence	Throughput (AOn) [part/min]	En. Cons. (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%	$3.52 \pm 0.08\%$	3%	$3.11 \pm 0.05\%$
ii	B-B-A-A-A	2.17 ± 0.02	2.29 ± 0.02	4.43%	$4.57 \pm 0.07\%$	3%	$3.23 \pm 0.07\%$
iii	A-A-B-A-A	2.20 ± 0.02	2.34 ± 0.01	4.68%	$4.92 \pm 0.11\%$	3%	$3.42 \pm 0.08\%$
iv	A-A-A-B-B	2.16 ± 0.03	2.23 ± 0.01	4.99%	$5.11 \pm 0.10\%$	3%	$3.29 \pm 0.06\%$

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. As a consequence, the control problem actually consists in properly selecting two thresholds for each machine defined as n_{ij}^{off} and n_{ij}^{on} to respectively switch off and on M_{ij} in S_i. Secondly, the obtained EEC policies are *exhaustive*, i.e. the machine busy period is not interrupted by the control and each station produces until its upstream buffer is empty. Therefore, at least one machine is kept working until there is at least a part in the buffer, i.e., $n_{i1}^{off} = 0$.

Table 4.5: Results of 15-stage production line experiments.

Case	Stages Sequence	Throughput (AOn) [part/min]	En. Cons. (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%	$11.91 \pm 0.09\%$	3%	$3.96 \pm 0.09\%$
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%	$15.39 \pm 0.11\%$	3%	$4.15 \pm 0.11\%$
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%	$14.34 \pm 0.23\%$	3%	$4.43 \pm 0.08\%$
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%	$15.59 \pm 0.22\%$	3%	$4.89 \pm 0.10\%$

Numerical evidence shows that the control policy might be simplified without affecting the results. Threshold values for the analyzed cases are reported in Tables 4.6 and 4.7 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 5-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 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 $n_1 = 0$. In order to give a benchmark, the *AOn* policy would indicate to maintain always 2 machines switched on in each stage S_i; in this case, $n_{i1}^{on} = n_{i2}^{on} = 0$, while there is no n_{i1}^{off} and n_{i2}^{off} for both stages, since for any n_i value both machines must in working state in S_i. 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, as previously stated, $(n_j^{off}, n_j^{on}) = (-, 0)$ 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 and, for this reason, they cannot be switched off for energy saving purposes.

Lastly, regarding long production lines as the cases with 15-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.

4.6 Conclusions

This chapter proposes 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 con-

Table 4.6: Thresholds obtained for the 5-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 4.7: Thresholds obtained for the 15-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}	$(0,1)$ $(1,2)$	$(0,1)$ $(1,2)$	$(0,1)$ $(1,2)$	$(0,1)$ $(1,2)$

straints. The proposed method, exact for 2-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.

Furthermore, the model is also applicable when the line has workstations with a single machine, as it is a specific sub-case of the parallel-machines layout. 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 model are related to the assumptions regarding the system under control, especially the exclusive presence of Markovian processes, and the approximations introduced for long lines. Also, to apply this approach the system to be controlled must be fully observable, with steady and defined transition probabilities and system dynamics. An RL-based algorithm that is both general and adaptive has the potential to overcome the mentioned limitations and to adapt the control to non-stationary system dynamics without relying on complete knowledge of transition probabilities. This idea will be further explored and examined in detail in the subsequent chapters of the work.

Chapter 5

General Reinforcement Learning Model for EEC of Single Workstations

Chapters 3 and 4 explored the possibility of effectively apply EEC to single workstations and multi-stage production lines by exploiting MDP formulations. However, several constraints and limitation arose, as the exclusive presence of Markovian processes and the fully observability of the system to be controlled. Therefore, Chapter 5 explores the use of a general Reinforcement Learning framework for solving control problems in high stochasticity scenarios, with a focus on the energy-efficient control of single workstations composed of parallel machines. The objective is to develop a novel RL-based algorithm that can simultaneously reduce energy consumption and maintain system throughput, without relying on complete knowledge of system dynamics. This model aims at being a direct evolution of Chapter 3, where the most of the associated limitations are overcame.

The chapter is structured as follows. In Section 5.1 the system under investigation is described in detail. Section 5.2 provides a complete description of the proposed RL algorithm for EEC. Section 5.3 presents the numerical experiments carried out, showing the resulting benefits of the algorithm when applied to a real industrial case along with the numerical validation for general cases of parallel machines systems. Section 5.4 closes the chapter with the respective conclusions.

5.1 System Design

The system to be controlled is a workstation composed by a single upstream buffer of finite capacity serving multiple parallel machines (layout in Figure 5.1). The system is described by the assumptions presented in Section 5.1.1 and the state model described in Section 5.1.2.

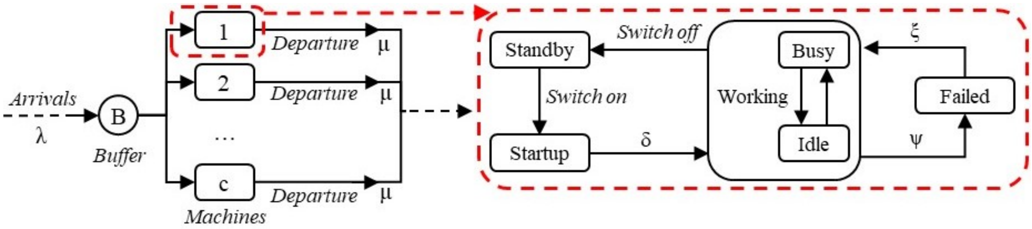


Figure 5.1: The system under analysis along with the machine state model.

5.1.1 Assumptions

The following assumptions are considered valid in this work and are introduced to describe the system to be controlled. The system is composed by a single workstation with c parallel machines and one shared upstream buffer B. The latter has a finite holding capacity $0 < K < \infty$. Furthermore, different stochastic processes characterize the system: arrival rate of parts to buffer B, machines processing and startup times, time between failures (*TBF*), and time to repair (*TTR*) of the machines. All of them are assumed to be independent of each other and stationary. In detail, machines startup and processing times follow stochastic process with expected values, respectively, equal to δ and μ; on the other hand, since machines are unreliable and can be subject to operation-dependent failures, they are characterized by stochastic *TBF* and *TTR* with expected values, respectively, equal to ψ and ξ. Lastly, arrival of parts to B follows a stochastic process with expected value λ and this process stops when B is full.

Further assumptions can be introduced describing the machines behavior: these are starved if they are ready to process parts but B is empty; on the other hand, they cannot be blocked since there is an infinite capacity buffer downstream the system, i.e. finished parts leave the system immediately after the process completion. This assumption simplifies the control problem since the blocking effect is not taken into account. Moreover, machines work on a single part type and first come first served rule is applied to extract parts from buffer B. Lastly, all the system devices cannot be switched off while operating on items, i.e. part processing cannot be interrupted by the control; also, the startup procedure cannot be interrupted by the control either.

5.1.2 Machine State Model

Machine state model is shown in Figure 5.1. Each machine in the system is characterized by the following states: *working*, *standby*, *startup*, and *failed* (w, sb, su, and f, respectively). Furthermore, *working* state is composed by two sub-states: *idle* (id) and *busy* (b). The machine state modelling is consistent with Section 3.1.1 and Table 3.1, but in this case failure are explicitly associated to the respective *failed* state. Indeed, from the *working* state, the machine can fall into the *failed* state. This transition is determined by the device stochastic *TBF* with expected value ψ. From the *failed* state, machines can go back into the *working* state when fully repaired. It is worth pointing out that a complete repair process encompasses the extreme scenario where the machine requires a complete shutdown before the actual repair. Subsequently, it must be turned back on and tested. If the repair is successful, i.e., the machine operates correctly, it will be restored to a functional state. The stochastic time-to-repair (TTR) of the machine governs this transition, with an expected value of ξ.

All the aforementioned states require a constant non-negative amount of power w_s, depending on the visited state or sub-state $s = \{w, sb, su, f, id, b\}$. It is assumed that when machine is *failed* it only requires a negligible power consumption to notify to the operators that a failure is going on: also w_f is assumed to be almost null and $w_b > w_{su} > w_{id} > w_{sb} \simeq w_f \simeq 0$ (consistently with Section 3.1.1).

5.2 Reinforcement Learning Modeling for EEC

This Section describes the proposed RL algorithm for EEC. The single workstation described in Section 5.1 is the environment in the proposed algorithm. For which concerns the RL agent modeling and implementation, its key features are introduced as follows: the action space in Section 5.2.1, the state space in Section 5.2.2, and the reward function in Section 5.2.3. For the design phase of agent components, the methodological approach of [103] is used.

5.2.1 Action Space Modeling

In the EEC approach, the action the agent has to perform is quite straightforward: it has to select, at any time, how many machines should be in working state in the workstation. In this way, starting from the actual number of working machines in the system, some machines might then be switched on or switched off, according to the agent action. The action space, i.e. the set of all possible actions the agent can implement, is therefore defined as $\mathbb{A} = \{0, 1, \ldots, c\}$, since the control action $a \in \mathbb{A}$ indicating the optimal number of working machines can only assume integer values ranging from 0 (all machines in the station must be switched off and not working) to c (all machines in the station must be switched on and working). The selected action space allows the agent to be able to change the environment.

Nevertheless, not all the actions are allowable at any moment. The admissibility of actions is indeed determined by system assumptions i.e. part processing and startup procedure cannot be interrupted by the control (Section 5.1.1). This means, for instance, that if a machine is not working because still executing startup procedure, the same machine cannot be directly set in working state by the agent until the startup procedure is not finished. At the same time, the admissibility of actions is indeed also determined by the system operating nature: if a machine is not working because in *failed* state and not fully repaired, it cannot be switched on until the repair procedure is not finished. Hence, the agent selects the action to perform also depending on its observation of the environment through the system state as presented in Section 5.2.2.

5.2.2 State Space Modeling

For the agent the state space $\mathbb{S}$ represents the set of possible agent observations of the environment. Hence, in the proposed algorithm, the system state is defined as **s** $\in \mathbb{S}$ and denoted by the ordered vector **s** $= \{\frac{n}{K}, x_1, \ldots, x_c\}$. Herein, n represents the number of parts in buffer B, so that $0 \leq \frac{n}{K} \leq 1$ is the variable giving information to the agent on how many parts are waiting to be processed by the workstation. Furthermore, each binary variable $x_i \in \{0, 1\}$ with $i \in \{1, \ldots, c\}$ provides information to the agent on the working state of each machine: $x_i = 1$ means machine i is in working state while if $x_i = 0$ machine i is not in working state. Thus, system state also provides information to the agent on the allowable actions.

The elements of the state vector are sufficient for the agent to select the optimal action and, at the same time, it is ensured that any change in the state resulting from the action-execution can be easily and immediately detected from the agent.

5.2.3 Reward Function Modeling

The reward function is the key element that directs the agents behavior to determine the optimal policy. EEC problems are characterized by two KPIs to be considered in the search of the optimal strategy: the energy consumption and the system throughput. The goal is to find the optimum trade-off between them and the optimization process is multi-objective.

The reward function in the proposed algorithm is therefore based on two non-negative and finite elements:

(i) The productivity partial-reward R_p, defined as:

$$R_p = \frac{\theta}{e} \tag{5.1}$$

with:

$$\theta = e^{\frac{TH_t}{TH_{Max,t}}} \tag{5.2}$$

and:

$$\begin{aligned} TH_t &= \frac{Nr.\ Produced\ Parts\ from\ 0\ to\ t}{t} \\ TH_{Max,t} &= \frac{Nr\ Produced\ Parts\ with\ AOn\ Policy\ from\ 0\ to\ t}{t} \end{aligned} \tag{5.3}$$

TH, t is the system throughput at the actual time t: the number of parts produced in the time period from the start of the simulation to the actual time t. $TH_{Max,t}$ is the maximum throughput achievable by the system in the same time period when all the machines are always switched on (i.e. with *AOn* policy applied). Since $0 \leq \frac{TH,t}{TH_{Max,t}} \leq 1$, then $0 \leq \theta \leq e$ and, consequently, $0 \leq R_p = \frac{\theta}{e} \leq 1$: the productivity partial-reward is a finite parameter always pertaining to the $[0, 1]$ interval. Moreover, R_p grows when TH, t gets closer to $TH_{Max,t}$. R_p guides the agent behavior to increase the productivity and maintain the machines more frequently in working state to produce more and more parts.

(ii) The consumption partial-reward R_c, defined as:

$$R_c = e^{-z\Delta_C} \tag{5.4}$$

with:

$$\Delta_C = C_t - C_{t-1} \tag{5.5}$$

and:

$$C_t = \sum_{i=1}^{c} \sum_{\mathbf{s} \in \mathbb{S}} w_s y_i, s(t) \tag{5.6}$$

$\Delta_C \geq 0$ is the consumption-growth, i.e. the difference between C_t and C_{t-1}. C_t is the overall energy consumption in the time period from the start of the simulation to the actual time t. In this time period, each machine i spent a certain amount of time in each of its states, this quantity is referred to $y_i, s(t)$, with $s = \{w, sb, su, f, id, b\}$. Therefore C_t is the overall sum, for each machine i , throughout all its states and sub-states, of the power requested in the respective state w_s multiplied by $y_i, s(t)$ (see Equation 5.6). Similarly holds for C_{t-1} at $t-1$, but in this case the consumption is measured in the time period from the start of the simulation to the time $t-1$. The latter is the time instant where the previous reward was given to the agent. Equation 5.6 is characterized by the presence of the scale factor z that is used to have $10^{-2} \leq z\Delta_C \leq 1$ and $0 \leq R_c = e^{-z\Delta_C} \leq 1$. In this way, also the consumption partial-reward is a finite parameter always pertaining to the $[0, 1]$ interval. R_c increases when Δ_C decreases. R_c guides the agent behavior to decrease the productivity and maintain the machines more frequently not in working state to produce less parts and save more energy.

It must be noted that for both R_p and R_c the exponential function is used because it allows an increasing gradient towards the optimal point and facilitates the learning process of the RL algorithm [7].

Once R_p and R_c are introduced, the reward function R can be defined as:

$$R = \phi R_p + (1-\phi)R_c \tag{5.7}$$

$\phi \in [0, 1]$ is a hyper-parameter defined in this work as the *multi-objective balancing factor* of the reward function. Since ϕ, R_p and R_c belong to the $[0, 1]$ interval, also $R \in [0, 1]$ and the agent goal will be to obtain the maximum achievable R, i.e. $R = 1$.

ϕ represents a crucial element for the proposed algorithm. When $\phi = 1$, the weight of R_c in the reward function is null, i.e. the only objective is to maximize R_p: the agent will implement actions only functional to the maximization of the productivity without taking into consideration the energy consumption aspect. On the opposite side, when $\phi = 0$ the weight of R_p in the reward function is null: the agent will not care about the system productivity and will implement actions only functional to the minimization of the energy consumption. Thus, when $\phi = 0$ or $\phi = 1$ the EEC optimization problem is no longer multi-objective; but, when $0 < \phi < 1$ the problem maintains its multi-objective nature. Finally, ϕ determines how “important" the productivity and the energy consumption in the optimization process are: the more ϕ grows the more the production rate becomes predominant on the consumption aspect. The calibration of ϕ becomes therefore a crucial aspect for the proposed RL algorithm and must be considered with great attention. At the same time, being the goal of any EEC problem the reduction of the energy consumption but without jeopardizing the production rate, it is expected that the optimal ϕ will always have a high value and closer to 1, leading to a major weight for R_p in the reward evaluation and consequent null or almost-null productivity drop.

5.3 Numerical Experiments

Different experiments are performed with the proposed RL-based model for EEC to evaluate its performance. The initial goal is to find the most suitable algorithm configuration able to effectively apply EEC to a real industrial case of workstation with parallel machines (system described in Section 5.3.1). For this layout, the most suitable agent-type to optimally apply the RL for EEC is identified in Section 5.3.2 and the optimal ϕ is detected in Section 5.3.3 with a calibration process. Subsequently, the goal becomes the validation of the proposed RL framework for the same industrial case but when general stochastic distributions are involved (Section 5.3.4). Lastly, in Section 5.3.5, the algorithm is also validated assessing the resulting benefits when applied on general cases of parallel machines manufacturing workstations.

The experiments utilized as environment the system described in Section 5.1 for which a discrete event simulator is built-up using the *SimPy* Python library. On the other hand, the agent has features described in Section 5.2 and is implemented through the *TensorForce* Python library [103]. They interacting through Python environment. A flow chart for the implemented RL algorithm used for all the numerical experiments is shown in Figure 5.2.

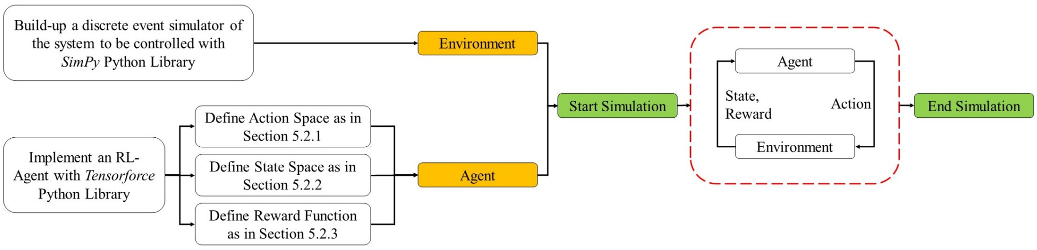

Build-up a discrete event simulator of the system to be controlled with *SimPy* Python Library
Environment
Implement an RL-Agent with *Tensorforce* Python Library
Define Action Space as in Section 5.2.1
Define State Space as in Section 5.2.2
Define Reward Function as in Section 5.2.3
Agent
Start Simulation
Agent
State, Reward
Action
Environment
End Simulation

Figure 5.2: Flow chart of the proposed RL algorithm implemented for the experimental campaign.

All the experiments are characterized by the evaluation of two KPIs: the throughput loss and the energy saving. For each analyzed case, the former is evaluated as the difference between the system throughput when the *AOn* policy is applied and when the RL-based model for EEC is implemented. Similarly, the energy saving is the difference between the overall system energy consumption when the *AOn* policy is applied and the same indicator when the RL for EEC is implemented. Discrete event simulation is used for performance evaluation: for each case 10 replications of the experiment are carried out and each of them has a simulation length of 30 days. All the experiments are characterized by independent random generation of numbers. The same warm-up period is imposed for each experiment, which is equal to 10 days of production period: this represents an overestimation, for computational-accuracy reasons, of the transient period identified with the Welch method ([98]). In all the experiments the scale factor z is imposed as equal to 10^{-4}. This choice is rather straightforward: being the consumption measured in kJ, either Δ_C is null or the absolute value of Δ_C is in the order of magnitude of several hundreds or few thousands, i.e. $10^2 \leq \Delta_C \leq 10^4$. Consequently, with $z = 10^{-4}$, $10^{-2} \leq z\Delta_C \leq 1$ and $0 \leq R_c = e^{-z\Delta_C} \leq 1$

All the reported KPI-values are extracted with a confidence level of 95% on the mean value. Experiments are performed in Python environment with a single laptop equipped with 4.90GHz i7 Intel Core and 16GB RAM.

5.3.1 Real Industrial System

The real industrial case under analysis is consistent with the modeling and the assumptions of Section 5.1. It is the manufacturing system producing cylinder heads in the automotive sector described in Section 3.3. The production line is composed by 20 total workstations and the one analyzed is Op.2 used for machining operations. The characteristics of Op.2 are reported in the following. In the workstation there are 6 parallel machines ($c = 6$) and one upstream buffer with finite capacity equal to 10 ($K = 10$). All the stochastic processes characterizing the workstation are Poisson processes: arrival rate of parts to buffer, machines processing times, startup times, *TBF* and *TTR* are exponentially distributed with expected values as in the following. Each machine is characterized by: mean processing time of 83.70 seconds ($\mu = 0.012$), mean startup time of 30 seconds ($\delta = 0.033$), *MTBF* of 3600 seconds ($\psi = 0.001$), and *MTTR* of 30 seconds ($\xi = 0.033$). The failure-related values describe a recurrent micro-stoppage for the machines required for a set-up procedure to be executed hourly on the device. A part arrive to buffer B, on average, each 20 seconds ($\lambda = 0.050$). Each machine, in working state is characterized by w_b and w_{id} equal to, respectively 15 and 9.30 kW; the startup state has an associated power consumption w_{su} of 10 kW and, lastly, both the standby and the failed state generate null power request $w_{sb} = w_f = 0$ kW. An overview on these system parameters, provided by the company owning the industrial system under study, are reported in Table 5.1.

A discrete event simulator is built-up for this system and the latter is validated by comparing its performance with data coming from the real industrial system.

Table 5.1: System parameters for the industrial case under investigation.

Parameter	c	K	μ	δ	ψ	ξ
Value	6	10	0.012	0.033	0.001	0.033
Parameter	λ	w_b	w_{id}	w_{su}	w_{sb}	w_f
Value	0.050	15 kW	9.30 kW	10 kW	0 kW	0 kW

5.3.2 Agent Selection

Four common types of RL agents are compared to assess which offers the best performance when applying the EEC to the industrial use-case:

1. The Tensorforce-General agent: the default agent of the Tensorforce library. It acts in accordance to a policy parametrized by a neural network, leveraging periodic updates based on batches of experience [104].
2. The TRPO agent [56].
3. The PPO agent [57].
4. The DQN agent [58].

See Section 2.2 for further details on TRPO, PPO, and DQN agents. All agents have a design consistent with the modeling described in Section 5.2. In each case, ϕ is varied between 0 and 1 and both throughput loss and the energy saving are evaluated. Results are shown in Figure 5.3. A detailed overview of the numerical results is in Table B.7 of Appendix B.

As expected, when $\phi = 0$ all the agents maintain all the machines always switched off provoking null throughput and a consequent 100% of throughput loss but also 100% of saving in respect to the *AOn* policy. Then, when ϕ increases, also the productivity increases leading to a decreasing throughput loss but also to a decreasing saving. Finally, when $\phi = 1$, all the agents maintain all the machines always switched on, i.e. they apply the *AOn* policy, leading to a null throughput loss but also to a null saving in respect to the *AOn* itself.

Among all agents, the DQN appears to be the most performing one since for decreasing throughput loss is able to guarantee the highest energy savings, and this occurs especially for high ϕ-values ($\phi \geq 0.75$). For all the agents, the default *TensorForce* library hyperparameters are used [103] (e.g. learning rate $\alpha = 0.001$, batch size $= 34$). Note that the learning rate $\alpha = 0.001$ was varied with limited effect. Literature shows that the agent type and reward, action, and state formulation have a significantly higher impact in production control [82, 76, 77]. Since the EEC goal is to reduce consumption while assuring low

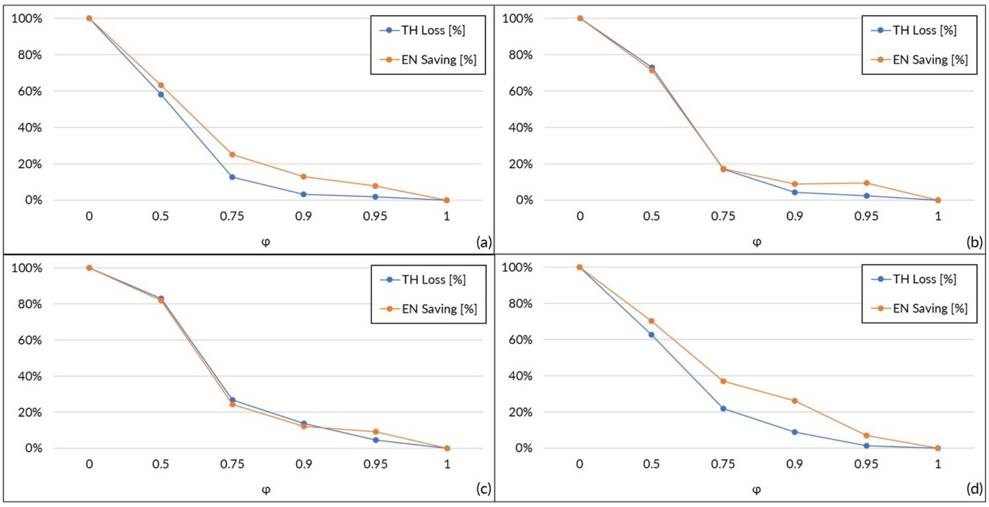


Figure 5.3: Comparison of different agents applied the industrial case: (a) Tensorforce-General, (b) PPO, (c) TRPO, (d) DQN. Mean values have a confidence level of 95% on the mean value; however, being all the confidence intervals strict, i.e. width lower than 2%, they are not visible with the selected figure scale.

productivity drops, the DQN with default *TensorForce* library hyperparameters is selected as the most suitable agent for the EEC problem to be addressed.

5.3.3 ϕ Calibration

Once the DQN agent has been selected, the optimal value of ϕ must be found. In accordance to what is stated in Section 5.2.3, the calibration process is executed by considering only high values of ϕ: from $\phi = 0.93$ to $\phi = 1$ with a step of 0.01. Results of this analysis are shown in Figure 5.4. A detailed overview of the numerical results is in Table B.8 of Appendix B.

The range of values in the $[0.95, 0.97]$ interval appears to be the most performing ones: they ensure a slight productivity drop of around 1% but at the same time the energy saving is remarkable and almost equal to 8%. In particular, the optimal ϕ is selected equal to 0.97 since it generates a saving of $7.72 \pm 0.10\%$ and a slight throughput loss of $1.20 \pm 0.05\%$. Both values are considered promising and significant from an EEC point of view, strongly improving the sustainability of the workstation without jeopardizing its productivity.

It must be noted that if the company owning the manufacturing workstation would prefer to avoid also the slight throughput loss and produce that $1.20 \pm 0.05\%$ of *lost* parts during extra working hours, the workstation would be subject to additional energy consumption. In this case, the corresponding savings will be decreased to, on average, $6.18 \pm 0.04\%$ but with a corre-

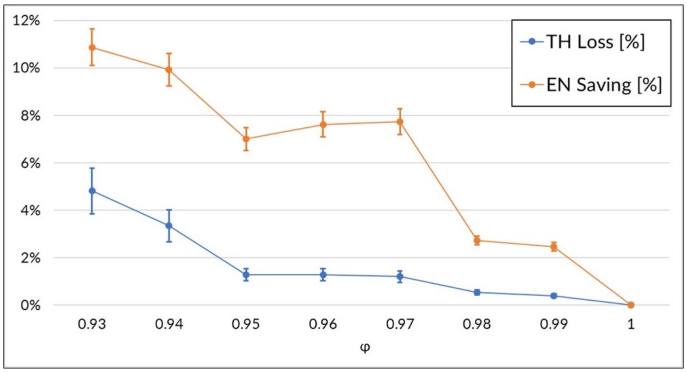


Figure 5.4: Comparison of different ϕ-values with DQN agent applied to the industrial case. Values are shown considering a confidence level of 95% on the mean value.

sponding null throughput loss. This managerial choice will anyway improve the workstation sustainability without any productivity drop.

It is interesting to understand how the RL modifies the operation of the workstation under analysis. By considering the best-identified solution, i.e. DQN agent with $\phi = 0.97$, Figure 5.5 shows the percentages of time that the controlled workstation spends with a certain number of switched on (working) machines. Trivially, when the *AOn* policy is applied all 6 machines in the workstations are always working. When RL algorithm for EEC is applied, instead, the amount of time spent with 6 working machines is reduced while for long periods the workstation operates with 3, 4 or 5 working machines and, for small periods, it works with only 2, 1 or 0 switched on machines.

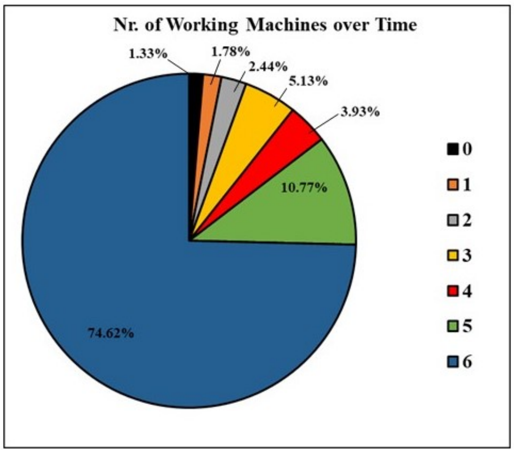


Figure 5.5: Percentages of time in which a specific number of machines are working (switched on) in the industrial case under analysis.

Figure 5.6 shows the reward received by the agent along the training episodes (5 replications while applying DQN agent with $\phi = 0.97$). The received reward reaches in a short amount of training episodes (less than 3000) a really high value close to 1, i.e. the maximum value of the reward function. This value corresponds to less than 2 hours of simulated production time. This indicates a fast convergence of the RL agent in determining and applying the optimal policy to the environment, since the agent only needs a short period to learn how to maximize the reward and optimally control the system. In this way, by using a DQN agent with $\phi = 0.97$ the design and implementation of a suitable and effective in short time RL-based algorithm for EEC for the industrial case under analysis are obtained.

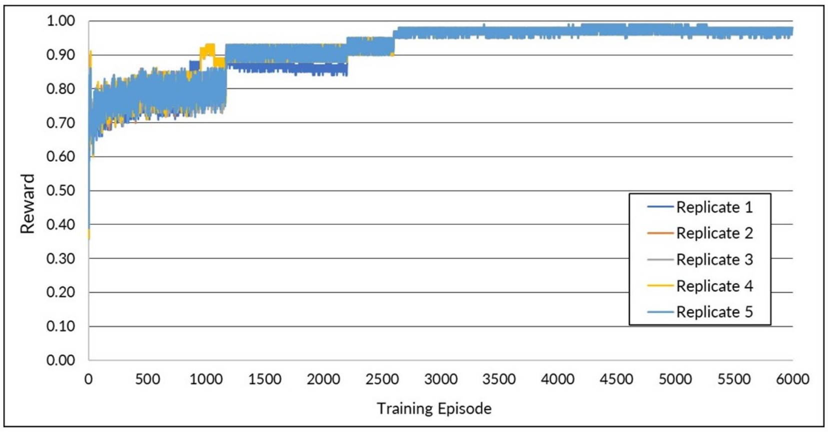


Figure 5.6: Reward trend for the industrial case with DQN agent applied and selected $\phi = 0.97$. High reward is reached in short time.

Being system dynamics of this numerical case characterized by only Poisson processes, it is possible to apply a special and reduced sub-version of the 2S-Model (Section 4.2), imposing $m = 1$ and, through LP, a constraint on the throughput loss being not higher than 1.15%, i.e. the lower boundary of the solution with DQN agent and $\phi = 0.97$. The constrained optimal solution for the EEC can be determined with the MDP-based model, corresponding to an energy saving of 8.86%. This value is slightly higher than the one obtained with DQN agent and $\phi = 0.97$ but it proves that the solution provided by the RL framework is really close to the real optimum obtained with an exact mathematical formalization. This proves the accuracy of the solution provided with the proposed RL algorithm for EEC. It must be noted that it is really rare to be able to have an exact mathematical definition of the system dynamics, and, therefore, dealing with unknown system dynamics is often required as in the RL approach such as through closely investigating the exploration-exploitation dilemma. However, in this regarded industrial scenario the very good performance and already fast conversion as shown in Figure 5.6 do not necessitate a thorough investigation. With this analysis, however, also the accuracy of the proposed algorithm is confirmed.

A last interesting analysis on the industrial case can be done by comparing the RL algorithm performance with an EEC *time-based* policy. The latter is defined as follows: switch off the machine after a time interval τ_{off} has elapsed from the last departure due to processor completion and, then, switch on back the machine after a time interval τ_{on} has elapsed from the last departure, i.e., when τ_{on}-τ_{off} has elapsed from the switch-off command, or when a part arrives [17]. *Time-based* policies are commonly used in machine tools [31]. In literature, it is demonstrated that, in the special case of exponentially distributed arrivals as the analyzed industrial case, the optimal control is immediately switching off the machine as soon as the departure occurs, and switching it on only when the next part arrives (i.e. $\tau_{off} = 0, \tau_{on} \rightarrow \infty$) [17]. These results are valid for a single-buffer-single-machine layout but can be applied also to the analyzed industrial parallel machines case with some approximation level. Applying the optimal EEC *time-based* policy to each machine in the real workstation case, the resulting saving with respect to the *AOn* policy is equal to 5.72 ± 0.23% and the throughput loss is 3.34 ± 0.14%. Hence, the performance is less effective than the one of DQN agent and $\phi = 0.97$ and, therefore, RL is proven to be more effective for this case than the optimal EEC *time-based* policy.

5.3.4 Validation for General Stochastic Distributions

A further analysis is carried out considering the same industrial case but with general stochastic distributions, in order to verify and extend the effectiveness of the proposed algorithm. In detail, the stochastic processes involved in the system are modified in this analysis: machines processing and startup times, along with *TBF* and *TTR* values are imposed following lognormal distributions with parameters in Table 5.2.

Table 5.2: Lognormal parameters for the general stochastic distribution analysis to be applied to the industrial case.

Parameter	μ	σ^2	Expected Value [s]	c_v
Processing Time	3.79	1.25	83.70	0.50
Startup Time	2.77	1.25	30	0.50
MTBF	2.77	1.25	3600	0.50
MTTR	7.56	1.25	30	0.50

All the considered distributions lead to the same coefficient of variation c_v (0.50); furthermore, all the mean values are the same as in Section 5.3.1. In this way, it is possible to compare two cases with aligned parameters where the only difference is represented by the stochastic distributions involved. The lognormal distribution is selected because it is always statistically plausible to model a stochastic time distribution with a lognormal if its coefficient of variation is lower than 1 [97]. Because of this property, the lognormal distribution allows a system modeling undoubtedly aligned to real manufacturing systems. All the other system parameters are unchanged in respect to Section 5.3.1. A DQN agent is used and ϕ is selected equal to 0.97 as in Section 5.3.3. Under these conditions, the resulting average energy saving and the throughput loss are equal, respectively, to 7.14 ± 0.21% and 2.14 ± 0.12% in comparison to the same case with *AOn* policy applied. Results are slightly worse than the original case analyzed in Section 5.3.3 (saving equal to 7.72 ± 0.10% and throughput loss equal to 1.20 ± 0.05%) and this is caused by the different stochastic distribution involved; however, the computed energy saving and the throughput loss still represent significant values. A complete overview on this analysis is presented in Table 5.3. This analysis confirms the effectiveness of the proposed RL algorithm for EEC also when general stochastic distributions are involved.

Table 5.3: Comparison of results when lognormal distributions are considered for the industrial case.

Distribution	Agent	ϕ	EN. Saving [%]	TH. Loss [%]
Exponential	DQN	0.97	1.20 ± 0.05	7.72 ± 0.10
Lognormal	DQN	0.97	2.14 ± 0.12	7.14 ± 0.21

5.3.5 2^k Factorial Analysis

A last numerical analysis is carried out to show the validity of the proposed RL-based algorithm for general cases of parallel machines manufacturing systems. A 2^k factorial design with a center point [96] with five factors at two levels is used to generate 33 different system configurations where the proposed RL algorithm has been applied. In all the experiments the DQN agent is used with ϕ equal to 0.97; moreover λ is set as fixed and equal to 0.05, while $w_{sb} = w_f = 0$ kW. The first three factors are: the startup expected value δ, the buffer capacity K, and the numbers of machines c. The fourth factor is the station saturation level ρ (see Section 4.4.1 for the definition). Therefore, μ is computed for each experiment given λ and the factors c, and ρ. The last factor is the *Power-Request Configuration Ratio* or *PCR* defined in Section 4.4.1. It is recalled that the *PCR* assesses machine energy saving potential: low *PCR* indicates power during the working states similar to that in EEC related states, i.e. lower saving potential, and for the opposite reason high *PCR* indicates major potential. An overview of factors and levels is provided in Table 5.4.

Table 5.4: Factors and levels for the 2^k factorial design along with the center point experiment value.

2^k **Factorial Design**					
Factor	K	δ	c	*PCCR*	ρ
Low Level	5	0.016	4	1.2	0.75
High Level	20	0.100	8	4	0.95
Center Point					
Values	12	0.058	6	2.6	0.85

Figure 5.7 shows the results of this analysis. In all the analyzed cases, the application of the proposed RL algorithm is able to provide a significant energy saving (from 4 to 10 %) with a correspondent throughput loss that varies from 0 to 4 %. It is worth noting that these values are rough and might be improved with a proper ϕ-calibration and/or different agent-type selection for each specific case. However, in all cases RL is able to impose EEC to the system in a proper way and, for this reason, the algorithm effectiveness is verified. A detailed overview of the numerical results is in Table B.9 of Appendix B.

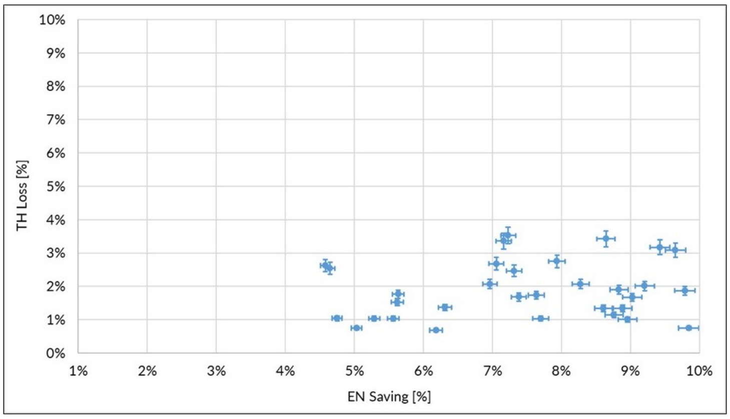
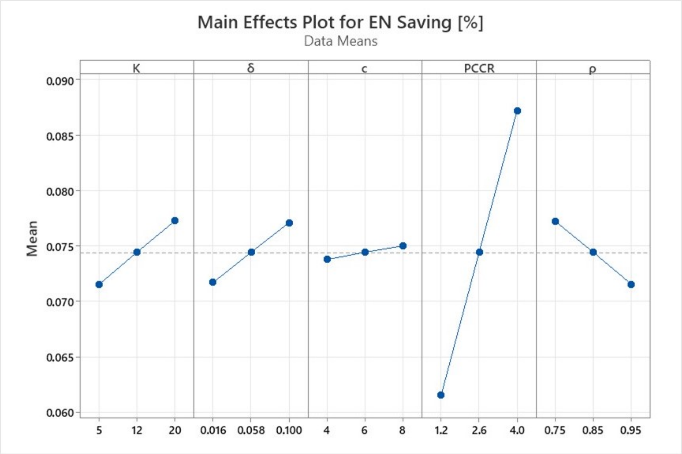


Figure 5.7: Results for the 2^k factorial analysis. Values are shown considering a confidence level of 95% on the mean value.

The main effects plot for the resulting energy saving is in Figure 5.8 with the Kruskall-Wallis test results to assess the significance of each factor. The high p-value for c indicates that this is not significant in this analysis. This means that it is possible to reduce the environmental impact of any parallel machine workstation regardless of whether it is composed by a large or small number of devices. Moreover, the main effect plot shows that higher savings can be achieved when: the startup expected value δ is high (i.e., the startup time is low), the saturation level is low, the buffer capacity K is high and, especially, the *PCR* is high. The last indication is consistent with the definition of *PCR* since high *PCR* leads to great energy saving potential.



Factor	Nr Samples	p-value
K	33	0.00
δ	33	0.00
c	33	0.55
PCCR	33	0.00
ρ	33	0.00

Figure 5.8: 2^k factorial analysis: main effects plot along with Kruskall-Wallis test p-values for each factor (results obtained with Minitab software).

5.4 Conclusion and Further Developments

This chapter presents a novel algorithm to reduce the environmental impact of single workstations composed by parallel machines with a common finite capacity buffer that are commonly used in manufacturing systems. The proposed approach exploits Reinforcement Learning techniques to effectively apply energy-efficient control actions to the regarded system. Being RL-based, the proposed model is general not limited by the assumption of having full knowledge about system dynamics and parameters: it adapts over time according to the varying system dynamics. Numerical results are presented and they confirm model validity and effectiveness for general cases of manufacturing systems and also when applied to a real industrial workstation operating in the automotive sector. Moreover, the carried out experiments indicate a fast convergence of the RL agent in determining and applying the optimal policy to the environment. Lastly, the accuracy of the proposed approach is verified by showing that the identified solution by the algorithm is really close to the real optimum obtained with an exact mathematical formalization of the energy-efficient control problem.

From a managerial point of view, the great potential of the approach along with the rising knowledge and user-friendliness of reinforcement learning techniques, might lead to a direct and successful application in industry in short-time and with low effort. The industrial application of the proposed method is also aided by the expansion and assimilation of Industry 4.0 Key Enabling Technologies by companies, and this may easily lead to improving industrial processes sustainability without jeopardizing system productivity.

The principal limitations of this model are related to the absence of focus on the workstation-interactions with the surrounding shop floor. In the developed algorithm, indeed, the agent observes an environment that is only a part of the overall production system and this leads to a lack of information about other machines or buffers in the line. The agent might implement actions that are optimal for the EEC of the controlled station but might not bring benefits for the other workstations. These conclusions have been the starting point leading to the last proposed model of this Thesis and discussed in Chapter 6.

Chapter 6

General Reinforcement Learning Model for EEC of Multi-Stage Production Lines

Chapter 4 showed the potential of the EEC approach for multi-stage production lines, but the associated results are limited by the exclusive presence of Markovian processes and the fully observability of the system under control. On the other hand, Chapter 5 highlighted the potential benefits of applying RL for the effective EEC of manufacturing equipment but was limited by the control of a single workstation and the consequent absence of focus on the workstation-interactions with the surrounding shop floor. Therefore, this chapter aims at overcoming the limitation of both mentioned chapters and to give a closure to the thesis work. The goal is to create a novel RL-based model capable of applying the EEC to manufacturing lines with multiple stages and parallel machine workstations. The model aims to reduce the energy consumption while maintaining system throughput, even without full knowledge of the actual system dynamics. The proposed algorithm can also be applied when workstations with a single machine are present, since they represent a sub-case of the parallel-machines category.

The chapter structure is the following. Section 6.1 describes the system to be controlled. Section 6.2 provides a complete description of the proposed RL-based algorithm. Section 6.3 presents the results of the numerical experiments carried out, demonstrating the benefits of the algorithm when applied to an industrial use-case and validated for general cases of multi-stage production lines. Conclusions are in Section 6.4.

6.1 System Design

The system to be controlled is a production line composed of m stages with parallel machine workstations (see Figure 6.1). It represents the environment in the proposed RL algorithm. It is possible to select how many stages must be controlled: this quantity is equal to y with $1 \leq y \leq m$. Each stage S_i with $i \in \{1, \ldots, m\}$, is characterized by an upstream buffer B_i with $i \in \{1, \ldots, m\}$, with a finite capacity $0 < K_i < \infty$, followed by c_i parallel machines M_{ij} with $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, c_i\}$. Machines M_{ij} are considered starved if they have the ability to process parts, but there are none available in B_i. Conversely, they are blocked if they can process parts, but B_{i+1} is already full. It is worth noting that machines $M_{ij} \in S_m$ cannot be blocked, as there is a downstream infinite capacity buffer that allows processed parts to exit the system immediately after they are completed. Another important detail is that all the machines in the line operate on a single type of part, blocking after service rule and first-come-first-served rule is enforced. Also, machines cannot be switched off while they are operating on items: part processing cannot be interrupted by the control. The system itself is characterized by different stochastic processes, including the arrival rate of parts to the first stage S_1, machines processing and startup times, and, also time between failures (TBF), and time to repair (TTR) of the machines. All of these are assumed to be independent of each other and stationary. The machines' startup and processing times follow a stochastic process, with expected values equal to δ_{ij} and μ_{ij}, respectively Additionally, the machines are unreliable and can be subject to operation-dependent failures. They are characterized by stochastic TBF and TTR, with expected values equal to ψ_{ij} and ξ_{ij}, respectively. All the mentioned expected values vary for each stage in the production line. Lastly, the arrival of parts to S_1 follows a stochastic process with an expected value of λ.

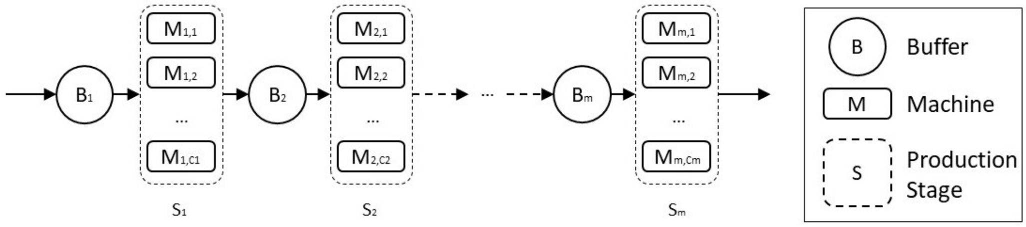


Figure 6.1: Layout of a multi-stage production line with parallel machines workstations.

Machine state model is consistent with Section 5.1.2 and Figure 5.1: each machine in is characterized by the following states: *working*, *standby*, *startup*, and *failed* (w, sb, su, and f, respectively). Furthermore, *working* state is composed by two sub-states: *idle* (id) and *busy* (b). Lastly, all the states require a constant non-negative amount of power w_s, depending on the visited state or sub-state $s = \{w, sb, su, f, id, b\}$ and it is assumed that $w_b > w_{su} > w_{id} > w_{sb} \simeq w_f \simeq 0$.

6.2 Reinforcement Learning Modeling

The elements of the proposed RL-based model are introduced in this section as follows: action space in Section 6.2.1, state space in Section 6.2.2, and reward function in Section 6.2.3. These elements compose the key features for the agent in the RL model, along with the environment described in section 6.1. The design phase is performed according to the approach of [103].

6.2.1 Action Space Modeling

The action space $\mathbb{A}$ is the set of all possible actions that an agent can take. It is important for the action space to be finite, so that the agent can explore all possible actions and learn to select the one that maximizes its reward. Let us define the vector $l = \{l_1, \ldots, l_z, \ldots, l_y\}$ of size y containing the indexes of the y stages to be controlled. Trivially, if $y = m$, then $l = \{l_1, \ldots, l_m\} = \{1, \ldots, m\}$. In the proposed model, the control action **a** $\in \mathbb{A}\}$ is applied to control the numbers of machines to be in working state in each of the y controllable stages. **a** is therefore a vector of size y defined as **a** $= \{a_{l_1}, \ldots,\ a_{l_z}, \ldots, a_{l_y}\}$ where a_{l_z} indicates the number of working machines to be present in the controllable stage $S_{i=l_z}$. Consequently, $0 \leq a_{l_z} \leq c_{i=l_z}$: it can only assume integer values ranging from 0 (all machines in the stage must be switched off and not working) to $c_{i=l_z}$ (all machines in the stage must be switched on and working).

For ease of understanding, let us use in Figure 6.2. Herein, $m = 3$, with $c_1 = 3, c_2 = 2$, and $c_3 = 4$. Let us assume that $y = 2$ stages are controllable, e.g. S_2 and S_3, leading to $l = \{l_1, l_2\} = \{2, 3\}$. The action vector is **a** $= [a_{l_1}, a_{l_2}] = [a_2, a_3]$, defining how many working machines there must be in stages $S_{i=l_1} = S_2$ and $S_{i=l_2} = S_3$. Therefore, $0 \leq a_2 \leq c_2 = 2$ and $0 \leq a_3 leq c_3 = 4$.

Consistently with Section 6.2.1, note that system assumptions dictate the allowable actions, such as not interrupting part processing or startup procedures. Therefore, the agent cannot immediately set a machine undergoing startup to a working state. Also, a failed machine cannot be turned on until fully repaired, as operational characteristics of the system determine the permissibility of actions.

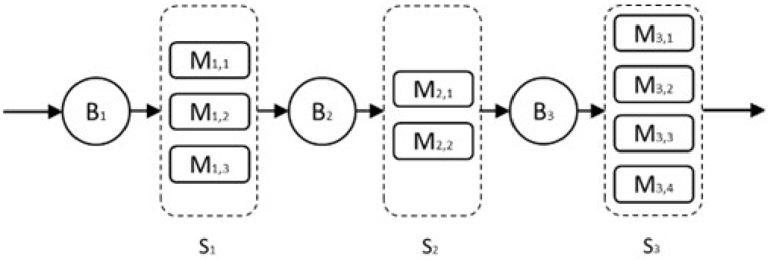


Figure 6.2: Example of a 3-stage manufacturing line where S_2 and S_3 are controllable.

6.2.2 State Space Modeling

In the RL framework, the state is the representation of the environment at a particular time, which includes all the relevant information necessary for the agent to make decisions about what action to take next. Consequently, the state space $\mathbb{S}$ represents the set of possible agent observations of the environment. In the proposed model the state **s** $\in \mathbb{S}$ is represented with the ordered vector of size $2m$: **s** $= \{\frac{n_1}{K_1}, \frac{x_1}{c_1}, \dots, \frac{n_m}{K_m}, \frac{x_m}{c_m}\}$. In **s**, $n_i \in \{0, \dots, K_i\}$ is the number of parts in buffer B_i while $x_i \in \{0, \dots, c_i\}$ represents the number of machines M_{ij} in working state in S_i. $x_i = 0$ indicates that in S_i no machine is working, $x_i = 1$ indicates that in S_i 1 machine is working and so on. Therefore, for each stage, the agent collects two information from **s**: (1) the number of parts present in B_i in respect to its capacity K_i, and, (2) the number of machines in working state in respect to the total number of machines c_i in the stage. Thus, $0 \leq \frac{n_1}{K_1} \leq 1$ and $0 \leq \frac{x_1}{c_1} \leq 1$. By doing so, the agent can quickly detect any state changes caused by executing an action.

6.2.3 Reward Function Modeling

The reward function is an essential component of RL, as it provides the feedback that guides the agent's learning process. Therefore it must be designed to reflect the goals of the task being learned. The EEC problem is characterized by a multi-objective nature since there are two goals, or Key Performance Indicators (KPIs), to be considered: reducing the energy consumption and maintaining the system throughput. Consistently with Chapter 5, the designed reward function is based on two elements:

(i) The productivity partial-reward R_p, defined as:

$$R_p = \frac{\theta}{e} \tag{6.1}$$

with:

$$\theta = e^{\frac{TH_t}{TH_{Max,t}}} \tag{6.2}$$

and:

$$\begin{aligned} TH_t &= \frac{Nr.\ Produced\ Parts\ at\ t}{t} \\ TH_{Max,t} &= \frac{Nr\ Produced\ Parts\ with\ AOn\ Policy\ at\ t}{t} \end{aligned} \tag{6.3}$$

It must be noted that the formulation of R_p is similar to Section 5.2.3, but in this case TH, t and $TH_{Max,t}$ are referred to the parts produced by the overall production system. Also, $0 \leq \frac{TH,t}{TH_{Max,t}} \leq 1$, then $0 \leq \theta \leq e$, and $0 \leq R_p = \frac{\theta}{e} \leq 1$. R_t grows as TH_t approaches $TH_{t,max}$. This means that, through R_p, the agent is directed to increase productivity by maintaining the machines in working state to produce a larger quantity of parts to increase R_t.

(ii) The consumption partial-reward R_c, defined as:

$$R_c = e^{-z\Delta_C} \tag{6.4}$$

with:

$$\Delta_C = C_t - C_{t-1} \tag{6.5}$$

and:

$$C_t = \sum_{i=1}^{c} \sum_{\mathbf{s} \in \mathbb{S}} w_s y_i, s(t) \tag{6.6}$$

where $\Delta_C \geq 0$ is the consumption-growth seen as the difference between: the overall energy consumption at the actual time t, i.e. C_t, and the overall energy consumption when the previous reward was given to the agent, i.e C_{t-1} at time $t-1$. Again, the formulation of R_c is similar to Section 5.2.3, but in this case the consumption values are referred to the overall production system. The computation of C_t and the role of the scale factor z are consistent with Section 5.2.3. Also, since $10^{-2} \leq z\Delta_C \leq 1$, then $0 \leq R_c = e^{-z\Delta_C} \leq 1$ where R_c grows as Δ_C

approaches 0. This means that, through R_c, the agent is directed to decrease productivity and save energy by maintaining the machines not in working state to produce fewer parts.

Lastly, for both R_p and R_c the exponential function is used because it allows an increasing gradient towards the optimal point and facilitates the learning process of the RL algorithm [7].

R_p and R_c compose the reward R as:

$$R = \phi R_p + (1 - \phi) R_c \tag{6.7}$$

Where $\phi \in [0, 1]$ is defined in Section 5.2.3. It is recalled that ϕ determines how “important" the productivity and the energy consumption in the optimization process are. Being ϕ, R_p and R_c in the $[0, 1]$ interval, also $R \in [0, 1]$ and the agent goal will be to obtain the maximum achievable R. Again, it is possible to affirm that the optimal ϕ will tend to 1, since this will lead to a null or almost-null productivity drop.

6.3 Numerical Experiments

The primary objective of the experimental study is to assess the effectiveness of the proposed RL for EEC model when applied to a real-world industrial system (see Section 6.3.1). In Section 6.3.2 different agent-types are compared to identify the most suitable for the use case while in Section 6.3.3, with the selected agent, ϕ is optimally calibrated for the production line under study. Section 6.3.4 focuses on validating the model by evaluating the benefits obtained when it is applied to multi-stage production lines with parallel machine workstations. Section 6.3.5 compares the results of applying the proposed model of this chapter with the resulting system performance where each controllable workstation is managed by a single agent, i.e. model proposed in Chapter 5.

The system described in Section 6.1 is the environment for all the experiments. A discrete event simulator has been developed using the *SimPy* Python library for this purpose. Meanwhile, the agent is equipped with features outlined in Section 5.2 and implemented using the *TensorForce* Python library [103]. Agent and environment interact through the Python environment. In all the experiments, two KPIs are evaluated: throughput loss and energy saving. Throughput loss is calculated as the difference between the system throughput when the *AOn* policy was applied and when the RL-based model was implemented. Similarly, energy saving is the difference between the overall system energy consumption when the *AOn* policy is applied and when the RL for EEC is implemented. For performance evaluation, discrete event simulation is employed, with 10 replications of the experiment carried out for each case, and each replication spanning a simulation length of 30 days. All experiments involve independent random number generation. To ensure consistency, the same warm-up period of 10 days of production period is imposed on each experiment. This duration represents an overestimation of the transient period identified with the Welch method ([98]), and is necessary for computational-accuracy reasons. In all the experiments the scale factor z is imposed as equal to 10^{-4}. All the reported KPI-values are extracted with a confidence level of 95% on the mean value. Experiments are performed in Python environment with a single laptop equipped with 4.90GHz i7 Intel Core and 16GB RAM.

6.3.1 Case Study Description

A real industrial system is used as reference case study where the proposed model can be applied and its effects can be analyzed. The production line under investigation is described in Section 3.3 and the layout is here reported again in Figure 6.3 to ease the readability. System parameters are in Table A.1 of Appendix A.

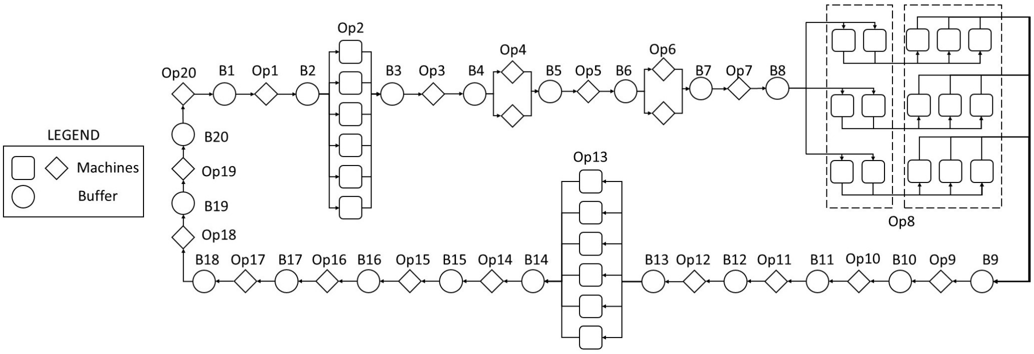


Figure 6.3: Layout of the industrial system under investigation.

The production process consists of 20 total operations performed in as many stages ($m = 20$). All the machines have stochastic processing times, TBF, and TTR. First-come-first-serve and blocking after service rules are applied. All the buffers have finite capacity. 5 stages are characterized by parallel machines ($c_i > 1$) while the remaining have a single-buffer-single-machine layout ($c_i = 1$). 12 stages are controllable ($y = 12$) since they are fully automated while 9 are not controllable due to the presence of human operators involved in the operation performed. In detail, $l = \{2, 3, 4, 7, 8, 9, 10, 12, 13, 14, 15, 19\}$. A discrete event simulator is built-up for this system and, subsequently, it is validated by comparing its performance with real data from the industrial system.

6.3.2 Agent Selection

To evaluate the most effective approach for applying the EEC in the real-world case, 4 commonly used types of RL agents are implemented and compared (results in Figure 6.4):

1. The Tensorforce-General agent [104].
2. The TRPO agent [56].
3. The PPO agent [57].
4. The DQN agent [58].

See Section 2.2 for further details on TRPO, PPO, and DQN agents.

To not jeopardize the throughput, only high values of ϕ ($\phi \geq 0.90$) are tested. Among all agents, PPO appears to be the best performing since at the same it is characterized by lower throughput loss and higher savings for high ϕ-values ($\phi \geq 0.95$). It must be also noted that when $\phi = 1$, all the agents maintain all the machines always switched on, i.e. they apply the *AOn* policy, leading to a null throughput loss but also to a null saving in respect to the AOn itself. For all the agents, the default TensorForce library hyperparameters are used (e.g. learning rate $\alpha = 0.001$, batch size $= 34$). The PPO with default *TensorForce* library hyperparameters is selected as the most suitable agent for the EEC problem to be addressed. A detailed overview of the numerical results is in Table B.1 of Appendix B.

6.3.3 ϕ Calibration

The subsequent step regarded the optimal calibration of ϕ for the real case when the PPO agent is applied. ϕ has been calibrated only for high values, varying it from 0.95 to 1 with a step of 0.01. Results of this analysis are shown in Figure 6.5. A detailed overview of the numerical results is in Table B.2 of Appendix B.

The optimal value appears to be $\phi = 0.98$, leading to a slight throughput loss ($0.85 \pm 0.10\%$) and a corresponding significant energy saving, equal to $5.79 \pm 0.44\%$. Considering the annual system performance, this would lead, on average, to a throughput decrease of 4472 units over a total production volume of more than 500000 parts, while saving more than 112 barrels of oil equivalent (or more than 690 GJ). It must be noted that, if the company owning the manufacturing line would prefer to avoid also the slight throughput loss and produce

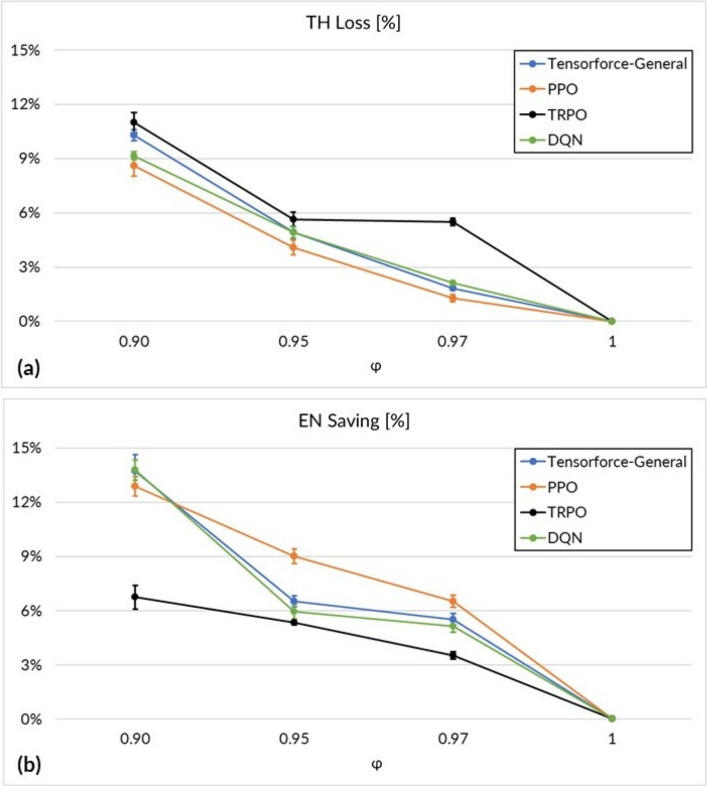


Figure 6.4: Comparison of different agents when applied to the industrial case: throughput loss (a) and energy saving (b) in respect to the AOn Policy. Mean values are shown considering a confidence level of 95% on the mean value.

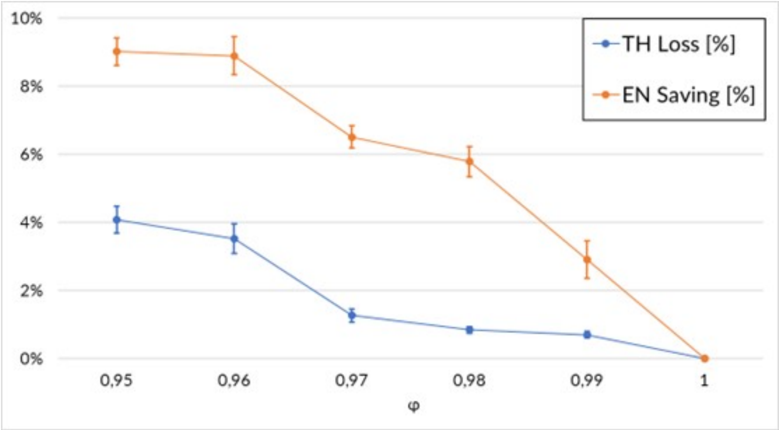


Figure 6.5: Energy saving and throughput loss when ϕ varies for the industrial case. A PPO agent is used. Values are shown considering a confidence level of 95% on the mean value.

that $0.85 \pm 0.10\%$ of lost parts during extra working hours, the system would be subject to additional energy consumption. In this case, the corresponding savings will be decreased, on average, to $5.11 \pm 0.14\%$ but with a corresponding null throughput loss. This managerial choice will anyway improve the system sustainability by saving, on average, more than 98 barrels of oil equivalent (or more than 603 GJ) but without any productivity drop. This confirms that, by using a PPO agent with $\phi = 0.98$, the successful implementation of the designed RL-based algorithm for EEC has been shown to significantly enhance the sustainability of the industrial use-case, while maintaining its productivity.

Figure 6.6 shows the reward received by the agent along the training episodes (5 replications while applying PPO agent with $\phi = 0.98$). The received reward reaches in a medium amount of training episodes (around 225000) a really high value close to 1, i.e. the maximum value of the reward function. This value corresponds to less than 6 days of simulated production time. This indicates a not-really fast convergence of the RL agent in determining and applying the optimal policy to the environment, since the agent needs about a one week period to learn how to maximize the reward and optimally control the system.

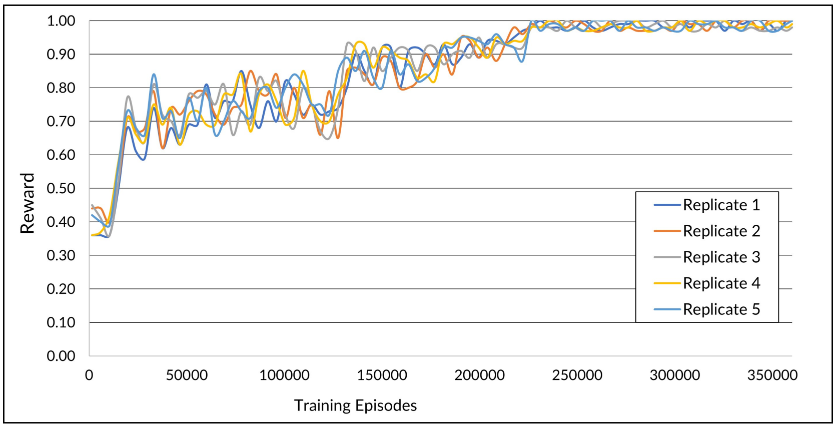


Figure 6.6: Reward trend for the industrial case with PPO agent applied and selected $\phi = 0.98$.

It is also possible to introduce a level of approximation and consider all the stochastic processes in the system as Markovian. In this case, it is possible to use the MDP-based model of Chapter 4 to the system under analysis. By imposing a constraint of maximum throughput loss equal to 0.23%, the resulting system KPIs in comparison to the *AOn* policy are: estimated throughput loss of $0.89 \pm 0.08\%$ (higher than the constraint due to the approximation introduced increases with the increasing number of stages, see Section 4.5) and an associated energy saving of $3.13 \pm 0.32\%$. The throughput loss is basically equal to the one obtained with the proposed RL-based model but the energy saving is lower. This proves that the effectiveness of the EEC approach for for multi-stage production lines is greater with a general RL-based framework than with an MDP-based model.

6.3.4 Validation for General Cases of Production Lines

The next step of the experimental campaign consists of a numerical analysis carried out to show the validity of the proposed model for general cases of multi-stage production lines with parallel machine workstations. Two types of layout are considered: in the first all the lines are composed of 5 stages ($m = 5$) while in the second they have 15 stages ($m = 5$). In both cases, all the stages are controllable: $y = m$. Also, the production lines are composed of stages of three types: A, B, and C (parameters in Table 6.1). The choice of the parameters is done accordingly to Chapter 4, where it is demonstrated that, for multi-stage production lines with parallel machine workstations, the EEC guarantees higher savings for stages with: high buffer capacity (high K), low saturation or fast processing time (high c and μ), fast startup time (high δ), and high power consumption (high w_s), Thus, it is expected that: stage-type A leads to the highest savings, C to the lowest and B to an intermediate situation.

Table 6.1: Parameters for the stage-types A, B, and C.

Stage Type	c	K	μ [1/s]	δ [1/s]	ψ [1/s]	x [1/s]	$[w_b, w_{su}, w_{id}, w_{sb}, w_f]$ [kW]
A	6	10	0.1	0.05	0.0001	0.02	$[15, 10, 12, 0, 0]$
B	4	5	0.03	0.02	0.0002	0.015	$[12, 7, 10, 0, 0]$
C	2	2	0.02	0.01	0.0003	0.010	$[10, 6, 8, 0, 0]$

In the 5 stages experiments, a full factorial design [96] with 5 factors at 3 levels is used to create 243 different manufacturing systems made up of workstations of type A, B, or C. The 5 factors are the 5 stages, with each stage having 3 possible levels: A, B, or C. Similarly, for the 15 stages, all the 14348907 possible layout combinations are generated. Among these, a *Latin Square Hypercube* [96], is used to select a subset of 25 cases from the 243 combinations of the 5 stages and another subset of 25 cases from the 14348907 of the 15 stages. The proposed model is therefore tested on a total of 50 production lines with varied parameters, 25 with 5 stages and 25 with 15 stages. In all the experiments, a PPO agent with $\phi = 0.98$ is used. Results are in Figures 6.7 and 6.8. A detailed overview of the numerical results for the 5 and 15 stages experiments is in Tables B.13 and B.14 of Appendix B.

In all the cases the proposed RL algorithm is able to provide a significant energy saving (from 3.5 to 12%) with a correspondent throughput loss that

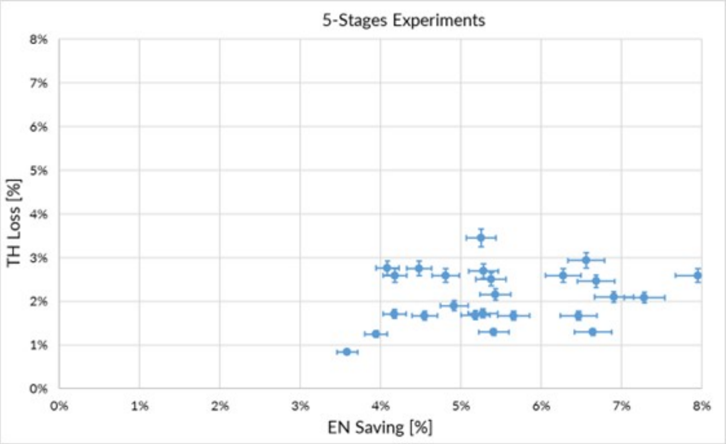


Figure 6.7: Results for the 5-stages experiments. Values are shown considering a confidence level of 95% on the mean value.

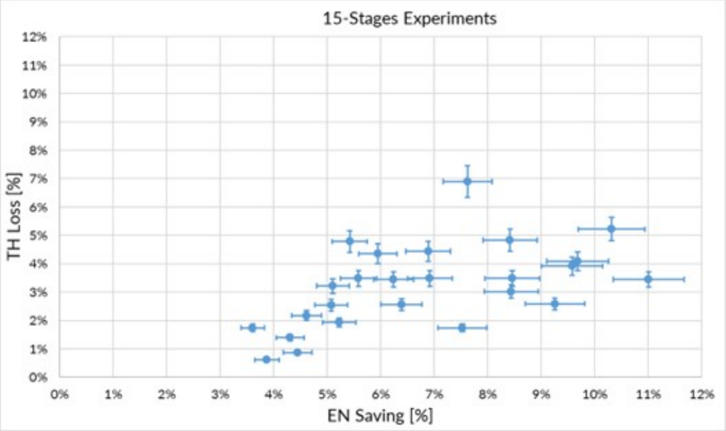


Figure 6.8: Results for the 15-stages experiments. Values are shown considering a confidence level of 95% on the mean value.

varies from 0 to 5% (only in 2 experiments exceeding this 5% threshold). It must be also noted that all the results are rough and might be improved with a proper calibration of ϕ and/or an alternative choice of the agent-type for each specific case. However, in all cases the proposed RL-based model is able to impose EEC to the system in a proper way and, for this reason, the algorithm effectiveness is verified.

6.3.5 Comparison with Multiple Single Agents

One last experimental analysis is worth carrying out: comparing the results obtained with a single agent controlling the overall production system with the system performance when more single agents control single workstations of the line. In particular, the benchmark is given by the results of Section 6.3.3, where the proposed model with a PPO agent and $\phi = 0.98$ led to a saving of $5.79 \pm 0.44\%$ and a throughput loss of $0.85 \pm 0.10\%$. In this case, the overall line, and in particular $y = 12$ stages of it, are controlled by a single RL agent. These results will be compared with the case where the RL-based model for the EEC of single workstations shown in Chapter 5 is applied independently to all the 12 controllable stages in the line. It means that, in this case, each of the 12 single agents is controlling only the respective workstation observing an environment that is only a part of the overall production system. A DQN agent is chosen, consistently with the results of Section 5.3.2 and ϕ is varied from 0.95 to 1 with a step of 0.01; in each experiment, the same ϕ is applied to all the 12 single agents. Results are shown in Figure 6.9. A detailed overview of the numerical results is in Table B.3 of Appendix B.

$\phi = 0.99$ appears to be the optimal value, since it guarantees the best trade-off: a significant energy saving of $4.91 \pm 0.17\%$ but with an associated remarkable throughput loss of $2.11 \pm 0.07\%$. It is clear how the solution with multiple agents is dominated from the one with one single agent controlling the overall line: the single agent leads to higher savings and lower productivity drop. This is confirmed with Figure 6.10, where it is porvied a visualization of how the single agent solution dominates the one with multiple agents. Having more agents taking control actions but without interacting with leads to a lack of information about other machines or buffers in the line and decreases the EEC effectiveness.

6.4 Conclusions

This chapter presents a novel model increase the energy-efficiency of manufacturing lines with multiple stages and parallel machine workstations, a layout commonly used in industry. The proposed method utilizes reinforcement learning strategies to efficiently implement energy-efficient control actions for

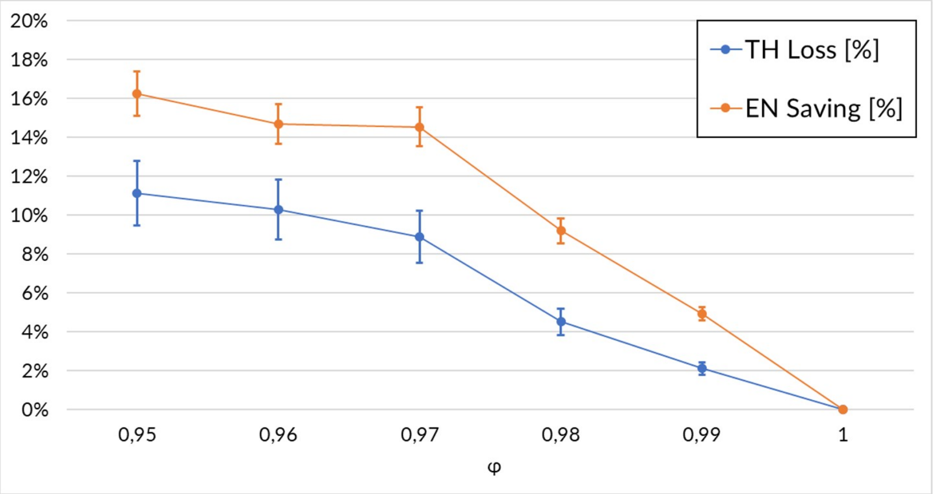


Figure 6.9: Comparison of different ϕ-values with 12 single DQN agents applied to the industrial case. Values are shown considering a confidence level of 95% on the mean value.

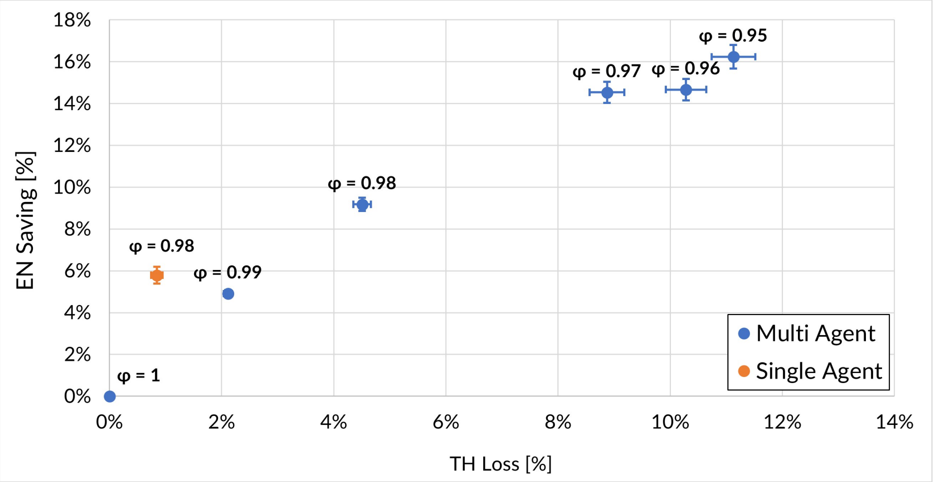


Figure 6.10: Comparison of different ϕ-values with 12 single DQN agents applied to the industrial case versus the best solution with a single agent. Values are shown considering a confidence level of 95% on the mean value.

the considered system. As it is RL-based, the proposed model does not rely on the assumption of complete knowledge of the system's dynamics and parameters; instead, it adapts and evolves over time, dealing with the varying system dynamics during the training process. Also, it takes control actions based on the observation of the overall production system, without having any lack of information about system operations and features.

The potential of the proposed approach, combined with the increasing ease of use and knowledge of reinforcement learning techniques, presents a promising opportunity for direct and successful application in industry. This application can be achieved in a short time and with minimal effort, making it appealing from a managerial standpoint. The validity and effectiveness of the model for manufacturing systems are confirmed by the presented numerical results, including real-world applications in the automotive sector. These results demonstrate the model's effectiveness for general cases and practical scenarios. Furthermore, it is also proven how a single agent controlling the overall production system is way more effective and accurate that multiple single agents controlling independently the single workstations. This, in turn, can improve the sustainability of industrial processes without negatively impacting system productivity.

Chapter 7

Conclusions

7.1 Research Impact

Industries around the world are increasingly pressured by customers, governments, and stakeholders to reduce their environmental impact and promote energy efficiency and sustainability. The manufacturing industry, in particular, has made significant strides in reducing its environmental impact through the adoption of renewable energy sources and more efficient production processes, resulting in both cost savings and improved efficiency. The goal of the thesis is to contribute to the research area of industrial systems planning, control and optimization by developing new models and methods for overcoming existing barriers to the application of effective energy-efficient control strategies to production systems.

The research work presented has focused on the development of novel RL-based models that could apply the switch off/on control to one or more manufacturing workstations, composed of single or parallel machines, while reducing the overall energy consumption and maintaining the system throughput. The four innovative models addressed the EEC problem with increasing complexity of the system to control and/or overcoming the assumption of having complete knowledge of system dynamics. All the proposed methods have been successfully tested on real industrial cases and validated for general cases and practical scenarios with significant and effective results.

From a scientific point of view, the RL-based algorithms developed in this research provide an adaptive and general EEC approach that can deal with unknown system dynamics and parameters in real-time from data, reducing the need for complete prior knowledge of the system. This can lead to significant time and cost savings in terms of system modeling and control. The proposed methods address effectively a multi-objective optimization problem: the simultaneous minimization of the energy consumption and productivity maximization must be ensured. They achieve this goal by incorporating the stochasticity of manufacturing systems, the limited control actions available, and the unknown system dynamics. Therefore this research overcomes the limitations of the existing EEC methods and improve the accuracy, effectiveness, and generality of the models.

Furthermore, another scientific impact is represented by the first-time application of a well-proven method, i.e. the MDP, to obtain an analytical model for a workstation composed of parallel machines. This provides an analytical tool that can serve as a reference point for future research on parallel machine workstations.

The industrial impact of this research is related to the great potential of the approach along with the rising knowledge and user-friendliness of reinforcement learning techniques, that might lead to a direct and successful application in industry in short-time and with low effort. The proposed RL models can help manufacturers to reduce their energy consumption and increase their competitiveness by optimizing the production process. This is achieved without compromising production rates or the quality of the product. Additionally, the models developed in this research are general and can be applied to various types of production systems, providing a framework for future research and development. Hence, the results of this research can help companies to reduce operational costs, enhance competitiveness, and contribute to achieving sustainability goals. Finally, this thesis demonstrates the effectiveness and potential of RL in industrial settings and open up new avenues for research and development in this field.

The results of this thesis have led to the following publications: [105, 106, 107, 108, 109, 110].

7.2 Limitations

While this thesis has achieved its primary objective of developing novel reinforcement learning models for energy-efficient control of manufacturing equipment, it is essential to acknowledge its limitations.

The first limitation of this study is that it focuses solely on the electrical energy usage of the machines and does not consider other elements such as compressed air and cooling liquids that are involved in the manufacturing operations. While the proposed switch on/off approach demonstrates the associated saving in electrical energy, it may also save other resources, leading to further improvements in the overall system energy-efficience.

Another limitation of this study is that the proposed RL models have been applied to systems characterized by stationary and unchanging dynamics. However, in real-world scenarios, the system dynamics may vary with time due to changes in operating conditions or maintenance schedules. It is necessary to investigate how reinforcement learning agents can adapt to these changes in system dynamics and learn new control strategies to maintain energy-efficient operation.

An additional limitation of the work is the lack of emphasis on the explainability aspects of the proposed RL models. Explainability in RL pertains to the ability to comprehend and interpret the decision-making process of an RL agent, which is often considered a black box, making it challenging to comprehend how the agent makes decisions. To establish trust in the agent's decision-making process and ensure that it aligns with the intended goals, it is crucial to develop methods for understanding and interpreting the agent's decision-making process. Thus, a thorough examination of this issue would offer better insights into RL actions for the EEC task.

A last limitation of this study is that it uses a centralized RL approach, where a single agent controls the entire production line or multiple agents control locally the single workstations but without interacting. In contrast, many real-world systems consist of multiple agents that interact with each other to achieve a common goal. Furthermore, distributed approaches can provide greater resilience and fault tolerance. In a centralized approach, a failure in the central agent can result in the entire system failing. However, in a distributed approach, each agent can continue operating independently, even if other agents fail or malfunction. This also allows for easier scaling of the system, as new agents can be added as needed without requiring significant changes to the overall system architecture.

7.3 Further Developments

Future research should investigate the possibility of overcoming the aforementioned limitations. For instance, it should investigate the impact of energy-efficient control actions on the machine overall energy consumption, including other resources, to develop more comprehensive energy-efficient control strategies.

Moreover, future research should explore the use of distributed multi-agent RL approaches, where multiple agents control the system locally and communicate with each other to achieve global objectives. This can lead to more efficient control strategies, as each agent can specialize in a particular task, leading to a more optimized overall control strategy. In a distributed approach, each agent controls a specific subset of the system and communicates with other agents to coordinate their actions. This allows for more parallelization and reduces the amount of information that needs to be processed by each agent, potentially leading to faster and more efficient decision-making. Overall, while the centralized RL approach used in this thesis provides a useful starting point, future research could explore the potential benefits of distributed multi-agent RL approaches for energy-efficient control in manufacturing systems.

Future research should analyze deeply the explainability of the proposed RL-based approaches and develop methods for understanding and interpreting the decision-making process of the RL agent. This can involve techniques such as visualizing the agent's policy or value function, identifying important features or states that the agent relies on to make decisions, or analyzing the impact of different factors on the agent's behavior. Through these methods, insights into the decision-making process of the RL agent can be provided, helping to understand reasons behind the agent decisions and enabling the identification of any potential issues and addressing them accordingly.

Furthermore, exploring the potential benefits of integrating RL techniques for both optimizing the energy consumption of machines and accomplishing other tasks could be valuable. As an example, RL can be utilized in a combined application of predictive maintenance and EEC. In this case, the agent can predict the potential failure of the equipment, enabling the companies to schedule maintenance during the most favorable time to minimize energy costs and downtime. This integration of actions among EEC policies can lead to more efficient and sustainable manufacturing processes. Other examples regard the simultaneous EEC of machines and energy use optimization in the manufacturing plant: the algorithm can learn to adjust temperature, lighting, and other settings in real-time to minimize energy consumption while still maintaining a comfortable environment. Finally, RL might be used for joint EEC and demand response management: the agent can optimize a company energy usage in response to changes in demand. For example, the algorithm can learn to shift energy-intensive tasks to times when electricity is less expensive, or to reduce usage during peak demand periods.

Bibliography

[1] Bipartisan Policy Center. Annual energy outlook 2022. *Energy Information Administration, Washington, DC*, 2020.

[2] Alperen Can, Gregor Thiele, Jörg Krüger, Jessica Fisch, and Carsten Klemm. 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, 2019.

[3] Luoke Hu, Chen Peng, Steve Evans, Tao Peng, Ying Liu, Renzhong Tang, and Ashutosh Tiwari. Minimising the machining energy consumption of a machine tool by sequencing the features of a part. *Energy*, 121:292–305, 2017.

[4] Jeffrey B Dahmus and Timothy G Gutowski. An environmental analysis of machining. In *ASME international mechanical engineering congress and exposition*, volume 47136, pages 643–652, 2004.

[5] Davide Anghinolfi, Massimo Paolucci, and Roberto Ronco. A bi-objective heuristic approach for green identical parallel machine scheduling. *European Journal of Operational Research*, 289(2):416–434, 2021.

[6] Christian Gahm, Florian Denz, Martin Dirr, and Axel Tuma. Energy-efficient scheduling in manufacturing companies: A review and research framework. *European Journal of Operational Research*, 248(3):744–757, 2016.

[7] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[8] Paolo Renna and Sergio Materi. A literature review of energy efficiency and sustainability in manufacturing systems. *Applied Sciences*, 11(16):7366, 2021.

[9] Claudio Francesco Angelo Cornaggia. On-line algorithms for energy efficient control of machine tools: a step towards practical implementation. *Master Thesis, Politecnico di Milano*, 2020.

[10] Yves Dallery and Stanley B Gershwin. Manufacturing flow line systems: a review of models and analytical results. *Queueing systems*, 12:3–94, 1992.

[11] Gilles Mouzon, Mehmet B Yildirim, and Janet Twomey. Operational methods for minimization of energy consumption of manufacturing equipment. *International Journal of production research*, 45(18-19):4247–4271, 2007.

[12] Xueying Guo, Sheng Zhou, Zhisheng Niu, and PR Kumar. Optimal wake-up mechanism for single base station with sleep mode. In *Proceedings of the 2013 25th International Teletraffic Congress (ITC)*, pages 1–8. IEEE, 2013.

[13] Zhisheng Niu, Jianan Zhang, Xueying Guo, and Sheng Zhou. On energy-delay tradeoff in base station sleep mode operation. In *2012 IEEE International Conference on Communication Systems (ICCS)*, pages 235–239. IEEE, 2012.

[14] Wei Wei and Zheng Wang. Integrated optimal production and energy control of a single machine and single product-type manufacturing system. In *2016 13th International Workshop on Discrete Event Systems (WODES)*, pages 343–348. IEEE, 2016.

[15] Nicla Frigerio, George J. Shanthikumar, and Andrea Matta. Dynamic programming for energy state control of machine tools in manufacturing with part admission control. In *11th Conference on Stochastic Models of Manufacturing and Service Operations*, pages 85–92. ITA, 2017.

[16] Nicla Frigerio and Andrea Matta. Energy-efficient control strategies for machine tools with stochastic arrivals. *IEEE Transactions on Automation Science and Engineering*, 12(1):50–61, 2014.

[17] Nicla Frigerio and Andrea Matta. Energy efficient control strategy for machine tools with stochastic arrivals and time dependent warm-up. *Procedia CIRP*, 15:56–61, 2014.

[18] Nicla Frigerio and Andrea Matta. Modelling the startup of machine tools for energy efficient multi-sleep control policies. *Journal of Manufacturing Systems*, 60:337–349, 2021.

[19] Nicla Frigerio and Andrea Matta. 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, 2015.

[20] Lorenzo Marzano, Nicla Frigerio, and Andrea Matta. Energy efficient state control of machine tools: A time-based dynamic control policy. In *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, pages 596–601. IEEE, 2019.

[21] Nicla Frigerio, Claudio Francesco Angelo Cornaggia, and Andrea Matta. An adaptive policy for on-line energy-efficient control of machine tools under throughput constraint. *Journal of Cleaner Production*, 287:125367, 2021.

[22] Eliana Torres Duque, Zicheng Fei, Junfeng Wang, Shiqi Li, and Y.F. Li. Energy consumption control of one machine manufacturing system with stochastic arrivals based on fuzzy logic. In *2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 1503–1507. IEEE, 2018.

[23] Anshul Gandhi, Mor Harchol-Balter, and Ivo Adan. Server farms with setup costs. *Performance Evaluation*, 67(11):1123–1138, 2010.

[24] Anshul Gandhi, Mor Harchol-Balter, and Ivo Adan. Analysis of an M/M/k system with exponential setup times under staggered boot up. In *Madrid Conference on Queueing Theory*, volume 2, page 6, 2010.

[25] Isi Mitrani. Managing performance and power consumption in a server farm. *Annals of Operations Research*, 202(1):121–134, 2013.

[26] Xiuli Xu and Naishuo Tian. The M/M/c queue with (e, d) setup time. *Journal of Systems Science and Complexity*, 21(3):446–455, 2008.

[27] Vincent J Maccio and Douglas G Down. On optimal control for energy-aware queueing systems. In *2015 27th International Teletraffic Congress*, pages 98–106. IEEE, 2015.

[28] Vincent J Maccio and Douglas G Down. Exact analysis of energy-aware multiserver queueing systems with setup times. In *2016 IEEE 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 11–20. IEEE, 2016.

[29] Vincent J Maccio and Douglas G Down. Structural properties and exact analysis of energy-aware multiserver queueing systems with setup times. *Performance Evaluation*, 121:48–66, 2018.

[30] Vincent J Maccio and Douglas G Down. Asymptotic performance of energy-aware multiserver queueing systems with setup times. In *2018 Annual American Control Conference (ACC)*, pages 6266–6272. IEEE, 2018.

[31] Nicla Frigerio and Andrea Matta. Analysis of an energy oriented switching control of production lines. *Procedia CIRP*, 29:34–39, 2015.

[32] Huiting Su, Nicla Frigerio, and Andrea Matta. Energy saving opportunities and value of information: a trade-off in a production line. *Procedia CIRP*, 48:301–306, 2016.

[33] Guorong Chen, Liang Zhang, Jorge Arinez, and Stephan Biller. Feedback control of machine startup for energy-efficient manufacturing in bernoulli serial lines. In *2011 IEEE International Conference on Automation Science and Engineering*, pages 666–671. IEEE, 2011.

[34] Qing Chang, Guoxian Xiao, Stephan Biller, and Lin Li. Energy saving opportunity analysis of automotive serial production system. *IEEE Transactions on Automation Science and Engineering*, 10(2):334–342, 2012.

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

[36] Zeyi Sun and Lin Li. Dynamic energy control for energy efficiency improvement of sustainable manufacturing systems using markov decision process. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 43(5):1195–1205, 2013.

[37] Michael P Brundage, Qing Chang, Yang Li, Guoxian Xiao, and Jorge Arinez. Energy efficiency management of an integrated serial production line and hvac system. *IEEE transactions on automation science and engineering*, 11(3):789–797, 2013.

[38] Michael P Brundage, Qing Chang, Yang Li, Jorge Arinez, and Guoxian Xiao. Implementing a real-time, energy-efficient control methodology to maximize manufacturing profits. *IEEE transactions on systems, man, and cybernetics: systems*, 46(6):855–866, 2015.

[39] Jing Zou, Jorge Arinez, Qing Chang, and Yong Lei. Opportunity window for energy saving and maintenance in stochastic production systems. *Journal of Manufacturing Science and Engineering*, 138(12), 2016.

[40] Jing Zou, Qing Chang, Jorge Arinez, and Guoxian Xiao. Data-driven modeling and real-time distributed control for energy efficient manufacturing systems. *Energy*, 127:247–257, 2017.

[41] JF Wang, Jin Xue, Y Feng, SQ Li, Y Fu, and Q Chang. Active energy saving strategy for sensible manufacturing systems operation based on real time production status. In *2016 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 1001–1005. IEEE, 2016.

[42] Zhiyang Jia, Liang Zhang, Jorge Arinez, and Guoxian Xiao. 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, 2016.

[43] Yunchao Zhang, Zeyi Sun, Ruwen Qin, and Haoyi Xiong. 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*, 2019.

[44] Peng-Hao Cui, Jun-Qiang Wang, Yang Li, and Fei-Yi Yan. Energy-efficient control in serial production lines: Modeling, analysis and improvement. *Journal of Manufacturing Systems*, 60:11–21, 2021.

[45] Junfeng Wang, Zicheng Fei, Qing Chang, and Shiqi Li. Energy saving operation of manufacturing system based on dynamic adaptive fuzzy reasoning petri net. *Energies*, 12(11):2216, 2019.

[46] Yang Li, Jun-Qiang Wang, and Qing Chang. Event-based production control for energy efficiency improvement in sustainable multistage manufacturing systems. *Journal of Manufacturing Science and Engineering*, 141(2), 2019.

[47] Paolo Renna. Energy saving by switch-off policy in a pull-controlled production line. *Sustainable Production and Consumption*, 16:25–32, 2018.

[48] Carlos Monteiro, Luís P Ferreira, Nuno O Fernandes, FJG Silva, and Ivo Amaral. Improving the machining process of the metalwork industry by upgrading operative sequences, standard manufacturing times and production procedure changes. *Procedia Manufacturing*, 38:1713–1722, 2019.

[49] Stuart J Russell and Peter Norvig. Artificial intelligence: a modern approach. malaysia, 2016.

[50] Bernd Waschneck, Thomas Altenmüller, Thomas Bauernhansl, and Andreas Kyek. Production scheduling in complex job shops from an industry 4.0 perspective: A review and challenges in the semiconductor industry. *SAMI@ iKNOW*, pages 1–12, 2016.

[51] Yuxi Li. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*, 2017.

[52] Tamás Kegyes, Zoltán Süle, and János Abonyi. The applicability of reinforcement learning methods in the development of industry 4.0 applications. *Complexity*, 2021, 2021.

[53] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

[54] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8:279–292, 1992.

[55] Petia Georgieva, Lyudmila Mihaylova, and Lakhmi C Jain. *Advances in Intelligent Signal Processing and Data Mining*. Springer, 2013.

[56] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

[57] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[58] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[59] Hongzi Mao, Mohammad Alizadeh, Ishai Menache, and Srikanth Kandula. Resource management with deep reinforcement learning. In Bryan Ford, Alex C. Snoeren, and Ellen Zegura, editors, *Proceedings of the 15th ACM Workshop on Hot Topics in Networks - HotNets '16*, pages 50–56, New York, New York, USA, 2016. ACM Press.

[60] Sheng Li, Maxim Egorov, and Mykel Kochenderfer. Optimizing collision avoidance in dense airspace using deep reinforcement learning, 2019.

[61] Jens Kober and Jan Peters. Reinforcement learning in robotics: A survey. In Jens Kober and Jan Peters, editors, *Learning Motor Skills*, volume 97 of *Springer Tracts in Advanced Robotics*, pages 9–67. Springer International Publishing, 2014.

[62] Mariya Yao. Breakthrough research in reinforcement learning from 2019, 2019.

[63] Marcel Panzer and Benedict Bender. Deep reinforcement learning in production systems: a systematic literature review. *International Journal of Production Research*, 60(13):4316–4341, 2022.

[64] H-P Wiendahl, Hoda A ElMaraghy, Peter Nyhuis, Michael F Zäh, H-H Wiendahl, Neil Duffie, and Michael Brieke. Changeable manufacturing-classification, design and operation. *CIRP annals*, 56(2):783–809, 2007.

[65] H-P Wiendahl, Hoda A ElMaraghy, Peter Nyhuis, Michael F Zäh, H-H Wiendahl, Neil Duffie, and Michael Brieke. Changeable manufacturing-classification, design and operation. *CIRP annals*, 56(2):783–809, 2007.

[66] Hans-Peter Wiendahl, Jürgen Reichardt, and Peter Nyhuis. *Handbook factory planning and design*. Springer, 2015.

[67] Chun-Cheng Lin, Der-Jiunn Deng, Yen-Ling Chih, and Hsin-Ting Chiu. Smart manufacturing scheduling with edge computing using multi-class deep q network. *IEEE Transactions on Industrial Informatics*, 15(7):4276–4284, 2019.

[68] Chien-Liang Liu, Chuan-Chin Chang, and Chun-Jan Tseng. Actor-critic deep reinforcement learning for solving job shop scheduling problems. *Ieee Access*, 8:71752–71762, 2020.

[69] Schirin Baer, Danielle Turner, Punit Mohanty, Vladimir Samsonov, Romuald Bakakeu, and Tobias Meisen. Multi agent deep q-network approach for online job shop scheduling in flexible manufacturing. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, volume 5, pages 1–9, 2020.

[70] Longfei Zhou, Lin Zhang, and Berthold KP Horn. Deep reinforcement learning-based dynamic scheduling in smart manufacturing. *Procedia Cirp*, 93:383–388, 2020.

[71] Chen-Xin Wu, Min-Hui Liao, Mumtaz Karatas, Sheng-Yong Chen, and Yu-Jun Zheng. Real-time neural network scheduling of emergency medical mask production during covid-19. *Applied Soft Computing*, 97:106790, 2020.

[72] In-Beom Park, Jaeseok Huh, Joongkyun Kim, and Jonghun Park. A reinforcement learning approach to robust scheduling of semiconductor manufacturing facilities. *IEEE Transactions on Automation Science and Engineering*, 17(3):1420–1431, 2019.

[73] Jorge A Palombarini and Ernesto C Martínez. Closed-loop rescheduling using deep reinforcement learning. *IFAC-PapersOnLine*, 52(1):231–236, 2019.

[74] Christian D Hubbs, Can Li, Nikolaos V Sahinidis, Ignacio E Grossmann, and John M Wassick. A deep reinforcement learning approach for chemical production scheduling. *Computers & Chemical Engineering*, 141:106982, 2020.

[75] Jinling Leng, Chun Jin, Alexander Vogl, and Huiyu Liu. Deep reinforcement learning for a color-batching resequencing problem. *Journal of Manufacturing Systems*, 56:175–187, 2020.

[76] Bernd Waschneck, André Reichstaller, Lenz Belzner, Thomas Altenmüller, Thomas Bauernhansl, Alexander Knapp, and Andreas Kyek. Deep reinforcement learning for semiconductor production scheduling. In *2018 29th annual SEMI advanced semiconductor manufacturing conference (ASMC)*, pages 301–306. IEEE, 2018.

[77] Andreas Kuhnle, Marvin Carl May, Louis Schaefer, and Gisela Lanza. Explainable reinforcement learning in production control of job shop manufacturing system. *International Journal of Production Research*, pages 1–23, 2021.

[78] Kenneth N McKay and Vincent CS Wiers. Planning, scheduling and dispatching tasks in production control. *Cognition, Technology & Work*, 5:82–93, 2003.

[79] Martijn Mes, Matthieu Van Der Heijden, and Aart Van Harten. Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems. *European journal of operational research*, 181(1):59–75, 2007.

[80] Thomas Altenmüller, Tillmann Stüker, Bernd Waschneck, Andreas Kuhnle, and Gisela Lanza. Reinforcement learning for an intelligent and autonomous production control of complex job-shops under time constraints. *Production Engineering*, 14:319–328, 2020.

[81] Nicole Stricker, Andreas Kuhnle, Roland Sturm, and Simon Friess. Reinforcement learning for adaptive order dispatching in the semiconductor industry. *CIRP Annals*, 67(1):511–514, 2018.

[82] Andreas Kuhnle, Jan-Philipp Kaiser, Felix Theiß, Nicole Stricker, and Gisela Lanza. Designing an adaptive production control system using reinforcement learning. *Journal of Intelligent Manufacturing*, 32(3):855–876, 2021.

[83] Tomé Silva and Américo Azevedo. Production flow control through the use of reinforcement learning. *Procedia Manufacturing*, 38:194–202, 2019.

[84] Andreja Malus, Dominik Kozjek, et al. Real-time order dispatching for a fleet of autonomous mobile robots using multi-agent reinforcement learning. *CIRP annals*, 69(1):397–400, 2020.

[85] Niclas Feldkamp, Soeren Bergmann, and Steffen Strassburger. Simulation-based deep reinforcement learning for modular production systems. In *2020 Winter Simulation Conference (WSC)*, pages 1596–1607. IEEE, 2020.

[86] Hao Hu, Xiaoliang Jia, Qixuan He, Shifeng Fu, and Kuo Liu. Deep reinforcement learning based agvs real-time scheduling with mixed rule for flexible shop floor in industry 4.0. *Computers & Industrial Engineering*, 149:106749, 2020.

[87] Max Hildebrand, Rasmus S Andersen, and Simon Bøgh. Deep reinforcement learning for robot batching optimization and flow control. *Procedia Manufacturing*, 51:1462–1468, 2020.

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

[89] J George Shanthikumar, Shengwei Ding, and Mike Tao Zhang. Queueing theory for semiconductor manufacturing systems: A survey and open problems. *IEEE Transactions on Automation Science and Engineering*, 4(4):513–522, 2007.

[90] Cong Zhao and Jingshan Li. Analysis of multiproduct manufacturing systems with homogeneous exponential machines. *IEEE Transactions on Automation Science and Engineering*, 11(3):828–838, 2013.

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

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

[93] ISO Central Secretary. Machine tools — environmental evaluation of machine tools — part 1: Design methodology for energy-efficient machine tools. Standard ISO/TC 39/WG 12., International Organization for Standardization, Geneva, CH, 2017.

[94] Steven A Lippman. Applying a new device in the optimization of exponential queuing systems. *Operations research*, 23(4):687–710, 1975.

[95] Richard Bellman. A markovian decision process. *Journal of mathematics and mechanics*, 6(5):679–684, 1957.

[96] Douglas C Montgomery. *Design and analysis of experiments*. John wiley & sons, 2017.

[97] Dan Trietsch and Kenneth R Baker. Pert 21: Fitting pert/cpm for use in the 21st century. *International journal of project management*, 30(4):490–502, 2012.

[98] Peter D Welch. The statistical analysis of simulation results. *The computer performance modeling handbook*, 22:268–328, 1983.

[99] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[100] Atul Bhandari, Alan Scheller-Wolf, and Mor Harchol-Balter. An exact and efficient algorithm for the constrained dynamic operator staffing problem for call centers. *Management Science*, 54(2):339–353, 2008.

[101] Emre Nadar, Mustafa Akan, and Alan Scheller-Wolf. Experimental results indicating lattice-dependent policies may be optimal for general assemble-to-order systems. *Production and Operations Management*, 25(4):647–661, 2016.

[102] Behnaz Hosseini and Barış Tan. Modelling and analysis of a cooperative production network. *International Journal of Production Research*, 57(21):6665–6686, 2019.

[103] Andreas Kuhnle, Louis Schäfer, Nicole Stricker, and Gisela Lanza. Design, implementation and evaluation of reinforcement learning for an adaptive order dispatching in job shop manufacturing systems. *Procedia CIRP*, 81:234–239, 2019.

[104] Alexander Kuhnle, Michael Schaarschmidt, and Kai Fricke. Tensorforce: a tensorflow library for applied reinforcement learning. *Web page*, 9, 2017.

[105] Alberto Loffredo, Nicla Frigerio, Ettore Lanzarone, and Andrea Matta. Energy-efficient control policy for parallel and identical machines with availability constraint. *IEEE Robotics and Automation Letters*, 2021.

[106] Alberto Loffredo, Nicla Frigerio, Ettore Lanzarone, Mani Ghassempouri, and Andrea Matta. Energy-efficient control of parallel and identical machines: Impact on the overall production system. *Procedia CIRP*, 105:739–744, 2022.

[107] Alberto Loffredo, Nicola Frigerio, Ettore Lanzarone, Andrea Matta, et al. Energy-efficient control in a two-stage production line with parallel machines. In *Abstracts of the 18th IEEE International Conference on Automation Science and Engineering (CASE 2022)*, number Paper MoAM7. 6, pages 1–2, 2022.

[108] Alberto Loffredo, Nicla Frigerio, Ettore Lanzarone, and Andrea Matta. Energy-efficient control in multi-stage production lines with parallel machine workstations and production constraints. *IISE Transactions*, 2023.

[109] Alberto Loffredo, Marvin Carl May, Louis Schaefer, Andrea Matta, and Gisela Lanza. Reinforcement learning for energy-efficient control of parallel and identical machines. *CIRP Journal of Manufacturing Science and Technology*, 2023 - Under Review.

[110] Alberto Loffredo, Marvin Carl May, and Andrea Matta. Reinforcement learning for energy-efficient control of multi-stage production lines with parallel machine workstations. *XVI AITEM Conference*, 2023 - Under Review.

Appendix A

Industrial Case Parameters

Table A.1: Industrial use case parameters, system described in Section 3.3 and presented in Figure 3.4.

Stage Nr.	**Operation Nr.**	**Controllable**	K	c	μ [1/s]	δ [1/s]	ψ [1/s]
1	Op. 1	No	20	1	0.018	0.043	-
2	Op. 2	Yes	10	6	0.012	0.033	0.001
3	Op. 3	Yes	21	1	0.028	0.010	0.009
4	Op. 4	Yes	10	2	0.030	0.100	-
5	Op. 5	No	5	1	0.018	-	-
6	Op. 6	No	3	2	0.008	0.017	0.001
7	Op. 7	Yes	5	1	0.030	0.100	-
8	Op. 8	Yes	10	15	0.002	0.033	0.001
9	Op. 9	Yes	21	1	0.028	0.010	0.009
10	Op. 10	Yes	10	1	0.018	0.100	0.002
11	Op. 11	No	5	1	0.018	0.043	-
12	Op. 12	Yes	10	1	0.018	0.040	0.001
13	Op. 13	Yes	10	6	0.012	0.033	0.001
14	Op. 14	Yes	21	1	0.028	0.010	0.009
15	Op. 15	Yes	8	1	0.012	0.013	0.001
16	Op. 16	No	10	1	0.018	0.040	0.001
17	Op. 17	No	5	1	0.018	0.043	-
18	Op. 18	No	10	1	0.018	-	-
19	Op. 19	Yes	10	1	0.018	0.040	0.001
20	Op. 20	No	10	1	0.018	-	-

Stage Nr.	**Operation Nr.**	ξ [1/s]	w_b [kW]	$w_i d$ [kW]	$w_s u$ [kW]	$w_s b$ [kW]	w_f [kW]
1	Op. 1	-	1.50	0.40	1.10	0	0
2	Op. 2	0.033	15.00	9.30	10.00	0	0
3	Op. 3	0.025	148.75	64.50	102.67	0	0
4	Op. 4	-	2.00	0.91	1.60	0	0
5	Op. 5	-	0.00	0.00	0.00	0	0
6	Op. 6	0.017	3.80	0.30	0.50	0	0
7	Op. 7	-	2.00	0.91	1.60	0	0
8	Op. 8	0.033	30.00	18.50	21.00	0	0
9	Op. 9	0.025	148.75	64.50	102.67	0	0
10	Op. 10	0.05	9.60	3.05	5.33	0	0
11	Op. 11	-	1.50	0.40	1.10	0	0
12	Op. 12	0.017	9.00	5.62	6.40	0	0
13	Op. 13	0.033	15.00	9.30	10.00	0	0
14	Op. 14	0.025	148.75	64.50	102.67	0	0
15	Op. 15	0.02	23.79	8.72	10.14	0	0
16	Op. 16	0.017	9.00	5.62	6.40	0	0
17	Op. 17	-	1.50	0.40	1.10	0	0
18	Op. 18	-	0.00	0.00	0.00	0	0
19	Op. 19	0.017	9.00	5.62	6.40	0	0
20	Op. 20	-	0.00	0.00	0.00	0	0

Appendix B

Detailed Values for Each Experimental Analysis

B.1 Detailed Results for Chapter 3

Table B.1: MDP-based model for single workstations validation results, values displayed in Figure 3.3.

Experiment Nr.	c	μ [1/s]	δ [1/s]	w_b [kW]	w_i [kW]	$w_s u$ [kW]	w_h [kW]	λ [1/s]	Energy Saving [%]
1	3	0.1	0.02	20	0.5	5.5	0.5	0.1	0.20%
2	3	0.1	0.02	20	0.5	5.5	0.5	0.02	0.00%
3	3	0.1	0.02	20	0.5	5.5	25	0.1	0.14%
4	3	0.1	0.02	20	0.5	5.5	25	0.02	0.00%
5	3	0.1	0.02	20	5	5.5	0.5	0.1	1.72%
6	3	0.1	0.02	20	5	5.5	0.5	0.02	0.00%
7	3	0.1	0.02	20	5	5.5	25	0.1	1.26%
8	3	0.1	0.02	20	5	5.5	25	0.02	0.00%
9	3	0.1	0.02	60	0.5	5.5	0.5	0.1	0.14%
10	3	0.1	0.02	60	0.5	5.5	0.5	0.02	0.00%
11	3	0.1	0.02	60	0.5	5.5	25	0.1	0.12%
12	3	0.1	0.02	60	0.5	5.5	25	0.02	0.00%
13	3	0.1	0.02	60	5	5.5	0.5	0.1	1.33%
14	3	0.1	0.02	60	5	5.5	0.5	0.02	0.00%
15	3	0.1	0.02	60	5	5.5	25	0.1	1.04%
16	3	0.1	0.02	60	5	5.5	25	0.02	0.00%
17	3	0.1	0.1	20	0.5	5.5	0.5	0.1	0.10%
18	3	0.1	0.1	20	0.5	5.5	0.5	0.02	0.00%
19	3	0.1	0.1	20	0.5	5.5	25	0.1	0.07%
20	3	0.1	0.1	20	0.5	5.5	25	0.02	0.00%
21	3	0.1	0.1	20	5	5.5	0.5	0.1	0.95%
22	3	0.1	0.1	20	5	5.5	0.5	0.02	0.00%
23	3	0.1	0.1	20	5	5.5	25	0.1	0.65%
24	3	0.1	0.1	20	5	5.5	25	0.02	0.00%
25	3	0.1	0.1	60	0.5	5.5	0.5	0.1	0.08%
26	3	0.1	0.1	60	0.5	5.5	0.5	0.02	0.00%
27	3	0.1	0.1	60	0.5	5.5	25	0.1	0.05%
28	3	0.1	0.1	60	0.5	5.5	25	0.02	0.00%
29	3	0.1	0.1	60	5	5.5	0.5	0.1	0.70%
30	3	0.1	0.1	60	5	5.5	0.5	0.02	0.00%
31	3	0.1	0.1	60	5	5.5	25	0.1	0.52%
32	3	0.1	0.1	60	5	5.5	25	0.02	0.00%
33	3	0.02	0.02	20	0.5	5.5	0.5	0.1	0.35%
34	3	0.02	0.02	20	0.5	5.5	0.5	0.02	0.14%
35	3	0.02	0.02	20	0.5	5.5	25	0.1	0.20%
36	3	0.02	0.02	20	0.5	5.5	25	0.02	0.09%
37	3	0.02	0.02	20	5	5.5	0.5	0.1	3.24%
38	3	0.02	0.02	20	5	5.5	0.5	0.02	1.34%
39	3	0.02	0.02	20	5	5.5	25	0.1	1.77%
40	3	0.02	0.02	20	5	5.5	25	0.02	0.82%
41	3	0.02	0.02	60	0.5	5.5	0.5	0.1	2.20%

42	3	0.02	0.02	60	0.5	5.5	0.5	0.02	0.09%
43	3	0.02	0.02	60	0.5	5.5	25	0.1	0.39%
44	3	0.02	0.02	60	0.5	5.5	25	0.02	0.07%
45	3	0.02	0.02	60	5	5.5	0.5	0.1	4.03%
46	3	0.02	0.02	60	5	5.5	0.5	0.02	0.90%
47	3	0.02	0.02	60	5	5.5	25	0.1	1.60%
48	3	0.02	0.02	60	5	5.5	25	0.02	0.62%
49	3	0.02	0.1	20	0.5	5.5	0.5	0.1	0.26%
50	3	0.02	0.1	20	0.5	5.5	0.5	0.02	0.08%
51	3	0.02	0.1	20	0.5	5.5	25	0.1	0.16%
52	3	0.02	0.1	20	0.5	5.5	25	0.02	0.05%
53	3	0.02	0.1	20	5	5.5	0.5	0.1	2.39%
54	3	0.02	0.1	20	5	5.5	0.5	0.02	0.72%
55	3	0.02	0.1	20	5	5.5	25	0.1	1.43%
56	3	0.02	0.1	20	5	5.5	25	0.02	0.44%
57	3	0.02	0.1	60	0.5	5.5	0.5	0.1	14.87%
58	3	0.02	0.1	60	0.5	5.5	0.5	0.02	0.05%
59	3	0.02	0.1	60	0.5	5.5	25	0.1	7.06%
60	3	0.02	0.1	60	0.5	5.5	25	0.02	0.22%
61	3	0.02	0.1	60	5	5.5	0.5	0.1	14.76%
62	3	0.02	0.1	60	5	5.5	0.5	0.02	0.48%
63	3	0.02	0.1	60	5	5.5	25	0.1	8.00%
64	3	0.02	0.1	60	5	5.5	25	0.02	0.22%
65	3	0.1	0.02	20	0.5	19.5	0.5	0.1	0.15%
66	3	0.1	0.02	20	0.5	19.5	0.5	0.02	0.00%
67	3	0.1	0.02	20	0.5	19.5	25	0.1	0.11%
68	3	0.1	0.02	20	0.5	19.5	25	0.02	0.00%
69	3	0.1	0.02	20	5	19.5	0.5	0.1	1.32%
70	3	0.1	0.02	20	5	19.5	0.5	0.02	0.00%
71	3	0.1	0.02	20	5	19.5	25	0.1	0.97%
72	3	0.1	0.02	20	5	19.5	25	0.02	0.00%
73	3	0.1	0.02	60	0.5	19.5	0.5	0.1	0.11%
74	3	0.1	0.02	60	0.5	19.5	0.5	0.02	0.00%
75	3	0.1	0.02	60	0.5	19.5	25	0.1	0.09%
76	3	0.1	0.02	60	0.5	19.5	25	0.02	0.00%
77	3	0.1	0.02	60	5	19.5	0.5	0.1	1.02%
78	3	0.1	0.02	60	5	19.5	0.5	0.02	0.00%
79	3	0.1	0.02	60	5	19.5	25	0.1	0.80%
80	3	0.1	0.02	60	5	19.5	25	0.02	0.00%
81	3	0.1	0.1	20	0.5	19.5	0.5	0.1	0.08%
82	3	0.1	0.1	20	0.5	19.5	0.5	0.02	0.00%
83	3	0.1	0.1	20	0.5	19.5	25	0.1	0.05%
84	3	0.1	0.1	20	0.5	19.5	25	0.02	0.00%
85	3	0.1	0.1	20	5	19.5	0.5	0.1	0.73%
86	3	0.1	0.1	20	5	19.5	0.5	0.02	0.00%
87	3	0.1	0.1	20	5	19.5	25	0.1	0.50%
88	3	0.1	0.1	20	5	19.5	25	0.02	0.00%
89	3	0.1	0.1	60	0.5	19.5	0.5	0.1	0.06%
90	3	0.1	0.1	60	0.5	19.5	0.5	0.02	0.00%
91	3	0.1	0.1	60	0.5	19.5	25	0.1	0.04%
92	3	0.1	0.1	60	0.5	19.5	25	0.02	0.00%
93	3	0.1	0.1	60	5	19.5	0.5	0.1	0.54%
94	3	0.1	0.1	60	5	19.5	0.5	0.02	0.00%
95	3	0.1	0.1	60	5	19.5	25	0.1	0.40%
96	3	0.1	0.1	60	5	19.5	25	0.02	0.00%
97	3	0.02	0.02	20	0.5	19.5	0.5	0.1	0.27%
98	3	0.02	0.02	20	0.5	19.5	0.5	0.02	0.11%
99	3	0.02	0.02	20	0.5	19.5	25	0.1	0.15%
100	3	0.02	0.02	20	0.5	19.5	25	0.02	0.07%
101	3	0.02	0.02	20	5	19.5	0.5	0.1	2.49%
102	3	0.02	0.02	20	5	19.5	0.5	0.02	1.03%
103	3	0.02	0.02	20	5	19.5	25	0.1	1.36%
104	3	0.02	0.02	20	5	19.5	25	0.02	0.63%
105	3	0.02	0.02	60	0.5	19.5	0.5	0.1	1.69%
106	3	0.02	0.02	60	0.5	19.5	0.5	0.02	0.07%
107	3	0.02	0.02	60	0.5	19.5	25	0.1	0.30%
108	3	0.02	0.02	60	0.5	19.5	25	0.02	0.05%
109	3	0.02	0.02	60	5	19.5	0.5	0.1	3.10%
110	3	0.02	0.02	60	5	19.5	0.5	0.02	0.69%
111	3	0.02	0.02	60	5	19.5	25	0.1	1.23%
112	3	0.02	0.02	60	5	19.5	25	0.02	0.48%
113	3	0.02	0.1	20	0.5	19.5	0.5	0.1	0.20%
114	3	0.02	0.1	20	0.5	19.5	0.5	0.02	0.06%
115	3	0.02	0.1	20	0.5	19.5	25	0.1	0.12%
116	3	0.02	0.1	20	0.5	19.5	25	0.02	0.04%
117	3	0.02	0.1	20	5	19.5	0.5	0.1	1.84%
118	3	0.02	0.1	20	5	19.5	0.5	0.02	0.55%
119	3	0.02	0.1	20	5	19.5	25	0.1	1.10%

120	3	0.02	0.1	20	5	19.5	25	0.02	0.34%
121	3	0.02	0.1	60	0.5	19.5	0.5	0.1	14.53%
122	3	0.02	0.1	60	0.5	19.5	0.5	0.02	0.04%
123	3	0.02	0.1	60	0.5	19.5	25	0.1	5.43%
124	3	0.02	0.1	60	0.5	19.5	25	0.02	0.17%
125	3	0.02	0.1	60	5	19.5	0.5	0.1	15.60%
126	3	0.02	0.1	60	5	19.5	0.5	0.02	0.37%
127	3	0.02	0.1	60	5	19.5	25	0.1	6.15%
128	3	0.02	0.1	60	5	19.5	25	0.02	0.17%
129	6	0.1	0.02	20	0.5	5.5	0.5	0.1	0.36%
130	6	0.1	0.02	20	0.5	5.5	0.5	0.02	0.14%
131	6	0.1	0.02	20	0.5	5.5	25	0.1	0.29%
132	6	0.1	0.02	20	0.5	5.5	25	0.02	0.13%
133	6	0.1	0.02	20	5	5.5	0.5	0.1	3.14%
134	6	0.1	0.02	20	5	5.5	0.5	0.02	1.27%
135	6	0.1	0.02	20	5	5.5	25	0.1	2.55%
136	6	0.1	0.02	20	5	5.5	25	0.02	1.05%
137	6	0.1	0.02	60	0.5	5.5	0.5	0.1	0.27%
138	6	0.1	0.02	60	0.5	5.5	0.5	0.02	0.11%
139	6	0.1	0.02	60	0.5	5.5	25	0.1	0.22%
140	6	0.1	0.02	60	0.5	5.5	25	0.02	0.10%
141	6	0.1	0.02	60	5	5.5	0.5	0.1	2.37%
142	6	0.1	0.02	60	5	5.5	0.5	0.02	0.98%
143	6	0.1	0.02	60	5	5.5	25	0.1	2.02%
144	6	0.1	0.02	60	5	5.5	25	0.02	0.84%
145	6	0.1	0.1	20	0.5	5.5	0.5	0.1	0.20%
146	6	0.1	0.1	20	0.5	5.5	0.5	0.02	0.07%
147	6	0.1	0.1	20	0.5	5.5	25	0.1	0.15%
148	6	0.1	0.1	20	0.5	5.5	25	0.02	0.06%
149	6	0.1	0.1	20	5	5.5	0.5	0.1	1.78%
150	6	0.1	0.1	20	5	5.5	0.5	0.02	0.69%
151	6	0.1	0.1	20	5	5.5	25	0.1	1.40%
152	6	0.1	0.1	20	5	5.5	25	0.02	0.55%
153	6	0.1	0.1	60	0.5	5.5	0.5	0.1	0.14%
154	6	0.1	0.1	60	0.5	5.5	0.5	0.02	0.06%
155	6	0.1	0.1	60	0.5	5.5	25	0.1	0.11%
156	6	0.1	0.1	60	0.5	5.5	25	0.02	0.04%
157	6	0.1	0.1	60	5	5.5	0.5	0.1	1.29%
158	6	0.1	0.1	60	5	5.5	0.5	0.02	0.50%
159	6	0.1	0.1	60	5	5.5	25	0.1	1.08%
160	6	0.1	0.1	60	5	5.5	25	0.02	0.42%
161	6	0.02	0.02	20	0.5	5.5	0.5	0.1	0.99%
162	6	0.02	0.02	20	0.5	5.5	0.5	0.02	0.32%
163	6	0.02	0.02	20	0.5	5.5	25	0.1	0.64%
164	6	0.02	0.02	20	0.5	5.5	25	0.02	0.22%
165	6	0.02	0.02	20	5	5.5	0.5	0.1	8.83%
166	6	0.02	0.02	20	5	5.5	0.5	0.02	2.83%
167	6	0.02	0.02	20	5	5.5	25	0.1	5.87%
168	6	0.02	0.02	20	5	5.5	25	0.02	1.97%
169	6	0.02	0.02	60	0.5	5.5	0.5	0.1	9.02%
170	6	0.02	0.02	60	0.5	5.5	0.5	0.02	0.20%
171	6	0.02	0.02	60	0.5	5.5	25	0.1	4.13%
172	6	0.02	0.02	60	0.5	5.5	25	0.02	0.15%
173	6	0.02	0.02	60	5	5.5	0.5	0.1	13.47%
174	6	0.02	0.02	60	5	5.5	0.5	0.02	1.78%
175	6	0.02	0.02	60	5	5.5	25	0.1	7.69%
176	6	0.02	0.02	60	5	5.5	25	0.02	1.40%
177	6	0.02	0.1	20	0.5	5.5	0.5	0.1	1.81%
178	6	0.02	0.1	20	0.5	5.5	0.5	0.02	0.15%
179	6	0.02	0.1	20	0.5	5.5	25	0.1	0.53%
180	6	0.02	0.1	20	0.5	5.5	25	0.02	0.11%
181	6	0.02	0.1	20	5	5.5	0.5	0.1	6.78%
182	6	0.02	0.1	20	5	5.5	0.5	0.02	1.40%
183	6	0.02	0.1	20	5	5.5	25	0.1	4.02%
184	6	0.02	0.1	20	5	5.5	25	0.02	0.98%
185	6	0.02	0.1	60	0.5	5.5	0.5	0.1	8.71%
186	6	0.02	0.1	60	0.5	5.5	0.5	0.02	0.10%
187	6	0.02	0.1	60	0.5	5.5	25	0.1	6.02%
188	6	0.02	0.1	60	0.5	5.5	25	0.02	0.07%
189	6	0.02	0.1	60	5	5.5	0.5	0.1	11.69%
190	6	0.02	0.1	60	5	5.5	0.5	0.02	0.90%
191	6	0.02	0.1	60	5	5.5	25	0.1	8.40%
192	6	0.02	0.1	60	5	5.5	25	0.02	0.70%
193	6	0.1	0.02	20	0.5	19.5	0.5	0.1	0.26%
194	6	0.1	0.02	20	0.5	19.5	0.5	0.02	0.10%
195	6	0.1	0.02	20	0.5	19.5	25	0.1	0.21%
196	6	0.1	0.02	20	0.5	19.5	25	0.02	0.09%
197	6	0.1	0.02	20	5	19.5	0.5	0.1	2.24%

198	6	0.1	0.02	20	5	19.5	0.5	0.02	0.91%
199	6	0.1	0.02	20	5	19.5	25	0.1	1.82%
200	6	0.1	0.02	20	5	19.5	25	0.02	0.75%
201	6	0.1	0.02	60	0.5	19.5	0.5	0.1	0.19%
202	6	0.1	0.02	60	0.5	19.5	0.5	0.02	0.08%
203	6	0.1	0.02	60	0.5	19.5	25	0.1	0.16%
204	6	0.1	0.02	60	0.5	19.5	25	0.02	0.07%
205	6	0.1	0.02	60	5	19.5	0.5	0.1	1.69%
206	6	0.1	0.02	60	5	19.5	0.5	0.02	0.70%
207	6	0.1	0.02	60	5	19.5	25	0.1	1.44%
208	6	0.1	0.02	60	5	19.5	25	0.02	0.60%
209	6	0.1	0.1	20	0.5	19.5	0.5	0.1	0.14%
210	6	0.1	0.1	20	0.5	19.5	0.5	0.02	0.05%
211	6	0.1	0.1	20	0.5	19.5	25	0.1	0.11%
212	6	0.1	0.1	20	0.5	19.5	25	0.02	0.04%
213	6	0.1	0.1	20	5	19.5	0.5	0.1	1.27%
214	6	0.1	0.1	20	5	19.5	0.5	0.02	0.49%
215	6	0.1	0.1	20	5	19.5	25	0.1	1.00%
216	6	0.1	0.1	20	5	19.5	25	0.02	0.39%
217	6	0.1	0.1	60	0.5	19.5	0.5	0.1	0.10%
218	6	0.1	0.1	60	0.5	19.5	0.5	0.02	0.04%
219	6	0.1	0.1	60	0.5	19.5	25	0.1	0.08%
220	6	0.1	0.1	60	0.5	19.5	25	0.02	0.03%
221	6	0.1	0.1	60	5	19.5	0.5	0.1	0.92%
222	6	0.1	0.1	60	5	19.5	0.5	0.02	0.36%
223	6	0.1	0.1	60	5	19.5	25	0.1	0.77%
224	6	0.1	0.1	60	5	19.5	25	0.02	0.30%
225	6	0.02	0.02	20	0.5	19.5	0.5	0.1	0.71%
226	6	0.02	0.02	20	0.5	19.5	0.5	0.02	0.23%
227	6	0.02	0.02	20	0.5	19.5	25	0.1	0.46%
228	6	0.02	0.02	20	0.5	19.5	25	0.02	0.16%
229	6	0.02	0.02	20	5	19.5	0.5	0.1	6.31%
230	6	0.02	0.02	20	5	19.5	0.5	0.02	2.02%
231	6	0.02	0.02	20	5	19.5	25	0.1	4.19%
232	6	0.02	0.02	20	5	19.5	25	0.02	1.41%
233	6	0.02	0.02	60	0.5	19.5	0.5	0.1	6.44%
234	6	0.02	0.02	60	0.5	19.5	0.5	0.02	0.14%
235	6	0.02	0.02	60	0.5	19.5	25	0.1	2.95%
236	6	0.02	0.02	60	0.5	19.5	25	0.02	0.11%
237	6	0.02	0.02	60	5	19.5	0.5	0.1	9.62%
238	6	0.02	0.02	60	5	19.5	0.5	0.02	1.27%
239	6	0.02	0.02	60	5	19.5	25	0.1	5.49%
240	6	0.02	0.02	60	5	19.5	25	0.02	1.00%
241	6	0.02	0.1	20	0.5	19.5	0.5	0.1	1.29%
242	6	0.02	0.1	20	0.5	19.5	0.5	0.02	0.11%
243	6	0.02	0.1	20	0.5	19.5	25	0.1	0.38%
244	6	0.02	0.1	20	0.5	19.5	25	0.02	0.08%
245	6	0.02	0.1	20	5	19.5	0.5	0.1	4.84%
246	6	0.02	0.1	20	5	19.5	0.5	0.02	1.00%
247	6	0.02	0.1	20	5	19.5	25	0.1	2.87%
248	6	0.02	0.1	20	5	19.5	25	0.02	0.70%
249	6	0.02	0.1	60	0.5	19.5	0.5	0.1	6.22%
250	6	0.02	0.1	60	0.5	19.5	0.5	0.02	0.07%
251	6	0.02	0.1	60	0.5	19.5	25	0.1	4.30%
252	6	0.02	0.1	60	0.5	19.5	25	0.02	0.05%
253	6	0.02	0.1	60	5	19.5	0.5	0.1	8.35%
254	6	0.02	0.1	60	5	19.5	0.5	0.02	0.64%
255	6	0.02	0.1	60	5	19.5	25	0.1	6.00%
256	6	0.02	0.1	60	5	19.5	25	0.02	0.50%

Table B.2: MDP-based model for single workstations: sensitivity analysis for the industrial case when u_{target} varies. Values displayed in Figure 3.5.

Target Availability [%]	Throughput [Parts/h]	λ [Parts/h]	Energy Saving [%]	Real Availability [%]
75%	211.32	240.12	9.38%	82%
76%	211.32	240.12	9.38%	82%
77%	211.32	240.12	9.38%	82%
78%	211.32	240.12	9.38%	82%
79%	211.32	240.12	9.38%	82%
80%	211.32	240.12	9.38%	82%
81%	211.32	240.12	9.38%	82%

82%	216.72	240.12	9.14%	84%
83%	216.72	240.12	9.14%	84%
84%	223.20	240.12	8.76%	86%
85%	223.20	240.12	8.76%	86%
86%	223.20	240.12	8.76%	86%
87%	228.24	240.12	8.38%	89%
88%	228.24	240.12	8.38%	89%
89%	233.28	240.12	8.07%	90%
90%	233.28	240.12	8.07%	90%
91%	237.96	240.12	7.76%	92%
92%	237.96	240.12	7.76%	92%
93%	240.12	240.12	7.01%	94%
94%	240.12	240.12	5.78%	95%
95%	240.12	240.12	5.17%	96%
96%	240.12	240.12	3.78%	97%
97%	240.12	240.12	2.45%	98%
98%	240.12	240.12	1.77%	98%
99%	240.12	240.12	0.75%	99%
100%	240.12	240.12	0.00%	100%

Table B.3: MDP-based model for single workstations: sensitivity analysis for the industrial case when w_h varies. Values displayed in Figure 3.6.

w_h [kW]	**Throughput [Parts/h]**	**Target Availability [%]**	**Energy Saving [%]**	**Real Availability [%]**
0.43	223.20	85%	8.97%	86%
0.54	223.20	85%	8.89%	86%
0.64	223.20	85%	8.82%	86%
0.75	223.20	85%	8.74%	86%
0.86	223.20	85%	8.67%	86%
0.96	223.20	85%	8.60%	86%
1.07	223.20	85%	8.53%	86%
1.18	223.20	85%	8.46%	86%
1.29	228.24	85%	7.69%	89%
1.39	228.24	85%	7.39%	89%
1.50	233.28	85%	6.58%	90%
1.61	233.28	85%	6.28%	90%
1.71	237.96	85%	5.47%	92%
1.82	237.96	85%	5.17%	92%
1.93	240.12	85%	4.01%	94%
2.00	240.12	85%	2.79%	95%
2.10	240.12	85%	2.17%	96%

B.2 Detailed Results for Chapter 4

Table B.4: MDP-based model for production lines: time duration of the experiments vs number of system states. Values displayed in Figure 4.3.

L - Nr States	**Time Duration [s]**
108	3.72 ± 0.83
504	19.07 ± 4.26
900	39.31 ± 2.80
1296	130.67 ± 1.00

1764	159.54 ± 2.45
2025	203.04 ± 11.62
2592	621.34 ± 4.32
3025	855.56 ± 14.88
3528	1322.02 ± 11.53
3969	1665.00 ± 25.00
4410	2090.00 ± 22.00
5040	3200.00 ± 6.00
5390	3822.00 ± 37.00
5929	4833.00 ± 67.00

Table B.5: MDP-based model for production lines: 2^k factorial analysis results, values displayed in Figure 4.4.

Experiment Nr.	**Balanced**	**Constraint**	δ [1/s]	c_1	c_2	ρ	PCR	w_h [kW]	K	**EN Saving [%]**
1	Yes	Yes	0.02	2	2	0.3	1.2	0.5	2	0.41%
2	Yes	Yes	0.02	2	2	0.3	1.2	0.5	6	2.67%
3	Yes	Yes	0.02	2	2	0.3	1.2	10	2	0.00%
4	Yes	Yes	0.02	2	2	0.3	1.2	10	6	0.25%
5	Yes	Yes	0.02	2	2	0.3	4	0.5	2	3.32%
6	Yes	Yes	0.02	2	2	0.3	4	0.5	6	5.60%
7	Yes	Yes	0.02	2	2	0.3	4	10	2	1.51%
8	Yes	Yes	0.02	2	2	0.3	4	10	6	0.94%
9	Yes	Yes	0.02	2	2	0.9	1.2	0.5	2	0.50%
10	Yes	Yes	0.02	2	2	0.9	1.2	0.5	6	1.60%
11	Yes	Yes	0.02	2	2	0.9	1.2	10	2	0.00%
12	Yes	Yes	0.02	2	2	0.9	1.2	10	6	0.00%
13	Yes	Yes	0.02	2	2	0.9	4	0.5	2	0.60%
14	Yes	Yes	0.02	2	2	0.9	4	0.5	6	1.80%
15	Yes	Yes	0.02	2	2	0.9	4	10	2	0.00%
16	Yes	Yes	0.02	2	2	0.9	4	10	6	0.00%
17	Yes	Yes	0.02	2	6	0.3	1.2	0.5	2	1.00%
18	Yes	Yes	0.02	2	6	0.3	1.2	0.5	6	3.33%
19	Yes	Yes	0.02	2	6	0.3	1.2	10	2	0.00%
20	Yes	Yes	0.02	2	6	0.3	1.2	10	6	0.53%
21	Yes	Yes	0.02	2	6	0.3	4	0.5	2	3.55%
22	Yes	Yes	0.02	2	6	0.3	4	0.5	6	6.23%
23	Yes	Yes	0.02	2	6	0.3	4	10	2	0.41%
24	Yes	Yes	0.02	2	6	0.3	4	10	6	0.00%
25	Yes	Yes	0.02	2	6	0.9	1.2	0.5	2	0.00%
26	Yes	Yes	0.02	2	6	0.9	1.2	0.5	6	0.00%
27	Yes	Yes	0.02	2	6	0.9	1.2	10	2	0.33%
28	Yes	Yes	0.02	2	6	0.9	1.2	10	6	0.44%
29	Yes	Yes	0.02	2	6	0.9	4	0.5	2	0.55%
30	Yes	Yes	0.02	2	6	0.9	4	0.5	6	0.11%
31	Yes	Yes	0.02	2	6	0.9	4	10	2	0.20%
32	Yes	Yes	0.02	2	6	0.9	4	10	6	0.11%
33	Yes	Yes	0.02	6	2	0.3	1.2	0.5	2	0.29%
34	Yes	Yes	0.02	6	2	0.3	1.2	0.5	6	4.10%
35	Yes	Yes	0.02	6	2	0.3	1.2	10	2	0.00%
36	Yes	Yes	0.02	6	2	0.3	1.2	10	6	0.22%
37	Yes	Yes	0.02	6	2	0.3	4	0.5	2	4.27%
38	Yes	Yes	0.02	6	2	0.3	4	0.5	6	6.70%
39	Yes	Yes	0.02	6	2	0.3	4	10	2	0.74%
40	Yes	Yes	0.02	6	2	0.3	4	10	6	0.83%
41	Yes	Yes	0.02	6	2	0.9	1.2	0.5	2	0.32%
42	Yes	Yes	0.02	6	2	0.9	1.2	0.5	6	0.35%
43	Yes	Yes	0.02	6	2	0.9	1.2	10	2	0.53%
44	Yes	Yes	0.02	6	2	0.9	1.2	10	6	0.62%
45	Yes	Yes	0.02	6	2	0.9	4	0.5	2	0.44%
46	Yes	Yes	0.02	6	2	0.9	4	0.5	6	0.46%
47	Yes	Yes	0.02	6	2	0.9	4	10	2	0.60%
48	Yes	Yes	0.02	6	2	0.9	4	10	6	0.70%
49	Yes	Yes	0.02	6	6	0.3	1.2	0.5	2	0.00%
50	Yes	Yes	0.02	6	6	0.3	1.2	0.5	6	4.76%
51	Yes	Yes	0.02	6	6	0.3	1.2	10	2	0.00%
52	Yes	Yes	0.02	6	6	0.3	1.2	10	6	0.00%
53	Yes	Yes	0.02	6	6	0.3	4	0.5	2	4.49%

54	Yes	Yes	0.02	6	6	0.3	4	0.5	6	7.53%
55	Yes	Yes	0.02	6	6	0.3	4	10	2	1.00%
56	Yes	Yes	0.02	6	6	0.3	4	10	6	0.29%
57	Yes	Yes	0.02	6	6	0.9	1.2	0.5	2	2.22%
58	Yes	Yes	0.02	6	6	0.9	1.2	0.5	6	1.22%
59	Yes	Yes	0.02	6	6	0.9	1.2	10	2	0.33%
60	Yes	Yes	0.02	6	6	0.9	1.2	10	6	0.50%
61	Yes	Yes	0.02	6	6	0.9	4	0.5	2	1.22%
62	Yes	Yes	0.02	6	6	0.9	4	0.5	6	1.20%
63	Yes	Yes	0.02	6	6	0.9	4	10	2	0.00%
64	Yes	Yes	0.02	6	6	0.9	4	10	6	0.00%
65	Yes	Yes	0.1	2	2	0.3	1.2	0.5	2	4.33%
66	Yes	Yes	0.1	2	2	0.3	1.2	0.5	6	6.62%
67	Yes	Yes	0.1	2	2	0.3	1.2	10	2	1.91%
68	Yes	Yes	0.1	2	2	0.3	1.2	10	6	3.03%
69	Yes	Yes	0.1	2	2	0.3	4	0.5	2	6.14%
70	Yes	Yes	0.1	2	2	0.3	4	0.5	6	9.17%
71	Yes	Yes	0.1	2	2	0.3	4	10	2	3.69%
72	Yes	Yes	0.1	2	2	0.3	4	10	6	4.04%
73	Yes	Yes	0.1	2	2	0.9	1.2	0.5	2	1.50%
74	Yes	Yes	0.1	2	2	0.9	1.2	0.5	6	2.55%
75	Yes	Yes	0.1	2	2	0.9	1.2	10	2	0.00%
76	Yes	Yes	0.1	2	2	0.9	1.2	10	6	0.63%
77	Yes	Yes	0.1	2	2	0.9	4	0.5	2	2.33%
78	Yes	Yes	0.1	2	2	0.9	4	0.5	6	3.39%
79	Yes	Yes	0.1	2	2	0.9	4	10	2	1.67%
80	Yes	Yes	0.1	2	2	0.9	4	10	6	2.67%
81	Yes	Yes	0.1	2	6	0.3	1.2	0.5	2	3.55%
82	Yes	Yes	0.1	2	6	0.3	1.2	0.5	6	6.09%
83	Yes	Yes	0.1	2	6	0.3	1.2	10	2	1.60%
84	Yes	Yes	0.1	2	6	0.3	1.2	10	6	2.72%
85	Yes	Yes	0.1	2	6	0.3	4	0.5	2	7.00%
86	Yes	Yes	0.1	2	6	0.3	4	0.5	6	9.32%
87	Yes	Yes	0.1	2	6	0.3	4	10	2	3.71%
88	Yes	Yes	0.1	2	6	0.3	4	10	6	3.32%
89	Yes	Yes	0.1	2	6	0.9	1.2	0.5	2	0.73%
90	Yes	Yes	0.1	2	6	0.9	1.2	0.5	6	2.77%
91	Yes	Yes	0.1	2	6	0.9	1.2	10	2	0.00%
92	Yes	Yes	0.1	2	6	0.9	1.2	10	6	0.00%
93	Yes	Yes	0.1	2	6	0.9	4	0.5	2	1.81%
94	Yes	Yes	0.1	2	6	0.9	4	0.5	6	2.95%
95	Yes	Yes	0.1	2	6	0.9	4	10	2	2.14%
96	Yes	Yes	0.1	2	6	0.9	4	10	6	2.47%
97	Yes	Yes	0.1	6	2	0.3	1.2	0.5	2	3.86%
98	Yes	Yes	0.1	6	2	0.3	1.2	0.5	6	6.19%
99	Yes	Yes	0.1	6	2	0.3	1.2	10	2	1.58%
100	Yes	Yes	0.1	6	2	0.3	1.2	10	6	3.36%
101	Yes	Yes	0.1	6	2	0.3	4	0.5	2	6.57%
102	Yes	Yes	0.1	6	2	0.3	4	0.5	6	9.63%
103	Yes	Yes	0.1	6	2	0.3	4	10	2	2.92%
104	Yes	Yes	0.1	6	2	0.3	4	10	6	4.29%
105	Yes	Yes	0.1	6	2	0.9	1.2	0.5	2	0.94%
106	Yes	Yes	0.1	6	2	0.9	1.2	0.5	6	1.85%
107	Yes	Yes	0.1	6	2	0.9	1.2	10	2	0.00%
108	Yes	Yes	0.1	6	2	0.9	1.2	10	6	1.07%
109	Yes	Yes	0.1	6	2	0.9	4	0.5	2	1.86%
110	Yes	Yes	0.1	6	2	0.9	4	0.5	6	3.27%
111	Yes	Yes	0.1	6	2	0.9	4	10	2	1.95%
112	Yes	Yes	0.1	6	2	0.9	4	10	6	2.42%
113	Yes	Yes	0.1	6	6	0.3	1.2	0.5	2	4.02%
114	Yes	Yes	0.1	6	6	0.3	1.2	0.5	6	7.33%
115	Yes	Yes	0.1	6	6	0.3	1.2	10	2	1.33%
116	Yes	Yes	0.1	6	6	0.3	1.2	10	6	2.95%
117	Yes	Yes	0.1	6	6	0.3	4	0.5	2	6.42%
118	Yes	Yes	0.1	6	6	0.3	4	0.5	6	9.44%
119	Yes	Yes	0.1	6	6	0.3	4	10	2	2.96%
120	Yes	Yes	0.1	6	6	0.3	4	10	6	3.71%
121	Yes	Yes	0.1	6	6	0.9	1.2	0.5	2	1.71%
122	Yes	Yes	0.1	6	6	0.9	1.2	0.5	6	2.52%
123	Yes	Yes	0.1	6	6	0.9	1.2	10	2	0.00%
124	Yes	Yes	0.1	6	6	0.9	1.2	10	6	0.00%
125	Yes	Yes	0.1	6	6	0.9	4	0.5	2	1.93%
126	Yes	Yes	0.1	6	6	0.9	4	0.5	6	3.67%
127	Yes	Yes	0.1	6	6	0.9	4	10	2	1.40%
128	Yes	Yes	0.1	6	6	0.9	4	10	6	2.32%
129	Yes	No	0.02	2	2	0.3	1.2	0.5	2	1.00%
130	Yes	No	0.02	2	2	0.3	1.2	0.5	6	6.62%
131	Yes	No	0.02	2	2	0.3	1.2	10	2	0.00%

132	Yes	No	0.02	2	2	0.3	1.2	10	6	0.60%
133	Yes	No	0.02	2	2	0.3	4	0.5	2	7.82%
134	Yes	No	0.02	2	2	0.3	4	0.5	6	13.88%
135	Yes	No	0.02	2	2	0.3	4	10	2	3.64%
136	Yes	No	0.02	2	2	0.3	4	10	6	2.17%
137	Yes	No	0.02	2	2	0.9	1.2	0.5	2	0.00%
138	Yes	No	0.02	2	2	0.9	1.2	0.5	6	0.00%
139	Yes	No	0.02	2	2	0.9	1.2	10	2	0.00%
140	Yes	No	0.02	2	2	0.9	1.2	10	6	0.00%
141	Yes	No	0.02	2	2	0.9	4	0.5	2	0.20%
142	Yes	No	0.02	2	2	0.9	4	0.5	6	0.15%
143	Yes	No	0.02	2	2	0.9	4	10	2	0.40%
144	Yes	No	0.02	2	2	0.9	4	10	6	0.00%
145	Yes	No	0.02	2	6	0.3	1.2	0.5	2	0.00%
146	Yes	No	0.02	2	6	0.3	1.2	0.5	6	7.76%
147	Yes	No	0.02	2	6	0.3	1.2	10	2	0.00%
148	Yes	No	0.02	2	6	0.3	1.2	10	6	1.18%
149	Yes	No	0.02	2	6	0.3	4	0.5	2	8.75%
150	Yes	No	0.02	2	6	0.3	4	0.5	6	15.23%
151	Yes	No	0.02	2	6	0.3	4	10	2	0.91%
152	Yes	No	0.02	2	6	0.3	4	10	6	0.00%
153	Yes	No	0.02	2	6	0.9	1.2	0.5	2	1.20%
154	Yes	No	0.02	2	6	0.9	1.2	0.5	6	2.00%
155	Yes	No	0.02	2	6	0.9	1.2	10	2	0.00%
156	Yes	No	0.02	2	6	0.9	1.2	10	6	0.00%
157	Yes	No	0.02	2	6	0.9	4	0.5	2	1.00%
158	Yes	No	0.02	2	6	0.9	4	0.5	6	1.50%
159	Yes	No	0.02	2	6	0.9	4	10	2	0.00%
160	Yes	No	0.02	2	6	0.9	4	10	6	0.00%
161	Yes	No	0.02	6	2	0.3	1.2	0.5	2	0.65%
162	Yes	No	0.02	6	2	0.3	1.2	0.5	6	9.30%
163	Yes	No	0.02	6	2	0.3	1.2	10	2	0.00%
164	Yes	No	0.02	6	2	0.3	1.2	10	6	0.00%
165	Yes	No	0.02	6	2	0.3	4	0.5	2	9.78%
166	Yes	No	0.02	6	2	0.3	4	0.5	6	15.26%
167	Yes	No	0.02	6	2	0.3	4	10	2	1.76%
168	Yes	No	0.02	6	2	0.3	4	10	6	2.03%
169	Yes	No	0.02	6	2	0.9	1.2	0.5	2	0.74%
170	Yes	No	0.02	6	2	0.9	1.2	0.5	6	0.84%
171	Yes	No	0.02	6	2	0.9	1.2	10	2	1.27%
172	Yes	No	0.02	6	2	0.9	1.2	10	6	1.38%
173	Yes	No	0.02	6	2	0.9	4	0.5	2	1.07%
174	Yes	No	0.02	6	2	0.9	4	0.5	6	1.10%
175	Yes	No	0.02	6	2	0.9	4	10	2	1.41%
176	Yes	No	0.02	6	2	0.9	4	10	6	1.58%
177	Yes	No	0.02	6	6	0.3	1.2	0.5	2	0.00%
178	Yes	No	0.02	6	6	0.3	1.2	0.5	6	11.04%
179	Yes	No	0.02	6	6	0.3	1.2	10	2	0.00%
180	Yes	No	0.02	6	6	0.3	1.2	10	6	0.00%
181	Yes	No	0.02	6	6	0.3	4	0.5	2	11.16%
182	Yes	No	0.02	6	6	0.3	4	0.5	6	17.97%
183	Yes	No	0.02	6	6	0.3	4	10	2	0.00%
184	Yes	No	0.02	6	6	0.3	4	10	6	0.66%
185	Yes	No	0.02	6	6	0.9	1.2	0.5	2	0.50%
186	Yes	No	0.02	6	6	0.9	1.2	0.5	6	1.22%
187	Yes	No	0.02	6	6	0.9	1.2	10	2	0.00%
188	Yes	No	0.02	6	6	0.9	1.2	10	6	0.00%
189	Yes	No	0.02	6	6	0.9	4	0.5	2	1.22%
190	Yes	No	0.02	6	6	0.9	4	0.5	6	1.44%
191	Yes	No	0.02	6	6	0.9	4	10	2	0.00%
192	Yes	No	0.02	6	6	0.9	4	10	6	0.00%
193	Yes	No	0.1	2	2	0.3	1.2	0.5	2	9.99%
194	Yes	No	0.1	2	2	0.3	1.2	0.5	6	15.92%
195	Yes	No	0.1	2	2	0.3	1.2	10	2	4.40%
196	Yes	No	0.1	2	2	0.3	1.2	10	6	7.49%
197	Yes	No	0.1	2	2	0.3	4	0.5	2	15.24%
198	Yes	No	0.1	2	2	0.3	4	0.5	6	22.78%
199	Yes	No	0.1	2	2	0.3	4	10	2	8.38%
200	Yes	No	0.1	2	2	0.3	4	10	6	9.51%
201	Yes	No	0.1	2	2	0.9	1.2	0.5	2	3.60%
202	Yes	No	0.1	2	2	0.9	1.2	0.5	6	5.82%
203	Yes	No	0.1	2	2	0.9	1.2	10	2	0.00%
204	Yes	No	0.1	2	2	0.9	1.2	10	6	1.52%
205	Yes	No	0.1	2	2	0.9	4	0.5	2	5.59%
206	Yes	No	0.1	2	2	0.9	4	0.5	6	8.05%
207	Yes	No	0.1	2	2	0.9	4	10	2	3.85%
208	Yes	No	0.1	2	2	0.9	4	10	6	5.97%
209	Yes	No	0.1	2	6	0.3	1.2	0.5	2	8.70%

210	Yes	No	0.1	2	6	0.3	1.2	0.5	6	14.35%
211	Yes	No	0.1	2	6	0.3	1.2	10	2	3.57%
212	Yes	No	0.1	2	6	0.3	1.2	10	6	6.15%
213	Yes	No	0.1	2	6	0.3	4	0.5	2	15.58%
214	Yes	No	0.1	2	6	0.3	4	0.5	6	23.06%
215	Yes	No	0.1	2	6	0.3	4	10	2	9.17%
216	Yes	No	0.1	2	6	0.3	4	10	6	8.05%
217	Yes	No	0.1	2	6	0.9	1.2	0.5	2	1.74%
218	Yes	No	0.1	2	6	0.9	1.2	0.5	6	6.34%
219	Yes	No	0.1	2	6	0.9	1.2	10	2	0.00%
220	Yes	No	0.1	2	6	0.9	1.2	10	6	0.00%
221	Yes	No	0.1	2	6	0.9	4	0.5	2	4.16%
222	Yes	No	0.1	2	6	0.9	4	0.5	6	6.67%
223	Yes	No	0.1	2	6	0.9	4	10	2	4.78%
224	Yes	No	0.1	2	6	0.9	4	10	6	6.00%
225	Yes	No	0.1	6	2	0.3	1.2	0.5	2	9.47%
226	Yes	No	0.1	6	2	0.3	1.2	0.5	6	14.31%
227	Yes	No	0.1	6	2	0.3	1.2	10	2	3.79%
228	Yes	No	0.1	6	2	0.3	1.2	10	6	7.96%
229	Yes	No	0.1	6	2	0.3	4	0.5	2	14.62%
230	Yes	No	0.1	6	2	0.3	4	0.5	6	22.30%
231	Yes	No	0.1	6	2	0.3	4	10	2	6.77%
232	Yes	No	0.1	6	2	0.3	4	10	6	10.39%
233	Yes	No	0.1	6	2	0.9	1.2	0.5	2	2.33%
234	Yes	No	0.1	6	2	0.9	1.2	0.5	6	4.54%
235	Yes	No	0.1	6	2	0.9	1.2	10	2	0.00%
236	Yes	No	0.1	6	2	0.9	1.2	10	6	2.46%
237	Yes	No	0.1	6	2	0.9	4	0.5	2	4.26%
238	Yes	No	0.1	6	2	0.9	4	0.5	6	8.13%
239	Yes	No	0.1	6	2	0.9	4	10	2	4.38%
240	Yes	No	0.1	6	2	0.9	4	10	6	5.41%
241	Yes	No	0.1	6	6	0.3	1.2	0.5	2	9.24%
242	Yes	No	0.1	6	6	0.3	1.2	0.5	6	16.35%
243	Yes	No	0.1	6	6	0.3	1.2	10	2	3.11%
244	Yes	No	0.1	6	6	0.3	1.2	10	6	6.77%
245	Yes	No	0.1	6	6	0.3	4	0.5	2	15.72%
246	Yes	No	0.1	6	6	0.3	4	0.5	6	22.31%
247	Yes	No	0.1	6	6	0.3	4	10	2	7.10%
248	Yes	No	0.1	6	6	0.3	4	10	6	8.66%
249	Yes	No	0.1	6	6	0.9	1.2	0.5	2	3.99%
250	Yes	No	0.1	6	6	0.9	1.2	0.5	6	5.79%
251	Yes	No	0.1	6	6	0.9	1.2	10	2	0.00%
252	Yes	No	0.1	6	6	0.9	1.2	10	6	0.00%
253	Yes	No	0.1	6	6	0.9	4	0.5	2	4.31%
254	Yes	No	0.1	6	6	0.9	4	0.5	6	8.42%
255	Yes	No	0.1	6	6	0.9	4	10	2	3.42%
256	Yes	No	0.1	6	6	0.9	4	10	6	5.68%
257	No	Yes	0.02	2	2	0.3	1.2	0.5	2	0.49%
258	No	Yes	0.02	2	2	0.3	1.2	0.5	6	3.22%
259	No	Yes	0.02	2	2	0.3	1.2	10	2	0.00%
260	No	Yes	0.02	2	2	0.3	1.2	10	6	0.36%
261	No	Yes	0.02	2	2	0.3	4	0.5	2	4.42%
262	No	Yes	0.02	2	2	0.3	4	0.5	6	7.64%
263	No	Yes	0.02	2	2	0.3	4	10	2	1.74%
264	No	Yes	0.02	2	2	0.3	4	10	6	1.42%
265	No	Yes	0.02	2	2	0.9	1.2	0.5	2	0.75%
266	No	Yes	0.02	2	2	0.9	1.2	0.5	6	2.43%
267	No	Yes	0.02	2	2	0.9	1.2	10	2	0.00%
268	No	Yes	0.02	2	2	0.9	1.2	10	6	0.00%
269	No	Yes	0.02	2	2	0.9	4	0.5	2	0.91%
270	No	Yes	0.02	2	2	0.9	4	0.5	6	2.71%
271	No	Yes	0.02	2	2	0.9	4	10	2	0.00%
272	No	Yes	0.02	2	2	0.9	4	10	6	0.00%
273	No	Yes	0.02	2	6	0.3	1.2	0.5	2	1.39%
274	No	Yes	0.02	2	6	0.3	1.2	0.5	6	4.89%
275	No	Yes	0.02	2	6	0.3	1.2	10	2	0.00%
276	No	Yes	0.02	2	6	0.3	1.2	10	6	0.59%
277	No	Yes	0.02	2	6	0.3	4	0.5	2	5.01%
278	No	Yes	0.02	2	6	0.3	4	0.5	6	7.71%
279	No	Yes	0.02	2	6	0.3	4	10	2	0.61%
280	No	Yes	0.02	2	6	0.3	4	10	6	0.00%
281	No	Yes	0.02	2	6	0.9	1.2	0.5	2	0.00%
282	No	Yes	0.02	2	6	0.9	1.2	0.5	6	0.00%
283	No	Yes	0.02	2	6	0.9	1.2	10	2	0.46%
284	No	Yes	0.02	2	6	0.9	1.2	10	6	0.62%
285	No	Yes	0.02	2	6	0.9	4	0.5	2	0.78%
286	No	Yes	0.02	2	6	0.9	4	0.5	6	0.13%
287	No	Yes	0.02	2	6	0.9	4	10	2	0.28%

288	No	Yes	0.02	2	6	0.9	4	10	6	0.13%
289	No	Yes	0.02	6	2	0.3	1.2	0.5	2	0.36%
290	No	Yes	0.02	6	2	0.3	1.2	0.5	6	4.67%
291	No	Yes	0.02	6	2	0.3	1.2	10	2	0.00%
292	No	Yes	0.02	6	2	0.3	1.2	10	6	0.31%
293	No	Yes	0.02	6	2	0.3	4	0.5	2	4.77%
294	No	Yes	0.02	6	2	0.3	4	0.5	6	10.27%
295	No	Yes	0.02	6	2	0.3	4	10	2	0.93%
296	No	Yes	0.02	6	2	0.3	4	10	6	0.97%
297	No	Yes	0.02	6	2	0.9	1.2	0.5	2	0.40%
298	No	Yes	0.02	6	2	0.9	1.2	0.5	6	0.47%
299	No	Yes	0.02	6	2	0.9	1.2	10	2	0.77%
300	No	Yes	0.02	6	2	0.9	1.2	10	6	0.93%
301	No	Yes	0.02	6	2	0.9	4	0.5	2	0.55%
302	No	Yes	0.02	6	2	0.9	4	0.5	6	0.62%
303	No	Yes	0.02	6	2	0.9	4	10	2	0.85%
304	No	Yes	0.02	6	2	0.9	4	10	6	1.02%
305	No	Yes	0.02	6	6	0.3	1.2	0.5	2	0.00%
306	No	Yes	0.02	6	6	0.3	1.2	0.5	6	7.09%
307	No	Yes	0.02	6	6	0.3	1.2	10	2	0.00%
308	No	Yes	0.02	6	6	0.3	1.2	10	6	0.00%
309	No	Yes	0.02	6	6	0.3	4	0.5	2	5.43%
310	No	Yes	0.02	6	6	0.3	4	0.5	6	8.39%
311	No	Yes	0.02	6	6	0.3	4	10	2	1.39%
312	No	Yes	0.02	6	6	0.3	4	10	6	0.43%
313	No	Yes	0.02	6	6	0.9	1.2	0.5	2	3.23%
314	No	Yes	0.02	6	6	0.9	1.2	0.5	6	1.85%
315	No	Yes	0.02	6	6	0.9	1.2	10	2	0.46%
316	No	Yes	0.02	6	6	0.9	1.2	10	6	0.57%
317	No	Yes	0.02	6	6	0.9	4	0.5	2	1.39%
318	No	Yes	0.02	6	6	0.9	4	0.5	6	1.49%
319	No	Yes	0.02	6	6	0.9	4	10	2	0.00%
320	No	Yes	0.02	6	6	0.9	4	10	6	0.00%
321	No	Yes	0.1	2	2	0.3	1.2	0.5	2	6.56%
322	No	Yes	0.1	2	2	0.3	1.2	0.5	6	10.31%
323	No	Yes	0.1	2	2	0.3	1.2	10	2	2.51%
324	No	Yes	0.1	2	2	0.3	1.2	10	6	4.27%
325	No	Yes	0.1	2	2	0.3	4	0.5	2	6.77%
326	No	Yes	0.1	2	2	0.3	4	0.5	6	12.11%
327	No	Yes	0.1	2	2	0.3	4	10	2	4.26%
328	No	Yes	0.1	2	2	0.3	4	10	6	4.69%
329	No	Yes	0.1	2	2	0.9	1.2	0.5	2	1.98%
330	No	Yes	0.1	2	2	0.9	1.2	0.5	6	3.21%
331	No	Yes	0.1	2	2	0.9	1.2	10	2	0.22%
332	No	Yes	0.1	2	2	0.9	1.2	10	6	0.83%
333	No	Yes	0.1	2	2	0.9	4	0.5	2	3.51%
334	No	Yes	0.1	2	2	0.9	4	0.5	6	4.79%
335	No	Yes	0.1	2	2	0.9	4	10	2	1.89%
336	No	Yes	0.1	2	2	0.9	4	10	6	3.60%
337	No	Yes	0.1	2	6	0.3	1.2	0.5	2	4.55%
338	No	Yes	0.1	2	6	0.3	1.2	0.5	6	9.36%
339	No	Yes	0.1	2	6	0.3	1.2	10	2	2.01%
340	No	Yes	0.1	2	6	0.3	1.2	10	6	3.76%
341	No	Yes	0.1	2	6	0.3	4	0.5	2	10.07%
342	No	Yes	0.1	2	6	0.3	4	0.5	6	10.64%
343	No	Yes	0.1	2	6	0.3	4	10	2	4.96%
344	No	Yes	0.1	2	6	0.3	4	10	6	3.76%
345	No	Yes	0.1	2	6	0.9	1.2	0.5	2	1.03%
346	No	Yes	0.1	2	6	0.9	1.2	0.5	6	3.49%
347	No	Yes	0.1	2	6	0.9	1.2	10	2	0.44%
348	No	Yes	0.1	2	6	0.9	1.2	10	6	0.10%
349	No	Yes	0.1	2	6	0.9	4	0.5	2	2.70%
350	No	Yes	0.1	2	6	0.9	4	0.5	6	4.59%
351	No	Yes	0.1	2	6	0.9	4	10	2	3.30%
352	No	Yes	0.1	2	6	0.9	4	10	6	3.17%
353	No	Yes	0.1	6	2	0.3	1.2	0.5	2	4.84%
354	No	Yes	0.1	6	2	0.3	1.2	0.5	6	7.54%
355	No	Yes	0.1	6	2	0.3	1.2	10	2	2.03%
356	No	Yes	0.1	6	2	0.3	1.2	10	6	5.11%
357	No	Yes	0.1	6	2	0.3	4	0.5	2	8.90%
358	No	Yes	0.1	6	2	0.3	4	0.5	6	13.38%
359	No	Yes	0.1	6	2	0.3	4	10	2	3.26%
360	No	Yes	0.1	6	2	0.3	4	10	6	5.50%
361	No	Yes	0.1	6	2	0.9	1.2	0.5	2	1.04%
362	No	Yes	0.1	6	2	0.9	1.2	0.5	6	2.60%
363	No	Yes	0.1	6	2	0.9	1.2	10	2	0.00%
364	No	Yes	0.1	6	2	0.9	1.2	10	6	1.48%
365	No	Yes	0.1	6	2	0.9	4	0.5	2	2.68%

366	No	Yes	0.1	6	2	0.9	4	0.5	6	3.98%
367	No	Yes	0.1	6	2	0.9	4	10	2	3.00%
368	No	Yes	0.1	6	2	0.9	4	10	6	3.75%
369	No	Yes	0.1	6	6	0.3	1.2	0.5	2	5.44%
370	No	Yes	0.1	6	6	0.3	1.2	0.5	6	8.81%
371	No	Yes	0.1	6	6	0.3	1.2	10	2	1.58%
372	No	Yes	0.1	6	6	0.3	1.2	10	6	4.18%
373	No	Yes	0.1	6	6	0.3	4	0.5	2	9.30%
374	No	Yes	0.1	6	6	0.3	4	0.5	6	14.39%
375	No	Yes	0.1	6	6	0.3	4	10	2	4.03%
376	No	Yes	0.1	6	6	0.3	4	10	6	5.59%
377	No	Yes	0.1	6	6	0.9	1.2	0.5	2	2.04%
378	No	Yes	0.1	6	6	0.9	1.2	0.5	6	3.73%
379	No	Yes	0.1	6	6	0.9	1.2	10	2	0.00%
380	No	Yes	0.1	6	6	0.9	1.2	10	6	0.00%
381	No	Yes	0.1	6	6	0.9	4	0.5	2	2.74%
382	No	Yes	0.1	6	6	0.9	4	0.5	6	5.65%
383	No	Yes	0.1	6	6	0.9	4	10	2	2.02%
384	No	Yes	0.1	6	6	0.9	4	10	6	2.85%
385	No	No	0.02	2	2	0.3	1.2	0.5	2	1.14%
386	No	No	0.02	2	2	0.3	1.2	0.5	6	8.58%
387	No	No	0.02	2	2	0.3	1.2	10	2	0.00%
388	No	No	0.02	2	2	0.3	1.2	10	6	0.83%
389	No	No	0.02	2	2	0.3	4	0.5	2	10.94%
390	No	No	0.02	2	2	0.3	4	0.5	6	18.72%
391	No	No	0.02	2	2	0.3	4	10	2	4.52%
392	No	No	0.02	2	2	0.3	4	10	6	3.28%
393	No	No	0.02	2	2	0.9	1.2	0.5	2	0.00%
394	No	No	0.02	2	2	0.9	1.2	0.5	6	0.00%
395	No	No	0.02	2	2	0.9	1.2	10	2	0.00%
396	No	No	0.02	2	2	0.9	1.2	10	6	0.00%
397	No	No	0.02	2	2	0.9	4	0.5	2	0.23%
398	No	No	0.02	2	2	0.9	4	0.5	6	0.18%
399	No	No	0.02	2	2	0.9	4	10	2	0.50%
400	No	No	0.02	2	2	0.9	4	10	6	0.00%
401	No	No	0.02	2	6	0.3	1.2	0.5	2	0.00%
402	No	No	0.02	2	6	0.3	1.2	0.5	6	10.43%
403	No	No	0.02	2	6	0.3	1.2	10	2	0.00%
404	No	No	0.02	2	6	0.3	1.2	10	6	1.69%
405	No	No	0.02	2	6	0.3	4	0.5	2	11.72%
406	No	No	0.02	2	6	0.3	4	0.5	6	17.93%
407	No	No	0.02	2	6	0.3	4	10	2	1.31%
408	No	No	0.02	2	6	0.3	4	10	6	0.00%
409	No	No	0.02	2	6	0.9	1.2	0.5	2	1.61%
410	No	No	0.02	2	6	0.9	1.2	0.5	6	2.57%
411	No	No	0.02	2	6	0.9	1.2	10	2	0.33%
412	No	No	0.02	2	6	0.9	1.2	10	6	0.55%
413	No	No	0.02	2	6	0.9	4	0.5	2	1.39%
414	No	No	0.02	2	6	0.9	4	0.5	6	1.86%
415	No	No	0.02	2	6	0.9	4	10	2	0.00%
416	No	No	0.02	2	6	0.9	4	10	6	0.00%
417	No	No	0.02	6	2	0.3	1.2	0.5	2	1.01%
418	No	No	0.02	6	2	0.3	1.2	0.5	6	11.74%
419	No	No	0.02	6	2	0.3	1.2	10	2	0.00%
420	No	No	0.02	6	2	0.3	1.2	10	6	0.00%
421	No	No	0.02	6	2	0.3	4	0.5	2	13.02%
422	No	No	0.02	6	2	0.3	4	0.5	6	19.64%
423	No	No	0.02	6	2	0.3	4	10	2	2.30%
424	No	No	0.02	6	2	0.3	4	10	6	2.98%
425	No	No	0.02	6	2	0.9	1.2	0.5	2	1.08%
426	No	No	0.02	6	2	0.9	1.2	0.5	6	1.00%
427	No	No	0.02	6	2	0.9	1.2	10	2	1.73%
428	No	No	0.02	6	2	0.9	1.2	10	6	1.86%
429	No	No	0.02	6	2	0.9	4	0.5	2	1.56%
430	No	No	0.02	6	2	0.9	4	0.5	6	1.21%
431	No	No	0.02	6	2	0.9	4	10	2	2.01%
432	No	No	0.02	6	2	0.9	4	10	6	2.19%
433	No	No	0.02	6	6	0.3	1.2	0.5	2	0.00%
434	No	No	0.02	6	6	0.3	1.2	0.5	6	15.91%
435	No	No	0.02	6	6	0.3	1.2	10	2	0.00%
436	No	No	0.02	6	6	0.3	1.2	10	6	0.00%
437	No	No	0.02	6	6	0.3	4	0.5	2	14.90%
438	No	No	0.02	6	6	0.3	4	0.5	6	22.56%
439	No	No	0.02	6	6	0.3	4	10	2	0.00%
440	No	No	0.02	6	6	0.3	4	10	6	0.93%
441	No	No	0.02	6	6	0.9	1.2	0.5	2	0.61%
442	No	No	0.02	6	6	0.9	1.2	0.5	6	1.81%
443	No	No	0.02	6	6	0.9	1.2	10	2	0.00%

444	No	No	0.02	6	6	0.9	1.2	10	6	0.00%
445	No	No	0.02	6	6	0.9	4	0.5	2	1.35%
446	No	No	0.02	6	6	0.9	4	0.5	6	1.79%
447	No	No	0.02	6	6	0.9	4	10	2	0.00%
448	No	No	0.02	6	6	0.9	4	10	6	0.00%
449	No	No	0.1	2	2	0.3	1.2	0.5	2	14.75%
450	No	No	0.1	2	2	0.3	1.2	0.5	6	20.28%
451	No	No	0.1	2	2	0.3	1.2	10	2	5.95%
452	No	No	0.1	2	2	0.3	1.2	10	6	9.90%
453	No	No	0.1	2	2	0.3	4	0.5	2	22.93%
454	No	No	0.1	2	2	0.3	4	0.5	6	31.30%
455	No	No	0.1	2	2	0.3	4	10	2	11.45%
456	No	No	0.1	2	2	0.3	4	10	6	11.28%
457	No	No	0.1	2	2	0.9	1.2	0.5	2	4.42%
458	No	No	0.1	2	2	0.9	1.2	0.5	6	9.05%
459	No	No	0.1	2	2	0.9	1.2	10	2	0.00%
460	No	No	0.1	2	2	0.9	1.2	10	6	2.07%
461	No	No	0.1	2	2	0.9	4	0.5	2	7.50%
462	No	No	0.1	2	2	0.9	4	0.5	6	12.35%
463	No	No	0.1	2	2	0.9	4	10	2	4.50%
464	No	No	0.1	2	2	0.9	4	10	6	7.00%
465	No	No	0.1	2	6	0.3	1.2	0.5	2	12.52%
466	No	No	0.1	2	6	0.3	1.2	0.5	6	18.55%
467	No	No	0.1	2	6	0.3	1.2	10	2	4.39%
468	No	No	0.1	2	6	0.3	1.2	10	6	8.31%
469	No	No	0.1	2	6	0.3	4	0.5	2	20.85%
470	No	No	0.1	2	6	0.3	4	0.5	6	28.68%
471	No	No	0.1	2	6	0.3	4	10	2	12.21%
472	No	No	0.1	2	6	0.3	4	10	6	9.27%
473	No	No	0.1	2	6	0.9	1.2	0.5	2	1.92%
474	No	No	0.1	2	6	0.9	1.2	0.5	6	9.32%
475	No	No	0.1	2	6	0.9	1.2	10	2	0.00%
476	No	No	0.1	2	6	0.9	1.2	10	6	0.00%
477	No	No	0.1	2	6	0.9	4	0.5	2	5.58%
478	No	No	0.1	2	6	0.9	4	0.5	6	7.36%
479	No	No	0.1	2	6	0.9	4	10	2	7.05%
480	No	No	0.1	2	6	0.9	4	10	6	7.92%
481	No	No	0.1	6	2	0.3	1.2	0.5	2	13.86%
482	No	No	0.1	6	2	0.3	1.2	0.5	6	21.28%
483	No	No	0.1	6	2	0.3	1.2	10	2	5.83%
484	No	No	0.1	6	2	0.3	1.2	10	6	11.64%
485	No	No	0.1	6	2	0.3	4	0.5	2	19.64%
486	No	No	0.1	6	2	0.3	4	0.5	6	30.01%
487	No	No	0.1	6	2	0.3	4	10	2	9.56%
488	No	No	0.1	6	2	0.3	4	10	6	16.15%
489	No	No	0.1	6	2	0.9	1.2	0.5	2	3.31%
490	No	No	0.1	6	2	0.9	1.2	0.5	6	5.95%
491	No	No	0.1	6	2	0.9	1.2	10	2	0.00%
492	No	No	0.1	6	2	0.9	1.2	10	6	3.15%
493	No	No	0.1	6	2	0.9	4	0.5	2	6.48%
494	No	No	0.1	6	2	0.9	4	0.5	6	12.12%
495	No	No	0.1	6	2	0.9	4	10	2	6.07%
496	No	No	0.1	6	2	0.9	4	10	6	7.44%
497	No	No	0.1	6	6	0.3	1.2	0.5	2	14.18%
498	No	No	0.1	6	6	0.3	1.2	0.5	6	21.70%
499	No	No	0.1	6	6	0.3	1.2	10	2	3.53%
500	No	No	0.1	6	6	0.3	1.2	10	6	9.67%
501	No	No	0.1	6	6	0.3	4	0.5	2	21.79%
502	No	No	0.1	6	6	0.3	4	0.5	6	33.68%
503	No	No	0.1	6	6	0.3	4	10	2	9.32%
504	No	No	0.1	6	6	0.3	4	10	6	12.23%
505	No	No	0.1	6	6	0.9	1.2	0.5	2	5.69%
506	No	No	0.1	6	6	0.9	1.2	0.5	6	7.28%
507	No	No	0.1	6	6	0.9	1.2	10	2	0.00%
508	No	No	0.1	6	6	0.9	1.2	10	6	0.00%
509	No	No	0.1	6	6	0.9	4	0.5	2	5.14%
510	No	No	0.1	6	6	0.9	4	0.5	6	13.10%
511	No	No	0.1	6	6	0.9	4	10	2	4.63%
512	No	No	0.1	6	6	0.9	4	10	6	7.13%
513	No	No	0.06	4	4	0.6	2.6	5.25	4	4.33%
514	Yes	Yes	0.06	4	4	0.6	2.6	5.25	4	4.22%
515	No	Yes	0.06	4	4	0.6	2.6	5.25	4	4.40%
516	Yes	No	0.06	4	4	0.6	2.6	5.25	4	4.11%

Table B.6: MDP-based model for production lines: sensitivity analysis results, values displayed in Figure 4.6.

Experiment Nr.	**Configuration**	w_h [kW]	**EN Saving [%]**
1	WORST	0.5	0.50%
2	WORST	1	0.46%
3	WORST	2	0.33%
4	WORST	3	0.22%
5	WORST	4	0.11%
6	WORST	5	0.05%
7	WORST	6	0.00%
8	WORST	7	0.00%
9	WORST	8	0.00%
10	WORST	9	0.00%
11	WORST	10	0.00%
12	MEDIUM -1	0.5	8.55%
13	MEDIUM - 1	1	7.99%
14	MEDIUM - 1	2	7.55%
15	MEDIUM - 1	3	7.25%
16	MEDIUM - 1	4	5.86%
17	MEDIUM - 1	5	5.01%
18	MEDIUM - 1	6	4.55%
19	MEDIUM - 1	7	4.02%
20	MEDIUM - 1	8	3.55%
21	MEDIUM - 1	9	2.88%
22	MEDIUM - 1	10	1.21%
23	MEDIUM - 2	0.5	18.88%
24	MEDIUM - 2	1	17.77%
25	MEDIUM - 2	2	17.33%
26	MEDIUM - 2	3	16.88%
27	MEDIUM - 2	4	13.44%
28	MEDIUM - 2	5	12.21%
29	MEDIUM - 2	6	10.55%
30	MEDIUM - 2	7	9.66%
31	MEDIUM - 2	8	8.77%
32	MEDIUM - 2	9	6.31%
33	MEDIUM - 2	10	4.33%
34	BEST	0.5	31.30%
35	BEST	1	30.44%
36	BEST	2	30.00%
37	BEST	3	29.55%
38	BEST	4	25.55%
39	BEST	5	22.88%
40	BEST	6	19.77%
41	BEST	7	15.66%
42	BEST	8	13.88%
43	BEST	9	12.55%
44	BEST	10	11.28%

Experiment Nr.	**Configuration**	***PCR***	**EN Saving [%]**
45	WORST	1.2	0.00%
46	WORST	2	0.00%
47	WORST	2.5	0.00%
48	WORST	3	0.00%
49	WORST	3.5	0.00%
50	WORST	4	0.00%
51	MEDIUM - 1	1.2	0.11%
52	MEDIUM - 1	2	0.44%
53	MEDIUM - 1	2.5	1.22%

54	MEDIUM - 1	3	2.33%
55	MEDIUM - 1	3.5	3.55%
56	MEDIUM - 1	4	4.33%
57	MEDIUM - 2	1.2	5.85%
58	MEDIUM - 2	2	6.66%
59	MEDIUM - 2	2.5	7.01%
60	MEDIUM - 2	3	8.99%
61	MEDIUM - 2	3.5	9.55%
62	MEDIUM - 2	4	10.44%
63	BEST	1.2	20.28%
64	BEST	2	21.55%
65	BEST	2.5	23.66%
66	BEST	3	26.66%
67	BEST	3.5	28.66%
68	BEST	4	31.30%

Experiment Nr.	**Configuration**	δ	**EN Saving [%]**
69	WORST	0.02	0.00%
70	WORST	0.02	0.00%
71	WORST	0.03	0.00%
72	WORST	0.03	0.00%
73	WORST	0.05	0.00%
74	WORST	0.10	0.00%
75	MEDIUM - 1	0.02	0.00%
76	MEDIUM - 1	0.02	0.22%
77	MEDIUM - 1	0.03	1.55%
78	MEDIUM - 1	0.03	3.33%
79	MEDIUM - 1	0.05	4.55%
80	MEDIUM - 1	0.10	7.88%
81	MEDIUM - 2	0.02	0.00%
82	MEDIUM - 2	0.02	1.22%
83	MEDIUM - 2	0.03	4.55%
84	MEDIUM - 2	0.03	6.77%
85	MEDIUM - 2	0.05	9.66%
86	MEDIUM - 2	0.10	13.55%
87	BEST	0.02	12.11%
88	BEST	0.02	18.72%
89	BEST	0.03	21.33%
90	BEST	0.03	23.44%
91	BEST	0.05	27.66%
92	BEST	0.10	31.30%

Experiment Nr.	**Configuration**	K	**EN Saving [%]**
93	WORST	2	0.00%
94	WORST	3	0.00%
95	WORST	4	0.00%
96	WORST	5	0.00%
97	WORST	6	0.00%
98	WORST	7	0.00%
99	WORST	8	0.00%
100	WORST	9	0.00%
101	WORST	10	0.00%
102	MEDIUM - 1	2	1.33%
103	MEDIUM - 1	3	2.44%
104	MEDIUM - 1	4	5.86%
105	MEDIUM - 1	5	6.66%
106	MEDIUM - 1	6	7.77%
107	MEDIUM - 1	7	8.01%
108	MEDIUM - 1	8	9.02%
109	MEDIUM - 1	9	10.11%

110	MEDIUM - 1	10	11.12%
111	MEDIUM - 2	2	10.44%
112	MEDIUM - 2	3	13.33%
113	MEDIUM - 2	4	17.77%
114	MEDIUM - 2	5	18.55%
115	MEDIUM - 2	6	19.33%
116	MEDIUM - 2	7	21.22%
117	MEDIUM - 2	8	22.33%
118	MEDIUM - 2	9	25.55%
119	MEDIUM - 2	10	26.66%
120	BEST	2	22.93%
121	BEST	3	24.33%
122	BEST	4	25.66%
123	BEST	5	28.66%
124	BEST	6	31.30%
125	BEST	7	33.00%
126	BEST	8	34.55%
127	BEST	9	35.55%
128	BEST	10	36.55%

B.3 Detailed Results for Chapter 5

Table B.7: RL-based model for single workstations: comparison of different agents. Values displayed in Figure 5.3.

Agent	ϕ	**TH Loss [%]**	EN Saving [%]
Tensorforce - General	0.00	100.00% ± 0.00%	100.00% ± 0.00%
Tensorforce - General	0.50	58.19% ± 0.58%	63.25% ± 0.63%
Tensorforce - General	0.75	12.51% ± 0.13%	24.94% ± 0.25%
Tensorforce - General	0.90	3.28% ± 0.03%	12.79% ± 0.13%
Tensorforce - General	0.95	1.67% ± 0.02%	7.79% ± 0.08%
Tensorforce - General	1.00	0.00% ± 0.00%	0.00% ± 0.00%
PPO	0.00	100.00% ± 0.00%	100.00% ± 0.00%
PPO	0.50	72.98% ± 0.73%	71.24% ± 0.71%
PPO	0.75	17.06% ± 0.17%	17.11% ± 0.17%
PPO	0.90	4.21% ± 0.04%	8.81% ± 0.09%
PPO	0.95	2.34% ± 0.02%	9.35% ± 0.09%
PPO	1.00	0.00% ± 0.00%	0.00% ± 0.00%
TRPO	0.00	100.00% ± 0.00%	100.00% ± 0.00%
TRPO	0.50	83.01% ± 0.83%	82.02% ± 0.82%
TRPO	0.75	26.62% ± 0.27%	24.39% ± 0.24%
TRPO	0.90	13.58% ± 0.14%	11.98% ± 0.12%
TRPO	0.95	4.62% ± 0.05%	9.08% ± 0.09%
TRPO	1.00	0.00% ± 0.00%	0.00% ± 0.00%
DQN	0.00	100.00% ± 0.00%	100.00% ± 0.00%
DQN	0.50	62.81% ± 0.63%	70.31% ± 0.70%
DQN	0.75	21.81% ± 0.22%	36.96% ± 0.37%
DQN	0.90	8.83% ± 0.09%	26.11% ± 0.26%
DQN	0.95	1.27% ± 0.05%	7.00% ± 0.07%
DQN	1.00	0.00% ± 0.00%	0.00% ± 0.00%

Table B.8: RL-based model for single workstations: comparison of different ϕ-values. Values displayed in Figure 5.4.

ϕ	TH Loss [%]	EN Saving [%]
0.93	4.82% ± 0.34%	10.87% ± 0.76%
0.94	3.34% ± 0.23%	9.92% ± 0.69%
0.95	1.27% ± 0.05%	7.00% ± 0.07%
0.96	1.27% ± 0.09%	7.62% ± 0.12%
0.97	1.20% ± 0.05%	7.72% ± 0.10%
0.98	0.54% ± 0.04%	2.72% ± 0.19%
0.99	0.40% ± 0.03%	2.46% ± 0.01%
1	0.00% ± 0.00%	0.00% ± 0.00%

Table B.9: RL-based model for single workstations: 2^k factorial analysis results. Values displayed in Figure 5.7.

Experiment Nr.	δ	c	ρ	*PCR*	K	EN Saving [%]	TH Loss [%]
1	0.100	4	0.75	1.2	5	7.23% ± 0.11%	3.52% ± 0.05%
2	0.016	4	0.75	1.2	5	5.64% ± 0.08%	1.76% ± 0.03%
3	0.100	8	0.75	1.2	5	6.96% ± 0.10%	2.07% ± 0.03%
4	0.016	8	0.75	1.2	5	4.75% ± 0.07%	1.04% ± 0.02%
5	0.100	4	0.95	1.2	5	4.65% ± 0.07%	2.54% ± 0.04%
6	0.016	4	0.95	1.2	5	5.62% ± 0.08%	1.53% ± 0.02%
7	0.100	8	0.95	1.2	5	4.58% ± 0.07%	2.62% ± 0.04%
8	0.016	8	0.95	1.2	5	5.04% ± 0.08%	0.75% ± 0.01%
9	0.100	4	0.75	4	5	9.43% ± 0.14%	3.17% ± 0.05%
10	0.016	4	0.75	4	5	7.31% ± 0.11%	2.46% ± 0.04%
11	0.100	8	0.75	4	5	8.28% ± 0.12%	2.07% ± 0.03%
12	0.016	8	0.75	4	5	7.64% ± 0.11%	1.74% ± 0.03%
13	0.100	4	0.95	4	5	8.83% ± 0.13%	1.90% ± 0.03%
14	0.016	4	0.95	4	5	8.76% ± 0.13%	1.15% ± 0.02%
15	0.100	8	0.95	4	5	9.79% ± 0.15%	1.87% ± 0.03%
16	0.016	8	0.95	4	5	9.85% ± 0.15%	0.75% ± 0.01%
17	0.100	4	0.75	1.2	20	9.03% ± 0.14%	1.67% ± 0.03%
18	0.016	4	0.75	1.2	20	7.16% ± 0.11%	3.36% ± 0.05%
19	0.100	8	0.75	1.2	20	7.38% ± 0.11%	1.68% ± 0.03%
20	0.016	8	0.75	1.2	20	7.06% ± 0.11%	2.68% ± 0.04%
21	0.100	4	0.95	1.2	20	5.29% ± 0.08%	1.03% ± 0.02%
22	0.016	4	0.95	1.2	20	5.57% ± 0.08%	1.03% ± 0.02%
23	0.100	8	0.95	1.2	20	6.32% ± 0.09%	1.37% ± 0.02%
24	0.016	8	0.95	1.2	20	6.19% ± 0.09%	0.68% ± 0.01%
25	0.100	4	0.75	4	20	9.21% ± 0.14%	2.01% ± 0.03%
26	0.016	4	0.75	4	20	8.61% ± 0.13%	1.34% ± 0.02%
27	0.100	8	0.75	4	20	8.96% ± 0.13%	1.01% ± 0.02%
28	0.016	8	0.75	4	20	8.89% ± 0.13%	1.34% ± 0.02%
29	0.100	4	0.95	4	20	7.70% ± 0.12%	1.03% ± 0.02%
30	0.016	4	0.95	4	20	7.94% ± 0.12%	2.75% ± 0.04%
31	0.100	8	0.95	4	20	9.65% ± 0.14%	3.08% ± 0.05%
32	0.016	8	0.95	4	20	8.65% ± 0.13%	3.42% ± 0.05%
33	0.058	6	0.85	2.6	12	7.44% ± 0.11%	1.90% ± 0.03%

B.4 Detailed Results for Chapter 6

Table B.10: RL-based model for production lines: comparison of different agents. Values displayed in Figure 6.4.

ϕ	**Agent**	**TH Loss [%]**	EN Saving [%]
0.95	Tensorforce - General	4.93% ± 0.35%	6.51% ± 0.46%
0.97	Tensorforce - General	1.83% ± 0.18%	5.52% ± 0.55%
1	Tensorforce - General	0.00% ± 0.00%	0.00% ± 0.00%
0.90	PPO	8.59% ± 0.40%	12.88% ± 0.40%
0.95	PPO	4.08% ± 0.29%	9.02% ± 0.40%
0.97	PPO	1.27% ± 0.20%	6.51% ± 0.33%
1	PPO	0.00% ± 0.00%	0.00% ± 0.00%
0.90	TRPO	10.99% ± 0.77%	6.73% ± 0.40%
0.95	TRPO	5.63% ± 0.39%	5.34% ± 0.37%
0.97	TRPO	5.49% ± 0.55%	3.53% ± 0.33%
1	TRPO	0.00% ± 0.00%	0.00% ± 0.00%
0.90	DQN	9.15% ± 0.64%	6.73% ± 0.40%
0.95	DQN	4.93% ± 0.35%	5.34% ± 0.37%
0.97	DQN	2.11% ± 0.21%	3.53% ± 0.33%
1	DQN	0.00% ± 0.00%	0.00% ± 0.00%

Table B.11: RL-based model for production lines: comparison of different ϕ-values. Values displayed in Figure 6.5.

ϕ	**TH Loss [%]**	**EN Saving [%]**
0.95	4.08% ± 0.29%	9.02% ± 0.40%
0.96	3.52% ± 0.43%	8.89% ± 0.55%
0.97	1.27% ± 0.20%	6.51% ± 0.33%
0.98	0.85% ± 0.10%	5.79% ± 0.44%
0.99	0.70% ± 0.10%	2.91% ± 0.55%
1	0.00% ± 0.00%	0.00% ± 0.00%

Table B.12: Multiple RL agents for production lines: comparison of different ϕ-values. Values displayed in Figure 6.9.

ϕ	**TH Loss [%]**	**EN Saving [%]**
0.95	11.13% ± 0.39%	16.24% ± 0.57%
0.96	10.28% ± 0.36%	14.67% ± 0.51%
0.97	8.87% ± 0.31%	14.53% ± 0.51%
0.98	4.51% ± 0.16%	9.18% ± 0.32%
0.99	2.11% ± 0.07%	4.91% ± 0.17%
1	0.00% ± 0.00%	0.00% ± 0.00%

Table B.13: RL-based model for production lines: results for the 5-stages experiments. Values displayed in Figure 6.7.

Experiment Nr.	**Stage Sequence**	**TH Loss [%]**	**EN Saving [%]**
1	B-C-B-C-A	2.59% ± 0.08%	7.95% ± 0.14%
2	B-B-B-B-A	2.46% ± 0.07%	6.68% ± 0.12%
3	C-B-A-B-C	1.67% ± 0.05%	5.66% ± 0.10%

4	A-C-C-C-B	1.72% ± 0.05%	5.28% ± 0.09%
5	B-A-C-B-B	2.16% ± 0.06%	5.43% ± 0.10%
6	A-C-C-C-C	2.59% ± 0.08%	4.81% ± 0.08%
7	A-C-B-A-A	2.59% ± 0.08%	6.28% ± 0.11%
8	A-A-A-C-B	2.93% ± 0.09%	6.56% ± 0.11%
9	C-C-A-A-A	1.68% ± 0.05%	5.18% ± 0.09%
10	A-B-A-C-A	1.90% ± 0.06%	4.92% ± 0.09%
11	C-A-A-B-B	1.67% ± 0.05%	4.55% ± 0.08%
12	B-A-C-A-A	1.29% ± 0.04%	5.41% ± 0.09%
13	A-B-A-B-B	2.10% ± 0.06%	6.91% ± 0.12%
14	C-B-B-A-B	2.08% ± 0.06%	7.28% ± 0.13%
15	C-C-C-A-A	2.69% ± 0.08%	5.28% ± 0.09%
16	C-A-B-B-B	2.50% ± 0.08%	5.38% ± 0.09%
17	A-A-B-B-C	2.75% ± 0.08%	4.48% ± 0.08%
18	A-B-C-C-A	2.76% ± 0.08%	4.09% ± 0.07%
19	C-B-B-A-B	1.67% ± 0.05%	6.47% ± 0.11%
20	B-C-B-A-C	2.59% ± 0.08%	4.18% ± 0.07%
21	B-A-C-A-C	3.45% ± 0.10%	5.25% ± 0.09%
22	C-A-B-A-C	1.25% ± 0.04%	3.94% ± 0.07%
23	B-B-B-C-A	1.29% ± 0.04%	6.64% ± 0.12%
24	C-A-C-B-B	1.71% ± 0.05%	4.17% ± 0.07%
25	B-A-A-A-C	0.83% ± 0.03%	3.59% ± 0.06%

Table B.14: RL-based model for production lines: results for the 15-stages experiments. Values displayed in Figure 6.8.

Experiment Nr.	Stage Sequence	TH Loss [%]	EN Saving [%]
1	B-C-B-C-A	2.59% ± 0.08%	7.95% ± 0.14%
2	B-B-B-B-A	2.46% ± 0.07%	6.68% ± 0.12%
3	C-B-A-B-C	1.67% ± 0.05%	5.66% ± 0.10%
4	A-C-C-C-B	1.72% ± 0.05%	5.28% ± 0.09%
5	B-A-C-B-B	2.16% ± 0.06%	5.43% ± 0.10%
6	A-C-C-C-C	2.59% ± 0.08%	4.81% ± 0.08%
7	A-C-B-A-A	2.59% ± 0.08%	6.28% ± 0.11%
8	A-A-A-C-B	2.93% ± 0.09%	6.56% ± 0.11%
9	C-C-A-A-A	1.68% ± 0.05%	5.18% ± 0.09%
10	A-B-A-C-A	1.90% ± 0.06%	4.92% ± 0.09%
11	C-A-A-B-B	1.67% ± 0.05%	4.55% ± 0.08%
12	B-A-C-A-A	1.29% ± 0.04%	5.41% ± 0.09%
13	A-B-A-B-B	2.10% ± 0.06%	6.91% ± 0.12%
14	C-B-B-A-B	2.08% ± 0.06%	7.28% ± 0.13%
15	C-C-C-A-A	2.69% ± 0.08%	5.28% ± 0.09%
16	C-A-B-B-B	2.50% ± 0.08%	5.38% ± 0.09%
17	A-A-B-B-C	2.75% ± 0.08%	4.48% ± 0.08%
18	A-B-C-C-A	2.76% ± 0.08%	4.09% ± 0.07%
19	C-B-B-A-B	1.67% ± 0.05%	6.47% ± 0.11%
20	B-C-B-A-C	2.59% ± 0.08%	4.18% ± 0.07%
21	B-A-C-A-C	3.45% ± 0.10%	5.25% ± 0.09%
22	C-A-B-A-C	1.25% ± 0.04%	3.94% ± 0.07%
23	B-B-B-C-A	1.29% ± 0.04%	6.64% ± 0.12%
24	C-A-C-B-B	1.71% ± 0.05%	4.17% ± 0.07%
25	B-A-A-A-C	0.83% ± 0.03%	3.59% ± 0.06%