U.PORTO

FEP FACULDADE DE ECONOMIA
UNIVERSIDADE DO PORTO

Energy-Aware Job Shop Scheduling: Optimizing Production Efficiency through Machine Power Saving Strategies

Pedro Sollari Allegro de Soveral Casal

Dissertation

Master in Economics of Business and Strategy

Supervised by
Dalila Martins Fontes

2023

Acknowledgments

First and foremost, I want to express my gratitude to all the individuals who consistently showed concern for the progress and completion of this dissertation, with special mention to my mother, Avó Laia, Tete, Carolina, and Sofia. I'd also like to acknowledge those who, despite my laid-back nature, fully trusted my progress, particularly my father, Gui, and Manelinho.

I extend my thanks to my supervisor, Dalila Fontes, for her guidance in selecting the topic, for addressing my doubts whenever they arose, and for granting me the freedom that I desired to complete this work.

Lastly, I want to offer a special note of appreciation to my friends who accompanied me throughout these two years master's program, Coutinho, Rita, Angela, and Kika, as well as to all my family and friends outside of the university who consistently provided me with support, laughter, and wonderful moments.

Abstract

The purpose of this work is to assess the feasibility and effectiveness of integrating a standby mode strategy into manufacturing processes as a means of reducing energy consumption during idle times while maintaining the total completion time of operations (makespan). The research aims to determine the potential energy savings and contribute to the development of energy-efficient manufacturing practices.

With the aim of establishing an efficient operation schedule, the study focuses on the Job Shop Scheduling Problem with a single objective model that minimizes the makespan, solved with IBM ILOG CPLEX Optimization Studio. Subsequently, following the computation of pertinent data related to the implementation of standby mode and the determination of the breakeven idle time thresholds for each machine, the idle intervals initially identified will be reevaluated and substituted by standby intervals, when considered energy efficient.

The calculations reveal a promising outlook. The incorporation of standby mode has the potential to deliver significant energy savings, estimated at approximately 46% of the idle energy and 4.2% of the total energy, encompassing both processing and idle energy. A sensitivity analysis reinforces the reliability of these results, showing minor variations. Furthermore, these findings surpass the performance of a bi-objective model that minimizes both makespan and idle time.

This research stands out for its specialized focus on integrating standby mode for energy conservation within job shop scheduling, effectively merging optimization techniques with sustainability considerations. This area is increasingly attracting more interest due to the growing awareness of resource shortage.

Despite the positive findings, some limitations should be considered. Firstly, assumptions and simplifications within the optimization models could diverge from real-world complexities. Moreover, variability in machine characteristics due to the manufacturing environment and the influence of external factors are not fully accounted for. Finally, time and resource constraints also played a role in the study's methodology.

Resumo

O objetivo deste trabalho consiste em avaliar a eficácia da integração de uma estratégia de *standby* nos processos de fabrico como forma de reduzir o consumo energético durante os tempos de inatividade, mantendo constante o tempo total de conclusão das operações (*makespan*).

Com o intuito de estabelecer um "horário" de operações eficiente, é desenvolvido um Problema de *Job Shop Scheduling* (JSP) com um modelo de objetivo único que minimiza o *makespan*, resolvido com recurso ao software IBM ILOG CPLEX Optimization Studio. Após a integração dos dados relacionados com a implementação do modo *standby* e a determinação dos pontos de equilíbrio (*breakeven*), a partir do qual é considerado vantajoso ativar o modo *standby*, os intervalos de inatividade são reavaliados e substituídos por tempos *standby*, quando considerado energeticamente eficiente.

Os cálculos revelam uma perspetiva promissora. A incorporação do modo de *standby* tem o potencial de proporcionar poupanças de energia estimadas em aproximadamente 46% da energia de períodos de inatividade e 4.2% da energia total, que abrange energia de processamento e de inatividade. A análise de sensibilidade reforça a confiabilidade dos resultados, revelando variações mínimas. Para além disso, os resultados superam os de um modelo que visa minimizar simultaneamente o *makespan* e os tempos de inatividade.

Esta pesquisa destaca-se pelo seu foco na integração do modo *standby* para conservação de energia, unindo técnicas de otimização com considerações de sustentabilidade, área cujo interesse tem vindo a aumentar em resultado da crescente conscientização relacionada com a escassez de recursos energéticos.

Apesar das descobertas positivas, algumas limitações devem ser tomadas em consideração. As simplificações nos modelos de otimização não refletem inteiramente as complexidades do mundo real e a variabilidade nas características das máquinas e a influência de fatores externos não são totalmente consideradas. Finalmente, limitações a nível de tempo e recursos também desempenharam um papel restritivo.

Table of Contents

Figure Index

Graph Index

Table Index

1. Introduction

In today's rapidly evolving manufacturing landscape, the need to reduce energy consumption is not only an environmental concern but also a strategic requirement, as industries worldwide deal with the two-fold challenge of sustainability and competitiveness.

In the area of manufacturing operations, the Job Shop Scheduling Problem (JSP), which consists of assigning a set of jobs with specific processing requirements to machines, has long been a focal point of optimization efforts. However, in the face of these escalating environmental concerns and energy constraints, the Green Job Shop Scheduling Problem (GSSP) has emerged as a critical extension of research and industrial focus, mainly due to increasing environmental awareness, rising energy costs, regulatory pressures, and corporate sustainability initiatives. The Energy Efficient Job Shop Scheduling Problem (EEJSP), a topic deeply interconnected to the GSSP, extends beyond traditional scheduling objectives, incorporating the imperative to minimizing energy consumption and reducing associated costs within manufacturing operations.

The focus of this dissertation will consist of analyzing if energy savings can be obtained with the incorporation of a machine power saving strategy, namely switching machines to standby mode while idle, without affecting the total completion time of the operations. These seemingly irrelevant gaps in production have far-reaching implications, affecting energy efficiency, cost management, and overall operational competitiveness. According to Mouzon (2008) a lot of energy is wasted when machines are left idle due to underutilization in manufacturing processes, as they are consuming energy but not producing.

The study involves utilizing IBM ILOG CPLEX Optimization Studio, a software specialized in mathematical optimization to solve complex optimization models, obtaining an efficient schedule of operations. Subsequently, after the computation of relevant data concerning the introduction of a standby mode and determining the breakeven idle time threshold for each machine, at which switching to standby becomes energy-efficient, the idle periods within the initially calculated schedule will be further analyzed. If viable, these idle times will be reassessed and replaced with standby intervals, that is, the machines will transition into a state where they remain operational but consume significantly less energy by powering down specific components. After applying this method to every machine involved in the JSP,

the energy savings, both in terms of idle energy and total energy, i.e., processing, and idle energy, will be calculated and illustrated.

Structurally, this dissertation is divided in 4 chapters, as outlined below. Chapter 1 serves as an introductory outline to the proposed topic of study. Chapter 2 offers a detailed literature review, focusing on analyzing the JSP, the GSSP, the EEJSP, the power savings mechanism during idle times and the solution approaches to the scheduling problem. Chapter 3 introduces the model and variables that will be employed and presents the data under examination. Furthermore, it presents and discusses the results obtained and reports on sensitivity analysis related to the data used for incorporating the standby mode. Finally, it also encompasses an alternative bi-objective model in which idle times are also minimized. The last and final chapter, Chapter 4, contains the conclusion.

2. Literature Review

2.1. The Classic Job Shop Scheduling Problem

The classic Job Shop Problem (JSP) is a well-known combinatorial optimization problem in the field of operations research and scheduling (Taillard, 1993). The scheduling problem involves the creation of a strategy for organizing the timing of specific tasks to meet the specified sequencing constraints (Manne, 1960), which dictate the order in which jobs or tasks are scheduled on machines or workstations, and equipment interference problems, such as scheduling constraints that arise when two tasks or jobs cannot be executed simultaneously due to a limitation imposed by the availability of a shared resource.

The JSP is defined by several fundamental characteristics, including Jobs, Machines, Operations, and Constraints. In this problem, there exists a set of jobs denoted as J = {J1, J2, …, Jn}, where 'n' represents the total number of jobs. Additionally, there is a set of machines, denoted as M = {M1, M2, …, Mw}, where 'w' represents the total number of available machines.

Each individual job, Jj (where j ∈ [1, n]), consists of a predetermined sequence of operations, Oij (where i ∈ [1, k]), with 'k' representing the total number of operations for each job. It is essential to note that each operation within a job must be executed on a specific machine, Mk.

Furthermore, the JSP incorporates two crucial types of constraints. First, there are precedence constraints that dictate the sequential order in which operations within a job must be executed. Second, there are machine constraints, which stipulate that each machine can handle only one operation at a time, and no interruptions are permitted during the execution of operations. These constraints play a pivotal role in shaping the complexity and nature of the Job Shop Problem. An additional constraint specifies that each job can only be processed by a single machine at any given time, i.e., the constraint dictates that a job cannot undergo processing simultaneously by multiple machines at any point in the schedule.

The primary objective of solving the basic Job Shop Scheduling Problem is to find an optimized schedule that assigns operations to machines in a manner that minimizes the overall makespan, representing the completion time of the last job (Pinedo, 2012).

The Job Shop Scheduling Problem is classified as an NP-hard problem, implying that finding an optimal solution for large instances becomes computationally demanding (Taillard, 1993).

The Job Shop Scheduling Problem is found in practical applications in diverse domains, including manufacturing process scheduling, production line planning, project management, and resource allocation in various industries. Efficiently solving the JSP can lead to enhanced productivity, reduced production costs, and optimal resource utilization.

According to Fernandes et al. (2022), in a job shop manufacturing environment, products are produced in small quantities, and each product or order requires a unique set of operations. This means that different machines, tools, and materials are used for each job, making the production process highly flexible and adaptable to varying production requirements. Moreover, the job shop production process is known for its exceptional flexibility and customization. The workflow revolves around individual jobs, each scheduled and processed separately, often following distinct routes through the production system. This type of manufacturing environment is frequently encountered in industries such as aerospace, automotive, and machine tool manufacturing, given that they are characterized for having a high demand for customized or low-volume products, but you can also see job shop manufacturing in smaller production settings like custom fabrication shops, machine shops, and repair facilities. It is also worth mentioning that the job shop manufacturing environment faces various challenges related to scheduling, planning, and control. The product mix and routing's variability demand a flexible and responsive scheduling system capable of handling unexpected changes and disruptions.

To solve the JSP, it is necessary to obtain the sequence of operations on each machine, as well as the processing starting time of each operation.

2.2. The Green Job Shop Scheduling Problem

The development of industry, in addition to an enormous economic and social development, caused a huge consumption of energy and resources. Nowadays, mankind faces unprecedented challenges related to global energy, resources, and climate change. To accomplish sustainable development, it is required that manufacturing adopts more efficient methods and technologies that enables energy saving and emission reduction.

In 2022, the industrial sector (factories) in the United States accounted for a significant portion of the country's total energy consumption, amounting to approximately 35% of the energy used, according to data from the United States Energy Information Administration (EIA, 2023).

Furthermore, 2022 witnessed a notable increase in average energy prices, which can be attributed to several factors, including rising demand, supply constraints, and geopolitical tensions such as the conflict in Ukraine. The EIA reported an average energy price increase of approximately 17.7% during the same period. Specifically, energy costs surged from 7.18 cents per kilowatt-hour (cent/kWh) in 2021 to 8.45 cent/kWh in 2022, as can be seen in Figure 1. The global impact of this surge in energy prices underscores the economic significance of adopting energy-efficient production processes.

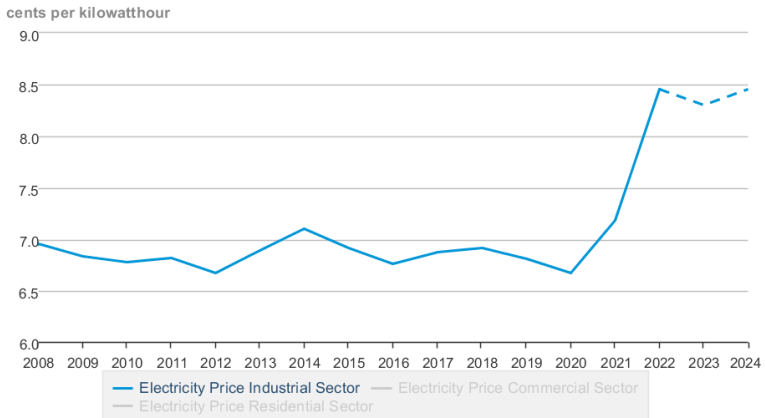


Figure 1. Average Electricity Price in the Industrial Sector in the United States from 2008 to 2024.

Source: U.S. Energy Information Administration (2023)

Adding an economic perspective to this scenario, data from the World Bank reveals that the inflation rate in 2022, representing the annual percentage change in consumer prices, stood at 8%. In fact, the increase in energy prices outpaced the rise in consumer prices by a significant margin, specifically by 9.7 percentage points. This highlights the disproportionate impact of rising energy costs compared to general consumer price inflation and it emphasizes the significance of developing strategies aimed at ensuring stable and sustainable energy expenses.

The sphere of green scheduling, which aims to assign jobs to machines in a way that

minimizes total costs, with a specific emphasis on promoting sustainable energy use, has been thoroughly evaluated as an effective approach for reducing energy consumption.

The green shop scheduling problem is an extension of the job shop scheduling problem that considers environmental factors like minimizing energy consumption, reducing greenhouse gas emissions, and optimizing the use of renewable resources (Hassini et al., 2013). Its objective is to create a production schedule that minimizes environmental impact while still meeting the operational constraints and goals of the manufacturing process. Addressing the green shop scheduling problem involves dealing with various conflicting objectives, such as reducing makespan, energy consumption, and carbon footprint.

The significance of the green shop scheduling problem has become increasingly apparent in today's world, with manufacturing industries facing growing pressure to enhance sustainability and minimize their environmental impact. Developing effective solutions for green shop scheduling not only helps achieve these environmental objectives but also improves operational efficiency and reduces costs.

In their 2022 study, Li and Wang discuss two primary strategies for enhancing energy efficiency and reducing emissions in the manufacturing sector. The initial approach involves directing efforts towards the design of products or machinery that demand reduced energy consumption and emit fewer pollutants. However, it is important to note that this approach implies substantial human resources, significant capital investments, and prolonged development timelines. The second option, as outlined by Li and Wang (2022), revolves around the implementation of green shop scheduling technology. This approach holds the promise of significantly enhancing energy efficiency, potentially at a more economical cost. In comparison to the first strategy, green shop scheduling technology represents a more resource-efficient and expedited means to achieve these environmental and efficiency goals (Li & Wang, 2022).

When contrasting the traditional Job Shop Scheduling Problem (JSPs) with the Green Shop Scheduling Problems (GSSPs), it becomes evident that GSSPs place a stronger emphasis on resource and environmental considerations. JSPs typically prioritize the optimization of economic factors like production time and cost, often without considering energy consumption and its resulting environmental implications.

In contrast, GSSPs have a dual objective of pursuing both economic and eco-friendly goals. They aim to enhance productivity while concurrently reducing energy usage and emissions of pollutants. This multifaceted approach is achieved through strategic resource allocation, the optimization of operational methods, and the meticulous organization of job sequences.

The objectives to be optimized by the GSSP can be divided into economic objectives, which aim to increase economic efficiency, and green objectives, which aim to attain energy conservation and environmental protection. As expected, economic and green objectives are mostly conflicting (Li & Wang, 2022).

For example, Gong et al. (2018) conducted research on a Flexible Job-shop problem with objectives centered around minimizing total worker cost, makespan, and maximizing a green production indicator. This green indicator considers various factors, including energy consumption, noise emissions, recycling of tool chips, and the safety of operations, reflecting a comprehensive approach to sustainability in production. The factors related to the green indicator are considered green objectives and likely conflict with the economic efficiency.

An associated concept with GSSP is Industry 4.0, which entails the incorporation of advanced digital technologies into manufacturing processes. Industry 4.0 empowers data-driven decision-making, boosts energy efficiency, optimizes resource allocation, and streamlines environmental reporting within manufacturing operations. According to Lu (2017), the objective is to establish a profoundly interconnected and smart production ecosystem.

It aims to revolutionize traditional manufacturing processes by leveraging technologies such as the Internet of Things (IoT) and Artificial Intelligence (AI), to enable a new level of automation, efficiency, and flexibility.

Li and Wang (2022) described the general process for solving GSSPs, which consists of two main steps. Firstly, it is necessary to establish a mathematical model that describes the objectives and its constraints. Subsequently, an appropriate solution algorithm is used to solve the problem.

As explained in Hidri et al. (2021), in the context of GSSP there are two viable approaches for addressing the minimization of energy consumption. One approach involves

explicitly incorporating energy consumption within the objective function, while the alternative approach entails setting energy consumption as a constraint.

Ichoua & Pechmann (2013) stated that there has been a growing interest in developing and expanding renewable energy equipment and facilities to offer alternative power sources for industries. However, there is still a lack of production planning methods that effectively use and manage renewable energy for day-to-day operations.

In their work, Golari et al. (2016) introduced a "green energy coefficient ϱ" as a parameter to guarantee that a minimum proportion of renewable energy is integrated into the overall energy consumption associated with production processes. Their objective was to ensure that at least ϱ percent of the total energy consumed is from renewable sources, all while optimizing a range of costs including those related to energy, production, inventory, backorder, and transportation.

The carbon footprint evaluates the carbon emissions throughout a product's entire life cycle, considering greenhouse gas emissions from both individual and enterprise activities (Matthews et al., 2008).

Most product carbon footprint calculation methods, such as gate to gate (evaluating environmental impact from raw material extraction to product completion) and life cycle analysis (assessing the entire product life cycle from extraction to disposal), are commonly used for single product/high volume manufacturing systems, where only one type of product is produced in large batches, making calculations relatively easy. However, accurately labelling product carbon emissions and calculating carbon footprints across their entire lifecycle can be challenging when carbon emissions are not explicitly assigned to individual products during the manufacturing phases. In cases of single product and high-volume manufacturing, researchers frequently employ an averaging method to allocate carbon emissions across products, yielding product carbon footprints as an average of the total emissions. Nevertheless, as pointed out by Liu et al. (2017), it is imperative not to disregard the distinctions among products, machining processes, and job shop sequencing, as these factors play a crucial role in carbon emissions and environmental impact.

Scheduling inherently influences setup times, idle periods, and handling distances in a job shop environment. These scheduling variations can consequently have a direct impact

on carbon emissions. Liu et al. (2017) stated that several research studies have explored the potential for optimizing scheduling to reduce carbon emissions.

In carbon footprint analysis, there are two types of mapping, direct and indirect. Direct mapping assigns carbon emissions solely to the related product, while indirect mapping attributes emissions caused by resource consumption to multiple products. Directly relating emissions from certain manufacturing activities, like a machine powering on or off, to specific products can be challenging. For example, as a machine serves multiple products with varying processing times, it is hard to allocate emissions accurately. To simplify, emissions from resources with indirect mapping are evenly divided among products using them.

The complexity of GSSPs is notably influenced by the manufacturing environments in which they are involved. According to Li & Wang (2022), these environments can be categorized into five fundamental types, according to the shop floor type: single machine, parallel machine, open shop, flow shop, and job shop.

The most basic scheduling problem entails a single machine handling all jobs, while a parallel machine environment involves multiple machines with similar capabilities processing tasks simultaneously. In an open shop problem (OSP), jobs consist of multiple steps with no specific order between them. Furthermore, manufacturing facilities can be categorized into two main types of scheduling problems: flow shop scheduling problems (FSPs) and job shop scheduling problems (JSPs), depending on whether the production process follows a fixed sequence for all jobs or not. In cases where multi-stage jobs are involved, and at least one stage requires more than one machine, the scheduling problem might be referred to as a hybrid flow shop problem (HFSP) or a flexible job shop problem (FJSP). In a hybrid flow shop problem (HFSP), jobs have multiple stages, and each stage may have multiple machines, creating a structured but complex process. In contrast, a flexible job shop problem (FJSP) allows for more diverse and adaptable machine sequences for different jobs, making it more flexible and less structured. These distinctions reflect the complexity and diversity of scheduling challenges in various manufacturing settings. Every manufacturing environment is characterized by facing different challenges, which forces the use of different and suitable scheduling methods and algorithms to efficiently tackle the complexities of GSSPs.

2.3. Energy Efficient Job Shop Scheduling Problem

The pursue of enhancing efficiency of the energy used in factories and consequently lowering the amount of total energy consumed presents plenty of advantages, including cost reductions, a positive environmental impact, resource conservation and an increase in competitiveness and productivity.

As explained in Fernandes et al. (2022), there are three main strategies that have been used to improve energy efficiency in the context of the Energy Efficient Job Shop Scheduling Problem (EEJSP):

- Controlling Machines Working Speed;
- Time-Dependent Energy Prices;
- Switching Machines to a Power-Saving Mode while Idle.

Despite the obvious importance of production efficiency, it should not be the only considered factor in manufacturing operations.

The research conducted by Luo et al. (2013) focuses on finding ways to lower energy costs within manufacturing at a system-level perspective. Using decision models and optimization techniques in production planning and scheduling, it becomes feasible to achieve a notable reduction in energy consumption without the necessity of altering processes or equipment.

Enhancing energy efficiency is a top priority for businesses looking to improve their energy performance, as noted by Gutowski et al. (2005). To boost the energy efficiency of production systems, Rager et al. (2015) propose two distinct categories of measures. Firstly, technological measures which involve the implementation of new machinery or manufacturing processes, and organizational measures, which concentrate on optimizing the current system, ultimately resulting in reduced energy consumption.

Additionally, it can prove beneficial for companies to express their interest in energy savings and, if feasible, modify contracts with energy providers to align with these objectives, all while ensuring that productivity remains uncompromised.

Salido et al. (2015) noted that energy-efficient systems tend to be more robust and less vulnerable to machine failures. This is because machines in these systems may have variable processing speeds, which directly impacts energy consumption. Fast machining, for instance, demands more energy but leads to shorter treatment times, while slow machining requires less energy. In a production system of this nature, downtime due to a breakdown can potentially be compensated for by adjusting the machine's processing speed, which also affects energy consumption.

Several researchers have tackled the Job Shop Scheduling Problem with energy considerations. Liu et al. (2014) proposed an NSGA-II approach to minimize total tardiness and energy consumption by reducing machine idle times.

In their study, He et al. (2015) investigated the Flexible Job-shop problem with the objective of minimizing the overall system energy consumption. This was achieved by efficiently assigning suitable machines to each operation while also reducing idle time. They presented two approaches: a linear model and a metaheuristic technique called the Nested Partition Algorithm (NPA). These methods were applied in two different scenarios: one aimed at minimizing total energy consumption, and the other focused on optimizing both energy consumption and makespan.

Wu and Sun (2018) addressed a Flexible Job Shop Scheduling Problem with the goal of minimizing makespan, total energy consumption, and the number of switch on/off events. They employed NSGA-II (Non-dominated Sorting Genetic Algorithm II) and incorporated a green heuristic into their approach.

Research on energy efficiency for industrial production has identified two main approaches. One option involves investing in energy-efficient production machines and designing new production processes. The other approach focuses on energy-oriented production planning (EOPP).

Terbrack et al. (2021) provide an overview of EOPP and categorize it into three key topics. The first topic is energy consumption, where the focus is on considering the amount of energy used during production planning. The second topic is load management, which addresses energy demand at specific points in time to stabilize the power grid and reduce costs associated with balancing energy supply and demand. The third topic involves different

energy sources and storage systems in the context of production planning, expanding the scope to energy supply and generation.

Manufacturing companies are highly motivated to reduce energy consumption costs, as utility expenses can significantly impact production costs. This cost-saving potential is not limited to energy-intensive industries but extends to non-energy-intensive enterprises as well.

Another economically beneficial aspect is the avoidance of penalty costs resulting from violating specific energy consumption thresholds. Certain approaches, like the generalized critical peak price concept, impose additional costs when consumption exceeds critical levels, encouraging manufacturers to shift energy usage and avoid consumption peaks.

Several articles have explored the potential of energy storage systems (ESSs) with the goal of obtaining surplus energy that can be use later. During periods with an excess of energy supply, the ESS acts as an energy consumer and stores the unused energy. Once charged, the ESS operates as an energy source, offering flexibility in energy utilization. Furthermore, by storing energy during low-cost periods, significant cost savings can be achieved during periods of higher energy prices, leading to overall reduced energy expenses.

Kawaguchi and Fukuyama (2017) stated that nowadays, there has been a shift in production schedules, moving away from single and mass production towards various forms of small quantity production. To address this evolving trend, the authors proposed a strategy aiming to minimize both total production time and electric power energy consumption during peak periods.

When electric power consumption is concentrated at peak times, electric power companies often face the necessity of significant capital investments. Consequently, such situations may result in increased electricity charges for the entire community, which could have negative effects on the well-being of the community.

2.4. Switching Machines to a Power-Saving Mode while Idle

This section will focus on the third strategy described in Fernandes et al. (2022), which involves switching machines to a power-saving mode while idle. This strategy can be divided into turning machines off entirely or switching them into a standby mode. This third strategy will be thoroughly examined since it forms the primary focus of this dissertation.

While the first and second strategies aim to attain a balance between energy consumption and production time (makespan, tardiness, and earliness), the third strategy focuses on optimizing energy savings from using power-saving modes while considering the energy requirements to restart and warm up the resources. The goal is to find an efficient trade-off that minimizes energy usage, optimizes production efficiency, and manages the costs associated with resource reactivation and warm-up periods. This strategy requires careful consideration of real-time production demands and energy pricing to make intelligent decisions on when to utilize power-saving modes and when to activate the resources to achieve the best overall energy efficiency.

In industries, a lot of energy is wasted when machines are left idle due to underutilization. To reduce this energy waste and lower the environmental impact of industrial plants, we can focus on minimizing energy consumption while making scheduling decisions. This means that the decision maker needs to determine when and for how long a machine should be turned off or switched into a standby mode, as well as to determine a sequence of jobs that minimizes the scheduling objectives, assuming that all jobs are not available simultaneously. This breakeven decision will be analysed in further detail in this section.

Hidri et al. (2021) explored the idea of a no-idle machine constraint. Idle machine times occur when a machine is ready to process jobs but there are no jobs to be processed. These idle periods lead to energy consumption without any productive output. Studies show that machines often remain idle for a significant portion of the time, with 80% of the total energy consumed during these idle phases (Hidri et al., 2021). This energy, known as idle energy, consists of the energy consumed by machines when they are running idle or when they are switched to a standby mode, and is the second most considered energy consumption type (Fernandes et al., 2022). Addressing idle energy with a better control of the idle machine periods can lead to potential significant energy savings.

According to Fernandes et al. (2022), most authors consider switch on time and switch on power consumption to calculate switch on energy consumption. Some studies even account for whether the machine was previously on standby mode or switched off when determining switch on time and power. Other authors pre-calculate a breakeven time for each machine, determining when it is economically justifiable to turn the machine off and on or switch it into standby. As expected, if the time between two consecutive operations on the

same machine exceeds the breakeven time, then the machine is switched off. Finally, some more realistic papers also include machine transition states like warming up and ramping down, often by considering sequence-dependent setup times (SDST).

The focus on minimizing energy consumption has been on the rise, especially within the realm of computer and embedded electronic systems. For instance, Swaminathan and Chakrabarty (2003) introduced a control system designed to reduce energy consumption and prolong battery life. Their research demonstrated that a substantial reduction in energy usage could be achieved by simply altering the state (on/off) of the devices.

Mouzon et al. (2007) stated that in manufacturing facilities, it is common to find non-bottleneck machines running idle, which may present an opportunity for energy-saving improvements. A case study of an aircraft parts supplier in Kansas, USA, showed that four Computer Numerical Control (CNC) machines, automated manufacturing tools controlled by computers, were left idle for about 16% of the time. If these machines were turned off during idle periods, energy savings of around 13% could be achieved, considering the energy consumed during actual cutting.

Leaving non-bottleneck machines idle is a common operating practice, but it has implications for energy consumption and overall efficiency. To address this, there are two possible types of decisions, either leave the machine idle or switch into a power saving state for a specific duration, aiming to minimize energy usage while meeting scheduling criteria.

Nevertheless, it is important not to forget that when a machine is turned on, it requires a warm-up time and start-up energy before being ready to process a part and that turning off the machine requires a stop time, and this process consumes stop energy.

As expected, during the initial startup phase, a motor accelerates and requires more electrical power than when it runs continuously at full load. According to the Advanced Manufacturing Office of the U.S. Department of Energy (2012), this increase in power demand during startup is generally observed in various motors, where the current drawn can be significantly higher than during regular operation, typically ranging from four to eight times the normal current level.

A critical aspect to consider when making decisions about switching machines into

power saving mode is predicting the inter-arrival time between jobs, or in other terms, when the next job will arrive for processing. Accurate predictions are vital for minimizing energy usage without causing delays in job processing. By incorporating these findings and optimizing the operation of non-bottleneck machines, significant energy savings can be achieved, contributing to a more sustainable and efficient manufacturing process.

Mouzon et al. (2007) put forward the concept of a break-even duration (S), which indicates the point at which it becomes economically viable to switch a machine off and then back on again, instead of letting it run idle:

$$S = \frac{\text{Turn OFF Energy + Turn ON Energy}}{\text{Idle power consumption per unit time}} . \quad (1)$$

The numerator of equation (1) refers to the energy required to turn off (Turn OFF Energy) and then turn back on (Turn ON Energy) a machine. The denominator of equation (1), on the other hand, denotes the energy consumed during idle intervals per unit of time (idle power consumption per unit time). Moreover, if we denote "y" as the inter-arrival time between jobs and "t_{off}" as the time needed to both turn off and turn on the machine, it becomes evident that when 'y' is greater than or equal to the maximum value between the suggested break-even duration (S) and the time required for machine cycling (t_{off}), it becomes feasible to power down the machine for a specific duration and subsequently turn it back on to process other jobs. In simpler terms, if the time between job arrivals (y) is equal to or exceeds the larger of the break-even duration (S) and the machine cycling time (t_{off}), this approach can be employed.

It has been observed that if the time between the arrival of the current job and the next job is longer than the breakeven duration, then turning off the machine until the next job arrives can result in significant energy savings (Mouzon & Yildirim, 2008). By considering these factors, we can make more eco-friendly scheduling decisions and help reduce the carbon footprint of industrial plants.

Che et al. (2017) conducted a case study that centered on addressing a single-machine scheduling problem featuring a power-down mechanism. Their objective was to simultaneously minimize both total energy use and the maximum tardiness. Their objective was to identify the best processing sequence for jobs and decide whether the machine should power

down between two consecutive jobs. The authors demonstrated the effectiveness of both their exact and approximation approaches in solving this problem.

This strategy exclusively pertains to the non-processing energy consumption aspect. Consequently, in theory, this strategy should not have any impact on completion times.

In the overall processing course, the total energy consumption consists of four components. It encompasses the energy needed for initial machine startup and ultimate shutdown, the energy consumed during job processing, the energy used when the machine is idle, and the energy required for all turn-off-on operations. Since the sum of the first two components remains constant and is independent of the job processing sequence, the focus should be on minimizing the energy consumption related to the idle machine and all turn-off-on operations. By optimizing these aspects, we can effectively reduce overall energy consumption during production.

While He et al. (2015) emphasizes significant idle energy waste in production, presenting an opportunity for improvement by considering energy associated with idle time, Du et al. (2011) argue that idle energy can be disregarded as it has insignificant overall impact. Terbrack et al. (2021) states that many articles (164 out of the 375 analysed) incorporate energy usage related to idle time in production planning, aiming to reduce idle energy either through enhanced machine utilization or powering down machines during idle periods, which suggests a significant attention to idle time and energy.

The power-down strategy is approached in diverse manners. For instance, Mashaei and Lennartson (2013) propose hot and cold idle modes for production machines, optimizing the trade-off between idle energy and transition time for state changes. According to Terbrack et al. (2021), some studies also account for the energy required for turning machines off and on, while others argue for its negligible impact.

In manufacturing, a buffer refers to a temporary storage area or space used to hold materials, work-in-progress items, or finished goods at various stages of the production process. A machine is powered down when either starved, in the case of a downstream machine due to an empty buffer, or blocked, in the case of an upstream machine by a full buffer (Terbrack et al, 2021).

Dong (2012) integrates both machine shutdown and a power-saving idle mode into a scheduling approach designed for a parallel-machine environment, with the primary objective of minimizing overall costs, specifically focusing on energy-related expenses.

The findings from the study conducted by Shrouf et al. (2014) suggest that substantial reductions in energy expenses can be achieved by avoiding periods of high energy prices. Furthermore, there is also a positive environmental impact by lowering energy consumption during peak periods, thus contributing to the potential reduction of CO2 emissions.

As explained by Hidri et al. (2021), there are two possible approaches to deal with idle energy. The first one consists of shutting down idle machines. Alternatively, the second approach consists of adopting idle machine time constraints which force each machine to process all the assigned jobs continuously, to avoid idle times.

It is important to mention that the first method is only relevant and potentially suitable for electric machines where it is simple to move from an idle state to an active state. For other shop environments where this mechanism is substantially energy consuming, such as the case of furnaces, the first strategy is not applicable. Furthermore, there are additional constraints regarding this strategy. Idle machines between operations are usually not turned off given that turning machines on and off frequently may cause faster deterioration. Also, some machines need time and cost to start up again, so shutting them down during short idle times may not be practical. Hence, factories need to find a balance between saving energy and keeping the machines in good condition. They must consider each machine's characteristics and their overall energy efficiency goals to make the best decision.

Some articles caution that the power-down strategy may not be universally suitable, considering factors such as warm-up time, additional energy consumption for state transitions, and potential machine deterioration from frequent switching. Certain research approaches incorporate a maximum allowable number of machine state switches to address this concern. For example, Wu and Sun (2018) present a flexible job shop scheduling approach that minimizes, besides makespan and energy consumption, the total number of machine turn-offs/turn-ons.

Liang et al. (2019) pointed out that in real-life manufacturing management, it is common for idle machines not to be shut down until all products have been processed. This

approach is often referred to as "batch processing". The rationale behind this practice is to ensure continuous production flow and avoid disruptions in the manufacturing process. Shutting down machines between product batches could lead to delays, increased setup time, and potential loss of productivity.

Terbrack et al. (2021), concluded that energy-supply-oriented production planning models, in which "idle/standby and transition states: power-down" are included in their frequently found characteristics, should and likely will attract more interest in future studies due to the growing shift toward renewable energies and increased awareness of resource shortage.

2.5. Solving Job Shop Scheduling Problems

As stated by Fernandes et al. (2022), the job shop scheduling problem is known to be a nondeterministic polynomial time (NP hard problem), which implies that there is no known algorithm that can solve the job shop scheduling problem optimally in polynomial time, given that it is at least as difficult as the hardest problems in NP.

Exact, heuristic, and metaheuristic algorithms are different approaches used to solve optimization problems, each with its own characteristics and trade-offs. Exact algorithms, such as branch and bound or integer programming, can find optimal solutions but, nevertheless, they may be computationally expensive and time-consuming, especially for large-scale problem instances. In the case of heuristic algorithms, such as the Nearest Neighbor Algorithm, they can find good solutions in reasonable time but might not assure optimality and are particularly useful for NP-hard problems, where finding an optimal solution is impractical within a reasonable time frame. Lastly, metaheuristic algorithms are generally problem-solving strategies that can be adapted and applied to various problem domains. They are typically based on concepts from nature-inspired processes, such as evolution, swarm intelligence, and local search. Like heuristic algorithms, metaheuristics algorithms do not guarantee optimality but are capable of efficiently exploring large solution spaces and finding good solutions. They can handle a diverse set of optimization problems and are more flexible and robust when compared to exact and heuristic algorithms.

IBM ILOG CPLEX Optimization Studio is a good software solution to solve complex mathematical programming models, due to its versatility, scalability, and robust capabilities.

This comprehensive optimization suite integrates both CPLEX and CP Optimizer, offering a well-rounded toolkit for addressing the complexities of a JSP mathematical model. Whether the JSP is formulated as a linear or a constraint programming model, CPLEX Optimization Studio provides the versatility required to tackle various JSP instances effectively.

According to Muller et al. (2022), solvers based on constraint programming have a strong reputation for delivering remarkable performances in the context of scheduling problems. The researchers explored the process of selecting an appropriate solver and considering a range of software options, including both commercial and non-commercial solutions that are acknowledged as the current state-of-the-art in the field, two solvers emerged as exceptional performers: IBM ILOG CPLEX CP Optimizer and Google's OR-Tools. These solvers displayed impressive capabilities, each with its unique strengths. Notably, IBM ILOG CPLEX CP Optimizer excelled in efficiently determining solutions that are probably optimal within realistic time constraints. Conversely, Google's OR-Tools showcased a remarkable agility in swiftly identifying high-quality feasible solutions across a diverse set of test instances.

Furthermore, Da Col & Teppan (2022) compared IBM's CP Optimizer, an optimization software package, with Google's OR-Tools (Operations Research Tools), an open-source software library. The results indicate that CP Optimizer outperformed OR-Tools in both classic and large-scale benchmark tests. These results highlight the effectiveness of CP solvers, like CP Optimizer, in tackling real-world industrial challenges, especially when it comes to large-scale optimization problems.

Constraint Programming (CP) has gained substantial recognition for its suitability in addressing the Job Shop Problem. CP's inherent flexibility allows for the precise modelling of complex constraint, such as task precedence, machine availability and resource limitations, commonly found in JSPs.

When it comes to choosing the data to be employed in solving the JSP, using benchmark problem instances, such as the Taillard problem instances, is highly advantageous when working on the Job Shop Problem (JSP) given that it provides a standardized basis for evaluating and comparing the performance of different algorithms and approaches, i.e., researchers and practitioners can use these benchmark instances as a common reference point to assess the effectiveness of their JSP-solving methods. Furthermore, benchmark problem instances sets

are often derived from real-world scenarios or inspired by practical production environments. This means that they capture the complexity and characteristics of actual JSP instances, making them representative and relevant for testing and development.

3. Methodology

3.1. Computational Settings

All the experiments were conducted in a desktop computer (DESKTOP-0I5GSPO) with a processor AMD Ryzen 7 3700U with Radeon Vega Mobile, 2.30 GHz CPU, 12.00G RAM, Windows 10 Education N, using IBM ILOG CPLEX Optimization Studio version 22.1.1, commonly referred to as CPLEX, a powerful software suite for solving complex optimization problems. Following the optimization conducted in CPLEX in which the objective consisted of makespan minimization, the output solutions were then transferred to and used in Microsoft Excel, known for having a more user-friendly interface, with the goal of analyzing if energy savings were obtainable.

3.2. Problem Instances and Data Sources

The benchmark instances designed by Taillard (1993) are employed as the test. The concept of Taillard instances revolves around a group of standardized scenarios that serve as benchmarks for assessing and contrasting the effectiveness of various algorithms and techniques in addressing the complexities of job shop scheduling problems. These instances were developed by E. Taillard and have gained substantial prominence within the operations research and optimization domain. Each individual Taillard instance encapsulates a distinct job shop scheduling problem, characterized by specific counts of jobs, machines, and tasks involved. For this dissertation, 14 Taillard problem instances where considered, seven of which include 15 jobs and 15 machines (15x15) and the remaining seven include 20 jobs and 15 machines (20x15).

Each one of the 14 problem instances were firstly collected in the format of 2 tables, with data related to the length of operations and the associated machine responsible for processing that certain operation (see Annex 1). To use this benchmark instances in CPLEX it was necessary to format them into a single matrix, with each entry of the matrix following the format of < machine that will process operation (x), length of operation (x), > (see Annex 2). Furthermore, given that this dissertation will compute the model in CPLEX using zero-based indexing, meaning that the first element or item in an array, list, or set is accessed with an index of 0, it was also necessary to reduce every value of every machine present in the

Taillard's instances by 1 unit. This practice is common in computer science and can simplify various algorithms and data structures. It can be more memory-efficient, especially in low-level programming languages. Additionally, when working with arrays or data structures, it is often more consistent to use zero-based indexing for all elements. This consistency can simplify code and reduce the potential for off-by-one errors. Many libraries and programming paradigms, such as linear programming or constraint programming, use zero-based indexing.

Given that the Taillard (1993) problem instances do not consider machine power saving modes, it is necessary to incorporate some other data in addition to the number of jobs, number of machines, the duration of the processing time of each task and the order of tasks and respective machine assignment that each job requires to be processed. The data required includes processing power per machine, idle power per machine, warmup power per machine, and standby power consumption per machine. To address the additional data, a table developed and employed by Wu & Sun (2018), designated "Table 4 – The power distribution for each machine", was considered. In the work of Wu & Sun (2018), in which different machine speed levels are considered, the authors were required to use three different levels of energy consumption based on the respective machine speed (i.e., an increase in the machine speed results in a rise of energy consumption per unit of time, ceteris paribus). For this dissertation's analysis, only one level of energy consumption is mandatory. Nevertheless, the other two will be further considered with the goal of testing the sensitivity of the obtained results.

There was, however, one major change to the initial table proposed in the work of Wu & Sun (2018) related to the energy consumption during the switching to standby mode. While the authors considered values of around two times the processing energy per unit of time, in this dissertation the values given to the energy consumed during the operation of switching to standby and then back on the machine will be eight times the processing power consumption, following the ratio illustrated by the Advanced Manufacturing Office of the U.S. Department of Energy (2012). The processing, idle and standby rates are illustrated in Annex 3.

3.3. Formulation of the JSP

Notation

Sets

J_j : Set of jobs, where j = 0, 1, …, n;

M_m : Set of machines, where m = 0, 1, …, w;

$O_{i,j}$: Set of predefined Operations i of Job j, where i = 0, 1, …, k;

Parameters

$pt_{i,j}^{m}$: Processing time of Operation i of Job j, to be processed in Machine m;

pp_m : Processing power consumption of Machine m;

ip_m : Idle power consumption of machine m;

sd_m : Duration of switching to standby state of Machine m;

wd_m : Duration of warmup of Machine m;

wp_m : Warmup power consumption of Machine m;

sbp_m : Standby power consumption of Machine m;

Decision and Auxiliary Variables

$st_{i,j}^{m}$: Starting time of Operation i of Job j, to be processed in Machine m;

TPT_m : Total processing time of Machine m;

TPE_m : Total processing energy consumption of Machine m;

$it_{i,j}^{m}$: Associated idle time of Operation i of Job j, to be processed in Machine m;

$sbt_{i,j}^{m}$: Associated standby time of of Operation i of Job j, to be processed in Machine m;

TIT_m : Total idle time of Machine m;

TIE_m : Total idle energy consumption of Machine m, before incorporating a standby state;

B_m : Breakeven idle duration threshold of Machine m;

Nsd_m : Number of switches to standby of Machine m;

nTIE_m : Total idle energy consumption of Machine m, after incorporating a standby state;

TEC_m : Total energy consumption of Machine m;

C_{max} : Makespan of all Operations.

This study addresses an optimization challenge related to task scheduling within a collection of jobs assigned to a group of machines. Each job comprises a series of distinct operations, each with varying sequences and processing times. Additionally, the machines possess unique characteristics, including specific processing rates, idle rates, warmup rates, and standby rates.

The primary objective of this research is to minimize the total makespan (C_{max}), which represents the completion time of all scheduled tasks. Achieving this goal requires the use of optimization techniques, namely trough CPLEX.

Following the optimization process, the interval times of the output will undergo further examination in Microsoft Excel. Additionally, each machine's operational status will be considered. Specifically, if the idle time between consecutive operations on a machine exceeds a predefined breakeven threshold (B_m) for that machine, it will be transitioned into a standby mode.

This transition aims to explore the potential for energy savings by putting machines into standby mode without adversely affecting the total makespan. The research will evaluate whether this energy-saving approach is viable and assess its impact on overall energy consumption.

The Constraint Programming model (see Annex 4) can be formulated as follows.

The first step consists of defining two crucial parameters, n+1 and w+1, which respectively represent the total number of jobs and machines involved in our manufacturing process. We also define two ranges, J and M, which are used to index the jobs and machines. These ranges facilitate the organization and manipulation of the problem data. The ranges are defined using zero-base indexing. To represent the scheduling of operations on machines, the Operations matrix is used. This matrix captures the assignment of each operation of each job to a machine and its associated processing time.

In addition to these fundamental definitions, we establish power consumption rates for each machine, following the power consumption rates illustrated on the work of Wu & Sun (2018), as previous stated.

Finally, to facilitate the modelling and optimization process, the FailLimit parameter was set to 10000 with the goal of controlling the maximum number of failures allowed during the solution process.

Regarding the model used in CPLEX, two decision variables were considered. Firstly, there is an Interval variable that represent the temporal aspects of the scheduling problem.

Specifically, they define the start times ($st_{i,j}^{m}$) for each manufacturing operation ($O_{i,j}$) on each machine. These variables are essential for capturing the timing of operations within the production process. Furthermore, there is a sequence variable that governs the order in which manufacturing operations are scheduled on individual machines. They are a fundamental element of the optimization model, as they dictate the sequence of work on each machine while adhering to constraints.

The objective function can be illustrated as:

$$Min\ C_{max}\ . \tag{2}$$

Subject to:

- ∀ (j ∈ [0, n], i ∈ [1, k], m ∈ [0, w]):

$$st_{i+1,j}^{m} \geq st_{i,j}^{m} + pt_{i,j}^{m}\ . \tag{3}$$

- ∀ (x ∈ [0, n], y ∈ [0, n], x ≠ y, i ∈ [1, k], r ∈ [0, w]):

$$st_{i,x}^{r} \neq st_{i,y}^{r}\ ; \tag{4}$$

$$st_{i,x}^{r} < st_{i,y}^{r} \Rightarrow st_{i,y}^{r} \geq \left(st_{i,x}^{r} + pt_{i,x}^{r}\right). \tag{5}$$

This first constraint, named Precedence Constraint (3), enforces that the starting time of the next operation (i+1) of any job j must be greater or equal to the sum of the starting time of the previous operation of that same job j and the duration of that previous operation, i.e., the processing time of the previous operation. This constraint is widely used in scheduling to sequence tasks logically and avoid starting something before its prerequisites are met.

The next constraint, called NoOverLap Constraint, imposes that given any two jobs with a different number (x and y) that need to be processed on the same machine (r), these two jobs must have different starting times on that particular machine (4) and, furthermore, if the starting time of operation i of job x is inferior to the starting time of operation i of job y on that particular machine (r), then the starting time of operation i of job y must be

greater or at least equal (it is equal if there is no idle time between the job x and y on machine r) to the sum of the starting time of operation i of job x and the respective duration of that operation i of job x on that same machine r (5).

This ensures that tasks, resources, or activities do not overlap or conflict with each other. They prevent situations where two events occur simultaneously, or two resources try to occupy the same space or time slot on the same machine. They are essential for optimizing resource allocation and maintaining feasible schedules.

Additional Assumptions

Deterministic Processing Times, i.e., it is assumed that the processing times for each operation on each machine are known with certainty and do not vary.

No Machine Breakdowns, i.e., it is assumed that machines do not break down, require maintenance, or experience downtime during the scheduling horizon.

No Resource Constraints, i.e., there are no resource constraints, such as machine capacity limitations or worker availability.

No Setup Times, i.e., there is no time required to switch a machine from processing one job to another.

Infinite Buffer Capacity, i.e., completed operations can wait indefinitely before moving to the next machine.

No Interruptions, i.e., once a job starts processing on a machine, it continues without interruptions until completion.

Following this, before running the optimization, for calculating the processing energy consumption per machine (6) it is just necessary to multiply the processing times of each machine by its respective processing power. To obtain the total processing energy consumption (7) it is simply needed to sum all the processing energy per machine.

$$TPE_m = \sum pt_{i,j}^{m} * pp_m ; \tag{6}$$

$$TPE = \sum TPE_m . \tag{7}$$

The script used for computing the idle time iterates through each machine, analysing the start and end time of each job scheduled to be processed on that machine, calculates the

total machine usage time, and then computes and prints the idle time for each machine.

$$TIT_m = (st_{i,l}^{m} + pt_{i,l}^{m}) - st_{i,f}^{m} - \sum pt_{i,j}^{m} . \tag{8}$$

Assuming that f identifies the first job that is processed on a certain machine m and that l refers to the last job that is processed on that same machine m, with $f \in J$ and $l \in J$, the total idle time per machine (8) is calculated by subtracting the starting time of the first job f processed on machine m and the sum of all the processing times of the tasks processed by that machine to the ending time of the last job (l), obtained by summing the starting time of that same job (l) and the associated processing time, processed by that same machine.

$$TIE_m = TIT_m * ip_m . \tag{9}$$

The total idle energy consumption per machine (9) is calculated by multiplying the total idle time per machine with the associated idle rate consumption. As expected, the total idle energy is simply the sum of all the idle energy consumption per machine.

$$TEC_m = TPE_m + TIE_m . \tag{10}$$

We are now able to calculate the total energy consumption per machine (10), which is obtained by adding the total processing energy consumption of a machine m with the total idle energy consumption of that same machine.

Figure 2 serves as a visual representation of the output generated by the Constraint Programming model computed in the CPLEX Studio. The analysis pertains to the Taillard's problem instance 15x15 (1). Each row in the first column corresponds to a unique machine, numbered from 0 to 14, each job is identified by one color (see the color legend). The horizontal axis quantifies the temporal span of the schedule.

For this specific instance, the makespan value, which denotes the total time required to complete all tasks, has been determined as 1393. When analyzing this chart its' easy to see that the amount of idle time is extremely significant. If we compute the ratio of total processing time divided by the total idle time it amounts to around 59%.

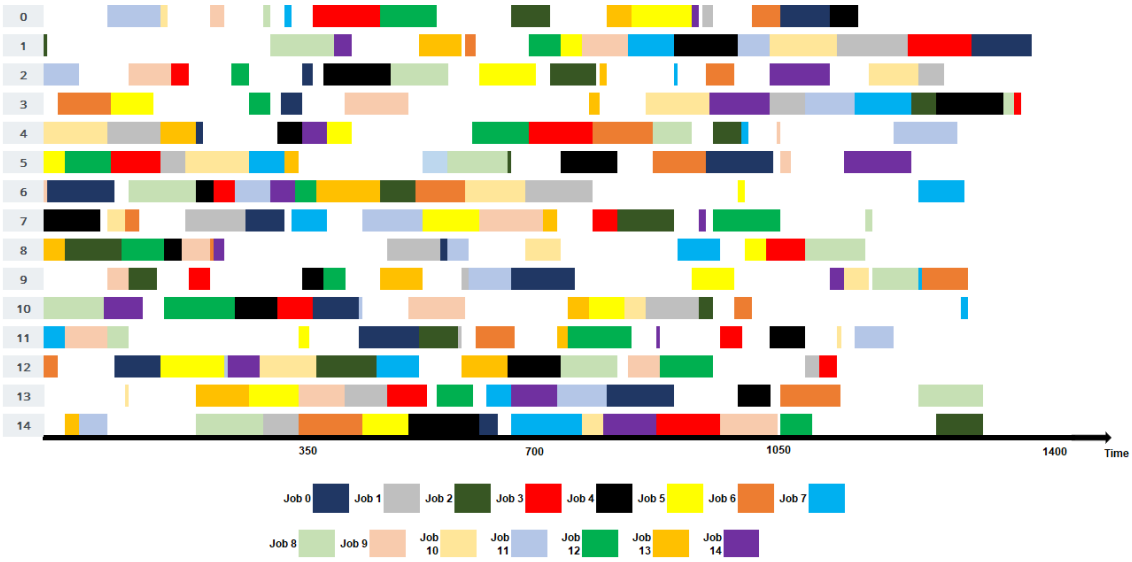


Figure 2. Gantt Chart of Schedule Solution, for Problem Instance 15x15 (1).

Source: Own Elaboration

This graphical representation of the job shop problem solution offers a valuable tool for enhancing our understanding of the presence of considerable idle time intervals during which machines await the start of subsequent tasks.

Following the optimization process conducted in CPLEX Studio, the subsequent analysis will be carried out in Microsoft Excel, mainly due to easier data visualization and its graphical user interface (GUI), well-suited for users who prefer a visual, interactive approach to data analysis.

After migrating all the solution output regarding the starting and ending time of each task on each machine and its associated idle time of the 14 benchmark problem instances used in this dissertation, it is now time to explore the possible energy savings in idle energy with the introduction of a standby mode.

It is only possible to turn a machine into a standby state if the 2 following conditions are respected:

∀ (j ∈ J, m ∈ M),

$$it_{i,j}^{m} \geq sd_{m} + wd_{m}\,; \tag{11}$$

$$it_{i,j}^{m} * (ip_m - sbp_m) > wp_m * wd_m . \quad (12)$$

Inequality (11) states that the idle time associated with a certain operation i of a job j to be processed by a particular machine m must be greater or equal to the length related to switching that machine into standby state (sd_m) and then turning back that same machine into active state (wd_m).

The condition imposes that the product of the idle time associated with a certain operation of a particular machine m and the difference between the idle power rate and standby power rate of that machine must be greater than the product of the warmup power rate, i.e., turning the machine back into active state and the amount of time necessary to turn the machine back into active state. For simplification purposes, in this dissertation both the shutting down and the warmup duration will be disregarded. Given that the idle time of an operation cannot assume negative values, the inequality (11) will not be important for this dissertation. Furthermore, expression (12) will be used to calculate the breakeven threshold (B_m), without the incorporation of the value of the time necessary to warmup the machine.

As previously stated, the data regarding the rates of processing power, idle power, standby power, and the warmup power will be based on Annex 3.

After checking that the processing times, the processing energy, the idle times, and the idle energy of each machine are the same as the values calculated in CPLEX Studio, the first step will consist of calculating the breakeven threshold of idle time (B_m) for each machine.

As formerly seen, Mouzon et al. (2007) proposed a break-even duration, illustrated in equation (1). This formula needs to be adjusted to include the possibility of having more than one machine and to incorporate the standby time power consumption, according to the respective standby rates of each machine. Given that without a standby rate the calculation of the minimum amount of idle time required to be economically reasonable to switch the machine into a standby mode simply consists of the proportion between the energy required to warm up again the machine and the idle energy that would be spent if the machine was not in standby, then incorporating a standby rate must increase the breakeven duration since it will become "more expensive" switch a machine into standby mode, ceteris paribus.

$$B_m = \frac{wp_{\,m} * wd_{\,m}}{ip_{\,m} - sbp_{\,m}}. \tag{13}$$

As can be seen in equation (13), the standby rate will be incorporated in the denominator of the quotient, subtracting itself to the idle rate per machine. It is easy to understand that an increase in the standby rate will cause an increase in the breakeven duration, ceteris paribus. The computed results for the B_m per machine are shown in Table 1.

Machine	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
B_m	47	59	56	56	43	50	48	54	48	50	48	49	52	50	43

Table 1. Breakeven Threshold per Machine (Level 1).

Source: Own Elaboration.

All the results illustrated in Table 1 were rounded up to the nearest integer. The breakeven results indicate the time length that an idle interval must at least have to be advantageous from an energy consumption perspective to switch a machine into standby mode. For example, the breakeven threshold for Machine 0 is 47, indicating that when idle periods on Machine 0 have a duration of 47 or more, the machine should be switched into standby mode during those periods.

The Standby Time per machine can be calculated as follows:

$\forall\ it_{\,i,j}^{\,m}$, such that $it_{\,i,j}^{\,m} \geq B_m$,

$$sbt_{\,i,j}^{\,m} = \sum it_{\,i,j}^{\,m}. \tag{14}$$

Equation (14) state that the standby time per machine consists of the sum of all idle times that are equal or larger than the threshold value B_m of each machine. In Excel this can be done by simply using the function: SUM.IF (Cells with the Idle Times of Machine m; "≥" & B_m).

The number of shutdowns per machine follows the same idea, but instead of being a sum, it is just necessary to count the number of times in which the inequality $it_{\,i,j}^{\,m} \geq B_m$ is true. In Excel this can be done by simply using the function: COUNT.IF (Cells with the Idle Times of Machine m; "≥" & B_m).

Following the context of the 15x15 (1) Taillard problem instance showed in Figure 2, this Gantt chart provides a visual representation of how the schedule would appear with the integration of a standby mode. Consequently, the idle periods are mitigated by transitioning machines into standby mode, when the idle period is greater than the threshold of the machine associated, resulting in a more efficient utilization of resources and a modified schedule.

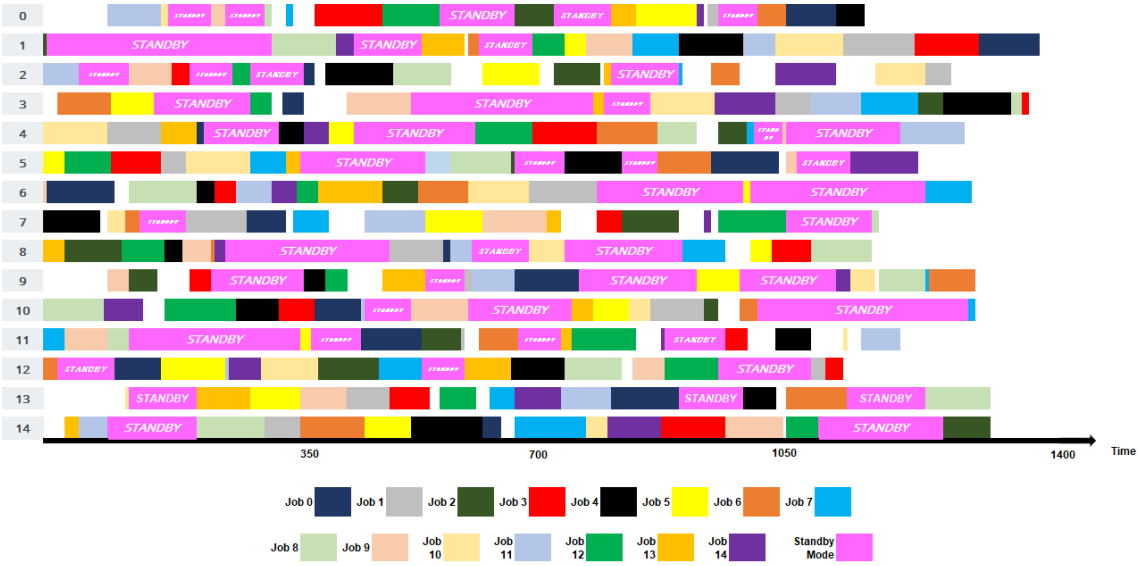


Figure 3. Gantt Chart of Schedule Solution with the Incorporation of the Standby Mode, for Problem Instance 15x15 (1).

Source: Own Elaboration

For a more enhanced representation of the remaining idle times, the chart illustrated in Figure 4 showcases the instances where only short idle durations, namely those falling below the breakeven threshold, remain unaffected and are not converted into standby mode.

Figure 4 clearly shows the distinction between short idle periods and those that exceed the threshold, thus providing clearer evidence of the scheduling optimization achieved through the incorporation of the standby mode.

Figure 4. Gantt Chart of the Idle Intervals of the Schedule Solution with the Incorporation of the Standby Mode, for Problem Instance 15x15 (1).

Source: Own Elaboration

We can finally compute the "new" total idle energy consumption per machine after introducing the possibility of turning machines into a standby state:

$$nTIE_m = \sum((TIT_m - sbt_m) * ip_m) + \sum(sbt_m * sbp_m) + \sum(Nsd_m * wp_m) \ . \tag{15}$$

The "updated" idle energy consumption per machine (15) is now computed by adding three components, namely: (i) the product of the idle time that is not turned into standby time (sum of all idle time inferior to the breakeven threshold) and the respective idle power consumption of each machine, (ii) the product of the standby time and the associated standby power consumption of each machine and (iii) the product of the number of shutdowns and the warmup power consumption per machine.

The last steps consist of calculating the difference between the total "updated" idle energy consumption per machine (15) and the total idle energy consumption per machine (9) and then add them over the 15 machines.

3.4. Discussion of Results

Now, to have a better understanding of the impact of incorporating the standby mode, the percentage energy savings in terms of both the idle energy and the total energy (processing and idle energy) are shown in Table 2. The reported values are calculated by dividing the sum of the 15 differences by the previous total of idle energy (before incorporating the standby state) and by the previous total of total energy, respectively.

Problem Instance	% of Idle Energy Saved	% of Total Energy Saved
15x15 (1)	45 %	4.4 %
15x15 (2)	44 %	4.8 %
15x15 (3)	48 %	5.2 %
15x15 (4)	52 %	5.8 %
15x15 (5)	50 %	5.3 %
15x15 (6)	46 %	4.4 %
15x15 (7)	52 %	5.4 %
20x15 (1)	44 %	3.2 %
20x15 (2)	42 %	2.7 %
20x15 (3)	42 %	3.3 %
20x15 (4)	46 %	3.9 %
20x15 (5)	50 %	4.0 %
20x15 (6)	41 %	3.0 %
20x15 (7)	43 %	3.8 %
Average	**46 %**	**4.2 %**

Table 2. Computed Results of Energy Savings (Level 1).

Source: Own Elaboration

On average, there is a 46% saving in terms of idle energy consumption and a 4.2% saving in terms of total energy consumption.

Based on data sourced from the EIA (2022), the average monthly energy consumption bill for the United States in 2021 stood at 81,573 kWh. To project the expected monthly energy cost for the year 2022, I've incorporated the findings from the Literature Review section pertaining to energy prices in the US. Accordingly, the estimated monthly energy bill would be calculated as follows: 81,573 kWh * 8.45 cents/kWh, which totals $6,893. In an annual perspective, this monthly electricity expenditure for US factories would accumulate to $82,716, considering all 12 months of the year. Accounting for the average total energy savings rate of 4.2%, this translates to annual savings of $3,474.

While these computations rely on estimations and statistical averages of the US, they do provide insights into the economic viability of implementing a standby mode. To be

considered cost-effective on average, the associated expenses should remain below the annual threshold of $3,474, assuming the accuracy of the total energy savings rate.

Moreover, it is crucial to underscore the sustainability benefits that accompany such energy consumption reductions. Beyond economic considerations, the reduction in energy usage contributes to a more environmentally responsible approach. This entails a reduction in greenhouse gas emissions, a decrease in the carbon footprint, and a positive impact on the overall ecological landscape. In essence, the incorporation of standby modes besides making possible economic sense also aligns with the broader goal of achieving a more sustainable and environmentally friendly energy landscape.

3.5. Sensitivity Analysis

In this section of the study, we will examine whether adjustments to the rates of processing and idle power have a substantial impact on the obtained results. As previously mentioned, in the work conducted by Wu & Sun (2018), the authors explored three distinct machine speed levels. They used a coefficient vector to scale the single speed level into three, contingent upon the respective machine speed. Essentially, an increase in machine speed results in higher energy consumption per unit of time, all other factors remaining constant. Specifically, the speed levels employed are as follows: (i) Level 1 corresponds to a machine speed of 1x, (ii) Level 2 corresponds to a machine speed of 1.2x, and (iii) Level 3 corresponds to a machine speed of 1.5x.

Subsequently, we meticulously examine whether there are significant discrepancies arising from the utilization of these different processing and idle power consumption rates (described in Annex 5). It is imperative to note that alterations in processing energy rates also influence the energy values associated with transitioning machines into standby mode. Consequently, modifications in one parameter will cause adjustments in others, including the breakeven threshold of each machine. Our aim is to assess the impact of variations in processing energy rates and idle rates on the overall results.

The same exact process used for the first analyzed level of processing and idle energy will be now conducted for the other two levels. Starting by computing the CP model in CPLEX Studio 14 different times, one for each Taillard problem instance used in this dissertation, we obtain the schedules that will be analyzed in Microsoft Excel. After calculating

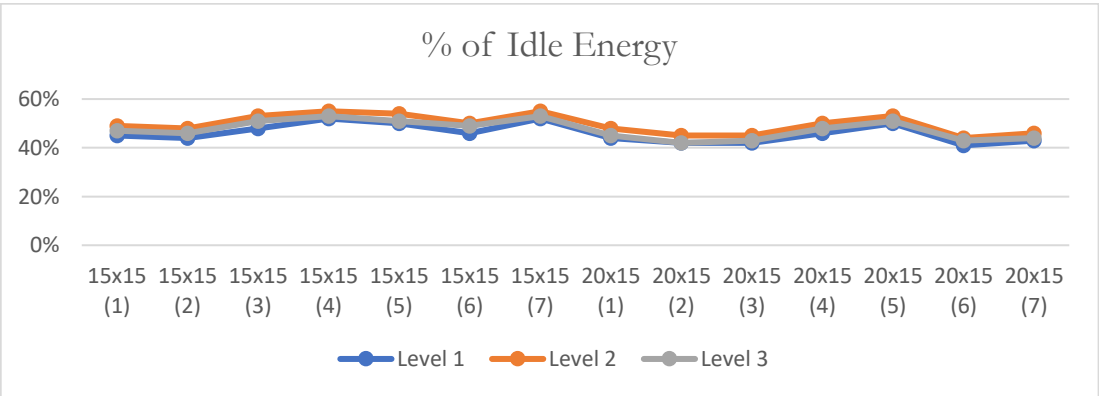
the breakeven threshold for the idle duration and incorporating the possibility of turning machines into standby mode, we obtain the energy savings as shown in Table 3.

	Level 2		Level 3	
	% of Idle Energy	% of Total Energy	% of Idle Energy	% of Total Energy
15x15 (1)	49 %	4.9 %	47 %	4.3 %
15x15 (2)	48 %	5.4 %	46 %	4.7 %
15x15 (3)	53 %	5.9 %	51 %	5.1 %
15x15 (4)	55 %	6.3 %	53 %	5.6 %
15x15 (5)	54 %	5.9 %	51 %	5.2 %
15x15 (6)	50 %	5.0 %	49 %	4.4 %
15x15 (7)	55 %	5.9 %	53 %	5.2 %
20x15 (1)	48 %	3.6 %	45 %	3.1 %
20x15 (2)	45 %	4.0 %	42 %	2.6 %
20x15 (3)	45 %	3.7 %	43 %	3.2 %
20x15 (4)	50 %	4.4 %	48 %	3.8 %
20x15 (5)	53 %	4.3 %	51 %	3.8 %
20x15 (6)	44 %	3.3 %	43 %	2.9 %
20x15 (7)	46 %	4.1 %	44 %	3.6 %
Average	**50 %**	**4.7 %**	**48 %**	**4.1 %**

Table 3. Computed Results of Energy Savings (Level 2 & Level 3).

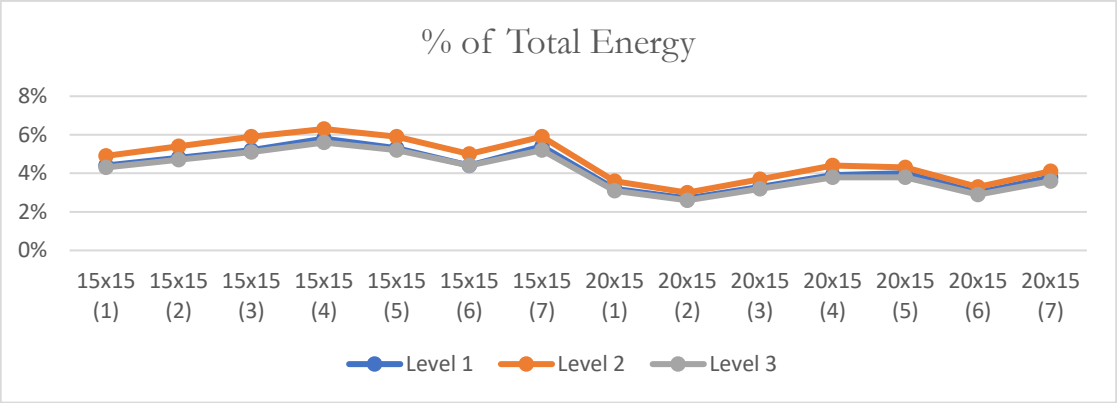
Source: Own Elaboration

Graph 1 and Graph 2 were created with the intention of providing a clearer and more comprehensive visualization of the results we have obtained. These visualizations focus on two key aspects: (i) the percentage of idle energy saved for each problem instance (Graph 1), (ii) the percentage of total energy saved per problem instance (Graph 2).



Graph 1 - Percentage of Idle Energy Saved per Problem Instance

Source: Own Elaboration



Graph 2 - Percentage of Total Energy Saved per Problem Instance

Source: Own Elaboration

Upon a close examination of the two graphs, it becomes apparent that no significant outliers emerge when comparing them to the trajectory of each curve. Furthermore, when we assess the averages of each level, we observe a high degree of similarity. This holds true for both the percentage savings in idle energy (46%, 50%, 48%) and the percentage savings in total energy (4.2%, 4.8%, 4.1%). Such consistency in the results is a positive signal related to the robustness of the research conducted within this dissertation.

3.6. Bi-Objective Model: Minimizing both Makespan & Idle Time

As previously outlined, this dissertation's primary objective is to assess the feasibility of achieving energy savings through the incorporation of a standby mode, all while maintaining the original makespan of the schedules intact. During this research, questions have surfaced regarding the practical implications on makespan when optimizing for minimized idle times in conjunction with the primary objective of minimizing the total completion time of all operations. This line of analysis arises from the consideration that if the impact on makespan is minimal, and the reduction in idle times yields substantial energy savings, it could potentially diminish the necessity of incurring in substantial costs associated with incorporating machines with the standby mode.

As a result, a similar model was formulated using CPLEX, with the primary distinction being the inclusion of the minimization of total idle time as part of the objective function, which originally prioritized the reduction of the makespan (see Annex 6). Furthermore, since it is possible to encounter situations where the complexity of the objective

function increases the probability of facing infeasible subproblems. In addressing such complexities within the optimization process, the FailLimit was adjusted upwards, recognizing that this adjustment results in longer computation times. The computed results are shown in Table 4.

	% of Idle Energy	% of Total Energy	% of Increase in Cmax
15x15 (1)	14.5%	1.4%	4.5%
15x15 (2)	21.2%	2.3%	4.7%
15x15 (3)	21.0%	2.3%	2.2%
15x15 (4)	17.8%	2.0%	5.7%
15x15 (5)	19.4%	2.1%	2.0%
15x15 (6)	18.7%	1.8%	3.4%
15x15 (7)	7.5%	0.8%	7.2%
20x15 (1)	13.3%	1.0%	8.5%
20x15 (2)	22.2%	1.4%	3.2%
20x15 (3)	4.6%	0.4%	10.9%
20x15 (4)	27.0%	2.3%	6.1%
20x15 (5)	12.6%	1.0%	5.8%
20x15 (6)	19.6%	1.4%	3.7%
20x15 (7)	12.5%	1.1%	8.0%
Average	**16.6%**	**1.5%**	**5.4%**

Table 4. Results of Bi-Objective Model.

Source: Own Elaboration

Based on the findings, by minimizing the total idle time of the schedule, it is possible to achieve an average reduction of 16.6% in idle energy consumption and a 1.5% decrease in overall energy usage, which includes both processing and idle energy. However, this energy-saving effort comes at the cost of a 5.4% increase in the makespan, meaning it takes 5,4% more time to complete all the tasks.

When we compare these results to the previously computed findings in this dissertation, it becomes apparent that the achieved savings are more favorable when incorporating the standby mode as opposed to solely minimizing idle times. This is evident in terms of the percentage reduction in idle energy and the percentage reduction in total energy consumption. However, it is important to note that implementing this standby mode is not a straightforward task, as it necessitates the installation and integration of such a mode into the system.

4. Conclusion

As previously stated, the central focus of this dissertation revolves around assessing the potential for energy savings by implementing a machine power-saving strategy, specifically switching machines to standby mode during idle periods, while ensuring that the total completion time of all operations (makespan) remains unaffected. The problem is defined as an optimization problem that seeks to minimize the makespan, and is solved using IBM ILOG CPLEX Optimization Studio, while introducing a standby mode that replaces idle times when those idle periods exceed the calculated breakeven threshold for each machine.

In the context of energy-saving strategies, it is crucial to consider the suitability of the approach for specific machines. The incorporation of a standby mode is more likely applicable to electric machines due to their ease of activation. However, it may not be suitable for energy-intensive equipment like furnaces. There are also practical constraints, such as the need to balance energy savings with machine deterioration and operational continuity. In real manufacturing settings, it is common for machines to remain active until all products are processed to maintain production flow and avoid disruptions.

Nonetheless, the results obtained from the calculations demonstrate a positive outlook. Based on the estimates used in this study, the incorporation of a standby mode has the potential to yield substantial energy savings. Specifically, it could save approximately 46% of the energy consumed during idle periods and 4.2% of the total energy, which includes both processing and idle energy. Furthermore, when subjecting the computed data to sensitivity analysis involving different rates of processing power, idle power, and the activation of standby mode, the results consistently demonstrate minor variations. These variations are within a range of 4 percentage points for idle energy and 0.6 percentage points for total energy, suggesting the robustness of the findings. Finally, the impact on projected potential energy savings outperforms the results obtained from a bi-objective model that focuses on minimizing both makespan and idle time was computed, using the same problem instances.

In the future it could be interesting to conduct in-depth case studies in real manufacturing environments to validate the effectiveness of energy-saving strategies and assess their practical applicability. Furthermore, investigate the viability of integration of energy storage

solutions, such as batteries or supercapacitors, to capture and store excess energy during idle periods for later use. This can enhance energy efficiency and reduce reliance on the grid.

References

Che, A., Wu, X., Peng, J., & Yan, P. (2017). Energy-efficient bi-objective single-machine scheduling with power-down mechanism. Computers & Operations Research, 85, 172–183. https://doi.org/10.1016/j.cor.2017.04.004

Da Col, G., & Teppan, E. (2022). Industrial-size job shop scheduling with constraint programming. Operations Research Perspectives, 9, 100249. https://doi.org/10.1016/j.orp.2022.100249

Dong, M. (2013). Parallel machine scheduling with limited controllable machine availability. International Journal of Production Research, 51(8), 2240–2252. https://doi.org/10.1080/00207543.2012.714002

Du, B., Chen, H., Huang, G. Q., & Yang, H. (2011). Preference Vector AnT Colony system for minimising make-span and energy consumption in a hybrid flow shop. In Springer eBooks (pp. 279–304). https://doi.org/10.1007/978-0-85729-652-8_9

Fernandes, J. M. R. C., Homayouni, S. M., & Fontes, D. B. (2022). Energy-Efficient Scheduling in Job shop Manufacturing Systems: A Literature review. Sustainability, 14(10), 6264. https://doi.org/10.3390/su14106264

Golari, M., Fan, N., & Jin, T. (2016). Multistage Stochastic Optimization for Production-Inventory Planning with Intermittent Renewable Energy. Production and Operations Management, 26(3), 409–425. https://doi.org/10.1111/poms.12657

Gong, G., Deng, Q., Gong, X., Liu, W., & Ren, Q. (2018). A new double flexible job-shop scheduling problem integrating processing time, green production, and human factor indicators. Journal of Cleaner Production, 174, 560–576. https://doi.org/10.1016/j.jclepro.2017.10.188

Gutowski, T. G., Murphy, C., Allen, D. T., Bauer, D., Bras, B., Piwonka, T. S., Sheng, P., Sutherland, J. W., Thurston, D. L., & Wolff, E. (2005). Environmentally benign manufacturing: Observations from Japan, Europe and the United States. Journal of Cleaner Production, 13(1), 1–17. https://doi.org/10.1016/j.jclepro.2003.10.004

Hassini, E., Surti, C., & Searcy, C. (2012). A literature review and a case study of sustainable supply chains with a focus on metrics. International Journal of Production Economics, 140(1), 69–82. https://doi.org/10.1016/j.ijpe.2012.01.042

He, Y., Li, Y., Wu, T., & Sutherland, J. W. (2015). An energy-responsive optimization method for machine tool selection and operation sequence in flexible machining job shops. Journal of Cleaner Production, 87, 245–254. https://doi.org/10.1016/j.jclepro.2014.10.006

Hidri, L., Alqahtani, A. S., Gazdar, A., & Youssef, B. B. (2021). Green Scheduling of Identical Parallel Machines with Release Date, Delivery Time and No-Idle Machine Constraints. Sustainability, 13(16), 9277. https://doi.org/10.3390/su13169277

IBM. (2021). Scheduling Examples. IBM Documentation. https://www.ibm.com/docs/en/icos/20.1.0?topic=programming-scheduling-examples

Ichoua, S., & Pechmann, A. (2013). Production scheduling for sustainable manufacturing systems. Key Engineering Materials, 572, 235–238. https://doi.org/10.4028/www.scientific.net/kem.572.235

Kawaguchi, S., & Fukuyama, Y. (2017). Parallel reactive tabu search for job shop scheduling problems considering peak shift of electric power energy consumption. In 2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE) (pp. 955-960). Kanazawa, Japan. doi:10.23919/SICE.2017.8105562.

Li, M., & Wang, G. (2022). A review of green shop scheduling problem. Information Sciences, 589, 478–496. https://doi.org/10.1016/j.ins.2021.12.122

Liang, P., Chen, W., Luo, M., He, W., & Liu, G. (2019). Job-shop scheduling problem considering power cost and makespan under time-of-use electricity price. Journal of Computational Methods in Sciences and Engineering, 19(4), 929–941. https://doi.org/10.3233/jcm-190001

Liu, Y., Dong, H., Lohse, N., Petrovic, S., & Gindy, N. (2014). An investigation into minimising total energy consumption and total weighted tardiness in job shops. Journal of Cleaner Production, 65, 87–96. https://doi.org/10.1016/j.jclepro.2013.07.060.

Liu, Q., Zhan, M., Chekem, F. O., Shao, X., Ying, B., & Sutherland, J. W. (2017). A hybrid fruit fly algorithm for solving flexible job-shop scheduling to reduce manufacturing carbon

footprint. Journal of Cleaner Production, 168, 668–678. https://doi.org/10.1016/j.jclepro.2017.09.037

Lu, Y. (2017). Industry 4.0: A survey on technologies, applications and open research issues. Journal of Industrial Information Integration, 6, 1–10. https://doi.org/10.1016/j.jii.2017.04.005

Luo, H., Du, B., Huang, G. Q., Chen, H., & Li, X. (2013). Hybrid flow shop scheduling considering machine electricity consumption cost. International Journal of Production Economics, 146(2), 423–439. https://doi.org/10.1016/j.ijpe.2013.01.028

Manne, A. S. (1960). On the Job-Shop Scheduling Problem. Operations Research, 8(2), 219–223. https://doi.org/10.1287/opre.8.2.219

Mashaei, M., & Lennartson, B. (2013). Energy reduction in a Pallet-Constrained flow shop through On–Off control of idle machines. IEEE Transactions on Automation Science and Engineering, 10(1), 45–56. https://doi.org/10.1109/tase.2012.2225426

Matthews, H. S., Hendrickson, C., & Weber, C. L. (2008). The Importance of Carbon Footprint Estimation Boundaries. Environmental Science & Technology, 42(16), 5839–5842. https://doi.org/10.1021/es703112w

Mouzon, G. C., & Yildirim, M. B. (2008). A framework to minimise total energy consumption and total tardiness on a single machine. International Journal of Sustainable Engineering, 1(2), 105–116. https://doi.org/10.1080/19397030802257236

Mouzon, G. C., Yildirim, M. B., & Twomey, J. M. (2007). Operational methods for minimization of energy consumption of manufacturing equipment. International Journal of Production Research, 45(18–19), 4247–4271. https://doi.org/10.1080/00207540701450013

Müller, D., Müller, M., Kress, D., & Pesch, E. (2022). An algorithm selection approach for the flexible job shop scheduling problem: Choosing constraint programming solvers through machine learning. European Journal of Operational Research, 302(3), 874–891. https://doi.org/10.1016/j.ejor.2022.01.034

Pinedo, M. (2012). Scheduling: Theory, Algorithms, And Systems (4th ed.). Springler. https://doi.org/10.1007/978-1-4614-2361-4

Rager, M., Gahm, C., & Denz, F. (2015). Energy-oriented scheduling based on Evolutionary Algorithms. Computers & Operations Research, 54, 218–231. https://doi.org/10.1016/j.cor.2014.05.002

Salido, M. A., Escamilla, J., Barber, F., Giret, A., Tang, D., & Dai, M. (2015). Energy efficiency, robustness, and makespan optimality in job-shop scheduling problems. Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 30(3), 300–312. https://doi.org/10.1017/s0890060415000335

Shrouf, F., Ordieres-Meré, J., García-Sánchez, Á., & Ortega-Mier, M. (2014). Optimizing the production scheduling of a single machine to minimize total energy consumption costs. Journal of Cleaner Production, 67, 197–207. https://doi.org/10.1016/j.jclepro.2013.12.024

Swaminathan, V., & Chakrabarty, K. (2003). Energy-conscious, deterministic i/o device scheduling in hard real-time systems. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 22(7), 847–858. https://doi.org/10.1109/tcad.2003.814245

Taillard, É. D. (1993). Benchmarks for basic scheduling problems. European Journal of Operational Research, 64(2), 278–285. https://doi.org/10.1016/0377-2217(93)90182-m

Terbrack, H., Claus, T., & Herrmann, F. (2021). Energy-Oriented Production Planning in Industry: A Systematic Literature Review and Classification Scheme. Sustainability, 13(23), 13317. https://doi.org/10.3390/su132313317

The World Bank. (2023). Inflation, consumer prices (annual %) - United States. https://data.worldbank.org/indicator/FP.CPI.TOTL.ZG?locations=US

U.S. Department of Energy. (2012). Turn Motors Off When Not in Use. Advanced Manufacturing Office. https://www.energy.gov/eere/amo/articles/turn-motors-when-not-use

U.S. Energy Information Administration. (2022). 2021 Average Monthly Bill- Industrial. https://www.eia.gov/electricity/sales_revenue_price/pdf/table5_c.pdf

U.S. Energy Information Administration. (2023). SHORT-TERM ENERGY OUTLOOK DATA BROWSER. https://www.eia.gov/outlooks/steo/data/browser/#/?v=8&f=M&s=0&start=199701&end=202412&map=&line

chart=ESICUUS~~~~ESCMUUS~ESRCUUS&ctype=linechart&maptype=0&chartin-dexed=0

U.S. Energy Information Administration. (2023). U.S. energy facts explained. https://www.eia.gov/energyexplained/us-energy-facts/

Wu, X., & Sun, Y. (2018). A green scheduling algorithm for flexible job shop with energy-saving measures. Journal of Cleaner Production, 172, 3249–3264. https://doi.org/10.1016/j.jclepro.2017.10.342

Annexes

Annex 1 - Taillard's Problem Instance 15x15 (1)

Times

94	66	10	53	26	15	65	82	10	27	93	92	96	70	83
74	31	88	51	57	78	8	7	91	79	18	51	18	99	33
4	82	40	86	50	54	21	6	54	68	82	20	39	35	68
73	23	30	30	53	94	58	93	32	91	30	56	27	92	9
78	23	21	60	36	29	95	99	79	76	93	42	52	42	96
29	61	88	70	16	31	65	83	78	26	50	87	62	14	30
18	75	20	4	91	68	19	54	85	73	43	24	37	87	66
32	52	9	49	61	35	99	62	6	62	7	80	3	57	7
85	30	96	91	13	87	82	83	78	56	85	8	66	88	15
5	59	30	60	41	17	66	89	78	88	69	45	82	6	13
90	27	1	8	91	80	89	49	32	28	90	93	6	35	73
47	43	75	8	51	3	84	34	28	60	69	45	67	58	87
65	62	97	20	31	33	33	77	50	80	48	90	75	96	44
28	21	51	75	17	89	59	56	63	18	17	30	16	7	35
57	16	42	34	37	26	68	73	5	8	12	87	83	20	97

Machines

7	13	5	8	4	3	11	12	9	15	10	14	6	1	2
5	6	8	15	14	9	12	10	7	11	1	4	13	2	3
2	9	10	13	7	12	14	6	1	3	8	11	5	4	15
6	3	10	7	11	1	14	5	8	15	12	9	13	2	4
8	9	7	11	5	10	3	15	13	6	2	14	12	1	4
6	4	13	14	12	5	15	8	3	2	11	1	10	7	9
13	4	8	9	15	7	2	12	5	6	3	11	1	14	10
12	6	1	8	13	14	15	2	3	9	5	4	10	7	11
11	12	7	15	1	2	3	6	13	5	9	8	10	14	4
7	12	10	3	9	1	14	4	11	8	2	13	15	5	6
5	8	14	1	6	13	7	9	15	11	4	2	12	10	3
3	15	1	13	7	11	8	6	9	10	14	2	4	12	5
6	9	11	3	4	7	10	1	14	5	2	12	13	8	15
9	15	5	14	6	7	10	2	13	8	12	11	4	3	1
11	9	13	7	5	2	14	15	12	1	8	4	3	10	6

Source: Taillard, É. D. (1993). Benchmarks for basic scheduling problems. European Journal of Operational Research, 64(2), 278–285

Annex 2 - Formatted Taillard's Problem Instance 15x15 (1)

[<6, 94>, <12, 66>, <4, 10>, <7, 53>, <3, 26>, <2, 15>, <10, 65>, <11, 82>, <8, 10>, <14, 27>, <9, 93>, <13, 92>, <5, 96>, <0, 70>, <1, 83>],

[<4, 74>, <5, 31>, <7, 88>, <14, 51>, <13, 57>, <8, 78>, <11, 8>, <9, 7>, <6, 91>, <10, 79>, <0, 18>, <3, 51>, <12, 18>, <1, 99>, <2, 33>],

[<1, 4>, <8, 82>, <9, 40>, <12, 86>, <6, 50>, <11, 54>, <13, 21>, <5, 6>, <0, 54>, <2, 68>, <7, 82>, <10, 20>, <4, 39>, <3, 35>, <14, 68>],

[<5, 73>, <2, 23>, <9, 30>, <6, 30>, <10, 53>, <0, 94>, <13, 58>, <4, 93>, <7, 32>, <14, 91>, <11, 30>, <8, 56>, <12, 27>, <1, 92>, <3, 9>],

[<7, 78>, <8, 23>, <6, 21>, <10, 60>, <4, 36>, <9, 29>, <2, 95>, <14, 99>, <12, 79>, <5, 76>, <1, 93>, <13, 42>, <11, 52>, <0, 42>, <3, 96>],

[<5, 29>, <3, 61>, <12, 88>, <13, 70>, <11, 16>, <4, 31>, <14, 65>, <7, 83>, <2, 78>, <1, 26>, <10, 50>, <0, 87>, <9, 62>, <6, 14>, <8, 30>],

[<12, 18>, <3, 75>, <7, 20>, <8, 4>, <14, 91>, <6, 68>, <1, 19>, <11, 54>, <4, 85>, <5, 73>, <2, 43>, <10, 24>, <0, 37>, <13, 87>, <9, 66>],

[<11, 32>, <5, 52>, <0, 9>, <7, 49>, <12, 61>, <13, 35>, <14, 99>, <1, 62>, <2, 6>, <8, 62>, <4, 7>, <3, 80>, <9, 3>, <6, 57>, <10, 7>],

[<10, 85>, <11, 30>, <6, 96>, <14, 91>, <0, 13>, <1, 87>, <2, 82>, <5, 83>, <12, 78>, <4, 56>, <8, 85>, <7, 8>, <9, 66>, <13, 88>, <3, 15>],

[<6, 5>, <11, 59>, <9, 30>, <2, 60>, <8, 41>, <0, 17>, <13, 66>, <3, 89>, <10, 78>, <7, 88>, <1, 69>, <12, 45>, <14, 82>, <4, 6>, <5, 13>],

[<4, 90>, <7, 27>, <13, 1>, <0, 8>, <5, 91>, <12, 80>, <6, 89>, <8, 49>, <14, 32>, <10, 28>, <3, 90>, <1, 93>, <11, 6>, <9, 35>, <2, 73>],

[<2, 47>, <14, 43>, <0, 75>, <12, 8>, <6, 51>, <10, 3>, <7, 84>, <5, 34>, <8, 28>, <9, 60>, <13, 69>, <1, 45>, <3, 67>, <11, 58>, <4, 87>],

[<5, 65>, <8, 62>, <10, 97>, <2, 20>, <3, 31>, <6, 33>, <9, 33>, <0, 77>, <13, 50>, <4, 80>, <1, 48>, <11, 90>, <12, 75>, <7, 96>, <14, 44>],

[<8, 28>, <14, 21>, <4, 51>, <13, 75>, <5, 17>, <6, 89>, <9, 59>, <1, 56>, <12, 63>, <7, 18>, <11, 17>, <10, 30>, <3, 16>, <2, 7>, <0, 35>],

[<10, 57>, <8, 16>, <12, 42>, <6, 34>, <4, 37>, <1, 26>, <13, 68>, <14, 73>, <11, 5>, <0, 8>, <7, 12>, <3, 87>, <2, 83>, <9, 20>, <5, 97>]

Source: Own Elaboration. Adapted from Taillard, É. D. (1993). Benchmarks for basic scheduling problems. European Journal of Operational Research, 64(2), 278–285

Annex 3 – Processing, Idle and Standby Rate (Level 1)

	Level 1			Standby Rate
	Processing Rate	Idle Rate	Switching to Standby Mode	
Machine 0	1230	230	9840	20
Machine 1	1160	180	9280	22
Machine 2	1150	190	9200	25
Machine 3	1380	230	11040	30
Machine 4	1040	220	8320	25
Machine 5	1270	230	10160	27
Machine 6	1170	220	9360	22
Machine 7	1000	170	8000	20
Machine 8	1300	250	10400	30
Machine 9	1360	250	10880	28
Machine 10	1350	250	10800	22
Machine 11	1030	190	8240	21
Machine 12	1310	230	10480	28
Machine 13	1060	200	8480	29
Machine 14	1450	300	11600	30

Source: Own Elaboration. Adapted from Wu, X., & Sun, Y. (2018). A green scheduling algorithm for flexible job shop with energy-saving measures. Journal of Cleaner Production, p. 3259

Annex 4 – Constraint Programming Model 1 (15 Machines)

```
using CP;


int nbJobs = ...;

int nbMchs = ...;

range Jobs = 0..nbJobs-1;

range Mchs = 0..nbMchs-1;

tuple Operation {

   int mch; // Machine

   int pt;  // Processing time

}


// Define energy consumption rates for each machine

int processingRate[Mchs] = [1230, 1160, 1150, 1380, 1040, 1270, 1170, 1000, 1300, 1360,
1350, 1030, 1310, 1060, 1450];

int idleRate[Mchs] = [230, 180, 190, 230, 220, 230, 220, 170, 250, 250, 250, 190, 230, 200,
300];


Operation Ops[j in Jobs][m in Mchs] = ...;


dvar interval itvs[j in Jobs][o in Mchs] size Ops[j][o].pt;

dvar sequence mchs[m in Mchs] in all(j in Jobs, o in Mchs : Ops[j][o].mch == m) itvs[j][o];


// Add energy consumption variables

float processConsumption[Mchs];

float idleConsumption[Mchs];

float energyConsumption[Mchs];
```

```
execute {
    cp.param.FailLimit = 10000;
}

minimize max(j in Jobs) endOf(itvs[j][nbMchs-1]);
subject to {
    forall (m in Mchs)
        noOverlap(mchs[m]);
    forall (j in Jobs, o in 0..nbMchs-2)
        endBeforeStart(itvs[j][o], itvs[j][o+1]);
}

int astart[j in Jobs][o in Mchs];
int aend[j in Jobs][o in Mchs];

execute {
    for (var j in Jobs)
    for(var o in Mchs){
      astart[j][o] = itvs[j][o].start;
      aend[j][o] = itvs[j][o].end;
  }
}

sorted {int} startItvM[m in Mchs] = {  astart[j][o] | j in Jobs, o in Mchs : Ops[j][o].mch
== m};
sorted {int} endItvM[m in Mchs] = {  aend[j][o] | j in Jobs, o in Mchs : Ops[j][o].mch ==
m};
```

```
assert forall(m in Mchs) card(startItvM[m]) == nbJobs;

assert forall(m in Mchs) card(endItvM[m]) == nbJobs;


int idleDuration[m in Mchs] = sum(j in 1..nbJobs-1) ( item(startItvM[m], j) -
item(endItvM[m], j-1) );


execute{

  idleDuration;

}


// Calculate processing energy consumption

execute {

    for (var m in Mchs) {

        processConsumption[m] = 0;

        for (var j in Jobs) {

            for (var o in Mchs) {

                if (Ops[j][o].mch == m) {

                    processConsumption[m] += (

                        itvs[j][o].size * processingRate[m]

                    );

                }

            }

        }

    }

}

// Calculate Idle Energy Consumption

execute {
```

```
    for (var m in Mchs) {
        idleConsumption[m] = 0;
        for (var j in Jobs) {
            for (var o in Mchs) {
                if (Ops[j][o].mch == m) {
                    idleConsumption[m] += (
                        idleDuration[m]/15  * idleRate[m]
                    );
                }
            }
        }
    }
}
//Calculate Total Energy Consumption
execute {
    for (var m in Mchs) {
        energyConsumption[m] = 0;
        for (var j in Jobs) {
            for (var o in Mchs) {
                if (Ops[j][o].mch == m) {
                    energyConsumption[m] += (
                        idleConsumption[m]/15 + processConsumption[m]/15
                    );
                }
            }
        }
```

```
    }
}
execute {
    for (var j = 0; j <= nbJobs-1; j++) {
        for (var o = 0; o <= nbMchs-1; o++) {
            write(itvs[j][o].start + " ", "- ", itvs[j][o].end + " ");
        }
        writeln("");
    }
}
execute {
    writeln("Total Energy Consumption for Machines:");
    for (var m = 0; m < nbMchs; m++) {
        writeln("Machine ", m, ": ", idleDuration[m], " - ", idleConsumption[m], " / ", pro-
cessConsumption[m], " / ", energyConsumption[m]);
    }
}
```

Source: Own Elaboration. Adapted from IBM (2021). Scheduling Examples. IBM Documentation.

Annex 5 - Processing, Idle and Standby Rate (Level 2 & 3)

	Level 2			Level 3			Standby Rate
	Processing Rate	Idle Rate	Switching to Standby	Processing Rate	Idle Rate	Switching to Standby	
Machine 0	1510	320	12080	2270	370	18160	20
Machine 1	1500	280	12000	1820	350	14560	22
Machine 2	1390	300	11120	1880	350	15040	25
Machine 3	1920	330	15360	2340	390	18720	30
Machine 4	1500	310	12000	2220	380	17760	25
Machine 5	1560	270	12480	2260	370	18080	27
Machine 6	1510	300	12080	2160	400	17280	22
Machine 7	1210	290	9680	1690	350	13520	20
Machine 8	1770	320	14160	2510	400	20080	30
Machine 9	1960	310	15680	2510	380	20080	28
Machine 10	1850	340	14800	2440	390	19520	22
Machine 11	1480	280	11840	1920	320	15360	21
Machine 12	1860	310	14880	2290	390	18320	28
Machine 13	1450	300	11600	1960	400	15680	29
Machine 14	2090	350	16720	2970	400	23760	30

Source: Own Elaboration. Adapted from Wu, X., & Sun, Y. (2018). A green scheduling algorithm for flexible job shop with energy-saving measures. Journal of Cleaner Production, p. 3259

Annex 6 – Objective Function of Bi-Objective Model

```
minimize max(j in Jobs) endOf(itvs[j][nbMchs-1]) +
    sum(m in Mchs) (max(j in Jobs, o in Mchs : Ops[j][o].mch == m) endOf(itvs[j][o], 0) -
                    min(j in Jobs, o in Mchs : Ops[j][o].mch == m) startOf(itvs[j][o],
99999) -
                    sum(j in Jobs, o in Mchs : Ops[j][o].mch == m) Ops[j][o].pt);;
```

Source: Own Elaboration. Adapted from IBM (2021). Scheduling Examples. IBM Documentation