EFFICIENT AND SUSTAINABLE SCHEDULING STRATEGIES FOR SMART MANUFACTURING SYSTEMS

SUN YIGE

PhD

The Hong Kong Polytechnic University

2024

The Hong Kong Polytechnic University

Department of Industrial and Systems Engineering

Efficient and Sustainable Scheduling Strategies for Smart Manufacturing Systems

Sun Yige

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

July 2024

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

____________________________ (Signed)

_______SUN Yige_____ (Name of student)

Abstract

Propelled by Industry 4.0 technical advancements, smart manufacturing is increasingly prioritized and implemented across various manufacturing systems. In this context, technology-supported scheduling plays crucial roles in ensuring smooth manufacturing processes, enabling agile responses to orders, and reducing operational costs. Two crucial aspects of this domain should be emphasized.

First, from a physical perspective, the introduction of autonomous mobile robots (AMRs) automates laborious material handling tasks, enhancing operational efficiency. However, this also significantly increases the system complexity and necessitates precise production scheduling to accommodate complicated machine-robot interactions. Additionally, analysis of the robot fulfillment process reveals wastes of energy raised due to the mismatching between robot operations and machine processing. Realizing the lack of studies considering energy elimination from the perspective of facilitating operational collaboration, this dissertation (in Chapter 3) investigates an energy-aware robotic job shop scheduling problem. To conquer the complexity induced by machine and robot operations and enhance the collaboration between processing and moving, network-based energy-aware modelling approaches are developed. Computational experiments show their capabilities in reducing carbon emissions and maintaining throughout.

Second, the integration of CPS has been instrumental in connecting the physical and digital worlds. Traditional scheduling methods, which typically rely on expert experiences, frequently overlook the complex interplay of various real-world factors, leading to impractical or inefficient schedules. The advent of IoT enables the collection of vast amounts of data from physical systems. It is thus promising to uncover useful

patterns from historical data and incorporate such data-driven insights into decision optimization processes to derive efficient schedules that are highly applicable to real-world scenarios. Motivated by scheduling challenges in real-world systems and the lack of studies considering the incorporation of multiple realistic factors on production efficiency into the scheduling process, how the multiple factors during the production process can influence job processing status are explored (see Chapters 4 and 5). Moreover, whether the influences can be captured and utilized to enhance production scheduling is explored. Specifically, the study in Chapter 4 aims to jointly predict the job processing time and processing rate level to facilitate resource allocation and timely reporting of production status. A multi-input modules-supported dual-task learning model is proposed, which achieves good performance by capturing influences from various aspects within the performing sequence and leveraging the synergy between dual learning tasks. The study in Chapter 5 further develops a context-based scheduling method, which integrates the prediction of context-based job processing rate (CBPR) under varying execution scenarios to the optimization process. A CBPR-guided branch-and-price-based scheduling approach is proposed, which can effectively identify promising execution positions for individual jobs so that overall production efficiency is substantially enhanced.

To conclude, this research is devoted to developing efficient and sustainable scheduling methods for smart manufacturing systems. The whole work focuses on two main perspectives: (i) coordinating operations in complex robotic production cells to achieve green production, and (ii) deriving data-driven prediction and scheduling optimization methods to timely inform and maximize production efficiency. Important academic and practical insights are generated.

Publications Arising from this Thesis

Journal Publications and Working Papers

1. **Sun, Y.**, Chung, S.-H., Choi, T.-M., & Wang, Y. (2024). Feature-driven production scheduling systems: Unveiling and exploiting job processing rate dependencies. *IEEE Transactions on Systems, Man and Cybernetics: Systems*, under review. (Related to Chapter 5)
2. **Sun, Y.**, Chung, S.-H., Choi, T.-M. (2024). Predictive analytics in manufacturing systems: Simultaneously forecasting processing time and rate level with a deep learning approach. *Working paper*. (Related to Chapter 4)
3. Wen, X., **Sun, Y.**, Ma, H.L., & Chung, S.-H. (2023). Green smart manufacturing: Energy-efficient robotic job shop scheduling models. *International Journal of Production Research*, *61*(17), 5791-5805. (Related to Chapter 3)

Other publications during the PhD period

4. Ma, H.L., **Sun, Y.**, Mo, D.Y., & Wang, Y. (2023). Impact of passenger unused baggage capacity on air cargo delivery. *Annals of Operations Research*. In press.
5. Ma, H.L., **Sun, Y.**, Chung, S.-H., & Chan, H.K. (2022). Tackling uncertainties in aircraft maintenance routing: A review of emerging technologies. *Transportation Research Part E: Logistics and Transportation Review*, *164*, 102805.

Conference

Sun, Y. (presenter) (with Chung, S.-H., Choi, T.-M.). Context-aware processing rate guided production scheduling: A forecasting embedded branch-and-price heuristic method. *33rd European Conference on Operational Research*, Copenhagen, Denmark, 30 June – 3 July 2024.

Acknowledgement

First, I would like to express my deepest gratefulness to my supervisor Dr. Sai-Ho Chung. Meeting him and being his student has been one of the greatest fortunes of my life. From that moment, I began to explore my potential and gradually developed into a researcher. Dr. Chung's unwavering support, guidance, and encouragement have been instrumental throughout my doctoral journey. His invaluable mentorship has always been pivotal in shaping my research, refining my ideas, and building my confidence to overcome challenges. The completion of my PhD would not be possible without his patience and meticulous instructions. The valuable opportunities he provided also have contributed immensely to my growth in various aspects. I would also extend my heartfelt gratitude to my supervisor, Professor Tsan-Ming Choi. His patient guidance and insightful comments have consistently sparked new thoughts that drive my research forward. His commitment to excellence and research attitude have been constantly inspiring me to strive for greatness. I am also deeply indebted to my supervisor Dr. Xin Wen for her guidance and support during my research, which helped me immeasurably.

Moreover, I also would like to thank my dear fellows for their insightful sharing, valuable discussions, and the great joys they brought me during this journey. Also, my deepest appreciation goes to many of my friends. The myriad encouragement they provided, and moments of happiness shared with them have been a vital source of support for me. This journey is greatly enriched by their presence.

Lastly, I am profoundly thankful to my father and mother. Their boundless love, understanding, and encouragement have been a constant source of nourishment and strength for me. Also, special thanks go to my fiancé. His emotional support and unconditional love have provided an anchor to my life, sustaining me to overcome many challenges during this journey. I am forever grateful.

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

1.1 Background of Research

1.1.1 Smart manufacturing and scheduling

Smart manufacturing is defined as *the integration of information technologies and robotic automation into production processes to create efficient, interconnected, and adaptive manufacturing systems* (Mourtzis, 2024). It is characterized by incorporating various advanced Industry 4.0 technologies, such as autonomous robots, Internet of Things (IoT), data analytics, and artificial intelligence (AI), to enhance production intelligence (Mourtzis, 2024; Schlemitz & Mezhuyev, 2024; Singh et al., 2023). These technologies are accelerating the fusion of devices in the physical world and virtual operations in cyberspace (Singh et al., 2024; Yang et al., 2019; Zheng et al., 2018). The so-formed cyber-physical systems (CPS) integrate sensors, actuators, devices, and computational algorithms to support decision making, accommodate complex emergent behaviours, enhance real-time adaptivity, and empower the workforce and environmental sustainability (Singh et al., 2024). In recent years, smart manufacturing has been increasingly appealing to production practitioners[1]. The global smart manufacturing market has been experiencing significant growth and is expected to reach $386.4 billion in 2025 and a further increase to $443.9 billion in 2026[2].

Examining the critical technologies propelling smart manufacturing, the markets for IoT devices, AI, and big data analytics are estimated to expand greatly[2]. Another notable progress from the physical aspect is the introduction of autonomous mobile

[1] Details are available at: https://www.oracle.com/industrial-manufacturing/smart-factory-and-smart-manufacturing/ (Accessed on 20 June 2024)

[2] Details are available at: https://scoop.market.us/smart-manufacturing-statistics/ (Accessed on 20 June 2024)

robots (AMR). The AMR market is estimated to expand from $3.88 billion in 2024 to $8.02 billion by 2029[3]. The growth demonstrates the increasing adoption of AMRs in streamlining manufacturing processes. In autonomous smart manufacturing systems, AMRs replace human beings in conducting repetitive and tedious tasks, such as picking, transporting, and sorting items[3]. Beyond providing a cost-efficient solution, applying AMR also largely enhances manufacturing resilience by providing continuous services and reducing labor-intensive tasks[4]. However, despite the great portion of research attention paid to robot-facilitated production system operations, seldom studies have considered the energy issue of the robot-facilitated production process (Gürel et al., 2019). Transportation of raw materials, work-in-progress pallets, and finished goods usually consumes substantial electrical power due to frequent movement and precise positioning. For example, in the automotive sector, robots account for 8% of total energy consumption[5]. Besides, machine idling/waiting also leads to significant energy consumption, as 30% of energy consumption by machines arises from stand-by-operations to maintain process stability[6]. This inefficient energy consumption raises practitioners' consciousness about carbon footprints and sustainability development demand.

Besides the adoption of autonomous devices, the deployment of IoT and sensors enables the collection of data from various sources including facilities, participants, and activities during the manufacturing process (e.g., product line, equipment, production process, labour activity, and environmental conditions) (Wang et al., 2018a). Thus,

[3] Details are available at: https://www.mordorintelligence.com/industry-reports/autonomous-mobile-robot-market (Accessed on 20 June 2024)
[4] Details are available at: https://roboticsandautomationnews.com/2021/02/01/the-business-case-for-autonomous-mobile-robots-in-manufacturing/40092/ (Accessed on 20 June 2024)
[5] Details are available at: https://www.roboticstomorrow.com/story/2021/03/3-trends-in-robotics-energy-consumption/16385/ (Accessed on 20 June 2024)
[6] Details are available at: https://www.iwu.fraunhofer.de/content/dam/iwu/en/documents/EffPro_en.pdf (Accessed on 20 June 2024)

sensors act as the crucial link between the tangible physical environment with the digital realm. In traditional production environments, decisions can only be made roughly according to prior experience, which may lead to a severe mismatching between personnel skills and job requirements, as well as a low utilization rate of materials and machine capacity (Qiao et al., 2021; Sotskov & Werner, 2014; Workneh & Gmira, 2022). In the new smart manufacturing era, access to a wide spectrum of data can thus facilitate the adoption of data analytics and AI methods to promote a better understanding of changeable production situations subjecting to the complex interplay of substantial factors within production process, such as machine conditions, material utilization, and operational properties. With the adoption of AI techniques, the influences of such factors can be captured and utilized to derive smarter decisions in areas such as planning, scheduling, maintenance, energy cost control, and supply chain management (Rossit et al., 2019a; Wu et al., 2021).

Among all the decisions in the manufacturing process, scheduling plays a pivotal role. It is dedicated to ensuring that all operations and procedures run smoothly and adhere to their designated timelines (Alemão et al., 2021). The scheduling problem investigates assigning and prioritizing a series of non-preemption jobs on a set of machines to maximize production efficiency with minimum costs. Such objectives can be implied by minimizing the production time, eliminating tardiness, maximizing the throughput, etc. (Mokhtari & Hasani, 2017; Zhang & Chiong, 2016). The new era of smart manufacturing proposes new contexts, requirements, and research directions for scheduling the shop floor (Parente et al., 2020). Combined with the new features of smart manufacturing as aforementioned, scheduling decisions thus should well address the interactions between different components of the complicated production systems induced by the incorporation of smart devices. Therefore, operations and specific timelines for activities of different participants or subsystems should be carefully modelled so as to ensure the generation of practical schedules, and further to realize efficient and sustainable objectives. Besides, by harnessing the power of Industrial IoT (IIoT)-enabled data analytics, it is likely to develop novel scheduling processes driven by real data based insights and thus achieve a more effective alignment of resource allocation (such as workforce allocation, material usage, and machine configurations) with tasks to promote scheduling accuracy and improve resource utilization.

Even though traditional scheduling methods proposed in the past decades remain effective in many production scenarios, they face significant limitations when applied to smart manufacturing systems. First, they tend to oversimplify the interactions among production components by neglecting some important subprocesses or operations, such as material transportation among the machines, setup processes for individual jobs, and other restrictions on storage, delivery, or machine buffer setting (Lee & Chen, 2001; Liu et al., 2018). Oversimplified schedules may result in infeasibility in real-world production (Sotskov & Werner, 2014; Workneh & Gmira, 2022). Secondly, vital parameters are often assumed to be deterministic or follow certain empirical distributions (Ramírez-Velarde et al., 2017). Consequently, jobs are often scheduled with given parameters (e.g., processing time based on a rough estimation by the planner). These empirical estimates, however, may not reflect the effect of real production situations on production performance due to the neglect of the potential interactions of factors involved (Sotskov & Werner, 2014; Wu et al., 2021b). Moreover, by doing so, it is implicitly assumed that production always proceeds at a normal status, which leads to the fragility of the system in dealing with unexpected downtime and delays (Lu et al., 2017). Additionally, traditional methods, relying on heuristics and stochastic modelling, struggle to grasp the complex and high-dimensional impacts of various elements within manufacturing systems, which can lead to system instability or failure (Sharp et al., 2018).

Motivated by the significance of smart manufacturing and the limitations of existing methodologies, this PhD study focuses on improving scheduling decisions of smart production floors from two main aspects at the operational level, namely physical operations collaboration and AI-empowered decision making, with advances in modelling, deep learning, and optimization algorithms. Considering the importance of integrating AMR into the production system and the challenges brought by such integration (which is detailed in Section 1.1.2), this PhD study concentrates on a green modelling approach to realize the integration of AMR into the system with the consideration of sustainability impact. Then, to explore how the influencing factors in the production environment affect production performance based on historical data (which is detailed in Section 1.1.3), this study is devoted to developing a novel deep learning (DL) architecture to capture the influences of production elements on performance. Then, to provide a data-driven scheduling decision process that mitigates discrepancies between the scheduled production timeline and the actual implementation, (which is detailed in Section 1.1.4), the study develops an AI-empowered scheduling method to find more accurate and efficient scheduling solutions.

1.1.2 Autonomous robotic delivery system[7]

As previously mentioned, AMRs have been increasingly adopted for intralogistics. For example, Amazon has widely deployed the Kiva system in logistics centres, where AMRs are used to move inventory pods. Large logistics companies like JD and Alibaba also largely use robots in their autonomous warehouses (Li et al., 2020a). In the manufacturing sector, the adoption of AMR is different from that in the logistics centres. The AMR adopted in the manufacturing process should accommodate the machine

[7] As a remark, a part of 1.1.2 has been published in paper Wen, X., Sun, Y., Ma, H. L., & Chung, S.H. (2023). Green smart manufacturing: energy-efficient robotic job shop scheduling models. *International Journal of Production Research*, *61*(17), 5791-5805.

performing states and fulfill the product requirements. For example, the robot can only perform delivery after an operation is completed and should deliver the job to its next designated machine (Dai et al., 2019). Thus, integrating AMR seamlessly into existing processes and workflows is challenging and may introduce layers of complexity. To achieve such integration, it should tackle the interactions among machines and the robotic delivery process (e.g., determining the performing sequence of jobs on machines and routing/priority of robot movement). Besides, the application of robots leads to higher energy (electricity) [8] consumption, which worsens the environmental concerns for the energy-intensive manufacturing industry (Zhang & Yan, 2021). The existing practice overlooks the coordination between the machinery production process and the robot movement process, which leads to excessive energy wastes (Gürel et al., 2019; Liu et al., 2019b). Implementing the aforementioned coordinated scheduling mechanism presents two major challenges. Firstly, the scheduling framework must identify the various interrelated factors that contribute to energy inefficiency in the production system. Secondly, the scheduling scheme must minimize energy waste resulting from speed mismatches between the two processes. This task is further complicated by the interactions between these processes. In the following section, the background of AMR-facilitated scheduling in a typical job shop setting (i.e., robotic job-shop scheduling) is first introduced, followed by the energy concerns in robotic cells and corresponding energy-reduction solutions.

Robotic job-shop scheduling problem

The robotic job-shop scheduling problem focuses on the scheduling and coordination of machines and a delivery robot in a robotic cell (Parente et al., 2020). It aims to identify the optimal production schedule for machines and the optimal delivery

[8] In a robotic cell, the energy consumed is generally electricity. In this study, “energy” and “electricity” are used interchangeably.

route for robots, with the objective of minimizing makespan to improve productivity (Brucker et al., 2012). In the basic setting, the machines are linearly arranged, and a robot is responsible for moving materials or semi-products among the machines for processing. A job usually consists of several operations that will be carried out on different machines. Different from the flowshop scheduling setting that is designed for the mass production of a single type of product, the job-shop setting allows different product types (i.e., with different operations or operating sequences) and one product may re-enter a machine (Demir & İşleyen, 2013). Besides, the involvement of robotic transportation process largely increases the problem complexity. That is because three decision sequences (i.e., sequence of operations in one job, sequence of operations on a machine, and sequence of operations carried out by the delivery tool) and corresponding restrictions (machine blocking, robot availability, robot delivery deadlock avoidance etc.) should be considered and fulfilled in the scheduling solution.

Energy concerns in robotic cells

The energy waste in robotic cells comes from several aspects. First, as the movement of the robot is subjected to diverse restrictions like robot availability, machines may stay idle for a long period (Koulamas & Panwalkar, 2019). Second, the robot/machines generally moves/work at a constant speed. It is thus commonly seen that (i) products are blocked on machines for a long time (i.e., the product is blocked on the processing machine after completion, waiting for the availability of the robot, named as *machine blocking*), or (ii) the robot arrives at a machine earlier than the completion of the current operation and has to wait there before conducting the next delivery (named as *robot partial-blocking*)[9]. These circumstances imply poor

[9] A *robot full-blocking* refers to the situation that the robot must wait at the machine for the whole operation process to deliver the same job (as instructed by the optimal schedule that minimizes the makespan while avoiding deadlocks).

coordination between the machine production process and the robot movement process, which causes nonnegligible energy waste. Considering the increasing energy costs and the growing public awareness of sustainability, the low energy efficiency in robotic cells greatly dampens the benefits brought by robotic technology (Mokhtari & Hasani, 2017). It is thus of great significance to enhance the energy efficiency of robotic cells via improving scheduling decisions (Jiang & Wang, 2019; Lamotte & Geroliminis, 2021).

Speed adjustment system for machinery and delivery coordination

According to (Zhang & Chiong, 2016), the energy consumption rate of machine production declines if it switches to a slower processing speed. Similarly, robots are shown to consume less electricity at a slower moving speed (Paryanto et al., 2015). Therefore, it is promising to achieve energy conservation through speed adjustments for both machines and robots. The mechanism of energy-reduction by speed adjustment is explained as follows. On one hand, to reduce machine idling and blocking periods, machines can process at a slower speed. In fact, as long as the processing finishes before the arrival of the robot, the overall makespan is not affected. In this way, the production energy consumption declines (as the processing speed is slower), while the energy waste caused by machine idling is also reduced. On the other hand, robot partial blocking can be eliminated/reduced if the robot moves at a slower speed, thus achieving movement energy reduction. Accordingly, the systematic energy consumption of a robotic cell can be reduced significantly through proper operating speed selection and better coordination between the machine production process and the robot movement process.

1.1.3 Production under multiple uncertain influencers

Traditional scheduling methods are commonly established based on deterministic information directly related to scheduling, e.g., operation processing time, delivery dates, etc., which assume that the production systems always operate normally (Rossit et al., 2019a). However, in the actual production process, operation delays, uncertain events, and abnormal disturbances occasionally appear (Malhotra et al., 2015). For example, machines need to experience downtime occasionally due to the necessity of setup works, material changes, and other activities of operations that need human intervention. Therefore, the traditional static view of deterministic problem settings may not suit the varying performing circumstances, leading to production deviation and seriously affecting the success of schedule implementation (Fang et al., 2019). Besides, conventional uncertainty-aware methods focus on involving stochasticity or robustness in the scheduling. However, the empirical-based depiction of uncertainty may not affect the real situation and ignores the interactions of production factors on the performance. Taking advantage of IIoT-enabled data-collection, real-time data can be obtained. AI-based methods can be developed to explore the joint effect of production factors on performance. In this section, the influencing factors are first identified. Then, the machine learning methods and production performance indicators are elaborated.

Influencing factors and effects on production performance[10]

The actual performance of many manufacturing sectors, such as printing, dyeing, and construction molding, rely on a portfolio of factors e.g., operator operations, machine operating status, resource use, operator proficiency, and even environmental

[10] As a remark, part of this section is included in paper Sun, Y., Chung, S.H., Choi, T.M., & Wang, Y. (2024). Feature-driven production scheduling systems: Unveiling and exploiting job processing rate dependencies. *IEEE Transactions on Systems, Man and Cybernetics: Systems*, under review.

factors. For example, in the dyeing process, the rates of dye absorption and fixation are controlled with dyeing temperature, liquor agitation, pH, and retarding agents. Also, the overall processing time (rate) of a dyeing task is affected by the depth of shade, the type of dyestuff, the nature of the textile material and the dyeing machine[11]. Similarly, the processing rate of an injection moulding process highly depends on the setup procedures, the shot size, the barrel temperature, and the injection speed[12]. Therefore, slight variations in the contexts of two identical jobs (i.e., jobs with the same quantity and quality requirements) may result in significant discrepancies in the processing rate, which can disturb the implementation of planned schedules. As a result, if the influences of the multiple related factors can be integrated into the planning and scheduling of such production activities, the schedules can more precisely accommodate the actual resource/processing time demands, which benefits cost control and quality enhancement.

To further illustrate the variabilities or influential factors that are defined by the specific context, take the investigated printing company in China as an example. First, job-related factors such as material usage (paper and ink), quality of similar materials by different suppliers, customer importance, and requested quality level vary and can directly affect the processing rate from the product standard and the ease of use of materials. Another important aspect is the operator in charge. Operators with higher professionals may be more experienced and proficient in controlling the machine, conducting preparation jobs before printing (e.g., cleaning the die, changing fixtures, etc.), and responding to various circumstances (e.g., adjusting machine parameters, replacing washers, etc.). Moreover, environmental elements, such as the temperature and humidity around the printing machine can make a difference in the materials (e.g.,

[11] Details are available at: https://textilelearner.net/dyeing-methods/ (Accessed on 22 June 2024)

[12] Details are available at: https://www.plasticstoday.com/injection-molding/troubleshooter-key-steps-stable-injection-molding-process (Accessed on 22 June 2024)

fluidity of the ink and the absorptivity of paper) and thus affect the quality of the product, which may consequently cause downtime for resetting the machine.

Besides the factors related to the specific job execution, through data analytics, it is discovered that the execution of the immediately preceding job may highly affect the processing rate of succeeding jobs, The rationale behind this is that the preceding job will affect the changeover operations of the next job, such as necessary setups and configurations of machine, material utilization, and consistency in human operations. In addition, the predecessors may also exert an implicit influence on the following job performance through material availability, machine vibration or wear, and operator status fluctuation. Accordingly, the influencing factors are summarized into three levels, the direct influences (which are owing to the changes of specific factors related to the job under processing), the adjacent influence (which is caused by the influences of the immediate predecessor), and sequential influence (which is due to a series of predecessors).

Indicators to track operating performance

To measure the effects of the above factors on production efficiency, two performance indicators can be applied. The first one is the job processing time, which directly reflects the absolute time of processing a job. Knowing processing time is vital for establishing schedules that enable a smooth production process with less deviation. However, relying solely on processing time may not provide a comprehensive view of processing performance, as it does not necessarily reflect the current processing status. For instance, it may not indicate whether a job is being completed at an acceptable efficiency level or if there are underlying inefficiencies causing the processing rate to lag behind the expected standard. To solve this concern, the other indicator adopted is the relative processing rate level (PR level for short), which signifies the relative production efficiency of a job (Owen & Blumenfeld, 2008).

In particular, the PR level embeds more information that cannot be indicated by the processing time. For instance, within a given timeframe, the PR level can reveal insights into cost efficiency, the alignment between personnel skills and job requirements, the smoothness of material flow, and the environmental advantages or the presence of errors and slowdown factors. Previous studies point out that the processing rate varies due to influential factors like the starting time of the job in the sequence, the number of jobs being processed simultaneously, and other settings in the system (Alidaee & Womer, 1999; Glock & Grosse, 2021; Schweitzer & Seidmann, 1991). They model the fluctuation in processing rates with linear or non-linear functions (Alidaee & Womer, 1999; Baldea & Harjunkoski, 2014). However, these approaches do not take into account the particular circumstances or conditions in which a job is performed. Therefore, by referring to the relative PR level, the decision makers can get knowledge about whether the performing factors are benefitting or impairing the job processing. Consequently, resource adjustments regarding machines, operators, materials, and environmental indicators (e.g., temperature and humidity etc.) can be applied to enable more jobs to be processed with a normal or even high PR level.

Machine learning methods for prediction and classification

Machine learning (ML) methods have been widely recognized for their ability to perform a variety of regression and classification tasks, offering the benefits of automating decision-making processes and uncovering patterns within large datasets. Traditional machine learning methods, such as the support vector machine and decision trees, often provide results with good explainability, as they typically require domain knowledge and human expertise in model construction and feature extraction (Mende et al., 2023). However, such methods also suffer notable limitations. For example, the shallow nature of traditional ML architectures restricts the sophistication and accuracy of the models to learn deeper data representations (Janiesch et al., 2021). Furthermore, as data points are usually treated as independent entities, it is challenging for these methods to capture complex temporal dependencies that exist between multiple features and different data points, such as time series trends (Han et al., 2019).

In recent years, advancements in statistics and optimization theory have significantly shifted the research focus toward deep learning, which largely transformed the methods to extract knowledge from high-dimensional data. With the deepening layers, DL models become more powerful in characterizing intricate relationships within the data (Han et al., 2019). By developing deep learning models that incorporate diverse functional layers, such as convolution layers (for extracting latent representations from structured inputs) as well as recurrent and transformer layers (for identifying temporal dependencies within time series data), deep learning architectures can be tailored to meet the specific requirements of various tasks.

Multi-input modules supported dual-task learning

Deep learning models process the flexibility of designing for suiting specific tasks. It is crucial to design input modules that can more explicitly reveal the intrinsic patterns, thereby enabling the learning model to capture the representation much more easily. Besides, most neural networks constructed in the literature only focus on training for a single task, such as predicting machine speed, and loading rate (Liu et al., 2019a). Notably, recent research has demonstrated the effectiveness of adopting multi-output learning in simultaneously predicting multiple outputs given an input (Xu et al., 2019). The rationale behind this is that many learning tasks share commonalities. Therefore, through training with neural networks sharing information between different tasks, the knowledge learned from one task can be transferred to another, which enhances the performance of joint learning with the synergy between the tasks (Xu et al., 2019).

To track the real-time production performance, it would be beneficial to know both processing time and the relative PR rate for a more comprehensive understanding of the production status. It is worth noting that these two indicators have inherent correlations as a relatively longer processing time regarding one performing task often represents a lower processing rate level. Thus, it can be inferred that the learning of both tasks may share inner representations. The adoption of multi-output architecture is promising in achieving good performance of both tasks.

1.1.4 Context-based production rate guided scheduling

As previously mentioned, most conventional scheduling approaches use predetermined parameters for scheduling. These parameters usually heavily rely on expert estimation (e.g., processing time and setup time), which may not be very accurate (Qiao et al., 2021). Such inaccurate estimations may cause the scheduling solutions to be fragile in real practice, leading to production delays and inefficiencies (Sotskov & Werner, 2014). Moreover, in normal circumstances, the processing time of a job will be positive to the quantity of production output and fluctuate around an average value, which can be described as normal distribution (depicted as *Origin schedule* in Figure 1-1). However, As demonstrated in Section 1.1.2, the interplay of multiple factors (e.g., the status fluctuation of machines, status of operators, material use and its performance in different environmental circumstances) may drive the actual processing time away from its normal value/range. Therefore, the actual performance of the processing time may follow some real distributions that cannot be described with well-known distributions, thus leading to delays or inefficiencies (depicted by *Scenario 1* and *Scenario 2* in Figure 1-1). Therefore, prior literature based on stochastic programming or robust programming with empirical distribution assumptions may not well capture the interrelationships between job and performing scenarios.

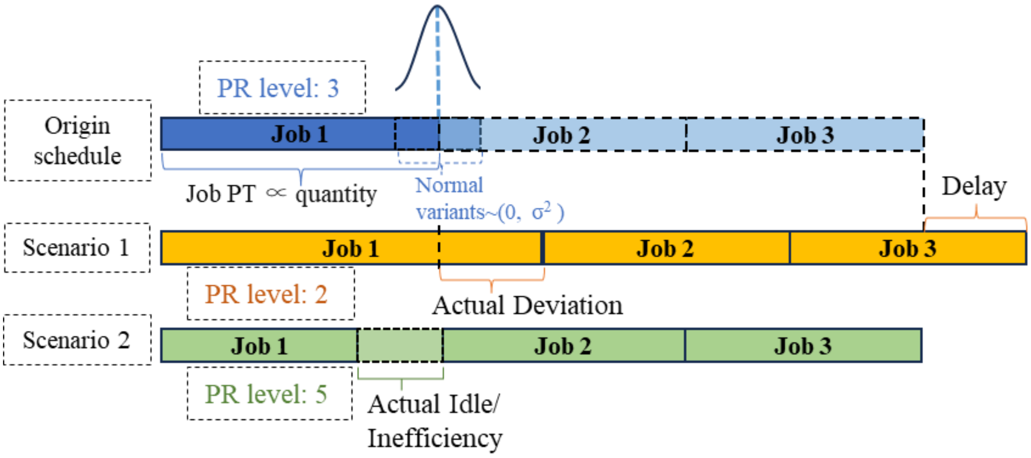


Figure 1-1. Demonstration of different circumstances in scheduling

Context-Based Production Rate (CBPR)-guided scheduling approach

In Section 1.1.2, two indicators are proposed, namely the processing time and PR level, which can be used to measure the processing efficiency of a job on a specified execution machine (which is charged by one operator) and under a combination of resources and environmental conditions. As illustrated, a better understanding of factors affecting PR can largely benefit prediction accuracy and schedule reliability.

Following the previous introduction, the portfolio of factors affecting the actual manufacturing process is defined as the processing context of a job (also the specific job execution scenario). The context involves critical job-specified features (e.g., job characteristics, machine setup, materials), operator in charge, and position in the execution sequence. A new index of context-based processing rate (CBPR) is proposed, which *represents the processing rate of a job (i.e., JPR) under a particular context generated along with the scheduling process*. The purposes of this index are to (i) provide a framework for interpreting factors influencing the JPR and (ii) enable a context-based scheduling method, which may capture JPR changes so that the scheduling algorithm can be sensitive to the varying context and adaptively produce efficient solutions. The CBPR generated along with the scheduling process can be used to guide the schedule generation process to derive schedules that can more flexibly accommodate the production requirements. For example, it can apply the CBPR to guide the positioning of a job to beneficial performing places with suitable operator and resource combinations so that the generated schedules can obtain enhanced execution efficiency, proper job allocation and execution arrangement, improved resource utilization efficiency and operator physical demanding, and reductions in excessive buffer allocation or potential delays.

1.2 Research Objectives

This dissertation aims to reach the following main objectives. From the problem aspect, it is devoted to (i) enhancing the collaboration between production machines and smart delivery tools for a robotic job shop floor; and (ii) incorporating the influences of multiple real-world factors to achieve effective predictive analytics of production status and efficient data-driven scheduling. From the method aspect, this dissertation focuses on addressing: (i) modelling interactions between production machines and autonomous delivery tools; (ii) extracting the influencing pattern of multiple factors as well as the dependencies of job processing rate on related factors; and (iii) developing efficient solution algorithms to tackle the formulated scheduling model. More specifically, the research questions are summarized in the following:

1. How can the operations of machines and the autonomous delivery tool (the mobile robot) be scheduled and effectively coordinated to achieve sustainable and efficient solutions that reduce carbon emissions while maintaining productivity? (Related to Chapter 3)
2. How can the various realistic operational factors (e.g., job characteristics, resource

utilization, production settings, environment, and preceding sequence) affect the actual production performance? With the availability of data collected from IoT devices, how can such influences be captured? (Related to Chapters 4 and 5)

3. How can the job processing status be captured with predictive analytics? What performance indicators should be tracked and how to better predict them by developing advanced AI techniques? (Related to Chapter 4)
4. How can the effects of multiple influencing factors on the processing rate be incorporated into the scheduling process to derive practical and efficient scheduling solutions? (Related to Chapter 5)

1.3 Research Methodology

The research methodologies employed in this dissertation focus on the development of optimization models & algorithms and deep learning methods to enhance the efficiency and sustainability of smart manufacturing systems. This dissertation first conducts a systematic review of the research development of production scheduling problems in the era of smart manufacturing. Then, based on the identified research gaps, three research studies were undertaken, employing methodologies covering the realms of mathematical modelling, real-world data processing, deep learning, and optimization algorithms.

The first study in Chapter 3 focuses on the development of novel mixed integer linear programming models that incorporate energy considerations. To verify the performance of the proposed model under various scenarios, extensive computational experiments are conducted based on hypothetical data to evaluate the computational efficiency and derive managerial insights.

Then, studies in Chapters 4 and 5 are based on a real-world production scenario. Comprehensive real-world processing data are collected from the company's production system. First, the raw data undergoes cleaning, processing, and encoding phases to prepare for analysis. Then, deep learning models are designed, which are tailored to capture the complex interrelationships existing in the data. Furthermore, the deep learning models are integrated with an optimization algorithm to create a novel solution architecture. The evaluation of these models is multi-faceted. Specifically, the performance of the proposed deep learning models is benchmarked against other leading models to establish their efficacy. A series of analyses involving ablation studies, benchmark comparisons, and sensitivity analysis, are conducted to deepen the understanding of the model performance and influencers. Additionally, computational experiments established with real-world data are conducted to validate the performance of the proposed integrated solution architecture and examine its applicability and effectiveness in practical settings.

1.4 Research Significance and Contributions

The study in Chapter 3. This study is one of the first attempts that focus on the energy consumption issue of robotic cells, which is becoming increasingly crucial in achieving sustainability within autonomous robot-enabled smart manufacturing systems. The detailed contributions are as follows:

1. It is the first study that integrates the energy consumption from both the machine side and robot side into the RJSP framework, which theoretically contributes to the JSP literature by proposing a new research direction.
2. The study is also the first to propose alleviating energy concerns of robotic cells by promoting better machinery-robot movement collaboration. To achieve so, two novel energy-efficient robotic job-shop scheduling models are proposed. A V-scale speed framework is applied for both machines and the mobile robot so that the

optimal speed for performing each operation/movement can be optimally selected to maximumly avoid machine idling and robot partial blocking circumstances. To be more specific, a robotic job-shop scheduling with energy consumption (i.e., RJSP-E) is first developed to minimize the total energy consumption. Then, a robotic job-shop scheduling with energy consumption and makespan limitation (i.e., RJSP-EM) is developed to simultaneously optimize energy consumption and system productivity.

3. Through computational experiments, the RJSP-E is shown to remarkably reduce energy consumption (with an average of 15%) by selecting slower operating speeds, but at a cost of productivity. In comparison, the RJSP-EM demonstrates superior ability in selecting the most proper operating speeds based on the evaluation of the production system. Notably, the RJSP-EM is shown to reduce energy consumption by a mean of 10% compared with the traditional model even without sacrifice in productivity (i.e., when the makespan is not allowed to increase). Other managerial implications for enhancing the green level of smart manufacturing are also derived.

The study in Chapter 4. Realizing the importance of production elements on actual production performance, this study is the first one that proposes a novel deep learning architecture to capture the dependencies of processing rate on utilized production resources, environmental factors, and the preceding jobs. The detailed contributions are as follows:

1. The study is the first to identify the influencing production elements on production performance. Through exploring the high-dimensional real-world production data, such influences are extracted in three levels: the direct influence (job-related factors, e.g., material utilization, machine, operators, environments), the adjacent influence

(the immediate predecessor impact), and the sequential influences (the impact of a series of predecessors).

2. This is the first study to propose utilizing the synergy between two production performance indicators (i.e., regression of processing time and classification of processing rate level), for a better measure of production performance, reflecting the performance of a single job and system status.
3. To capture hierarchical influences and shared information between training the two tasks, a multi-input-module supported dual-task learning (MMDT) model is proposed, which can adaptively learn the inner patterns of sequential and multivariate job information. The model with three input modules can largely mine and exploit the information within the time series sequence to predict job processing status. A joint loss function with a controllable weight parameter is applied to train the multi-output neural network simultaneously. Such a co-learning mechanism reduces overfitting and also utilizes the synergy between the two tasks to learn the right representation for each task.
4. Extensive experiments are conducted to validate the model performance. Compared with the single input module, the single output layer, and other state-of-the-art benchmarks, the proposed architecture shows significant effectiveness in improving the performance for both processing time prediction and PR level classification.

The study in Chapter 5. Realizing that realistic operational factors in the production process will affect production performance and a lack of studies considering incorporating such influences in scheduling, this study proposes to investigate a new production scheduling problem with varying processing rates. The novel property of varying processing rates (determined by different production scenarios) results in the studied problem being non-deterministic. To tackle the induced problem complexity, we propose an efficient solution architecture with tailored dominance rules. The detailed contributions are as follows:

1. The study first proposes an index named Context-Based Processing Rate (CBPR) to indicate the processing rate of a job (i.e., JPR) under a particular context generated along with the scheduling process, which is sensitive to the varying context. It enables the incorporation of multiple realistic and operational factors for an accurate JPR prediction to empower the scheduling process. Also, the scheduling algorithm can adaptively produce efficient solutions.
2. A DeepPR model is developed to capture the dependencies on CBPR with two attention mechanisms. A DeepPR integrated CBPR-guided branch-and-price (BnP) scheduling approach is further presented, which applies DeepPR to capture JPR changes under varying contexts and further uses the predictive CBPR to guide the optimization of assignment and sequencing of jobs to operators. The proposed CBPR-labelling algorithm with tailored dominance operations enables a CBPR-guided scheduling process, which enables the scheduling scheme to focus more on exploring promising positions for individual jobs.
3. Computational experiments show that the proposed DeepPR outperforms other state-of-the-art benchmarks in JPR prediction accuracy. Also, applying the CBPR-guided scheduling approach to the investigated printing company enables a substantial enhancement of their production efficiency by reducing their processing time by an average of 12.84%. Additional experiments show that this method can effectively improve the overall processing rate.

1.5 Dissertation Organization

The outline of this dissertation is as follows. The introduction of background, motivations, research questions, methodologies, and contributions are detailed in Chapter 1. Chapter 2 provides a comprehensive literature review of the research problems. Then, Chapters 3, 4, and 5 present the details of three studies. Finally, Chapter 6 concludes this dissertation.

Figure 1-2 provides the overall structure of this dissertation.

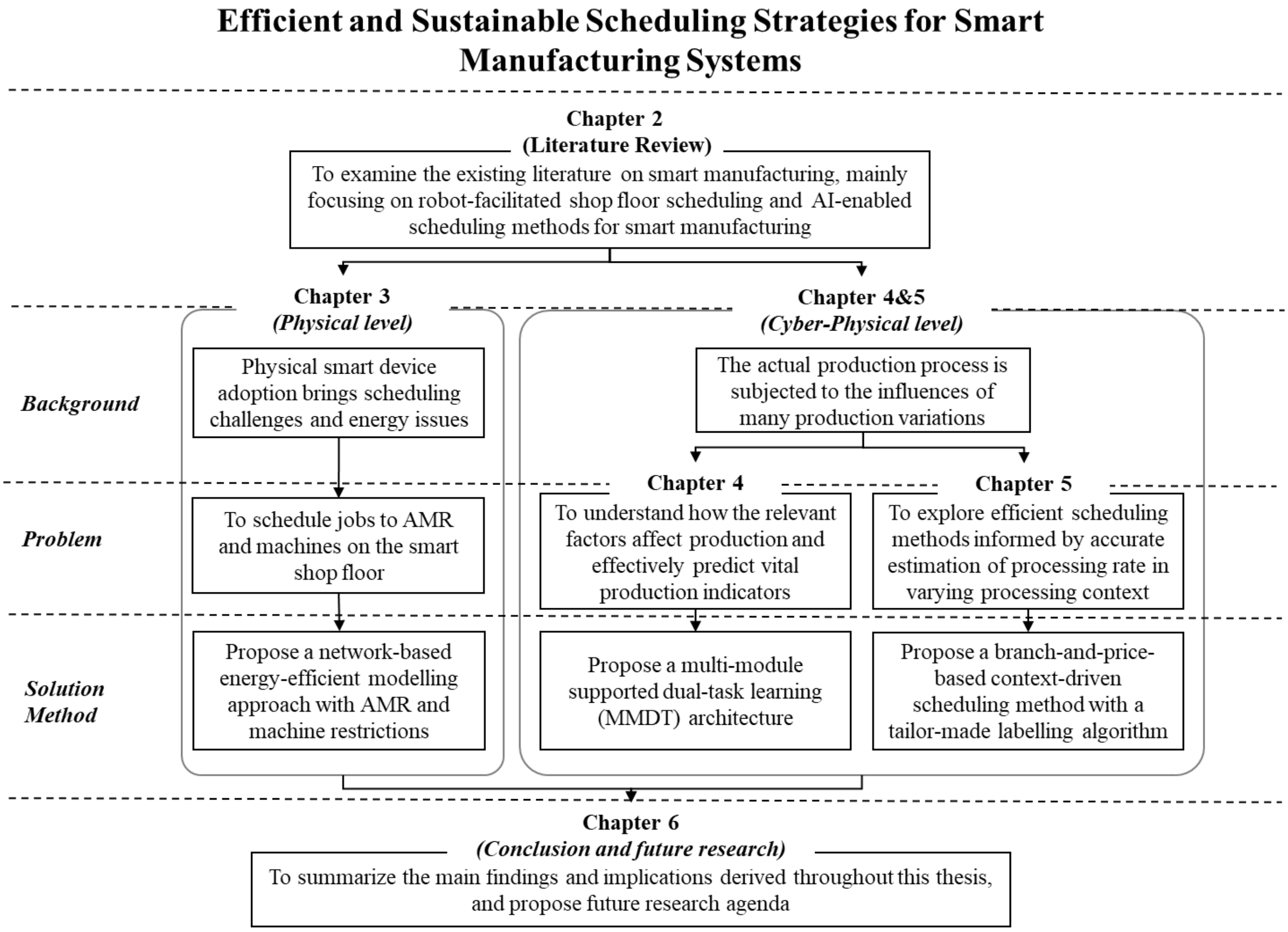

Efficient and Sustainable Scheduling Strategies for Smart Manufacturing Systems
Chapter 2
(Literature Review)
To examine the existing literature on smart manufacturing, mainly focusing on robot-facilitated shop floor scheduling and AI-enabled scheduling methods for smart manufacturing
Chapter 3
(Physical level)
Chapter 4&5
(Cyber-Physical level)
Background
Physical smart device adoption brings scheduling challenges and energy issues
The actual production process is subjected to the influences of many production variations
Chapter 4
Chapter 5
Problem
To schedule jobs to AMR and machines on the smart shop floor
To understand how the relevant factors affect production and effectively predict vital production indicators
To explore efficient scheduling methods informed by accurate estimation of processing rate in varying processing context
Solution Method
Propose a network-based energy-efficient modelling approach with AMR and machine restrictions
Propose a multi-module supported dual-task learning (MMDT) architecture
Propose a branch-and-price-based context-driven scheduling method with a tailor-made labelling algorithm
Chapter 6
(Conclusion and future research)
To summarize the main findings and implications derived throughout this thesis, and propose future research agenda

Figure 1-2. The outline of the overall dissertation

Chapter 2. Literature Review

In this chapter, the literature on production scheduling in the smart manufacturing context is detailly reviewed. Section 2.1 provides a fundamental review for this dissertation by overviewing the research on production scheduling, the impact of Industry 4.0 on production scheduling, and existing solution approaches. Then, more detailed reviews are conducted for separate research studies. Section 2.2 (related to Chapter 3) reviews the robot-facilitated production scheduling and the energy consumption issue of robotic cells. Section 2.3 (related to Chapter 4) examines the adoption of machine learning methods in smart manufacturing and production scheduling. Section 2.4 (related to Chapter 5) reviews the combination of machine learning methods and operations research techniques in solving production scheduling problems. Finally, Section 2.5 derives the research gaps through the above review, which are addressed in this study.

2.1 Evolution of Production Scheduling

2.1.1 Considerations in production scheduling problems

The main purpose of production scheduling is to transform the received order requirements into a series of ordered tasks to be performed on machines (Buxey, 1989). Due to the complicated nature of production processes, a good production schedule requires the coordination of multiple resources at various levels (Lohmer & Lasch, 2021). Basically, production scheduling is related to decisions from multiple levels from more long-term and high-level decisions (e.g., facility configuration) to short-term (such as the daily operations) which are summarized in Table 2-1.

Table 2-1. A summary of research focuses on production scheduling

Scheduling Decision Level	Research Focus
Strategic level	Layout design and capacity configuration (Guo et al., 2023; Karmarkar & Kekre, 1987; Wu et al., 2020)
Tactical level	Worker employment and purchasing plan (Cornwell et al., 2021; Ecer, 2022)
	Production mode (e.g., parallel machine processing, job-shop, flowshop, open shop, etc.) (Jamrus et al., 2017; Rossit et al., 2018; Workneh & Gmira, 2022; Wu & Che, 2019)
Operational level	Job priority, job allocation, job sequencing, and timely schedule adjustment (Özgüven et al., 2010; Raheja & Subramaniam, 2002; Rossit et al., 2018; B. Wang et al., 2022; Yanıkoğlu & Yavuz, 2022)

The strategic decision aspect covers layout design (i.e., plan for placing facilities or departments) and capacity configuration (e.g., machine type and mode selection) (Guo et al., 2023; Karmarkar & Kekre, 1987; Wu et al., 2020). Due to the significant influences on the interactions between facilities on the shop floor, these decisions fundamentally affect the operational process (Lohmer & Lasch, 2021). Then, the tactical decision level establishes middle-term decisions, such as the employment of workers and creating resource purchasing requirements from suppliers (Cornwell et al., 2021; Ecer, 2022). Besides, the mode of the production process can be determined according to system configurations. To be more specific, the production mode can be divided into single-stage or multiple-stage processing (Graves, 1981). The single-stage processing mode includes single machine scheduling (which is to determine the sequence of non-preemptive jobs on one processor) (Koulamas & Kyparisis, 2023) and parallel machine processing scheduling (which is to allocate a set of jobs on several parallel unrelated machines) (Wu & Che, 2019). Then, multiple-stage processing involves more complex processing patterns and thus requires good coordination between machines to jointly complete the processing steps. The representative processing modes of this mode involve flowshop and job-shop scheduling settings. The flowshop mode is designed for producing the same type of product on a series of machines, which can facilitate mass production with enhanced production efficiency (Rossit et al., 2018). However, due to the limitations of flowshops in fulfilling customized preferences and the demand for variability, job shop scheduling is adopted in many areas (e.g., circuit board printing and semiconductor) to achieve more production flexibility (Jamrus et al., 2017; Workneh & Gmira, 2022). The job-shop scheduling mode enables producing products to follow different processing steps and allows jobs to re-enter a processing centre/machine more than once. To further increase flexibility, flexible job-shop scheduling and open shop scheduling problems are also popular research focuses (Rahmani Hosseinabadi et al., 2019; Shen & Yao, 2015).

Based on the above production layout and configuration, the operational-level scheduling decisions involve establishing a specific daily processing timetable for machines and workforce (Xu & Hall, 2021). It mainly determines job priority, job allocation to machines, and sequencing on machines (Özgüven et al., 2010; Rossit et al., 2018; Wang et al., 2022a; Yanıkoğlu & Yavuz, 2022). Moreover, necessary monitoring systems may track the production performance and make quick adjustments to production plans to control the process (Raheja & Subramaniam, 2002).

2.1.2 Impact of Industry 4.0 on production scheduling

Industry 4.0 brings manufacturing into a new era with the involvement of many disruptive technologies, e.g., the cyber-physical system, AI-facilitated autonomous devices, cloud computing, and digital twin (Rossit et al., 2019b). Research on the related field is seen from a large spectrum. Zheng et al. (2019a) review the smart manufacturing systems for smart manufacturing and demonstrate their changes in design, machining, control, monitoring, and scheduling. Parente et al. (2020) summarize the impact of Industry 4.0 on modern production scheduling from the following aspects: cyber-physical systems (CPS), the Internet of Things (IoT), big data and cloud computing, integrated production systems and adaptive manufacturing. In Table 2-2, we summarize the main technical advancements and their applications in developing smart manufacturing systems.

Table 2-2. A summary of main technical advancements in smart manufacturing

Techniques	Main Focus	Applications
Cloud computing and data analytics	Management and storage of big data for production systems (Sharp et al., 2018; Xu, 2012)	Integrated cloud-based and data-driven solutions (Jiang et al., 2022; Xu, 2012)
Cyber-physical systems (CPS) and Digital twin	Reconstructs the devices and operations in the physical real world to its virtual digital representation (Fang et al., 2019; Zhang et al., 2019a).	Manufacturing process control, product development, production planning, machine failure detection, performance assessment (Serrano-Ruiz et al., 2021; Workneh & Gmira, 2022)
Industrial Internet of Things (IIoT)	Building interconnected production systems, with sensors, machines, materials, operators, robots, etc. (Yang et al., 2019).	Intelligent connection, real-time data processing, monitoring and autonomous control (Chen et al., 2023)
Autonomous robots	Automate the material handling process (Fragapane et al., 2021)	Robot-facilitated material retrieval/picking, material handling, and material storage (Fragapane et al., 2021; Keung et al., 2021)

As seen from Table 2-2, with the foundation of cyber-physical integration, data processing, and data analytics techniques, production scheduling and control can be conducted in a data-driven approach with enhanced coordination between participants/ components and better alignment between resources (Bueno et al., 2020; Jiang et al., 2022). Moreover, the adoption of autonomous robots in automating the material handling process is also emphasized by the literature in recent years. As the scheduling of the robot-facilitated material handling system is one of our main focuses, a more detailed review is available in Section 2.2.

2.1.3 Solution approaches for production scheduling problems

Modelling and solutions for deterministic scheduling problems

Operations research techniques have been widely investigated for solving production scheduling problems. In the well-established literature, a specific production scheduling problem can be formulated as mathematical optimization models (e.g., linear programming and mixed integer programming models) with objectives of minimizing the completion time /makespan, maximizing the throughput, or minimizing the delays in fulfilling due dates (Mokhtari & Hasani, 2017). To formulate the scheduling problems, different scheduling methods are proposed, e.g., the position-based modelling approach (Demir & İşleyen, 2013; Meng et al., 2020; Roshanaei et al., 2013), sequence-based modelling approach (Karimi et al., 2017; Özgüven et al., 2010), and time-interval-based modelling approach (Yan et al., 2018).

However, most production scheduling formulations (e.g., job shop scheduling problems) are very challenging combinatorial optimization problems and finding optimal solutions for large-scale problems in reasonable time limits is very challenging (Çaliş & Bulkan, 2015). Exact algorithms, such as branch-and-price, branch-and-bound, and benders decomposition, are popular research concentrations to increase the tractability of the formulated models (Avci et al., 2022; Gmys et al., 2020; Juvin et al., 2023; Naderi & Roshanaei, 2022; Quinton et al., 2020; Yanıkoğlu & Yavuz, 2022). Other scholars devote to developing numerous heuristic decision rules and metaheuristic algorithms (e.g., genetic algorithms, large neighbourhood search, simulated annealing, particle swarm, tabu-search) to obtain efficient scheduling approaches with relatively good results (Fan et al., 2021; Fontes et al., 2023; Gao et al., 2019; Tamssaouet & Dauzère-Pérès, 2023).

Uncertainty-aware production scheduling strategies

Many studies investigate establishing resilient schedules by accommodating various uncertain factors, e.g., machine failures, malfunctions, breakdowns, or other abnormalities. From the literature, the most direct choice is to take reactive or corrective actions, such as rescheduling or dynamic scheduling (Ghaleb et al., 2020; Petrovic & Duenas, 2006). However, these event-driven interventions may unavoidably disturb normal operations and cause unnecessary costs. Another stream of studies investigates robust programming methods. For example, Enginarlar et al. (2002) propose to incorporate buffer or robustness into the schedule generation process so that the schedules can be more resilient to disruptions. Yue et al. (2020) propose a robust optimization model to tackle the uncertainty of job due date by minimizing the maximum tardiness in the worst-case scenario. Wu et al. (2021a) consider the uncertainty of scenario-dependent job processing time for an assembly flowshop scheduling problem and propose a solution method that minimizes the maximum makespan for all scenarios.

Even though robust solutions may significantly enhance schedule resilience, adding buffers to schedules or considering the worst scenario can be very expensive. Therefore, other studies concentrating on production planning with uncertainty investigate stochastic or fuzzy optimization methods (He et al., 2021; Tirkolaee et al.,

2020) by considering non-deterministic processing parameters (e.g., processing time, idle time, anomaly, etc.) following some deterministic distributions based on empirical experience, e.g., processing time follows normal or exponential distribution (Birge et al., 1990; Mittenthal & Raghavachari, 1993) or take its value range (discrete or continuous) into account to enhance scheduling adaptivity or flexibility (Ramírez-Velarde et al., 2017; Sotskov, 2020). However, due to the complicated interactions among factors and intricate influences in the real-world production process, these assumptions may not hold, and it is difficult for the above approaches to model the uncertainty with the consideration of the influences of multiple real-world factors.

Scheduling approaches in Industry 4.0

In the context of Industry 4.0, the availability of data provides the potential to design scheduling solutions based on more information. Relevant studies are examined from three perspectives. First, several studies focus on building new scheduling architectures/frameworks for smart manufacturing systems supported by many advanced techniques (Qiao et al., 2021; Rossit et al., 2019a). The second category of studies focuses on online decision scheduling, which makes scheduling decisions in a real-time manner based on human-made decision rules or scheduling policies by reinforcement learning approaches (Gu et al., 2022; Park et al., 2019; Serrano-Ruiz et al., 2024; Wu et al., 2021b; Zhou et al., 2021). Last but not least, realizing the limitations of traditional scheduling approaches in making many unrealistic assumptions in many production-related factors, the importance of integrating data analytics and AI methods into production scheduling is increasingly emphasized (Workneh & Gmira, 2022). As the integration of AI and scheduling optimization methods is one of the main focuses of this dissertation, it will be more detailed in Sections 2.3 and 2.4.

To summarize, the IIoT systems enable the adoption of autonomous robots and recording of the production process, which enables access to real-time processing data. Production scheduling in this novel context can be devoted to coordinating the intelligent production network and utilizing recorded processing data for better scheduling or adjustment of production settings, plans, maintenance, and employees in a real-time or short-term manner.

2.2 Production Scheduling in Robotic Cells

2.2.1 Application of autonomous techniques in intralogistics

In recent years, the planning and control of AMRs in intralogistics (e.g., the manufacturing systems and warehousing environments) is a popular topic (Fragapane et al., 2021). In the domain of robotic fulfilment system scheduling, most studies focus on joint decisions of allocating/sequencing orders to workstations and delivery robots. Lee and Murray (2019) transform robotic order picking into a vehicle routing problem and formulate it as a mixed integer linear programming model. For the order assignment problem, Wang et al. (2022a) evaluate the order assignment performance of a robotic cell with multiple picking stations under a zoning policy. Cai et al. (2021) combine the goods location assignment, rack storage, and AGV path planning into one optimization problem to enhance collaborative optimization. To jointly optimise order sequencing and rack scheduling, Yang et al. (2021), Teck and Dewil (2022), and Shi et al. (2021) study the simultaneous assignment of orders and racks to multiple picking stations. Yang (2022) examines the joint impact of the item storage assignment policies and order batching policies on the order-picking process. Keung et al. (2021) study the IIoT-enabled storage location assignment in a resource synchronization and sharing-based RMFS.

Besides the above studies on robot-workstation or robot-order assignment, a few studies focus on coordinating several robots. Yuan et al. (2021) propose to tackle the multirobot task allocation (MRTA) in robotic mobile fulfillment systems. Their model proposed model considers both the picking time balance of picking stations and the load balance of robots. A four-stage balanced heuristic auction algorithm is designed to solve the task allocation model and the tasks with an execution sequence for each robot. Zhuang et al. (2021) focus on the cooperative task planning of heterogeneous multi-robots. They formulate the problem as open shop scheduling with sequence-dependent set-up and transportation times and developed a MILP model and a hybrid artificial bee colony algorithm. Qin et al. (2022) design dispatching algorithms to make real-time dispatching decisions among robots, racks, and workstations. Wang et al. (2022b) propose a stochastic dynamic program that scheduled robots and mobile racks with the consideration of the working state fluctuation of human pickers.

2.2.2 Robotic flowshop and job-shop scheduling problems

Robotic cells refer to production environments where autonomous mobile robot(s) (AMR) are deployed for material handling on an assembly line. The production modes are generally arranged as a flowshop or a job-shop environment and AMRs may perform the job transshipment among machines. The so-formed robotic flowshop cells is examined by Dawande et al. (2005). Similar to basic scheduling problems discussed in Section 2.1, the optimization objectives of scheduling in robotic cells are also to minimize the makespan, idle time, or tardiness and with considerations of job-specified processing sequence, and sequence on machines (Brucker et al., 2012; Dawande et al., 2005; Petrović et al., 2019). However, due to the integration of AMR routing in the scheduling framework, scheduling both machines and robots in robotic settings is rather complicated (Brucker et al., 2012). Besides, compared with robotic fulfillment services in order-picking systems, robot activities are largely different. Robots in robotic cells are required to visit one job several times (move to different workstations for moving goods, while in order-picking services, one task will be completed once the robot moves the goods from the storage area to the corresponding workstation. The additional restrictions involved are mainly indicated by the following aspects.

Buffer capacity. Machine buffer decides whether semi-products can stay on the machine after finished or should be removed before conducting the next operation (Liu et al., 2018), while robot buffer determines how many products an AMR can carry (Drobouchevitch et al., 2010).

Pickup criteria. Several pickup criteria can be adopted according to specific processing requirements, such as a blocking criterion (a semi-product stays on the current machine before the availability of the next machine), a no-wait criterion (a semi-product should be removed immediately after finishing), and a time window one (a semi-product can be picked up within a legal time window (Caumond et al., 2009; Cheng et al., 2019; Hurink & Knust, 2002; Zeng et al., 2014).

***Deadlock avoidance*.** Another important aspect is the guarantee of conflict-free robot delivery to ensure smooth completion. A deadlock can happen when a robot tries to place a semi-product on a machine (with a single buffer) that is already occupied, or oppositely forcing a robot to carry another job when it is with a single buffer and is already occupied (Caumond et al., 2009; Ham, 2021; Yan et al., 2018).

2.2.3 Energy consideration for robotic cells

As green production and sustainability are more emphasized in modern production systems, increasing research attention has been paid to incorporating green strategies and resolutions in production scheduling (Abedi et al., 2020; Hassani et al., 2019; Mokhtari & Hasani, 2017). Through examining the literature, several energy-reduction strategies are identified.

***Existing energy reduction strategies*.** The basic idea is to develop an energy-concerned multi-objective optimization model for scheduling in robotic cells (Dai et al., 2019). Besides, several research studies focus on increasing the scheduling of production activities to off-peak periods to reduce electricity costs (Masmoudi et al., 2019; Wang & Wang, 2019) or exploring a turning off/on strategy, which turns off the machine when a long period of idling appears (Meng et al., 2019). However, this on/off strategy may cause a negative impact on the lifetime of devices or produce extra energy due to frequent on/off operations (Zhang & Chiong, 2016). To alleviate this concern, many scholars explore the speed scaling strategy, leveraging the fact that many industrial machines are equipped with the capability to adjust their operating speeds. On this issue, Zhang and Chiong (2016), Wu and Che (2019), and Abedi et al. (2020) all study the scheduling for machines with a speed scaling mechanism.

Energy reduction from robot side. Even though many studies aim to tackle energy consumption from the machine side, there are very few studies on the energy consumption of the intra-transportation from the robot side. Only several studies are identified. Gürel et al. (2019) find that controlling robot speed is an effective method to reduce the energy consumption of a robotic cell. Their method can determine the robot performing sequence and robot speed. Similarly, Bukata et al. (2019) propose a method to reduce robotic cell energy consumption and meanwhile maintain system throughput by applying robot power-reduction modes and robot position adjustment. Barak et al. (2021) try to incorporate the energy efficiency of AGVs into flexible manufacturing systems by proposing an energy-efficient model that optimizes the allocation of operations to AGVs.

2.3 Machine Learning (ML)-Facilitated Smart Manufacturing

Besides the adoption of autonomous devices at the physical level, another important aspect of smart manufacturing is to utilize advanced data analytics to improve decision-making in production systems, which is becoming increasingly popular (Wang et al., 2018a). The following two subsections examine the applications of ML techniques for industrial applications and the adoption of ML techniques for production scheduling.

2.3.1 ML techniques for industrial applications

ML techniques are increasingly emphasized to support the decision support of the manufacturing system, such as cost estimation, tool utilization, and batch size plan (Sharp et al., 2018). On these issues, classical machine learning methods, such as support vector machines are seen to be adopted in characterizing the cost space and constructing more accurate and generalizable cost estimation functions (Deng & Yeh, 2011; Yeh & Deng, 2012). Nevertheless, the classical machine learning methods do work in performing classification and prediction tasks in many areas, these methods may require human expertise or prior knowledge for feature extraction or feature dimension reduction (Wang et al., 2018a) or cannot learn complicated non-linear relationships within data due to the shallow network (Jia et al., 2016).

In comparison, deep learning (DL) models enable more powerful learning of representations from imbalanced data and data with noises (Khan et al., 2017). The deep architecture with hidden layers performs multi-layer nonlinear operations, thus obtaining advantages in feature representation, relationship approximation, and training (Wang et al., 2018a). Convolution neural networks (CNN) have good capability to extract information from structural data, e.g., two-dimensional input of figures (Zhang et al., 2019b). To deal with the time-series-based input signal, recurrent neural networks

(RNN) and various variants (e.g., LSTM, GRU) are advantageous, which learn historical patterns from the input of a sequence of time steps and predict future steps (Hochreiter & Schmidhuber, 1997). However, due to the gradient vanishing effect, RNN models are limited in their ability to process long-sequential data (Vaswani et al., 2017). In 2017, the transformer model was developed and largely changed the landscape of dealing with multi-variant time series data, which can capture useful relationships from any two input steps through a self-attention mechanism (Grigsby et al., 2021; Vaswani et al., 2017).

2.3.2 ML applications in production scheduling

This section focuses on examining the adoption of ML methods in the production scheduling area. Wang et al. (2018a) summarize the main application of DL to industrial use into three aspects based on different analytic methods, namely, descriptive analytics (to identify useful relationships between production performance and environmental or operational settings); diagnostic analytics (to detect or examine the occurrence of disrupted events, e.g., machine breakdown); and predictive analytics (to learn from historical data and make predictions for future performance of components such as machines and robots so that preventive actions can be taken in advance of a failure). We identify the main application domains of ML to production scheduling as follows.

Forecasting of demand and order arrival. Demand uncertainty is a major source of disruptions to the implementation of schedules, which frequently produces the need for rescheduling (Tang & Grubbström, 2002). Thus, demand forecasting is very necessary in managing production capacity. Time-series forecasting of demand changes provides valuable references for establishing production schedules (Matsumoto & Komatsu, 2015). Ghaleb et al. (2020) investigate how real-time updates on unexpected job arrivals and other factors can be used for rescheduling in a flexible job-shop setting.

Fault diagnosis and anomaly detection. Timely machinery health diagnosis and anomaly detection are important to guarantee smooth production process (Zhao et al., 2017a). Many studies thus focus on using ML and DL methods to tackle these issues. Li et al. (2015) propose a multimodal deep support vector classification model to learn the deep representation from wide modalities and improve fault diagnosis ability. Jia et al. (2016), Lei et al. (2016), Guo et al. (2016), and Lu et al. (2017) develop deep learning methods for fault diagnosis in rotating machinery. To tackle the unplanned downtime, Lee et al. (2019) propose to use SVM and ANNs based on extracted meaningful features with domain knowledge for monitoring conditions of the cutting tool and the spindle motor. Another stream of studies manages machine health monitoring and detects anomalies with time-series-based data, with variants of recurrent neural networks to capture higher-level temporal features (Malhotra et al., 2015; Zhao et al., 2017a; Zhao et al., 2017b).

Forecasting maintenance requirements. Due to the severe outcomes of machine breakdown/unavailability, maintenance is a key component for machine scheduling. O'Donovan et al. (2015) classify maintenance into four categories: reactive (to act when a failure appears), corrective (to spot abnormality during processing and act before failing), preventive (to regularly adjust and avoid failing), and predictive (to forecast failure and take actions). Reactive maintenance would lead to unplanned downtime and cascading failures (Sharp et al., 2018), while preventive maintenance may induce extra costs. Therefore, timely identifying the maintenance requirements and incorporating them into predictive maintenance scheduling is important. On this issue, Zonta et al. (2022) propose a predictive maintenance model, which enables continuously adjusting the maintenance schedule based on predictions of machine operating conditions. Other studies can refer to Bencheikh et al., 2022 and Cardin et al., 2017.

Prediction for performance indicators. Several studies are identified to predict performance indicators for machine conditions and manufacturing processes. Wang et al. (2017) develop a deep belief network-based data-driven approach to uncover the relationship between material removal rate and operation parameters, such as pressure and rotational speed for a chemical mechanical polishing process. Considering the time series feature implied by machine conditions, Essien and Giannetti (2020) propose a deep convolutional LSTM end-to-end architecture to predict machine speed for production throughput optimization. Wang et al. (2018b) use a neural network to track the abnormal pattern of machine energy consumption. Lee et al. (2019) propose to use the flank wear and the bearing's Remaining Useful Life (RUL) as classification metrics to represent the machine tools' conditions. Chui et al. (2021) also predict RUL to avoid downtime or unnecessary checks. Other studies on RUL prediction can refer to Deutsch et al. (2017), Rathore and Harsha (2022), and Wu et al. (2018).

2.4 Synergy between AI and OR for Enhanced Scheduling

Realizing the importance of empowering production decision making with AI methods, several papers identified investigate integrating AI with optimization methods to tackle various challenges in production scheduling. Despite the great potential and importance, research in this area is still insufficient, leaving large research space for further exploration. The existing related studies mainly involve two aspects: (i) using AI methods to enhance scheduling decision making (prediction-based scheduling) and (ii) combining AI and OR algorithms for enhanced solution efficiency.

2.4.1 Prediction-based scheduling

Prediction-based scheduling focuses on leveraging AI techniques to optimize the decision-making processes in production so that the scheduling can be enhanced with the availability of data, e.g., historical demand data and operational information from the real production process (Zonta et al., 2022). Del Gallo et al. (2023) examine the application of AI in solving production scheduling problems. They highlight the adoption of particle swarm optimization, neural networks, and reinforcement learning.

Prediction for rescheduling. Wang et al. (2018b) design a big data-enabled intelligent immune system, which combines a neural network into a re-scheduling algorithm. It works once an anomaly is detected by solving a multi-objective optimization model. Li et al. (2020b) develop a rescheduling framework for a flexible job shop scheduling problem, where several ML methods are used to periodically learn from historical data and return predictive results of whether rescheduling is needed or not. Ghaleb et al. (2020) investigate the benefits of using real-time updates on the unexpected arrival of jobs and consider random machine breakdowns that follow exponential distribution to establish rescheduling decisions in a flexible job shop setting.

Prediction for production resource allocation. Kim et al. (2020) propose a deep neural network-based dynamic scheduling method that predicts the next target machine with the considerations of automated material handling constraints. Jacso et al. (2023) also present an ANN-based model, aiming to optimize feed rates in trochoidal milling to derive schedules with better tool load control. Rohaninejad et al. (2023) propose a machine learning-enabled data-driven predictive scheduling method for a capacitated lot-sizing and scheduling problem with a job shop setting. In a rolling horizon setting, they predict (before the scheduling for the next period) the values of two types of reserves in the schedule based on the newly revealed values (e.g., customer demand) and generate the future schedule based on the existing information of the system and the predicted reserves. Morariu et al. (2020) present a hybrid solution for large manufacturing systems using Big Data techniques and a Long Short-term Memory model for predicting instant power consumption of resources, which is then fed into a cloud-based scheduler that performs resource allocation and operations scheduling.

Predictive maintenance scheduling. Azab et al. (2021) propose a machine-learning-based simulation approach, which uses various ML methods to estimate predictive maintenance slots and incorporates the predictive maintenance into dynamic flow-shop scheduling to improve manufacturing efficiency. Ye et al. (2020) take machine speed, age, setups, and status into scheduling to improve production efficiency and maintenance decisions. Zonta et al. (2022) propose a predictive maintenance model to optimize the production schedule, which enables the adjustment of maintenance and scheduling based on machine conditions.

Incorporation of predicted human factors into scheduling. Several studies focus on enhancing scheduling decisions by factoring in human elements, which try to incorporate human fatigue, physical demands, and diverse individual characteristics, into the scheduling process to identify optimal job assignments and sequences for each operator, thereby maximizing their efficiency (Du et al., 2021; Wang et al., 2022b).

2.4.2 Combination of AI and optimization algorithms

Realizing the great potential of combining AI with classical optimization algorithms, in recent years, increasing studies have explored this direction to suitably combine AI with operations research methods, aiming to leverage both the computation advantage of AI methods and the reliability of classical combinatorial optimization algorithms. Such combinations on one hand enhance the robustness of AI results, and on the other provide more efficient solutions for MIP problems.

Enhancing AI performance with optimization methods. In terms of using OR methods to improve AI performance, a few examples are identified in the following.

Chui et al. (2021) use a non-dominated sorting genetic algorithm II to select the weights to combine RNN and LSTM in their model. Rathore and Harsha (2022) employ domain knowledge and particle swarm optimization for feature selection in performing the task of RUL prediction. Djenouri et al. (2023) present a hybrid method to predict traffic flow, which uses a graph convolutional neural network and RNN to capture spatial and temporal dependencies and a branch and bound algorithm to tune the hyperparameters.

Empowering optimization algorithms with AI methods. A few studies focus on enhancing the efficiency of solving complicated discrete or combinatorial problems with AI methods. Instead of involving two separate layers for making decisions in a "predict-then-scheduling" manner, studies such as Baty et al. (2024) encapsulate the machine learning and combinatorial optimization into a pipeline, which iteratively trains the machine learning to predict better parameters for the combinatorial optimization. Besides, the notorious computational complexity of discrete and combinatorial optimization problems motivates many valuable efforts to leverage AI methods to speed up classical branch-and-bound or branch-and-cut algorithms while maintaining the solution quality (Huang et al., 2021). Zhang et al. (2023) provide a comprehensive survey about the attempts that AI is used to facilitate the solution of MIP models with exact algorithms and heuristic algorithms. They point out that the application of ML in "learning to branch" for improving solution efficiency of exact algorithms includes branching variable selection, branching node selection, cutting plane, and even decomposition strategy selection.

2.5 Research Gaps

The previous literature review is further summarized in Table 2-3. Moreover, based on the review, we derive several significant research gaps.

Table 2-3. Summary of prior literature and research gaps

	Research Focus	Prior Literature	Research Gaps
Problem Domain	Production scheduling in robotic cells	(i) The introduction of AMR in production cells complicates the production operations (Brucker et al., 2012); (ii) most studies focus on reducing energy from the machine side (Wu & Che, 2019; Zhang & Chiong, 2016); (iii) only a few papers study the energy consumption of AMR (Gürel et al., 2019).	(i) Energy efficiency of robotic cells is under-explored; (ii) The robot movement process has been rarely considered; (iii) No prior studies explore the coordination of these two processes to reduce energy waste.
	Enhancing production scheduling by incorporating multiple influencing factors	Industry 4.0 empowers CPS systems and data-based predictive analytics (Rossit et al., 2019a). Many studies focus on predicting order arrival, machine anomaly detection, maintenance requirements, and remaining useful life prediction (Ghaleb et al., 2020; Li et al., 2015; Rathore & Harsha, 2022; Zonta et al., 2022)	Less concentration on exploring the influences of various operational production factors on processing performance, such as operator experience, material usage, material supply, product quality, environment, and performing sequence.
		Prediction for making rescheduling decisions, better scheduling of maintenance, and better production resource (e.g., machine) allocation (Jacso et al., 2023; Li et al., 2020b; Rohaninejad et al., 2023)	No prior studies explore how can multiple operational production factors affect the job processing rate and how to utilize such influences to derive better schedules.
Solution Method	Enhancing machine and robot-handling collaboration	Several energy-saving methods, such as multi-objective optimization, turning on/off methods, and scheduling to off-peak periods, speed scaling methods (Dai et al., 2019; Meng et al., 2019; Wang & Wang, 2019; Wu & Che, 2019)	There is no investigation into developing a new modelling approach to facilitate the collaboration between the two subsystems.
	Efficient integrated AI-empowered optimization solution approaches	Classical ML models (e.g., SVM and ANN) and variants of RNN models are explored for industrial prediction (Yeh & Deng, 2012; Malhotra et al., 2015; Zhao et al., 2017a; Wang et al., 2018b).	Developing powerful deep learning models for extracting useful patterns from relevant operational influencing factors is under-explored
		(i) Empowering scheduling results with AI techniques in a "prediction-then-scheduling" manner (Zonta et al., 2022). (ii) Improving the algorithmic efficiency of optimization algorithms with machine learning techniques (Baty et al., 2024; Huang et al., 2021)	(i) Few research on integrated machine learning and optimization methods to solve production scheduling; (ii) Few investigations on prediction-enabled optimization methods for better scheduling and enhanced solution efficiency

From the summary in Table 2-3, first, although robotic technologies have reshaped the manufacturing industry, the energy efficiency of robotic cells is under-explored. Besides, even though several energy- reduction strategies have been developed for job shops, most studies focus on reducing energy consumption from the side of machines. However, the robot movement process which occupies a significant portion of cycle time and consumes lots of energy has been rarely considered. Moreover, a great amount of energy is wasted during machine idling, blocking, and robot movement due to the mismatching between the machine production and the robot movement. However, no prior studies explore the coordination of these two processes to reduce energy waste. More specifically, little research investigates the benefits of simultaneously controlling the operating speeds of machines and mobile robot in enhancing systematic sustainability performances. (*These research gaps are addressed in Section 3*).

Previous studies using learning methods to extract knowledge from historical data often focus on the machinery side, such as anomaly detection, predictive maintenance, and remaining useful life prediction. These aspects are very important to get knowledge of when a machine may fail and suggest possible adjustment actions. However, from the operating level, they cannot directly indicate production status and guide the planning and resource allocation. Besides, very few studies concentrate on exploring the potential influences of factors in the operational processing circumstance on the production performance indicators such as the time consumption of carrying out tasks, and the tardiness of performing tasks with the resource plan. Moreover, no previous studies were found to consider the joint effect of multiple factors, such as operator experience, material usage, material supply, product quality, environment, and also the effect of previous job executions on production performance. (*These research gaps are addressed in Section 4*)

By reviewing the literature, it is also discovered that research on applying data-driven methods or integrated machine learning and optimization methods to solve production scheduling to enhance decision quality is far from sufficient. Furthermore, previous research efforts aimed at enhancing scheduling outcomes with AI techniques typically adopt a "prediction-then-scheduling" approach. This method involves first executing the prediction task to acquire essential parameters, which then inform the creation and solution of mathematical models (including objectives and constraints) to generate scheduling results. No studies investigate algorithmic methods that integrate the prediction engine into the optimization decision making process to simultaneously produce better scheduling results and enhance algorithmic solution efficiency. (*These research gaps are addressed in Section 5*).

Chapter 3. Collaborative Robotic Job-Shop Scheduling[13]

The study described in Chapter 3 concentrates on a green modelling approach to realize the integration of a mobile robot into the system with the consideration of sustainability impact. More specifically, a mobile robot is adopted for material handling in a production cell operated in the job shop setting. We propose an energy-efficient modelling approach that facilitates the operational collaboration between machines and mobile robot. By adopting the proposed method, energy reduction can be achieved by eliminating unnecessary energy consumption due to machine idling and robot waiting. In addition, the proposed method can achieve a balance between energy consumption and production efficiency by further taking production makespan into account.

3.1 Problem Description

The RJSP with energy considerations studied in this study is described as follows. In the robotic cell, there are $|M|$ processing machines and one mobile robot. Each machine can perform a specific type of operation. An input depot *D* and an output stock *S* are placed at the two ends of the cell. The robotic cell aims to process sets of jobs. The job features and machine execution follows the job shop setting. Each job i consists of several ordered operations denoted by $J_i = \{O_{i1}, O_{i2}, \ldots, O_{i|Ji|}\}$. The sequence of operations for a job is named as *in-job sequence*. Each O_i is a nonstop operation that will be performed by a designated machine M_{ij} with processing time PT_{ij}. Pre-emption is not allowed. Notations used in this study are summarized in Table 3-1.

[13] Most of this chapter is published in Wen, X., Sun, Y., Ma, H. L., & Chung, S.H. (2023). Green smart manufacturing: energy-efficient robotic job shop scheduling models. *International Journal of Production Research*, *61*(17), 5791-5805.

Table 3-1. Notations[14]

Parameters	
I	Set for jobs, $I = \{1,2,\dots,\lvert I\rvert\}$.
$\lvert I\rvert$	Total number of jobs.
i, m, h	Indexes for jobs, where $i, m, h \in I$.
J_i	Set for operations in job *I*, $J_i = \{1,2,\dots,\lvert J_i\rvert\}$.
$J_{i'}$	Set for operations with the stock operation in job *i*, $J_{i'} = \{1,2,\dots,\lvert J_i\rvert + 1\}$.
$\lvert J_i\rvert$	Number of operations in job *i*.
$\lvert J_{i'}\rvert$	Number of operations with the stock operation in job *i*, $\lvert J_{i'}\rvert = \lvert J_i\rvert + 1$.
j, n, g	Indexes for operations, where $j \in J_{i'}, n \in J_{m'}, g \in J_{h'}$.
O_{ij}, O_{mn}, O_{hg}	Index for the *j-th*, *n-th*, and *g-th* operation of job *i*, job *m*, and job *h*.
M	Set for machines, $M = \{1,2,\dots,\lvert M\rvert\}$.
$\lvert M\rvert$	Number of machines.
k	Index for the *k-th* machine, $k \in M$.
M_{ij}	Index for the machine to execute O_{ij}.
PM_{ij}	Position for machine to execute O_{ij} (Positions for D and S: $PM_D = 0$, $PM_{i\lvert J_{i'}\rvert} = \lvert M\rvert + 1$).
SPT_{ij}	Processing time of O_{ij} on the machine under normal speed (stock operation: $PT_{i\lvert J_{i'}\rvert} = 0$).
tl_{ij}	Moving time for a loaded movement from M_{ij-1} to M_{ij}.
tu_{ijmn}	Moving time for an empty movement from M_{ij} to M_{mn} under normal speed.
tu_{ijD}	Moving time for an empty movement from M_{ij} to D under normal speed.
σ_{ij}	Moving distance for a loaded movement from M_{ij-1} to M_{ij}.
σ_{ijmn}	Moving distance for an empty movement from M_{ij} to M_{mn}.
σ_{ijD}	Moving distance for an empty movement from M_{ij} to *D*.
V	Set for speed scales of machines and robot, $V = \{1,2,\dots,\lvert V\rvert\}$.
$\lvert V\rvert$	Number of speed scales of machines and robot.
v_k	Normal processing speed for machine *k*.
v_{kv}	Actual processing speed for machine *k* under speed scale *v*.
v_R	Normal robot moving speed (1 unit distance/minute).
v_r	Robot moving speed under speed scale r.
μ_k	Processing power of machine *k* under normal speed (unit: w).
μ_{kv}	Processing power of machine *k* under speed scale *v*.
α_k	Operating characteristics of the machine *k*.
$p\ (q)$	Parameters denote the positive relationships between the speed and machine (robot) power.
w	Robot loaded weight.
ξ	Operating characteristics of the robot.
C_0	Minimized makespan derived by the traditional model.
EI_k	Idling energy consumption per unit time of machine *k*.

[14] The meanings of notations listed here are only applicable to Chapter 3.

SPE_{ij}	Energy consumption to perform O_{ij} under normal speed.
ERE	Energy consumption per unit distance for robot empty movements under normal speed.
ERL	Energy consumption per unit distance for robot loaded movements.
SLE_{ij}	Energy consumption for robot loaded movement from M_{ij-1} to M_{ij}.
SEE_{ijmn}	Energy consumption for robot empty movement under normal speed from M_{ij} to M_{mn}.
β	A large positive number.
α	Makespan increase tolerance.
F	Dummy sink node that connects with the last operation in a job batch.
Decision Variables	
X_{ijmn}	Binary decision variable. It equals 1 when the robot leaves for O_{mn} after O_{ij} starts, where *i* and *m* are two different jobs; 0 otherwise.
$Y_{ij(j+1)}$	Binary decision variable. It equals 1 when the robot waits for the entire processing time of the current operation O_{ij} and goes to $M_{i(j+1)}$; 0 otherwise.
Z_{ijhg}	Binary decision variable. It equals 1 when both O_{ij} and O_{hg} are executed on the same machine, and O_{ij} precedes O_{hg} (not necessarily the immediate predecessor); 0 otherwise.
VM_{ijv}	Binary decision variable. It equals 1 when machine executes O_{ij} with speed scale *v*.
VR_{ijmnr}	Binary decision variable. It equals 1 when the robot selects speed scale *r* to execute the empty movement from M_{ij} to $M_{m(n-1)}$.
SM_{ij}	Starting time of O_{ij} on the assigned machine.
SM_F	Time to reach the sink node *F*.
RM_{ij}	Removing time of O_{ij} from machine after completion.
APE_{ij}	Energy consumption of O_{ij} under actual processing speed.
APT_{ij}	Time consumption of O_{ij} under actual processing speed.
AEE_{ijmn}	Energy consumption for moving from M_{ij} to M_{mn} under actual robot speed.
TI_k	Total idling time of machine *k*.
$Cmax$	Makespan.
LE	Total energy consumption for loaded movements.
EE	Total energy consumption for empty movements.
TE	Total energy consumption for movements.
PE	Total machine processing energy consumption.
IE	Total machine idling energy consumption.
AE	Total auxiliary energy consumption.

A single-gripper robot is involved for the in-facility movements of goods. For each job, the robot should first pick the initialized job up at *D* before moving it to the first machine *M_{i1}*. Also, the robot should deliver the job to *S* after all operations within it are completed. As there is no precedence restriction between operations from different jobs, the robot can flexibly turn to handle another job (after an upload action) if all *in-job sequences* are not violated. Two types of robot movement exist: *loaded movement* implies moving a job to a machine for uploading and processing, while *empty movement* indicates a deadhead movement that relocates the robot for job picking-up. Due to the linear layout, the moving time between any pair of machines is symmetrically determined by the absolute distance between their locations. Similar to Sun et al. (2021), this study considers the situation that machines have no buffer and the mobile robot has the capacity to hold one product each time. Therefore, machines and the robot can be occupied by only one job at any time. Besides, a job will be blocked on a machine after completing until the robot comes for releasing. Such periods are called *machine blocking*. The setup time of both the machines and the robots is incorporated into the processing times and movement times.

The problem aims to simultaneously determine the job schedules and the robot route. Following the above settings, two scenarios can be extracted after a loaded movement that places job i ($i\in I$) on one machine: the robot can (i) wait at the machine for the entire processing of the operation and then transport job i to its next handling machine (defined as a *robot full-blocking*); or (ii) turn to another job m ($m\in I; m\neq i$). In the second scenario, the robot first moves emptily to the machine currently holding job m (or to D). Then, three sub-scenarios may appear: (a) the robot arrives later than the completion time of job m's current operation (i.e., job m should experience a blocking period in this sub-scenario); (b) the robot arrives before the completion of that operation and the robot should experience a *robot partial-blocking (RPB))* before it can conduct the transport; and (c) the robot arrives exactly when the operation is finished and could pick up job m directly for the next move. Sub-scenario (c) is a synchronized process, where machine blocking and robot partial-blocking are avoided.

In the machine speed scaling framework, the processing speed of each machine can be selected from a finite and discrete set (Abedi et al., 2020; Hassani et al., 2019; Zhang and Chiong, 2016). This study proposes to improve the coordination of machines and the robot with a *V-scale* speed framework that enables speed adjustment for both machines and the robot. Generally, machines and the robot operate at normal speeds (denoted as v_k $(k \in M)$ for machine k, and v_R for the robot). For productivity, companies usually set machines to work at a high speed. Thus, this study considers the normal speed as the highest speed. While with the *V-scale* speed framework, for each operation, the actual processing speed level v_{kv} $(k \in M, v \in V)$ is a value selected from $|V|$ levels ($v_{kv} \leq v_k$). Also. The moving energy can be reduced by adjusting the robot speed. As the robot partial-blocking only occurs after empty movements, the empty movements can be decelerated to eliminate the original robot partial-blocking periods under normal speeds. Thus, for each empty movement, the robot selects a speed v_r $(r \in V)$ from the *V- scale* levels. Considering there is no robot partial-blocking after loaded movements or movements to *D* and *S*, these movements are conducted at the highest speeds to ensure productivity.

Here the energy calculation methods for machines and the robot are specified. Following Zhang and Chiong (2016), the machine processing energy equals the actual processing time (APT) multiplying the power under this speed. If the normal speed is applied for O_{ij}, the APT_{ij} equals to the normal processing time SPT_{ij}. While if a lower speed v_{kv} is selected, the APT_{ij} is proportionally changed to $\frac{v_k}{v_{kv}} SPT_{ij}$. μ_k denotes the power of machine k with normal speed, and μ_{kv} represents the power of Machine k under the selected speed v_k. $\mu_k > \mu_{kv}$ because the machine power is positively related to the processing speed. Besides, similar to Zhang and Chiong (2016), this study considers cases where energy consumption decreases with reduced speed despite the longer processing time. Therefore, $\mu_k \times SPT_{ij} > \mu_{kv} \times APT_{ij}$. Equations

(3.1) - (3.2) derive the machine power under the normal speed and the selected speed. Parameter p denotes the positive relationship between the speed and the power, which should be larger than 1 to guarantee that the speed growth leads to an increasing energy consumption rate. Note that α_k represents the operation characteristics of machine *k*. Equations (3.3) calculate the actual energy consumed for processing O_{ij} and derives the relationship between the actual energy consumption APE_{ij} and the normal consumption SPE_{ij}.

$$\mu_{ks} = \alpha_k {v_k}^p \tag{3.1}$$

$$\mu_{kv} = \alpha_k {v_{kv}}^p \tag{3.2}$$

$$APE_{ij} = \mu_{kv} \times APT_{ij} = SPT_{ij} \times (\frac{v_{\mathrm{kv}}}{v_{\mathrm{k}}})^{p-1} \times \mu_{ks} = SPE_{ij} \times (\frac{v_{\mathrm{kv}}}{v_{\mathrm{k}}})^{p-1} \tag{3.3}$$

Following Gürel et al. (2019), it is considered that the robot movement energy consumption depends on the robot moving speed, the travelled distance, the carrying load, and the operating characteristics of the robot. Equations (3.4) calculate the *loaded energy consumption per unit distan*ce (ERL), which is jointly determined by robot operating characteristics ξ, loaded weight *w*, and robot normal speed v_R. The exponential parameter q (*q*>1) forms the positive relationship between the moving speed and energy consumption, which indicates that higher speeds lead to larger energy consumption (but not a linear relationship). Equations (3.5) obtain the energy for loaded movement from $M_{i(j-1)}$ to M_{ij} (the travelling distance is σ_{ij}). Similarly, Equation (3.6) computes the unit consumption of empty movements under normal speed, based on which Equation (3.7) calculates the energy consumption of empty movements under normal speed for a distance σ_{ijmn}. Equations (3.8) derive the actual energy needed for an empty movement from M_{ij} to M_{mn} by applying the *V-scale* speed scales.

$$ERL = \xi w {v_R}^q \tag{3.4}$$

$$SLE_{ij} = \sigma_{ij} \times ERL \tag{3.5}$$

$$ERE = \xi {v_R}^q \tag{3.6}$$

$$SEE_{ijmn} = \sigma_{ijmn} \times \xi {v_R}^q \tag{3.7}$$

$$AEE_{ijmn} = \ \sigma_{ijmn} \times \xi {v_r}^q = \sigma_{ijmn} \times ERE \times (\frac{v_r}{v_R})^q \tag{3.8}$$

3.2 Model Development

As introduced, this study proposes two novel robotic job-shop scheduling models to enhance energy efficiency. In this section, the novel RJSP-E is first presented. Then, the RJSP-EM is constructed in Section 3.2.2.

3.2.1 Model RJSP-E

The model named *robotic job-shop scheduling with energy consumption* (i.e., RJSP-E) is first formulated with Equation (0) to Equations (44) in Table 3-2. Following Sun et al. (2021), a network-based modelling approach is applied to build the new models. The optimization objective is to minimize the total energy consumed by the robotic cell. The overall energy consumption (as shown in Equation (0)) consists of four parts: the machine processing energy, the machine idling energy, the robot movement energy, and the auxiliary energy consumption. In the following, the energy consumption constraints are first explained. Then, other traditional RJSP constraints are briefed.

Total machine processing energy consumption

The energy consumed by machine production is the product of the power of machines (in Watts, w) and the processing time (in seconds). Constraints (18-20) calculate the total machine processing energy consumption by summing the actual energy consumed by each operation. Specially, Constraints (18) ensure that each operation is assigned with one speed from the *V-scale* framework. Constraints (19) derive the actual processing energy (APE_{ij}) under the selected speed with the relationship between APE_{ij} and the normal energy consumption SPE_{ij} (refer to Equation (3.3) in Section 3.1). Constraints (20) then add up all APE_{ij} to obtain the total machine energy consumption PE.

Total robot movement energy consumption

The energy consumed by robot movement is the product of the electricity consumed for a unit distance[15] movement (in KJ) and the moving distance. Constraints (21-25) formulate the total robot movement energy consumption by summing up the robot energy consumed by loaded movements and empty movements. Constraints (21) make sure that the empty movement from O_{ij} to $O_{m(n-1)}$ will be assigned a speed level if an arc X_{ijmn} is selected. Constraints (22-24) derive the total energy consumed by empty movements (*EE*), which is further divided into empty movements to machines (Constraints (22)) and empty movements to the *D* (Constraints (23)). Constraints (22) apply the *V-scale* speed framework to empty movements and derive the energy consumption (refer to Equations (3.8) in Section 3.1). Constraints (23) add up all empty movements to the input depot for picking up the initialized jobs, which are conducted at the normal speed. Constraints (25) obtain the energy consumed for loaded movements (*LE*) by summing up the electricity used for every loaded movement, as calculated with Equations (3.5) in Section 3.1.

[15] A unit distance is the distance moved in a minute of the robot.

Total machine idling energy consumption

The total machine idling energy consumption is the electricity used during machine idling periods throughout the entire manufacturing process. Constraints (26) calculate the length of idling time encountered by each machine by subtracting the processing time of that machine from the lasting time *Cmax*. Constraints (27) obtain the total machine idling energy consumption by adding up the idling energy of each machine, and the latter is calculated by multiplying the individual idling power and the idling time.

Auxiliary energy consumption

The auxiliary energy is consumed by supporting activities in the robotic cell not directly related to production, such as keeping temperature and humidity. Following Meng et al. (2019), it is modelled as proportional to the total processing time (i.e., the makespan) by an auxiliary energy consumption coefficient *s* (Constraint (28)).

Table 3-2. The formulation of RJSP-E.

$Obj.\ Min\ \ PE + IE + LE + EE +\ AE$		(0)
$s.t.$		
$C_{max} \geq SM_F,$		(1)
$SM_F \geq SM_{ij}$	$\forall i,j \in \{1,2,\dots,\lvert J_{i'}\rvert\},$	(2)
$SM_{11} = \lvert PM_{11} - PM_D\rvert / v_R,$		(3)
$\sum_{m\in I}\sum_{n\in J_{m'}} X_{ijmn} + Y_{ij(j+1)} = 1,$	$\forall i, i \neq m, j \in \{1,2,\dots,\lvert J_i\rvert\},$	(4)
$\sum_{m\in I}\sum_{n\in J_{m'}} X_{mnij} + Y_{i(j-1)j} = 1,$	$\forall i, i \neq m, j \in \{2,3,\dots,\lvert J_{i'}\rvert\},$	(5)
$\sum_{m\in I}\sum_{n\in J_{m'}} X_{i\lvert J_{i'}\rvert mn} + X_{i\lvert J_{i'}\rvert F} = 1,$	$\forall i, i \neq m,$	(6)
$\sum_{m\in I}\sum_{n\in J_{m'}} X_{mni1} = 1,$	$\forall i, i \neq m, i \neq 1,$	(7)
$SM_{i1} \geq\ SM_{mn} + tu_{mnD} + tl_{i1} - (1 - X_{mni1}) \times \beta,$	$\forall i, m, i \neq m, n \in \{1,2,\dots,\lvert J_{m'}\rvert\},$	(8)
$RM_{ij} \geq\ SM_{mn} + \sum_{\mathrm{r}\in\mathrm{V}} VR_{mni(j+1)r} \times tu_{mnij} \times (v_R/v_r) - \left(1 - X_{mni(J+1)}\right) \times$ $\beta,$	$\forall i, m, i \neq m, j \in \{1,2,\dots,\lvert J_i\rvert\},, n \in \{1,2,\dots,\lvert J_{m'}\rvert\},$	(9)
$RM_{ij} \geq SM_{ij} + APT_{ij}$	$\forall i,j \in \{1,2,\dots,\lvert J_i\rvert\},$	(10)
$APT_{ij} = \sum_{v\in\mathrm{V}} VM_{ijv} \times PT_{ij} \times (v_{\mathrm{k}}/v_{\mathrm{kv}}),$	$\forall i,j \in \{1,2,\dots,\lvert J_i\rvert\},$	(11)
$SM_{ij+1} \geq RM_{ij} + tl_{ij+1}$	$\forall i,j \in \{1,2,\dots,\lvert J_i\rvert\},$	(12)
$Z_{ijhg} + Z_{hgij} = 1,$	$\forall i,h,j \in \{1,2,\dots,\lvert J_i\rvert\}, g \in \{1,2,\dots,\lvert J_h\rvert\}, M_{ij} = M_{hg},$	(13)
$SM_{ij} \geq SM_{h(g+1)} + tu_{h(g+1)i(j-1)} +\ tl_{ij} - Z_{ijhg} \times \beta,$	$\forall i,h,j \in \{2,3,\dots,\lvert J_i\rvert\}, g \in \{1,2,\dots,\lvert J_h\rvert\}, M_{ij} = M_{hg,},$	(14)

$SM_{hg} \geq SM_{i(j+1)} + tu_{i(j+1)h(g-1)} + tl_{hg} - (1 - Z_{ijhg}) \times \beta,$	$\forall i,h,j \in \{1,2,\dots,\vert J_i\vert\}, g \in \{2,3,\dots,\vert J_h\vert\}, M_{ij} = M_{hg},$	(15)
$SM_{i1} \geq SM_{h(g+1)} + tu_{h(g+1)D} + tl_{i1} - Z_{i1hg} \times \beta,$	$\forall i,h,g \in \{1,2,\dots,\vert J_h\vert\}, M_{i1} = M_{hg},$	(16)
$SM_{h1} \geq SM_{i(j+1)} + tu_{i(j+1)D} + tl_{h1} - (1 - Z_{ijh1}) \times \beta,$	$\forall i,h,j \in \{1,2,\dots,\vert J_i\vert\}, M_{ij} = M_{h1},$	(17)
Processing		
$\sum_{v\in V} VM_{ijv} = 1,$	$\forall i,j \in \{1,2,\dots,\vert J_i\vert\},$	(18)
$APE_{ij} = \sum_{v\in V}[VM_{ijv} \times SPE_{ij} \times (v_{kv}/v_k)^{p-1}],$	$\forall i,j \in \{1,2,\dots,\vert J_i\vert\},$	(19)
$PE = \sum_{i\in I}\sum_{j\in J_i} APE_{ij},$		(20)
Transportation		
$\sum_{r\in V} VR_{ijmnr} = X_{ijmn},$	$\forall i,m,m \neq i, j \in \{1,2,\dots,\vert J_{i'}\vert\}, n \in \{2,3,\dots,\vert J_{m'}\vert\},$	(21)
$AEE_{ijmn} \geq \sum_{r\in V}[VR_{ijmnr} \times \sigma_{ijm(n-1)} \times (v_r/v_R)^q \times ERE],$	$\forall i,m,m \neq i, j \in \{1,2,\dots,\vert J_{i'}\vert\}, n \in \{2,3,\dots,\vert J_{m'}\vert\},$	(22)
$AEE_{ijm1} \geq \sigma_{ijD} * ERE - (1 - X_{ijm1}) \times \beta,$	$\forall i,m,m \neq i, j \in \{1,2,\dots,\vert J_{i'}\vert\},$	(23)
$EE \geq \sum_{i\in I}\sum_{j\in J_{i'}}\sum_{m\in I}\sum_{n\in J_{m'}} AEE_{ijmn},$		(24)
$LE \geq \sum_{i\in I}\sum_{j\in\{2,3,\dots,\vert J_{i'}\vert\}}(\sigma_{ij} \times ERL) + \sum_i \sigma_{i1} \times ERL,$		(25)
Idling		
$TI_k \geq Cmax - \sum_{i\in I}\sum_{j\in J_{i'}} APT_{ij},$	$\forall k \in \{1,2,\dots,\vert M\vert\}, M_{ij} = k;$	(26)
$IE = \sum_{k\in M}(EI_k \times TI_k),$		(27)
Auxiliary		
$AE = s \times Cmax,$		(28)
$X_{iJmn} \in (0,1),$	$\forall i,m \in \{1,2,\dots,\vert I\vert + 1\}, i \neq m, j \in \{1,2,\dots,\vert J_{i'}\vert\}, \mathrm{n} \in \{1,2,\dots,\vert J_{m'}\vert\},$	(29)
$Y_{ij(j+1)} \in (0,1),$	$\forall i,j \in \{1,2,\dots,\vert J_i\vert\},$	(30)
$Z_{iJhg} \in (0,1),$	$\forall i,h,i \neq h, j \in \{1,2,\dots,\vert J_i\vert\}, g \in \{1,2,\dots,\vert J_h\vert\},$	(31)
$VM_{ijv} \in (0,1),$	$\forall i,j \in \{1,2,\dots,\vert J_i\vert\}, v \in \{1,2,\dots \vert V\vert\},$	(32)
$VR_{ijmnr} \in (0,1)$	$\forall i \quad , \quad m,j \in \{1,2,\dots,\vert J_{i'}\vert\}, n \in \{1,2,\dots,\vert J_{m'}\vert\}, r \in \{1,2,\dots \vert V\vert\},$	(33)
$SM_{ij} > 0,$	$\forall i,j \in \{1,2,\dots,\vert J_{i'}\vert\},$	(34)
$RM_{ij} > 0,$	$\forall i,j \in \{1,2,\dots,\vert J_{i'}\vert\},$	(35)
$APE_{ij} > 0,$	$\forall i,j \in \{1,2,\dots,\vert J_i\vert\},$	(36)
$APT_{ij} > 0,$	$\forall i,j \in \{1,2,\dots,\vert J_i\vert\},$	(37)
$AEE_{ijmn} > 0,$	$\forall i,\mathrm{m},j \in \{1,2,\dots,\vert J_{i'}\vert\}, n \in \{1,2,\dots,\vert J_{m'}\vert\},$	(38)
$TI_k > 0,$	$\forall k \in \{1,2,\dots,\vert M\vert\},$	(39)
$tu_{ijmn} = \vert PM_{ij} - PM_{mn}\vert/v_R,$	$\forall i,m,j \in \{2,3,\dots,\vert J_{i'}\vert\}, n \in \{1,2,\dots,\vert J_{m'}\vert\},$	(40)
$tl_{ij} = \vert PM_{ij} - PM_{i(j-1)}\vert/v_R,$	$\forall i,j \in \{2,3,\dots,\vert J_{i'}\vert\},$	(41)
$tl_{i1} = \vert PM_{i1} - PM_D\vert/v_R,$	$\forall i,$	(42)
$tu_{ijD} = \vert PM_{ij} - PM_D\vert/v_R$	$\forall i,j \in \{1,2,\dots,\vert J_{i'}\vert\}.$	(43)

Other constraints

Although the makespan minimization is not the optimization objective of the RJSP-E, Constraints (1-2) calculate the value of makespan to obtain the length of processing and idling time for machines. Specifically, makespan is no less than the time when the algorithm reaches the dummy sink node *F*. Constraint (3) provides the entry of the model by specifying O_{11} as the first operation to execute. Constraints (4-7) formulate the transportation network. Constraints (8) specify that the first operation of job i (i≠1) should be later than the starting time of O_{mn} plus the travelling time of (i) the empty movement from O_{mn} to *D* and (ii) the loaded movement from *D* to M_{i1}, as long as O_{mn} is linked to O_{i1} by an *X* arc. Constraints (9-10) integrate the speed scaling framework into the robot transportation and machine scheduling processes, which guarantees that the removing time of O_{ij} should satisfy two criteria: (i) if $O_{i(j+1)}$ is the next operation to be executed after O_{mn}, the removing action of O_{ij} can happen after the empty movement of the robot from M_{mn} to M_{ij} (the removing time here is denoted by T_1); (ii) the removing action of O_{ij} can take place after the operation is completed on the dedicated machine with the actual speed (the removing time here is denoted by T_2). Note that if $T_1 > T_2$, a machine blocking appears (the length of the blocking period is $T_1 - T_2$); if $T_1 < T_2$, a partial-blocking of the robot occurs (the length of the robot partial-blocking period is $T_2 - T_1$); and if $T_1 = T_2$, no blocking or robot partial-blocking happens since O_{ij} is finished exactly when the robot arrives, which is a synchronized situation. Constraints (11) obtain the actual processing time APT_{ij} of each operation under the selected speed. Constraints (12) regulate the operation execution sequence within each job. Constraints (13) make sure that there is only one execution sequence for two operations assigned to the same machine, while Constraints (14-17) forbid possible deadlock situations in the production process. Constraints (29-43) specify the value scope of variables and the calculation methods of parameters.

3.2.2 Model RJSP-EM

In the RJSP-E developed in the previous section, the makespan of the production system may increase significantly in order to reduce energy, which can cause additional delays in fulfilling the orders. Therefore, it is desired that the makespan vary in a certain range to avoid heavy impact on makespan while achieving significant energy reduction. To achieve so, this section further formulates the model named *robotic job-shop scheduling with energy consumption and makespan limitation* (RJSP-EM). An additional Constraint (44) is involved in the RJSP-EM, which restricts that the increase in makespan should not exceed an upper limit. Note that C_0 is the makespan obtained by the traditional model without energy considerations, while α represents the tolerance of the decision-maker on the increase in makespan. Through this approach, it is clear for the company to (i) identify how much energy they could reduce if encountering a certain makespan growth, and (ii) understand how the makespan tolerance will affect the speed selection and the trade-off between energy consumption and makespan.

$$Cmax \leq C_0 \times (1+\alpha) \tag{44}$$

The logic behind this constraint can be explained as follows. If there is no makespan restriction, both machine operation processing and robot empty movements will be carried out at a low speed to minimize energy consumption. However, as productivity is another important evaluator for the industry, it is crucial to ensure that the makespan will not be compromised much when we try to reduce energy.

Through numerical examples, the RJSP-E reduces the most energy by selecting slow production/moving speeds by sacrificing makespan. Differently, the RJSP-EM is able to reduce energy consumption by selecting the most appropriate speeds for both machines and the robot to realize coordination, thus achieving energy reduction without much compromise in productivity. Prominently, even when the makespan is not allowed to increase (α=0), the RJSP-EM can reduce the system energy consumption.

Table 3-3. Instances

Instance Code	Problem Scale	Instance Code	Problem Scale
1	5×3×4	6	6×3×4
2	6×3×4	7	8×3×4
3	6×4×4	8	6×4×4
4	5×5×4	9	5×4×4
5	5×3×4	10	6×4×4

3.3 Computational Experiments

In this section, computational experiments are conducted to examine the performances of the proposed models. The traditional RJSP model without energy considerations (see Appendix A), the RJSP-E, and the RJSP-EM are coded in OPL and use the IBM commercial solver CPLEX Studio IDE 12.10 to solve the models on a desktop MacBook Pro with 1.4 GHz Intel Core i5 processor and 8 GB of RAM. The running time limit is set as 3600s. Ten job set instances are tested, which are generated based on the classical work of Bilge and Ulusoy (1995). The problem scales are presented in Table 3-3, denoted by i×j×k (the number of jobs, the number of operations in the job, and the number of machines). Table 3-4 and Table 3-5 show energy-related parameters for machines and robot movements. Specially, a three-scale speed framework is applied to machines and robot. Specifically, the normal speed v_k is defined as level 3, which is the fastest. Machines can turn to slower speeds $\frac{5}{6}v_k$ and $\frac{2}{3}v_k$. While the robot is considered to have a larger adjustment range. It can change to

$\frac{2}{3}v_R$ and $\frac{1}{3}v_R$.

Table 3-4. Parameters for machine energy consumption

	Processing power (w)			Idling power (w)
	Level 3 (Normal speed v_k)	Level 2 ($\frac{5}{6}v_k$)	Level 1 ($\frac{2}{3}v_k$)	
machine 1	2270	1665	1139	370
machine 2	1820	1335	914	350
machine 3	1880	1379	944	350
machine 4	2340	1717	1175	383

Table 3-5. Parameters for robot movement energy consumption

	Loaded movement	Empty movement		
		Level 3 (Normal speed v_R)	Level 2 ($\frac{2}{3}v_R$)	Level 1 ($\frac{1}{3}v_R$)
Energy consumption (KJ/unit distance)	47	28	18	8

This study tests and compares the performance of the traditional model, the RJSP-E, and the RJSP-EM with four different increase tolerance levels (α=0, 5%, 10%, 15%). The models are evaluated from perspectives of the total energy consumption, the energy consumed by machine and transport processes, the makespan, and the CPU time. Major test results are listed in Table 3-6, which lists the comparison of the above models from three perspectives: overall energy, makespan, and CPU time. The following sections unveil the impact of incorporating energy considerations in the RJSP decision framework (the performance of RJSP-E) and the performance of RJSP-EM compared with RJSP-E and the traditional model.

Table 3-6. Experiment results

Metrics	Instance	RJSP-E (KJ)	RJSP-EM (KJ)				Traditional model (KJ)
			α =0	α = 5%	α = 10%	α = 15%	
Overall Energy	1	25402	26338	25727	25540	25422	30131
	2	26823	29383	28213	27514	27160	32062
	3	28943	31673	30459	30036	29494	34882
	4	27023	28188	27305	27029	27023	31201
	5	20186	20474	20333	20284	20196	23151
	6	31885	33911	33075	32349	31919	37857
	7	Uns.	29669	28407	27596	27073	32860
	8	38689	38984	38689	38689	38689	44836
	9	31795	33924	33142	32524	32182	36935
	10	37750	41008	39953	39077	38425	44766
Makespan	1	122	103	108	111	118	103
	2	131	103	108	113	118	103
	3	134	107	112	117	123	107
	4	139	123	129	135	139	123
	5	112	88	91	94	101	88
	6	155	130	136	143	148	130
	7	Uns.	108	113	118	124	108
	8	172	166	172	172	172	166
	9	166	133	139	146	152	133
	10	197	161	169	177	185	161
CPU Time	1	5	0.55	0.74	0.91	1.36	1.36
	2	124	0.91	3.21	4.19	2.26	2.26
	3	440	0.75	4.63	5.93	5.13	5.13
	4	140	8.34	6.82	11.23	3.82	3.82
	5	51	1.27	2.43	2.04	0.88	0.88
	6	33	3.02	7.73	6.4	2.04	2.04
	7	N/A	56.22	85	300	32.75	32.75
	8	540	108	317	694	3.81	3.81
	9	20	2.12	3.86	4.08	2.63	2.63
	10	227	5.69	8.59	12.56	6.08	6.08

3.3.1 Performance of RJSP-E in energy reduction

First, the performances of the RJSP-E and the traditional model is analyzed. Compared with the traditional model, RJSP-E can achieve a remarkable 15% (on average) reduction in energy. However, such an achievement is at the cost of the increase in makespan and CPU time. Since the machines and robots tend to select moderate speed, the makespan average grows by 20% based on the traditional model. Even though RJSP-E is superior in reduction overall energy consumption, the CPU time to reach optimality is 48 times longer than the conventional model. Besides, Instance 7 is unsolvable for the RJSP-E within the given time limit. Therefore, the productivity of the manufacturing system is impaired due to a sacrifice in makespan, and the solution efficiency is much lower.

3.3.2 Performance of RJSP-EM in energy and productivity

To test the performance of RJSP-EM, four makespan increase tolerance levels, 0, 5%, 10%, and 15%, are examined. Table 3-7 summarizes the main metrics comparison results between RJSP-EM vs. EJSP-E and the traditional model.

<u>RJSP-EM vs. RJSP-E</u>

The RJSP-EM alleviates the disadvantages of the RJSP-E in productivity loss with the insertion of the makespan increase restriction. It is reasonable that a tighter makespan increase tolerance level (i.e., a smaller α) leads to a shorter mean makespan (Column MSDE). Besides, the makespan increase constraint shows the potential to accelerate the solution process. Compared with RJSP-E (Column CRE), the RJSP-EM consumes much less CPU time. When α=0, the RJSP-EM even reduces the CPU time by an average of 96%. However, when α=15%, the figure decreases to 77%, which means the advantage in computing time is impaired along with the increase of α.

Obviously, the RJSP-EM consumes more energy than the RJSP-E due to the compressed makespan. Comparing the overall energy between RJSP-EM and RJSP-E (Column TECDE), 6% more energy is witnessed in RJSP-EM when the makespan is not allowed to increase. While the reduction discrepancy is narrowed along with the increase in α. When α equals 15%, the average difference in energy consumption between RJSP-EM and RJSP-E is reduced to 1%, demonstrating the energy-reduction efficacy of RJSP-EM approximates the RJSP-E when α increases to 15%.

Table 3-7. Main metrics comparison

Tolerance α	Energy					Makespan		CPU Time	
	TECDE	TECDT	PECDT	IECDT	TECDT	MSDE	MSIT	CRE	CRT
α = 0	6%	10%	10%	13%	7%	16%	0%	96%	7%
α = 5%	3%	12%	14%	11%	8%	13%	5%	93%	-279%
α = 10%	2%	14%	16%	8%	9%	9%	9%	85%	-899%
α = 15%	1%	15%	18%	3%	10%	6%	13%	77%	-1970%

TECDE: total energy consumption discrepancy compared with RJSP-E; TECDT: total energy consumption discrepancy compared with the traditional model; PECDT: processing energy consumption discrepancy compared with the traditional model; IECDT: idle energy consumption discrepancy compared with the traditional model; TECDT: transportation energy consumption discrepancy compared with the traditional model; MSIT: makespan increase compared with the traditional model; MSDE: makespan decrease compared with the RJSP-E; CRE: CPU time reduction based on the RJSP-E.

RJSP-EM vs. traditional model

The performance of the RJSP-EM over the traditional model is further examined to illustrate the significance of the proposed model in facilitating processing and transport collaboration. From column MSIT, it is obvious that the makespan obtained by the RJSP-EM equals that of the traditional model when α=0. For the RJSP-EM with the other three α, the average makespan increases are prone to reach the given upper bound (i.e., 5%, 9%, and 13% under α=5%, 10%, and 15%). This shows that the RJSP-EM is efficient in adjusting the operating speed for operations or empty movements by fully utilizing the allowed makespan relaxation.

From the perspective of energy reduction, the amount of energy reeduced by the RJSP-EM increases along with the growth of α (Column TECDT), which is reasonable because the relaxation in makesapn leaves more space for energy-reduction solutions. It is valuable to note that when α=0, the RJSP-EM outperforms the traditional model with a significant average energy reduction of 10%, demonstrating the merits of the EJSP-EM in speed coordination for reduction energy. However, with the rise in α, even though more energy can be reduced, the reduction efficacy declines. For example, the energy is reduced by 12% when α is set as 5%, while the figure only grows to 15% when α is 15%.

A closer look is taken into the decomposed energy consumption (i.e., the machine processing energy consumption (PE), the machine idling consumption (IE), and the robot movement energy consumption (TE)). The RJSP-EM is shown to consume less PE than the traditional model in all instances by switching to lower production speeds (Column PECDT). Besides, along with the increase in the allowed production time, more energy reduction from PE is witnessed, while the reduction rate is slowed down. For IE (Column IECDT), the largest reduction by RJSP-EM is achieved when α=0. This reduction efficacy is also weakened with the increased α. Moreover, the TE reduction achieved by the RJSP-EM overall witness a slight growth along with the increase in α (Column TECDT). But it does not show a necessary growing trend in individual instances, because under different α the robot can re-design the delivery route or accelerate the movement when necessary to better coordinate with the machine production process. Thus, the increase in TE can be counteracted by the reduced PE to achieve overall energy reduction.

By comparing the CPU time with the traditional model (Column CRT), it is seen that the RJSP-EM with the tightest makespan upper bound shows higher solution efficiency at an average of 7%. However, along with the growth in α, a much longer CPU time is required for the RJSP-EM.

3.3.3 Sustainability analysis

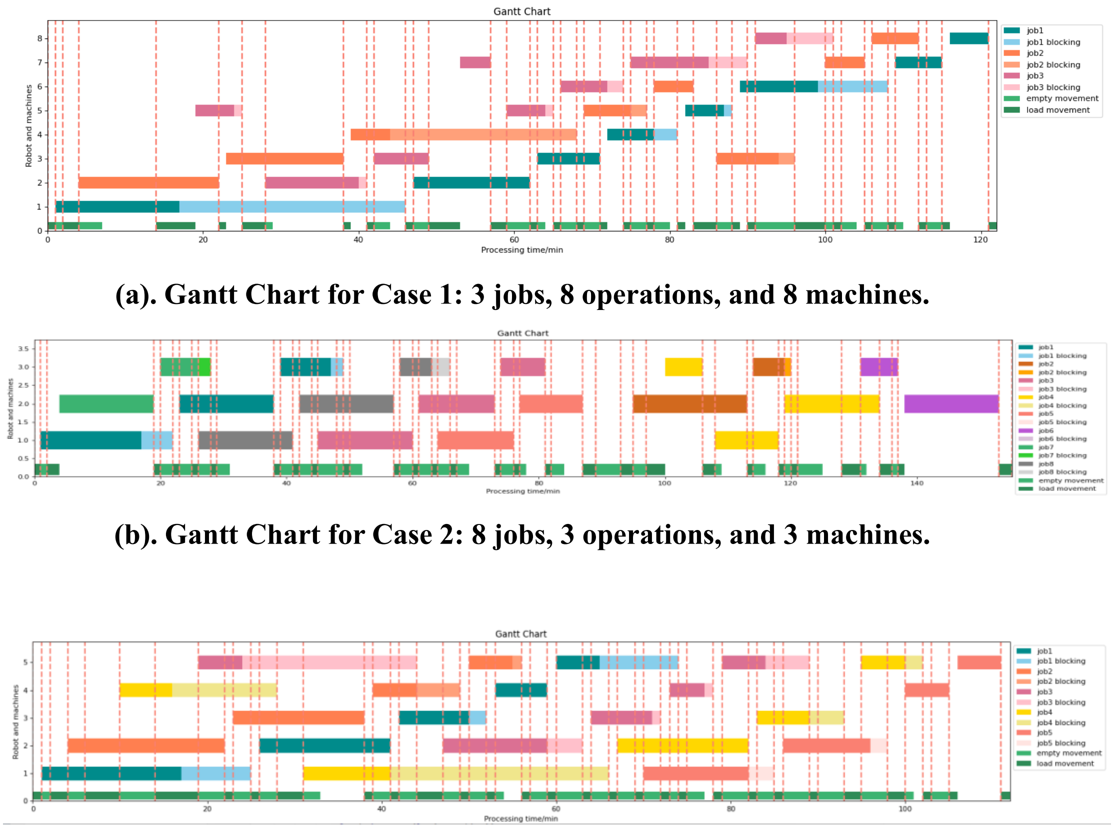
The RJSP-EM and RJSP-E can facilitate the coordination between machines and robot with the *V-scale* speed framework. However, in a more general view, the coordination of machines and robot depends on many factors. Basically, it relates to the number of jobs (batch size), the number of operations in each job (processes), and the number of machines. From the machine perspective, it also depends on the processing time of operations and machine speed. While from the transport perspective, it relies on the layout of machines and the speed of the robot. Therefore, to explore the sustainability of the robotic cell, in this section, a sustainability analysis is further conducted for the above covariates.

To evaluate the sustainability of the entire system (i.e., the coordination between machines and robot in performing batches of jobs), machine blocking and robot blocking (both full-blocking and partial-blocking) can be adopted as measurements. As the experiments vary in the number of machines and the optimal makespan, the average machine blocking rate and robot blocking rate are used as metrics. First, Figure 3-1(a-c) shows the optimal schedules of three different scenarios with a variation in the number of jobs and machines (operations) at the normal processing speed. Specifically, it plots the schedules of operations on machines and the schedule of robot movement. The X-axis is the processing time. The Y-axis includes the robot (bottom green bar) and the code of the machines. In the bottom green bar, the dark green is the loaded robot movement, and the light green is the empty robot movement. Each job is represented by two colours: the dark one denotes the processing, and the light one denotes the machine blocking. In case 1, there are three jobs (each job has eight operations) and eight involved machines. Case 2 oppositely schedules eight jobs (each job has three operations) on three machines. In the more balanced case 3, five jobs (each job has five operations) are planned on five machines.

Table 3-8 summarizes the sustainability indicators. As can be seen, in case 1 when machines are in a large number while the number of jobs is small (but each job has a large number of operations), the machine blocking rate and robot blocking rate are not very high, as the robot can readily handle the products. On the other hand, when more jobs are scheduled on a few machines in case 2, the machine blocking rate is reduced while the robot's blocking time increases. This is because jobs should always wait for the availability of machines and the robot is often blocked by machines (machines are usually occupied). In case 3, the growth in the number of jobs enables parallel processing. The increase in the number of machines facilitates the reduction of robot waiting caused by the high machine occupation rate and transport restrictions. Thus, the robot is more occupied. Nevertheless, the machine blocking rate becomes larger, showing the struggle of the robot to handle a larger workload but still maintain the system efficiency. Consequently, the number of machines and robots should be matched to ensure the sustainability of the system.

Table 3-8. Summary of sustainability indicators.

	Avg_machine_blocking_rate	**Robot_blocking_rate**
Case 1	0.08	0.15
Case 2	0.05	0.45
Case 3	0.15	0.12



(a). Gantt Chart for Case 1: 3 jobs, 8 operations, and 8 machines.

(b). Gantt Chart for Case 2: 8 jobs, 3 operations, and 3 machines.

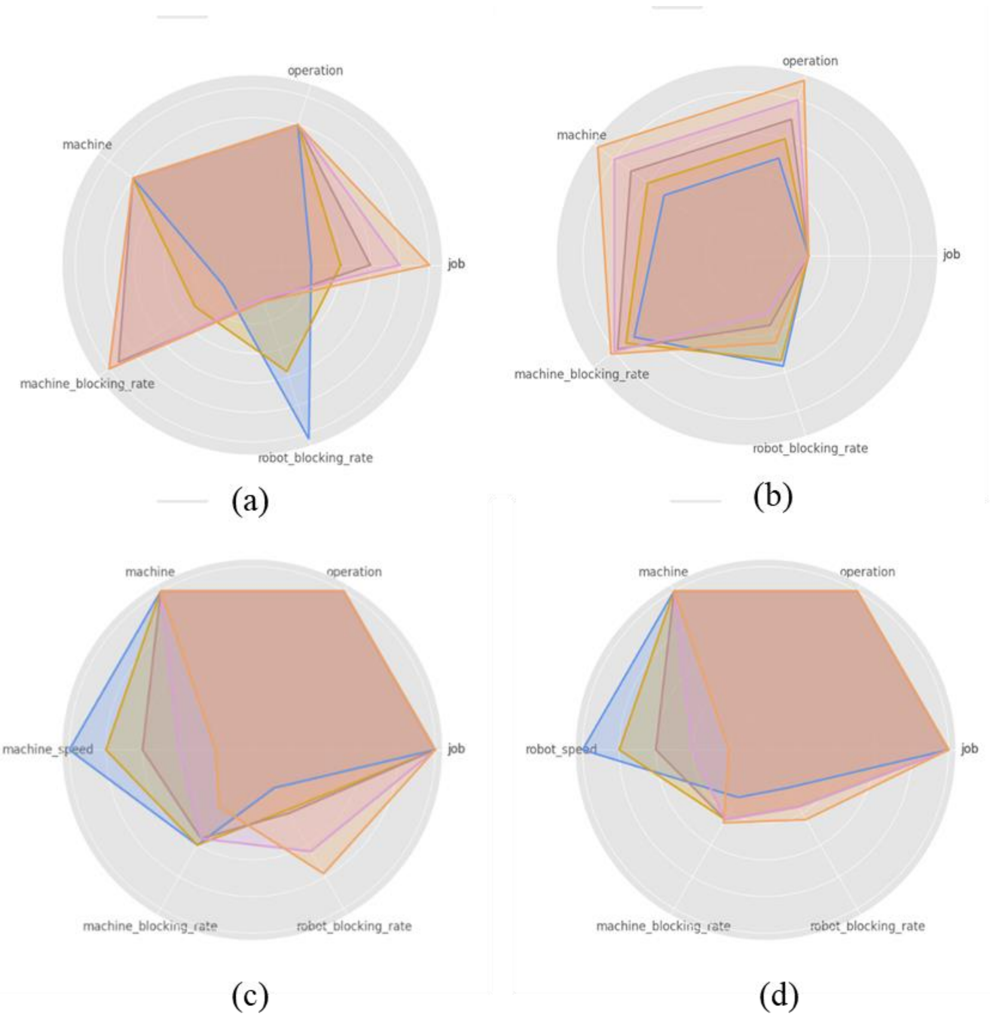
(c). Gantt Chart for Case 3: 5 jobs, 5 operations, and 5 machines.

Figure 3-1. Gantt chart for three representative cases

This study further explores how the speed changes of machines and robot will affect the production process. Figure 3-2 presents the changes of two metrics (machine blocking rate and robot blocking rate) along with the four factors: (a) controls the number of jobs, (b) controls number of machines (operations in jobs), (c) controls the machine speed, and (d) controls the robot speed. Similar to the above analysis, when the number of jobs increases, the robot blocking time is likely to be reduced (Figure 3-2 (a)). When the number of machines (also operations in jobs) increases, the robot blocking time decreases, while the machine blocking time increases (Figure 3-2 (b)).

Figure 3-2 (c) reflects the changes in two metrics with the machine speed adjustment. Decreasing the machine speed has a slight impact on machine blocking time but tends to increase the robot blocking rate. This is because the robot needs to wait longer in both full-blocking and partial-blocking. While in Figure 3-2 (d), decreasing the robot speed will increase the robot blocking rate as the robot is less capable of processing more operations simultaneously. Instead of turning to other operations, the robot will often be blocked by the current operation.

Therefore, the speed scale also has an impact on the system sustainability. However, by using the proposed models, instead of uniformly changing the speed levels, the machines and robot speeds are changed for separate operations or transport, which can be seen as system fine-tuning for sustainability improvement under the premise of maintaining productivity.



(a) (b)

(c) (d)

Figure 3-2. Sustainability measurement and general factors

3.4 Summary

Smart manufacturing has boosted the wide application of mobile robots in robotic cells. However, the mismatching between machine production and robot movement causes extensive energy waste. This study innovatively proposes to achieve energy reduction by enhancing the process coordination between machine production and robot movement. Two MILP models are developed with the application of a *V-scale* speed adjustment framework. The RJSP-E minimizes the overall energy consumption, while RJSP-EM simultaneously considers makespan and energy consumption. Computational experiments are conducted to verify the model performance. The RJSP-E demonstrates superior performances in reducing overall energy consumption (with an average of 15%) but at a loss of makespan (20% on average) due to the slow operating speeds. On the other hand, the RJSP-EM is able to select the most suitable operating speeds to achieve energy reduction without much sacrifice in productivity. Notably, the RJSP-EM reduces energy consumption by a mean of 10% with no compromise in makespan. However, the energy-reduction efficacy of the RJSP-EM declines with the enlarged permitted makespan duration, as the energy reduced from machine processing is counteracted by the additional prolonged idling consumption.

Managerial Implications

The novel RJSP approaches developed in this work can enhance the energy efficiency of modern robotic cells, thus promoting the healthy and sustainable development of smart manufacturing. The study shows that in a smart manufacturing environment with an automated material handling process, production energy can be significantly reduced by developing more powerful scheduling models to achieve intelligent and sustainable interactions/collaboration between machines and robots. Through dynamically adjusting operating speed to accommodate different scenarios (on one hand, adjusting to lower robot speed to reduce unnecessary waiting periods at machines, or on the other hand, slowing down machine speed to suit the transportation capacity of the robot), a significant portion of energy can be reduced without affecting the makespan or throughput of the system.

Also, the autonomous production system is complex and requires the coordination and adaptability of multiple participants and their operational settings. Through sustainability analysis, we show that the number of robots and the speed of machines or robots should be configured according to many factors, including the number of jobs, the number of operations involved in a job, the number of machines, and the layout of the shop floor (e.g., distances between machines), so that the delivery capacity of robots can maximumly satisfy the material handling requirements and reduce the unnecessary blocking periods of machines and robots.

Moreover, robotic cells with configurations similar to the problem settings of this study can apply the proposed model and fine-tune the parameters (like the distances among machines, machine production speeds, and robot movement speeds) to determine their own job assignments, job sequences, machine processing speeds, and robot moving speeds. In addition, job shops with more robots can take the model proposed as a benchmark to adjust their production schedules.

Chapter 4. AI-Enabled Production Status Prediction

This study is driven by the practical needs of a production company operating in the printing industry. Through their operations, it is observed that the time consumed to produce tasks with similar requirements (e.g., output quantity) can vary significantly, which heavily disturbs the establishment of production schedules or the implementation of production plans. This issue raises the demand for production planners to figure out the interrelationship between the job processing time (JPT) and related production factors. In addition, apart from gaining a deeper understanding of the individual JPT, decision-makers such as production line planners also desire a better estimate of the performance of the undergoing production activities. As mentioned in Section 1.1.3, the relative job processing rate (JPR) can provide decision-makers with insights into the system's operating status by indicating whether the performance is high, moderate, or delayed. Inspired by such benefits, this study explores the adoption of AI technology to forecast both the absolute JPT and relative JPR to provide insights for controlling production activities and enabling better planning. Specifically, JPT forecasting is formulated as a regression problem that obtains the estimation of the absolute value of processing time. JPR is cast into six levels, the higher the efficiency of the system. The estimation of JPR is thus a multi-class classification problem.

4.1 Dataset and Potential Dependencies

4.1.1 Dataset

This study adopts the real-world data collected from a partner printing company with IoT sensors (recording real-time processing data and environmental data) and ERP systems (for job specifications, materials, etc.). After data merging and preprocessing (e.g., handling outliers, missing values, different devices), the following categories of data are involved as shown in Table 4-1. The *engineering related data* records the production requirements for a specific printing job, such as the desired output quantity and the utilization of the material (paper and ink); *order related data* concerns information about the customer and other product preferences; *material inspection data* is about detailed specifications of materials; *technician information* provides the grades and service length (indicating experience) of each operator; *environmental data* records the temperature and humidity at each time step; and *production data* records production details (e.g., work shift, machine speed, and setup jobs) when a job is carried out.

Table 4-1. Involved data categories

Engineering Information Order ID * (*ido*); Sheet ID * (*ids*); Job Category (*tc*); Quantity/Workload (*w*); Paper Consumption (*pc*); Paper Supplementary (*ps*); Grams (*g*); Size (width) (*sw*); Size (length) (*sl*); Paper Brand (*pb*); Printing Ink (*pi*); Printing Color (*c*); Material Code (*mc*); Material Quantity (*mq*)
Order Information Order ID * (*ido*); Product Category (*prc*); Customer Category (*cc*); Quality Grade (*qg*)
Material Inspection Report Material Code (*mc*); Material Brand (*mb*); Material Batch Number (*mbn*); Proportion (*mp*); Whiteness (*w*); Thickness (*t*); Folding Resistance (*fr*); Roughness (*r*); Weight (*mw*); Material Suppliers (*ms*)
Technician Information Operator ID * (*idt*); Grade (*opg*); Rank (*opr*); Service Length (*sl*)
Operating Environment Record Time * (*rt*); Temperature (*et*); Humidity (*eh*)
Production Information Date *; Shift (*s*); Sheet ID * (*ids*); Operator ID * (*idt*); Output Quantity (*oq*); Processing Rate (*pr*); Machine Speed (*mss*); Preparation jobs (*pj*)

4.1.2 Dependencies within processing sequence

Upon scrutinizing the job processing workflow on a specific machine, three potential interdependencies between a job's performance and its corresponding circumstances can be identified (which is defined as "context" in the study of Chapter 5). First, to perform the tasks, factors directly related to the processing job are vital, the impact from which is defined as *"direct influence"*. Such factors include the material usage, the quality of materials and quantity of output, the operator in charge, and the environmental indicators like temperature and humidity. The interrelations and interactions among these elements to a large degree affect the JPT&JPR. Then, the actual performance of a job is largely affected by its immediate predecessor, e.g., the changeover time, and the set-up operations. For example, if the former job is labour-intensive, the operator may turn to a low status, which affects the processing rate of the current job. If the former job is largely different from the current one in terms of materials and settings, it may require more time to do the changeover between the two tasks, which will deteriorate the average processing rate of the current job. In this study, the influence coming from the immediate predecessor is defined as *"adjacent influence"*. Thirdly, the processing of one job may be influenced by its previous predecessors, e.g., the processing of a series of jobs on one machine may affect the running speed and the inherent status of the machines, which may exert a so-called *"sequential influence"* on the successors. It is therefore necessary to capture the impact from these three levels to fully utilize the historical data for estimating the production performance.

4.2 Proposed Architecture

To simultaneously account for the influence from three levels and undertake the dual tasks of predicting JPT and estimating JPR level, a multi-module supported dual-task learning (MMDT) architecture is proposed. As shown in Figure 4-1, the MMDT adopts three input modules, which are responsible for extracting the influence of the previously mentioned three levels. The proposed model also involves dual output layers to simultaneously learn the JPT and JPR. It will show that both learning tasks benefit from the hierarchical influence captured by three input modules, and also the co-learning mechanism enabled by the dual output layers. The details of each input module and output layer are detailed in the following.

4.2.1 Input modules

Direct influence module (DIM)[16]

The DIM module is responsible for extracting information from the individual job-related features. The module structure involves several linear layers each followed by a ReLU activation function (to identify nonlinear relationships) and dropout operations (to avoid overfitting). The connected layers enable extracting information from the input factors related to job characteristics (e.g., output quantity, material requirements), operational property (e.g., operator proficiency, machine setting), and environment (e.g., temperature). Then, the input is transformed into a feature representation of the desired dimension. The computation of the output of layer l and input to the next layer $l+1$ can be denoted as:

$$z^{(l+1)} = W^{(l)}a^{(l)} + b^{(l)}$$

$$a^{(l+1)} = f(z^{(l+1)})$$

where $W^{(l)}$ is the weight matrix of layer l, $b^{(l)}$ is the bias of layer l, and $f(.)$ is the activation function where ReLU is adopted: $f\left(z^{(l+1)}\right) = max\left\{0, W^{(l)}a^{(l)} + b\right\}$.

[16] As a remark, meanings of the notations appearing in Chapter 4 below are only applicable to this chapter.

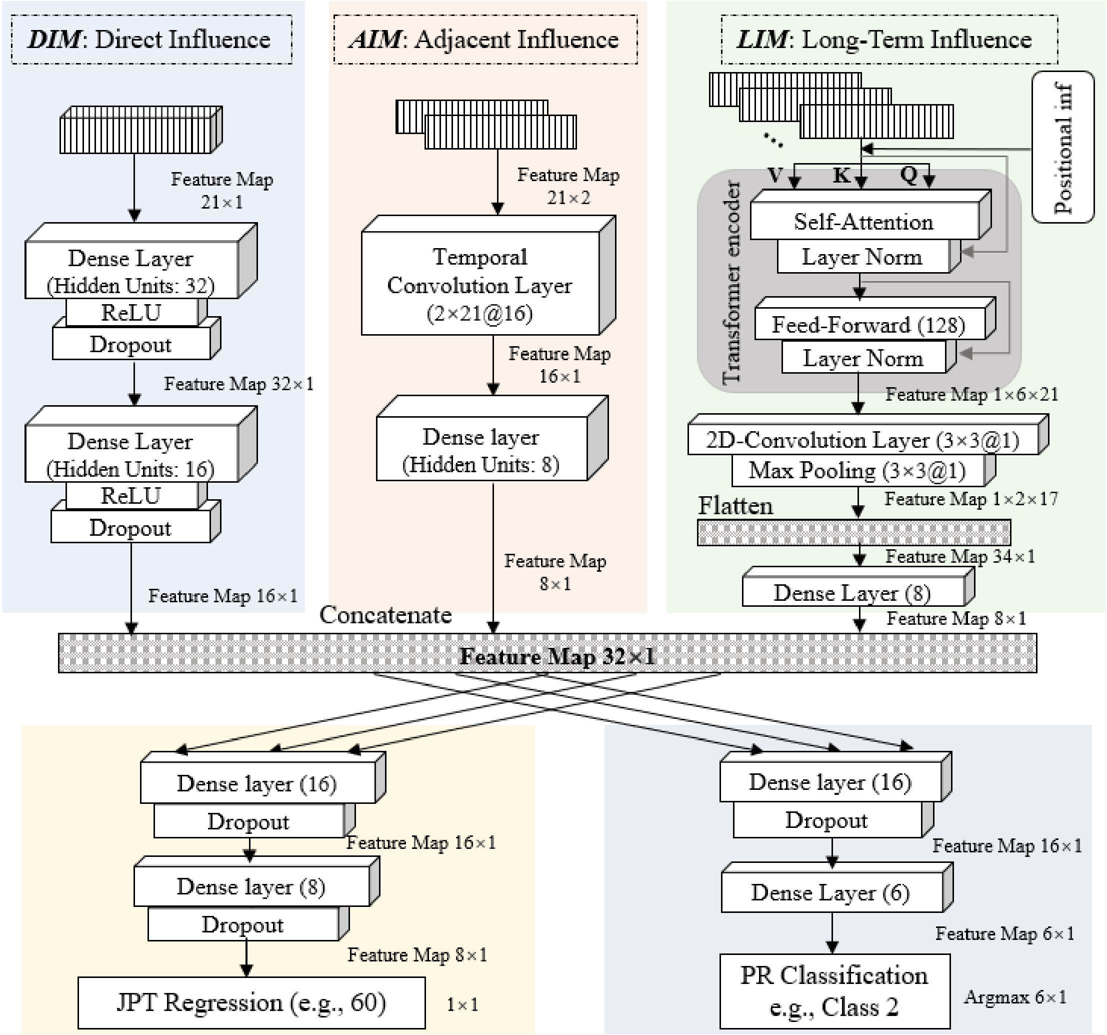


Figure 4-1. Proposed end-to-end MMDT deep learning framework

Adjacent influence module (AIM)

Then, the second input module (AIM) focuses on extracting dependency between adjacent jobs. This is achieved by implementing the temporal convolution operation. Convolution operation is in essence a mathematical operation that can be regarded as the integral of two functions with one function sliding on the input features and the other a shared weight matrix so that the features of different areas can be captured. The shared kernel is designed to highlight certain features in the input data.

For the sequential job processing data of this study, we care about the potential linkage between two jobs conducted on the same machine/by the same operator, thus we focus more on the time-series-based dimension. A set K of parallel 1D kernels is used to highlight such features by moving along the time-based dimension. The formula of 1D-convolution can be described as:

$$Z_k[i] = (I \cdot K)[i] = \sum_{j=0}^{k_l-1} I(i+\ j) \cdot k(j), \qquad k \in K$$

where I is the input matrix and k is one 1-D kernel belonging to the set of all used parallel kernels K. $k(j)$ is the weight vector of kernel k at position j and k_l is the length/size of the kernel k. Let L_{in} denotes the length of the input data, P, D, and S denote padding, dilation size, and stride, respectively. The length of Z_k at the time-series-based dimension can be derived as $\lfloor\frac{L_{in}+2P-D\cdot(k_l-1)-1}{S}+1\rfloor$. After the convolution, the extracted feature map Z_k is fed to a densely connected linear layer, which produces the output of

$$h_{W,b}(z) = Z_k W^T + b$$

where W is the weight matrix, b is the bias vector.

The implementation of the AIP module shown in Figure 4-1 is briefed in the following. Let *feature dim* denote the dimension of input features of the data. The adjacent influence is first extracted with a temporal convolution layer, which adopts 16 parallel 1D kernels with the size of (2, *feature dim*) to capture the inherent feature representation between the two adjacent jobs (the parallel kernels using different weights can capture different aspects of information). The reason for setting the kernel size as two is that the AIM only focuses on capturing the influence of the immediate predecessor on the current job under prediction. Then, the representation feature map goes through fully connected layers to further aggregate the information, which transforms the learned feature map of the adjacent influence as an 8-dimensional representation.

Long-term influence module (LIM)

To capture the long-term influence exerted by the previous job queue, deep learning models with the ability to capture temporal influences are required. While recurrent neural networks such as LSTM and GRU networks can be applied, they may suffer from gradient vanishing (Pascanu et al., 2013), which can result in a weakened impact of information from jobs far ahead. Thus, to capture the influence of the historically executed job queue more accurately, we use the transformer encoder layer, which applies the self-attention mechanism to relate different positions in one sequence, based on which to compute a feature representation with weighted significance for the entire job sequence. Compared with RNN models, it computes the weight for different positions of the entire input to understand the importance of different predecessors, so that the attention can be allocated to different preceding jobs. The core operation of self-attention is the scaled-dot product between queries and keys. The calculation of the self-attention is given by:

$$Attn(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

where $Q = XW_Q$, $K = XW_K$, and $V = XW_V$. X denotes the input matrix, and W_Q, W_K, and W_V are weight matrices. d_k is the dimension of the queries. By this operation, an attention score can be computed between each position and the job sequence, allowing the model to weigh different steps (preceding jobs) of the sequence simultaneously. The obtained attention vector thus contains the effect/ importance of each preceding job to the job under prediction.

Then, to further transform the attention feature map from the encoder layer into a higher-level representation, a 2D-convolution layer with Maxpooling operations is exerted. The first dimension of the convolution aggregates information on the job sequence level, and the second dimension works at the feature level. The formula of 2D-convolution can be described as:

$$Z_k[i,j] = (I \cdot k)[i,j] = \sum_{m=0}^{k_h-1} \sum_{n=0}^{k_w-1} I(i+m, i+n) \cdot k(m,n), \quad k \in K$$

where I is the input matrix. $k \in K$ is one 2-D kernel, with a height of K_h and width of K_w. After one convolution operation, the kernel slides to the next window until deriving the $(I \cdot k)$. After the 2D-convolution and Maxpooling, the obtained feature map is flattened and further fed into a linear layer to adjust the output shape.

It should be noted that only the transformer encoder is applied in the proposed architecture to extract the effect from previous jobs that have been executed and uses 2D-convolutional layer and Maxpooling to further extract information. This architecture is different from the traditional transformers adopting an encoder-decoder structure. The rationale behind this is that we only need to predict the JPT and JPR for a single job at each step, and consequently only the encapsulated information of predecessors is needed for predictions. The decoder layer thus becomes redundant.

The implementation details of our LIM in Figure 4-1 are as follows. First, the features of a series of preceding jobs and the current job under prediction are fed into the transformer encoder to generate a feature map (e.g., if five preceding jobs are involved, the size of the feature map is 1×6×21). Then, the extracted map is further extracted first with a 2D-convolution layer, using one 3×3 kernel. Then a Maxpooling operation is performed to subsample from the feature representation by using one 3×3 filter moving across each channel of the feature map to generate a statistical summary of the features for each nearby region.

4.2.2 Fusion of input modules

The output of the three input modules, i.e., DIM, AIM, and LIM are then fused with a concatenation layer. The extracted high-level feature map combining the knowledge from three input modules is then passed to the output layers for learning the parameters of output layers. It should be noted that the dimension of information (i.e., feature map) injected into the concatenated vector by different input modules can be controlled by adjusting the output dimension of the modules, which helps control the proportion or contribution of each input module and further manage the dimension of the learned feature representation vector. For example, in Figure 4-1, the output dimensions of DIM, AIM, and LIM are 16, 8, and 8, respectively.

4.2.3 Dual output layers

Traditional neural networks (NN) are commonly designed as single-task NN. However, it is shown that by sharing feature representations between two learning tasks with a synergy effect, the prediction performance for both models can be improved (Ruder, 2017). As shown in Figure 4-2, different from the single task network, a multi-output NN contains a few layers shared by different tasks (as shown at the bottom in Figure 4-2). The input data will first go through shared layers, and then the separate subnetworks are constructed to learn the unique feature map of individual tasks. With this approach, during the training process, the backpropagation algorithm can jointly optimize the parameters of the two output layers concurrently. For separate subnetworks, the parameters are trained separately, while for shared layers, both gradients passed from the two subnets will be used to correct the weights, so that the shared hidden units can learn the joint representation of the two tasks.

Following the above idea, the MMDT network incorporates two output layers, which learn the classification task and the regression task separately, while the joint feature representation map extracted from the three input modules is shared by both tasks. The common knowledge shared by the two tasks thus helps the model figure out the similarities and connections between the tasks and transfer knowledge learned from one task to another. As data representation learned by one task can be used by the other, more implicit feature information for one task can be learned with the help of the other through this mechanism. Besides, for each individual task, the training for the other one will eliminate the effects of the noise in the data and the overfitting, as the existence of the other training job forces the network to concentrate on features that really matter.

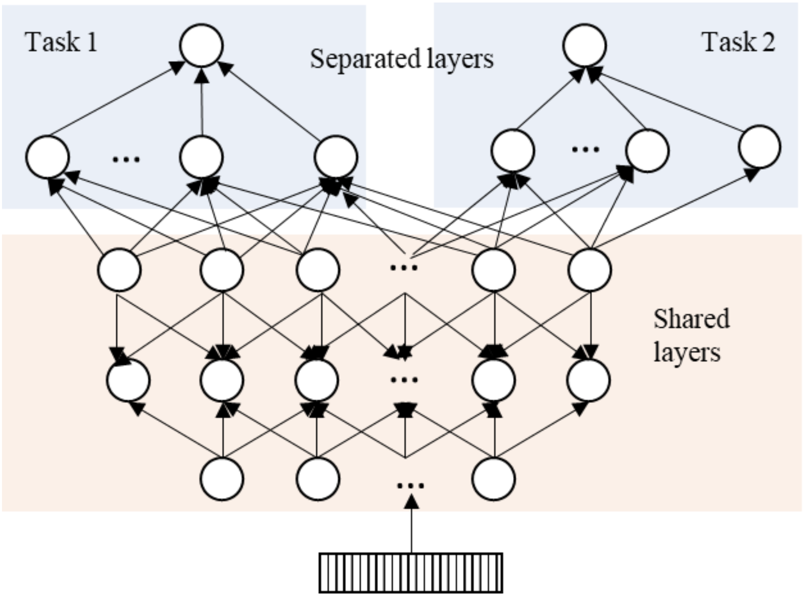


Figure 4-2. Dual-output NN that simultaneously tackles two tasks

It should be noted that one critical requirement to applying the above multi-task training is that the tasks should have commonalities in relevant features. That enables each task to provide meaningful training signals for training each other. In the joint learning of our problem, the prediction of processing time and the classification of processing rate have many connections. In a continuous perspective, the processing rate depends on the inverse of the processing time and the production quantity. The rate classification is therefore a higher-level viewpoint comprising more information. For example, considering performing two tasks with the same output quantity, the judgement of the processing rate level will produce a direct impression for decision-makers in terms of the system efficiency in carrying out the job under these two situations. Both the classification and prediction tasks are based on the utilization of internal and external influencing factors, making it possible for the two tasks concerned to work in a collaborative manner.

The implementation details of the dual output layers (shown in Figure 4-1) are as follows. The JPT prediction (regression problem) is task 1 and the PR classification problem is task 2. The inputs from three input modules are concatenated to a feature map with the size of 32×1, which is then fed into both the JPT prediction layer and the PR classification layer for separate training (or prediction/classification). For the JPT prediction layer, the combined feature vector goes through two linearly connected layers with dropout operations, which first transform the vector dimension to 16 and then to 8. Finally, the 8×1 vector is aggregated to a single value, which is the output for JPT prediction. On the other hand, for the PR classification task, the concatenated feature vector goes through two dense layers with dropout operations and is mapped to a 6×1 vector. After a SoftMax operation, the vector is changed to probability corresponding to 6 PR classes. The argmax function is finally applied to the probability vector and derives the class with the highest probability, which serves as the output of the PR output layer. The model is trained iteratively to mitigate the deviation from the predicted JPT and the real value as well as maximize the probability of the correct class that the job belongs to.

4.3 Training of the Proposed Architecture

The proposed architecture adopts a supervised learning training approach. The input samples to the network are denoted as $(x^{(i)}, \widehat{y_1}^{(i)}, \widehat{y_2}^{(i)})$, where $x^{(i)}$ is the feature space of the i-th sample, and $\widehat{y_1}^{(i)}$ (i.e., the JPT) and $\widehat{y_2}^{(i)}$ (i.e., the correct classification of PR) for sample i are two labels obtained under a certain execution circumstance. Parameters of the model are denoted with W, and bias is denoted as b.

The mean-square-error is applied as the loss function to train the task of JPT regression, which is described as:

$$J_1(W, b) = \frac{1}{n}\sum_{i=1}^{n}(y_1^{(i)} - \widehat{y_1}^{(i)})^2$$

where $y_1^{(i)}$ is the output of the first sub-network or the predicted JPT.

On the other hand, the cross-entropy loss (CE loss) is applied to train the multi-classification network, which is given as:

$$J_2(W, b) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{c=1}^{C}\widehat{y_{2(c)}}^{(i)} \log y_{2(c)}^{(i)}$$

where $y_{2(c)}^{(i)}$ is the output of the second sub-network (i.e., the PR classification) of the i-th instance for the classification c, which is calculated as $y_{2(c)}^{(i)} = \frac{\exp(z_{(c)}^{(i)})}{\sum_{j=1}^{C}\exp(z_{(j)}^{(i)})}$.

To efficiently train the joint model, a combined loss function is defined as:

$$J(W, b) = \alpha J_1(W, b) + J_2(W, b)$$

where α is a parameter that controls the contribution of two outputs of the sub-networks. The joint loss is a combination of two terms from the two subtasks, thus the loss of one task can be regarded as a regularization term to the other, which forces the model to find a balance between the two tasks and avoid overfitting to one specific task.

Feedforward propagation. In forward propagation of the MMDT architecture has been detailed in the previous illustration of the three input modules and the dual output layers. It can be generalized as $z^{(l+1)} = W^{(l)}z^{(l)} + b^{(l)}$, where $z^{(l)}$ denotes the output of layer l and the input to the layer $l+1$ as well.

A more critical step of model training lies in the backpropagation process, which tries to mitigate the discrepancy between the prediction value and the label by iteratively updating the parameters of the whole network so that the model can learn the approximation function between the input and the labels.

Backpropagation. After the feedforward propagation, a joint error $J(W,b)$ is obtained from the output units, which can be regarded as a function of parameters W,b. Therefore, the process to optimize the parameters of the network is the process to find the best solution of W,b that can minimize the joint error of the network, which is denoted as:

$$W,b = argmin_{W,b}J(W,b)$$

The error term of the output layer is described as $\delta^{(L)} = \alpha\delta_1 + \delta_2$, where L refers to the index of the final output layer; δ_1 and δ_2 are the backpropagated error terms for loss L_1 and L_2. Through the chain rule, the error terms for previous layers from $l=1$ to $l=L-1$ can be obtained by:

$$\delta^{(l)} = \left(W^{(l+1)}\right)^T\delta^{(l+1)}$$

where $W^{(l+1)}$ denotes the weight matrix of layer $l+1,$ $\delta^{(l+1)}$ is the loss at layer $l+1$. Thus, the gradients of the cost function $J(W,b)$ in terms of parameters W and b at layer l in the t iteration can be obtained by:

$$g_t\left(W^{(l)}\right) = \nabla_{W(l)}J(W,b;x,\widehat{y_1},\widehat{y_2}) = \delta^{(l+1)}(z^{(l)})^T$$

$$g_t\left(b^{(l)}\right) = \nabla_{b(l)}J(W,b;x,\widehat{y_1},\widehat{y_2}) = \delta^{(l+1)}$$

Let (μ, θ) denote the estimate of the first and second moments of the gradients g_t, which can be described as:

$$\mu_t\big(W(l)\big) = \beta_1\big(\mu_{t-1}\big) + (1-\beta_1)g_t\big(W^{(l)}\big)$$

$$\theta_t\big(W(l)\big) = \beta_2(\theta_{t-1}) + (1-\beta_2)g_t\big(W^{(l)}\big)^2$$

and

$$\mu_t\big(b(l)\big) = \beta_1\big(\mu_{t-1}\big) + (1-\beta_1)g_t\big(b^{(l)}\big)$$

$$\theta_t\big(b(l)\big) = \beta_2(\theta_{t-1}) + (1-\beta_2)g_t\big(b^{(l)}\big)^2$$

where $\beta_1, \beta_2 \in [0, 1]$, μ_0 and θ_0 are initialized as zero. Then, the bias-corrected first and second moment estimates are computed with:

$$\hat{\mu}_t\big(W(l)\big) = \frac{\mu_t\big(W(l)\big)}{1-\beta_1}, \hat{\theta}_t\big(W(l)\big) = \frac{\theta_t\big(W(l)\big)}{1-\beta_2}$$
$$\hat{\mu}_t\big(b(l)\big) = \frac{\mu_t\big(b(l)\big)}{1-\beta_1}, \hat{\theta}_t\big(b(l)\big) = \frac{\theta_t\big(b(l)\big)}{1-\beta_2}$$

and the parameters updated using adaptive learning rate with Adam rule is given by:

$$W_t^l = W_{t-1}^l - \frac{\eta}{\sqrt{\hat{\theta}_t\big(W(l)\big) + \epsilon}}\, \hat{\mu}_t\big(W(l)\big)$$

$$b_t^l = b_{t-1}^l - \frac{\eta}{\sqrt{\hat{\theta}_t\big(b(l)\big) + \epsilon}}\hat{\mu}_t\big(b(l)\big)$$

where η is the learning rate or step size and ϵ is a small positive number to avoid dividing by zero. With the Adam optimization algorithm and through a sufficient number of iterations, the loss will be reduced until reaching the stop condition.

With the above parameter updating methods, the training algorithm of the MMDT model is presented in the following:

```
Training of the proposed MMDT architecture
Input:
Training set ($x^{(i)}$, $\widehat{y_1}^{(i)}$, $\widehat{y_2}^{(i)}$), $i=1, ...n$;
Sequence length: $N + 1$
Separate data with sequence length: A set of ($N$ preceding jobs + 1 succeeding job)
Learning rate: $\eta$;
Joint loss function:
$J(W, b) = (-\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} y_{2(k)}^{(i)} \log \frac{\exp(z_{(k)}^{(i)})}{\sum_{j=1}^{K} \exp(z_{(j)}^{(i)})}) + \alpha \frac{1}{n}\sum_{i=1}^{n}(y_1^{(i)} - \widehat{y_1}^{(i)})^2$;
Decay rates: $\beta_1$, $\beta_2$.
Initialization:
Initialize parameters $W$ and $b$;
Initialize: $\mu_0 \leftarrow 0$, and $\theta_0 \leftarrow 0$;
Initialize step: $t \leftarrow 0$.
While $W$, $b$ do not converge:
$t \leftarrow t+1$;
Compute the output $a^{(l+1)}$ at each layer by forward propagation;
Obtain feature maps of three input modules;
Fusion of feature maps from three input modules;
Compute the error term $\delta^{(fl)}$ at the output layer;
Compute the error term $\delta^{(l)}$ at all hidden layers;
Compute the gradients $g_t(W^{(l)})$, $g_t(b^{(l)})$;
Update the $W^{(l)}$, $b^{(l)}$ with the Adam rule;
Until the stop criteria are reached.
Output:
A trained architecture with optimized parameters.
```

4.4 Computational Experiments

4.4.1 Experimental setup

The experimental data is collected from a collaborated printing company in China. The job production data are arranged according to the performed sequence, with a single operator working for a shift (eight hours) and then turning to another operator for the next shift. Printing jobs across two shifts are split into two jobs with corresponding operators and workload (calculated by actual time proportion).

The labels of the two tasks are generated as follows. The total operating time (JPT) of a job is the total time to complete a task, which is computed by:

$$JPT = setup\ time + machine\ processing\ time + human\ operating\ time$$

where the setup time of a job includes the changeover time between two succeeding tasks and preparation time, the second term takes the machine working time into account, and the third term is the time cost for operators performing activities like cleaning the box, changing the die, etc. to make sure the job is carried out smoothly (also taken as the necessary machine downtime during the operating period). These three time periods are recorded by sensors in the heartbeat data form.

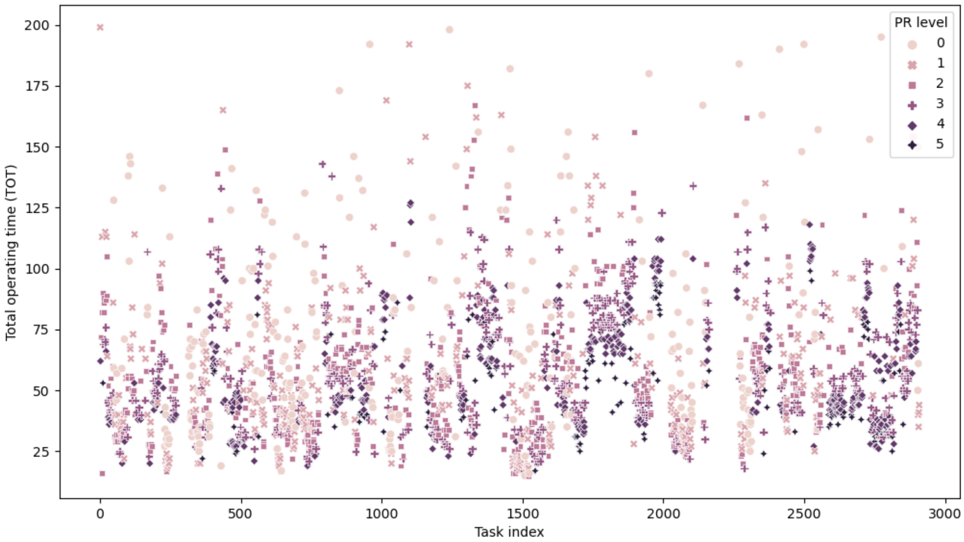


Figure 4-3. Distribution of collected data samples with two labels

The label of processing rate level is computed in two steps. First, the absolute continuous processing rate (CPR) is computed by:

$$CPR = \frac{Output\ quantity}{Job\ processing\ time\ (JPT)}$$

Then, through examining the region that all CPR falls in, the total region is classified into 6 areas that correspond to PR levels from 1-6 according to the distribution of the data. It is found that most CPR falls into the interval of 50-130. To balance the samples of each group and make the classification sensitive to the PR changes, the six groups of PR are set up as: Class 1: (CPR<50, label: 0, num: 322), Class 2: (50≤CPR<70, label: 1, num: 380), Class 3: (70≤CPR<90, label: 2, num: 503), Class 4: (90≤CPR<110, label: 3, num: 546), Class 5: (110≤CPR<130, label: 4, num: 418), Class 6: (CPR≥130, label: 5, num: 157). Figure 4-3 shows the distribution of the collected data samples in terms of two labels. Following conventions of DL literature, we divide the dataset into the training dataset and test dataset with a ratio of 8:2. After data processing, 2096 pieces of job records are obtained for training, and 524 job data records for testing.

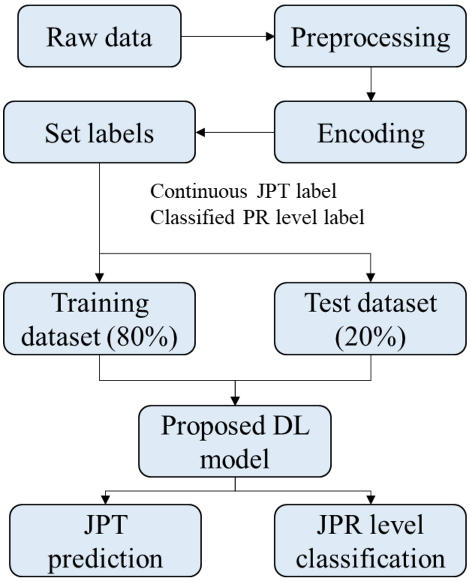


Figure 4-4. Flowchart of the training and testing process

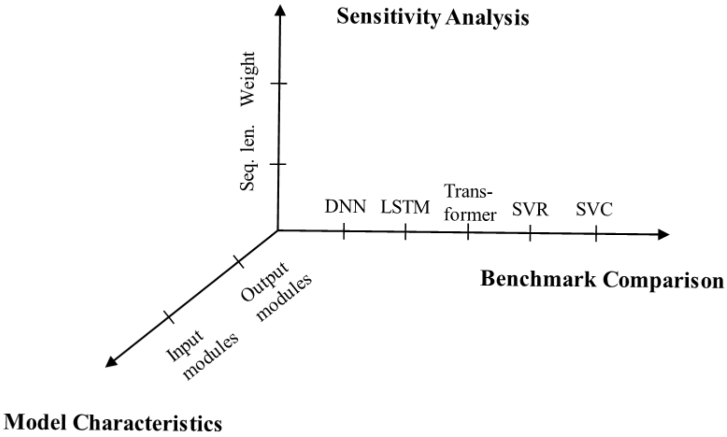


Figure 4-5. Experimental comparisons dimensions

4.4.2 Experimental results and comparisons

The overall flow of the model training and testing is shown in Figure 4-4. A series of computational experiments are conducted to examine the performance of the proposed MMDT in the following three perspectives. First, the effectiveness of components contained by MMDT is tested. More specifically, we validate the effect of incorporating the three input modules (DIM, AIM, and LIM, which capture the information from three levels of relationships) and the effect of simultaneously training for two output layers (dual-task learning). Then, in the second part of the experiments, the proposed MMDT is compared with other benchmarks that have been well established in the machine learning area, including the traditional machine learning models (SVM and SVR), multi-layer neural network models, recurrent neural network models (LSTM model), and the complete transformer model. Lastly, sensitivity analysis is conducted to examine the influences of some important parameters, which involve the weight that controls the contributions of two outputs to the final loss function and the length of the input sequence (which aims to see how changing the number of predecessors will influence the prediction). In addition, for the first and second parts of the experiments, the sequence length is chosen as five (i.e., four predecessors are involved as the preceding sequence to predict the following task). This setting is reasonable because the average processing time for the printing jobs is 61.79 minutes according to historical data. Thus, in general, seven to nine jobs can be completed in one shift. Therefore, a middle number of 4 preceding jobs is selected as the benchmark. Figure 4-5 shows the three dimensions of the following experiments.

Ablation studies of multi-modules and dual-task learning

First, ablation studies are conducted to validate the effect of the three input modules. To do so, the three modules are separately used to predict both the JPR level and the JPT with others blocked (i.e., only involving single DIM (S-DIM), only involving single AIM (S-AIM), and only involving single LIM (S-LIM)). Then, combinations of different input modules are applied to learn both tasks together. Each experiment is trained with the training dataset and then tested on the test dataset in 150 epochs. The results of the final 12 epochs were recorded. Table 4-2 shows the results of JPR regression using only a single input module, including the joint training loss and the MSE and MAPE on the test dataset. It can be seen that model S-DIM cannot achieve satisfactory performance. However, model S-AIM helps reduce the mean squared error (MSE) to 39.512 and the mean absolute percentage error (MAPE) to 7.52%, while model S-LIM achieves a significant reduction of MAPE to 7.13%. These results imply that the JPT is highly affected by the sequential relationship. Therefore, capturing the influence of different preceding jobs in the execution sequence on the JPT of succeeding ones is crucial for improving prediction accuracy.

Table 4-2. Comparisons of solely involving one input module for JPT prediction

	S-DIM			S-AIM			S-LIM		
	Joint Loss	Test MSE	Test MAPE	Joint Loss	Test MSE	Test MAPE	Joint Loss	Test MSE	Test MAPE
exp1	17.475	89.366	13.90%	34.636	41.106	7.60%	16.543	106.617	6.73%
exp2	19.378	99.531	15.00%	34.515	40.649	7.60%	18.731	137.639	6.44%
exp3	17.564	95.409	14.30%	34.385	40.409	7.60%	20.199	124.289	7.29%
exp4	17.681	80.54	13.50%	34.243	40.111	7.50%	42.464	109.839	6.37%
exp5	17.511	90.781	14.20%	34.095	39.589	7.50%	34.124	101.896	8.32%
exp6	18.11	94.453	14.50%	33.939	39.015	7.40%	16.833	110.111	6.70%
exp7	18.666	127.248	17.30%	33.779	38.685	7.40%	25.353	104.598	7.84%
exp8	17.82	156.439	19.70%	33.627	38.639	7.40%	16.395	109.747	6.67%
exp9	15.747	115.9	17.00%	33.484	38.811	7.50%	16.846	112.753	6.69%
exp10	17.517	98.805	15.70%	33.346	39.026	7.50%	21.135	108.240	7.84%
exp11	26.982	87.238	13.90%	33.216	39.103	7.60%	34.124	101.896	7.40%
exp12	20.298	189.053	21.60%	33.087	39.004	7.60%	24.145	98.603	7.29%
AVG	18.729	110.397	15.88%	33.863	39.512	7.52%	23.908	110.519	7.13%

Table 4-3 shows the results of using combined modules on JPT prediction. The ablation studies focus on examining the involvement of sequential influence to original features Thus, the combination of DIM and AIM (C-DIM&AIM, which adds the influence of adjacent preceding job) and combination of DIM and LIM (C-DIM&LIM, which adds the attended information from the performed sequence) are tested. It can be seen that C-DIM&AIM can achieve lower MAPE (7.41%) compared with S-DIM and S-AIM, and a lower MSE (52.790) compared with S-DIM. Similarly, model C-DIM&LIM shows advantages in reducing MAPE to 6.89% and MSE to 41.21. The proposed MMDT facilitates the reduction of MAPE to an average of 6.24%. The ablation studies show that the combined input modules are capable of approximating the inherent relationship within the JPT prediction.

Table 4-3. Comparisons of involving more than one input module for JPT prediction

	C-DIM&AIM			C-DIM&LIM			MMDT (C-DIM&AIM&LIM)		
	Joint Loss	Test MSE	Test MAPE	Joint Loss	Test MSE	Test MAPE	Joint Loss	Test MSE	Test MAPE
exp1	31.373	40.345	7.64%	10.38	41.045	7.50%	9.363	37.705	6.10%
exp2	12.776	54.141	7.30%	12.167	35.415	6.10%	11.429	37.733	5.80%
exp3	11.321	74.568	7.20%	11.208	46.087	7.15%	8.83	36.383	6.60%
exp4	19.052	70.921	7.90%	10.84	52.264	6.90%	9.247	52.727	6.00%
exp5	9.617	78.323	7.30%	10.46	36.321	7.20%	11.743	33.746	5.70%
exp6	10.775	57.399	6.90%	10.89	32.677	6.10%	12.937	36.023	5.90%
exp7	30.868	37.570	7.28%	7.166	38.313	6.90%	12.713	46.019	6.40%
exp8	31.264	39.636	7.55%	9.684	31.218	6.00%	11.56	36.023	5.90%
exp9	30.958	37.955	7.33%	9.471	46.935	7.52%	10.457	46.019	6.40%
exp10	19.844	65.200	7.60%	15.043	37.872	7.10%	11.204	36.383	6.60%
exp11	31.158	38.991	7.47%	10.02	44.06	7.30%	11.232	50.61	6.80%
exp12	31.055	38.428	7.39%	10.84	52.264	6.90%	9.107	54.98	6.70%
AVG	22.505	52.790	7.41%	10.68	**41.21**	6.89%	10.819	42.029	**6.24%**

This study then examines the effect of using the different modules for performing the PR classification task. It should be noted that the PR classification problem is a multi-class classification problem that estimates which degree (1-6) the production efficiency level belongs to. Multi-class classification problems can provide a more detailed view of the production efficiency, while it tends to be more complicated than binary classification in which the decision boundary is relatively easy. In this part, we measure and visualize the performance of PR classification with accuracy, which is calculated by the sum of the number of jobs correctly classified in each class dividing the number of samples in the test dataset. A deeper analysis of classification performance for each category is provided later.

Following the same logic of JPT prediction, the ablation experiments are conducted using only single modules or combined modules. Table 4-4 shows results involving single modules. Then, Table 4-5 shows the results of using combined modules. The results show that only combining DIM and AIM (model C-DIM&AIM) is less helpful, which even disturbs the capability of the model to understand the PR compared with using single modules. In comparison, combining DIM and LIM shows more effectiveness in reducing training loss and test loss and can improve the accuracy to 65.57%. It is noticeable that with the adoption of three input modules, the MMPT model further improves the classification accuracy to an average of 75.77%. The results suggest that the PR classification problem largely benefits from the multi-input-modules structure. The fusion of patterns extracted from different perspectives of the performing sequence helps the model understand the complex PR variations and correctly diagnose the PR level.

Table 4-4. Comparisons of solely involving one input module for PR classification

	S-DIM			S-AIM			S-LIM		
	Joint Loss	Test CEL	Accuracy	Joint Loss	Test CEL	Accuracy	Joint Loss	Test CEL	Accuracy
exp1	17.475	1.171	59.10%	28.155	0.889	61.96%	17.609	1.137	47.60%
exp2	19.378	1.136	63.50%	28.147	0.888	60.87%	18.467	1.17	48.50%
exp3	17.564	0.947	66.10%	28.138	0.884	60.43%	23.211	1.085	52.40%
exp4	17.681	1.053	62.60%	28.116	0.877	60.43%	25.378	1.086	53.70%
exp5	17.511	1.073	63.50%	28.085	0.869	60.65%	11.975	1.064	52.80%
exp6	18.11	1	63.70%	28.030	0.858	62.39%	12.466	1.12	49.30%
exp7	18.666	1.092	58.50%	27.961	0.847	62.83%	32.731	1.004	55.20%
exp8	18.688	1.085	62.00%	27.894	0.868	61.52%	20.055	1.036	57.20%
exp9	17.82	1.151	58.50%	27.846	0.886	58.48%	12.881	0.987	55.00%
exp10	15.747	1.242	54.10%	27.806	0.900	57.39%	17.092	0.954	57.20%
exp11	17.517	1.148	55.70%	27.769	0.907	56.74%	18.026	0.944	61.70%
exp12	26.982	0.999	62.00%	27.735	0.904	56.96%	17.071	1.035	57.40%
AVG	18.595	1.091	60.78%	27.974	0.882	60.05%	18.914	1.052	54.00%

CEL stands for cross-entropy loss.

Table 4-5. Comparisons involving more than one input module for PR classification

	C-DIM&AIM			C-DIM&LIM			MMDT (C-DIM&AIM&LIM)		
	Joint Loss	Test CEL	Accuracy	Joint Loss	Test CEL	Accuracy	Joint Loss	Test CEL	Accuracy
exp1	17.567	1.29	50.70%	10.38	0.903	65.20%	9.363	0.732	74.80%
exp2	12.776	1.033	55.70%	12.167	0.998	63.30%	11.429	0.724	76.30%
exp3	11.321	0.942	57.20%	10.635	0.866	64.80%	8.83	0.736	76.30%
exp4	19.052	0.951	60.70%	10.84	0.996	63.90%	9.247	0.678	77.83%
exp5	9.617	0.919	61.30%	10.46	0.873	67.20%	10.457	0.731	76.10%
exp6	10.775	0.933	63.00%	10.89	0.843	68.00%	11.204	0.766	76.70%
exp7	9.953	1	58.30%	7.166	0.877	64.10%	11.232	0.701	75.20%
exp8	12.382	1.198	56.30%	9.684	0.849	67.00%	9.107	0.758	77.20%
exp9	12.483	1.356	55.00%	11.275	0.882	66.30%	11.059	0.830	73.30%
exp10	19.844	0.886	63.30%	15.043	0.789	68.30%	12.713	0.694	76.09%
exp11	12.382	1.198	56.30%	10.635	0.866	64.80%	18.614	0.748	74.60%
exp12	10.226	1.121	52.60%	10.84	0.996	63.90%	11.56	0.784	74.80%
AVG	13.198	1.069	57.53%	10.835	0.895	65.57%	11.235	**0.740**	**75.77%**

CEL stands for cross-entropy loss.

Table 4-6. Comparisons between single or dual output(s) with the proposed architecture

	Single JPT output		MMDT (JPT)		Single PR output		MMDT (PR)	
	Test MSE	Test MAPE	Test MSE	Test MAPE	Test CEL	Accuracy	Test CEL	Accuracy
exp1	42.038	6.80%	37.705	6.10%	1.354	66.30%	0.732	74.80%
exp2	103.283	10.80%	37.733	5.80%	1.365	65.00%	0.724	76.30%
exp3	53.779	8.50%	36.383	6.60%	1.234	67.20%	0.736	76.30%
exp4	35.857	7.00%	52.727	6.00%	1.404	67.20%	0.678	77.83%
exp5	34.195	6.60%	33.746	5.70%	1.379	67.00%	0.731	76.10%
exp6	53.473	9.90%	36.023	5.90%	1.816	57.80%	0.766	76.70%
exp7	35.783	7.30%	46.019	6.40%	1.521	65.90%	0.701	75.20%
exp8	40.882	7.30%	36.023	5.90%	1.331	66.50%	0.758	77.20%
exp9	45.371	8.30%	46.019	6.40%	1.97	59.80%	0.830	73.30%
exp10	46.009	7.40%	36.383	6.60%	1.59	62.40%	0.694	76.09%
exp11	43.632	8.40%	50.61	6.80%	1.682	64.10%	0.748	74.60%
exp12	43.887	8.60%	54.98	6.70%	1.575	65.20%	0.784	74.80%
AVG	48.182	8.08%	**42.029**	**6.24%**	1.518	64.53%	0.740	**75.77%**

After examining the input modules, ablation studies are further conducted to test the effect of simultaneously training two outputs on the performance. Table 4-6 shows the effect of involving one or two outputs for our proposed architecture. It can be seen that for both tasks, training with the combination of three input modules and only one output shows a significant disadvantage compared with dual-task training. The results demonstrate that the application of the co-learning mechanism enabled by the dual-task training largely benefits a more accurate approximation of both tasks. Also, it implies that knowledge learned from training JPT prediction and PR level classification can also be used to figure out the dependence between the inputs and the other tasks.

<u>Comparisons with other benchmarks</u>

In this section, the proposed model is compared with other state-of-the-art models to further demonstrate the model performance. The benchmark models can be divided into two categories, i.e., models with sequence information and models without sequence information. The former category involves recurrent neural network LSTM and a simplified transformer-based model, which takes the job under prediction and a set of preceding jobs as input so that the dependencies between JPT and PR level on previous jobs can be taken into account. The latter category, DNN and SVR, however, solely captures the JPT and PR level based on the job characteristics as they only take the features of the job under prediction as input. The SVC and SVR models were implemented using the scikit-learn package with default settings, while the other models were developed using PyTorch. We perform a grid search to tune the hyperparameters for models of interest (as shown in Table 4-7). The best parameter combinations of models are summarized in Table 4-8.

Table 4-7. Grid search for model hyperparameters

Model	Hyperparameters	Range
Transformer-based model	Number of encoder layers	[1, 2]
	Number of decoder layers	[1, 2]
	Feedforward dimensions	[128, 64]
	Number of heads	[1, 3]
	Learning rate	[1e-2, 1e-3]
LSTM	Number of hidden units	[16, 32]
	Number of LSTM layers	[1, 2]
	Dropout rate	[0.2, 0.3]
	Learning rate	[1e-2, 1e-3]
DNN	Number of layers	[2, 3, 4, 5]
	Number of neurons on each layer	[64, 32], [32, 62, 32], [128, 64, 32, 16], [128, 64, 32, 16, 8]
	Dropout rate	[0.2, 0.3]
	Learning rate	[1e-2, 1e-3]

MMDT	Concatenating units of DIM	[1, 2, 4, 8, 16]
	Concatenating units of AIM	[1, 2, 4, 8, 16]
	Concatenating units of LIM	[1, 2, 4, 8, 16]
	Number of AIM channels	[16, 32]
	Number of encoder layer	[1, 2]
	Number of multi-heads	3
	2D-kernel size	[[2, 2], [3, 3]]

Table 4-8. Parameter specifications

Model	Parameters specifications
Transformer-based model	Number of encoder layers: 1, Number of decoder layers: 1, Feedforward dimensions: 128, Number of heads: 3, Learning rate: 1e-3
LSTM	Number of layers: 1, Number of hidden units: 32, Dropout rate: 0.2, Learning rate: 1e-3
DNN	Number of layers: 4, Number of neurons on each layer: [128, 64, 32, 16], dropout rate: 0.2, learning rate: 1e-3
MMDT	Concatenating units of DIM: 4, Concatenating units of AIM: 2, Concatenating units of LIM: 2, Number of AIM channels: 16, Number of encoder layer: 1, Number of heads: 3, 2D-kernel size: [3, 3], Learning rate: 1e-3

Table 4-9 summarizes the results of comparisons between different DL models for JPT prediction. In terms of performing the JPT prediction task, models with sequential information perform better than models without sequential information. The transformer-based model plays better than the LSTM model, which reduces MSE to 91.377 and MAPE to 7.62%, While the proposed model demonstrates superiority in reducing both the MSE (42.029) and MAPE (6.24%).

Table 4-9. Comparisons between different DL models for JPT prediction

	SVR		DNN		LSTM		Transformer-based model		Proposed MMDT	
	Test MSE	Test MAPE	Test MSE	Test MAPE	Test MSE	Test MAPE	Test MSE	Test MAPE	Test MSE	Test MAPE
exp1	353.850	12.46%	153.594	12.40%	202.487	8.90%	90.207	8.40%	37.705	6.10%
exp2	267.661	10.37%	197.466	15.90%	203.37	8.80%	89.530	7.94%	37.733	5.80%
exp3	393.653	10.29%	247.566	17.40%	208.633	8.60%	92.539	6.77%	36.383	6.60%
exp4	421.453	14.13%	181.432	13.20%	221.809	8.40%	93.444	7.59%	52.727	6.00%
exp5	620.496	11.02%	226.255	16.00%	197.125	8.90%	92.521	7.21%	33.746	5.70%
exp6	245.331	11.62%	577.445	15.00%	181.41	8.00%	76.820	6.67%	36.023	5.90%
exp7	216.888	10.07%	423.864	17.30%	194.404	7.30%	85.709	7.68%	46.019	6.40%
exp8	463.341	10.08%	368.365	15.00%	167.231	6.60%	84.457	8.07%	36.023	5.90%
exp9	1284.951	11.64%	366.017	20.70%	171.722	9.60%	105.537	7.97%	46.019	6.40%
exp10	887.014	12.29%	326.926	15.90%	213.005	9.10%	95.667	7.22%	36.383	6.60%
exp11	693.183	11.59%	339.478	18.60%	202.974	8.50%	88.303	7.88%	50.61	6.80%
exp12	715.033	11.35%	311.318	18.50%	193.123	8.80%	101.786	8.08%	54.98	6.70%
AVG	546.904	11.41%	309.977	16.33%	196.441	8.46%	91.377	7.62%	**42.029**	**6.24%**

Table 4-10. Comparisons between different models for PR classification

	SVC	DNN		LSTM		Transformer-based model		Proposed MMDT	
	Accuracy	Test CEL	Accuracy	Test CEL	Accuracy	Test CEL	Accuracy	Test CEL	Accuracy
exp1	58.20%	0.676	67.60%	1.611	27.00%	1.83	68.00%	0.732	74.80%
exp2	64.40%	0.685	68.50%	1.633	27.80%	1.631	68.90%	0.724	76.30%
exp3	63.10%	0.683	68.30%	1.649	28.50%	1.649	69.10%	0.736	76.30%
exp4	61.30%	0.676	67.60%	1.664	28.90%	1.671	67.00%	0.678	77.83%
exp5	59.20%	0.663	66.30%	1.678	28.90%	1.845	66.10%	0.731	76.10%
exp6	60.80%	0.646	64.60%	1.694	27.80%	1.65	68.00%	0.766	76.70%
exp7	60.30%	0.62	62.00%	1.718	27.00%	1.845	67.40%	0.701	75.20%
exp8	58.70%	0.602	60.20%	1.751	26.10%	1.758	66.50%	0.758	77.20%
exp9	60.80%	0.617	61.70%	1.792	26.10%	2.052	63.70%	0.830	73.30%
exp10	59.30%	0.667	66.70%	1.83	26.10%	2.214	64.30%	0.694	76.09%
exp11	58.20%	0.654	65.40%	1.858	26.30%	1.735	67.80%	0.748	74.60%
exp12	58.40%	0.626	62.60%	1.874	25.40%	1.979	65.40%	0.784	74.80%
AVG	60.23%	0.651	65.13%	1.729	27.16%	1.822	66.85%	0.740	**75.77%**

CEL stands for cross-entropy loss.

Furthermore, Table 4-10 shows the performance of different DL models in conducting PR-level classification. It can be seen that the proposed MMDT demonstrates significant superiority in PR classification compared with other state-of-the-art models. For other models, similar to the JPT prediction, the transformer-based model performs better and achieves an accuracy of 66.85%. While it is noticeable that still with the sequential information, the LSTM model performs poorly and can be barely trained. DNN is seen as more competitive in performing the PR level than JPT. While it is noted that the DNN model achieves a less cross-entropy loss (0.651) compared with the proposed MMDT, the accuracy of the DNN model is not very high, which suggests the overconfidence of the model in making incorrect classifications.

Sensitivity analysis of important parameters

In this section, the impact of important parameters on the model performance of both tasks is examined. First, we test the effect of changing the weight α that determines the contribution of the two output layers to the joint loss. The default value of α in precious experiments is 1. As the two errors are of different magnitudes, it is desired to identify the best α that facilitates the execution of the two tasks. Specifically, we conduct a series of computational experiments with different values of α = {0.25, 0.5, 0.75, 2, 3}. The results of α = 1 are omitted, as it has been displayed in previous sections. Table 4-11 and Table 4-12 show the effect of variation of weight α on the JPT prediction and PR level classification. Figure 4-6 further plots the changes of (i) MSE and MAPE of JPT prediction and (ii) CE loss and classification accuracy of PR level classification along with varying α.

(1) Effect of the weight in joint loss on the performance

From Table 4-11, it can be seen that when α = 0.75, the MAPE reaches the minimal value of 6.15%. Also, the MSE reaches a minimal value. When α increases to 2, the performance significantly worsens. However, when α = 3, the prediction accuracy increased compared with when α = 2. This result suggests that adding the proportion of MSE loss in the joint loss may slightly benefit JPT prediction performance. However, the performance is worse compared with placing the other loss with proper weight.

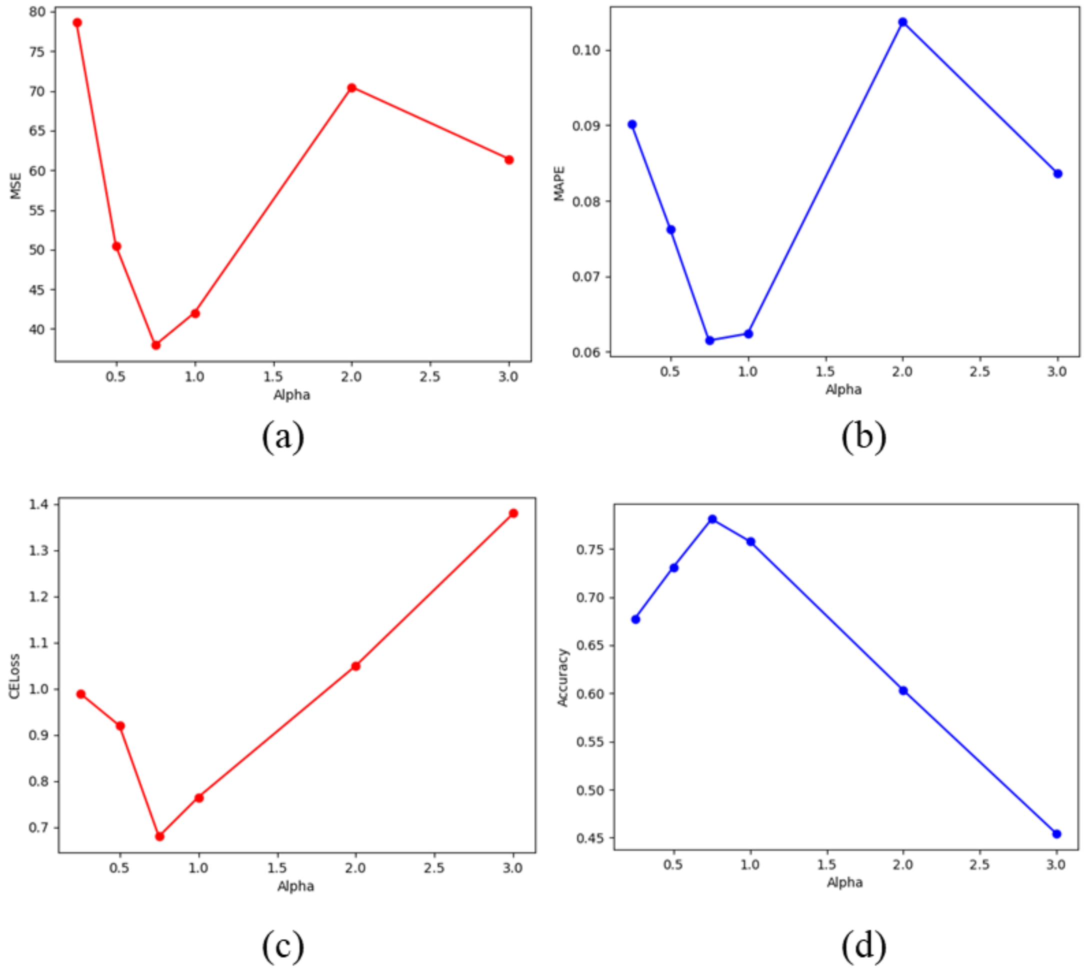
Table 4-12 shows the effect of changing α on the PR classification task. It can be seen that when α takes a small value, the classification accuracy is poor. Along with the increase in weight, the loss is narrowed and the accuracy is greatly improved. Especially, when α = 0.75, the prediction accuracy reaches the highest and when α = 1, the CE loss is reduced to the minimum level. However, when the α further increases (to 2, 3), the prediction performance largely deteriorates. The results suggest that the CE loss should have a weight slightly less than the weight of the MSE loss of the JPT prediction so that the model can achieve superior performance in PR classification. This finding justifies the significance of co-learning. When the JPT loss contribution is too small, the PR classification task cannot effectively learn from the training of the JPT prediction task. However, when the contribution of JPT loss is too large, the learning of PR classification is disturbed. A middle value (e.g., 0.75, 1) benefits both tasks.

Table 4-11. Sensitivity analysis of weight α on JPT prediction

	α = 0.25		α = 0.5		α = 0.75		α = 2		α = 3	
	JPT MSE	JPT MAPE	JPT MSE	JPT MAPE	JPT MSE	JPT MAPE	JPT MSE	JPT MAPE	JPT MSE	JPT MAPE
exp1	66.49	8.37%	70.43	9.95%	33.23	6.41%	59.50	9.35%	57.73	9.17%
exp2	51.68	7.00%	44.54	6.99%	49.33	7.33%	92.56	11.66%	69.46	9.42%
exp3	77.63	10.08%	46.36	7.31%	38.84	7.45%	61.69	10.43%	68.28	7.86%
exp4	64.22	8.38%	39.17	6.68%	35.45	6.12%	60.89	10.35%	53.69	7.47%
exp5	69.71	7.87%	43.40	6.18%	33.83	5.26%	76.45	10.13%	45.61	7.91%
exp6	192.67	13.76%	47.70	7.56%	39.20	6.46%	50.41	8.36%	78.78	9.57%
exp7	71.52	9.18%	65.45	8.32%	35.50	5.19%	100.20	13.00%	67.69	7.87%
exp8	47.85	7.37%	47.68	7.29%	42.84	6.23%	59.50	9.35%	52.35	8.58%
exp9	65.69	9.22%	49.87	8.39%	33.36	4.90%	73.02	10.73%	59.11	7.36%
AVG	78.61	9.02%	50.51	7.63%	**37.95**	**6.15%**	70.47	10.37%	61.41	8.36%

Table 4-12. Sensitivity analysis of weight α on PR classification

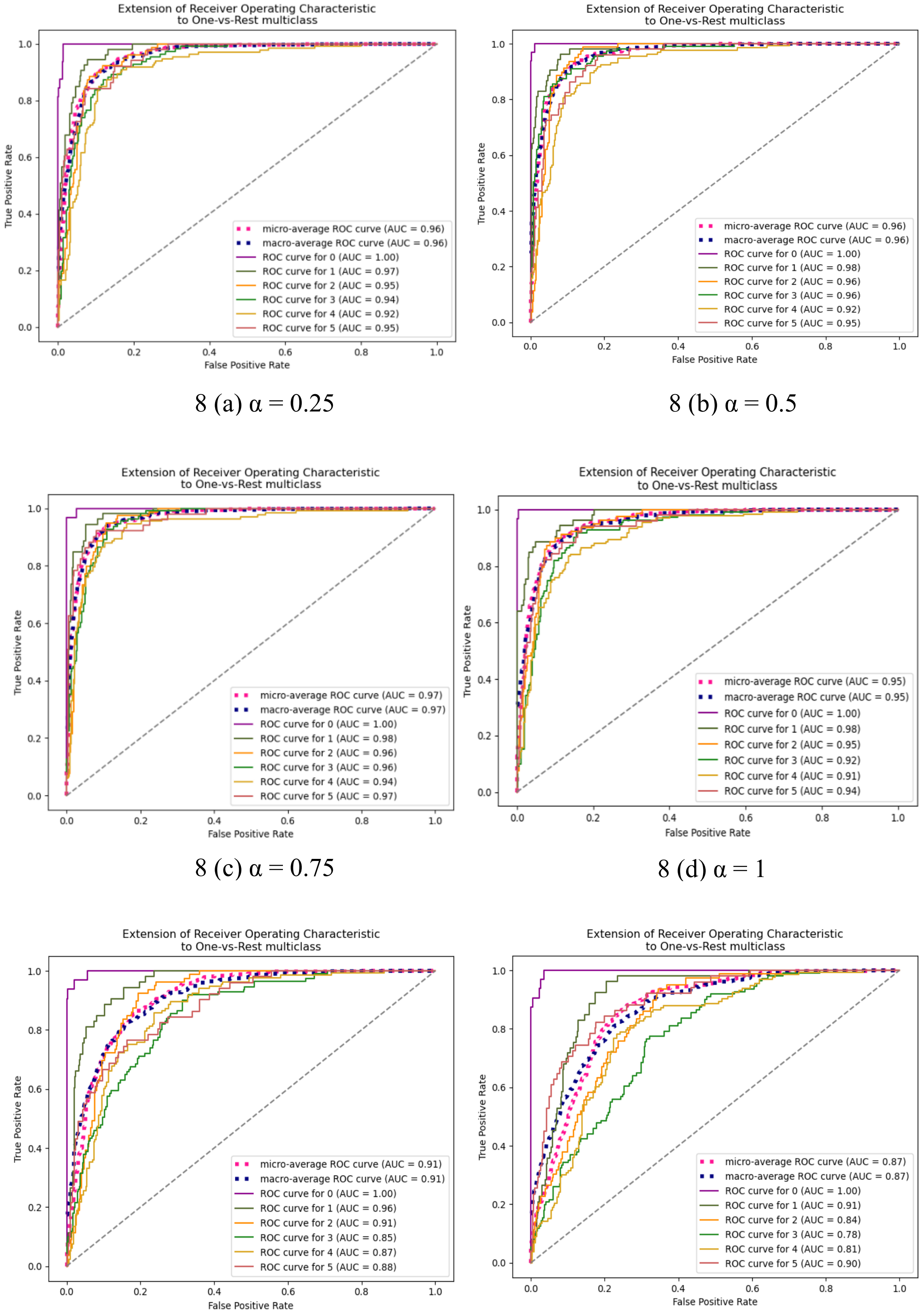
	α = 0.25		α = 0.5		α = 0.75		α = 2		α = 3	
	PR loss	PR accuracy	PR loss	PR accuracy	PR loss	PR accuracy	PR loss	PR accuracy	PR loss	PR accuracy
exp1	0.96	68.04%	0.88	75.00%	0.760	76.74%	1.07	60.43%	1.29	49.35%
exp2	0.77	74.13%	0.77	78.26%	0.860	75.87%	1.02	60.65%	1.28	47.17%
exp3	0.95	69.35%	0.95	72.17%	0.661	78.91%	1.05	58.26%	1.43	42.17%
exp4	0.85	69.57%	1.09	67.61%	0.850	76.09%	1.05	60.00%	1.41	42.83%
exp5	1.08	65.22%	0.97	71.96%	0.702	78.26%	1.09	58.91%	1.36	46.30%
exp6	1.54	54.57%	0.88	73.26%	0.665	79.57%	1.04	61.74%	1.39	45.22%
exp7	1.08	67.17%	0.90	73.26%	0.930	79.57%	1.02	61.30%	1.49	43.04%
exp8	0.79	71.96%	0.77	76.96%	0.642	78.26%	1.07	60.43%	1.37	46.30%
exp9	0.88	69.78%	1.05	69.35%	0.800	79.57%	1.01	61.09%	1.42	46.09%
AVG	0.99	67.75%	0.92	73.09%	**0.763**	**78.09%**	1.05	60.31%	1.38	45.39%



(a) (b)

(c) (d)

Figure 4-6. MSE, MAPE, CE loss, Accuracy changes with varying weight α



8 (a) α = 0.25

8 (b) α = 0.5

8 (c) α = 0.75

8 (d) α = 1

8 (e) α = 2

8 (f) α = 3

Figure 4-7. ROC curves for multiple classes

A deeper look is then paid to the PR classification result of each PR level. ROC curves are generally used to evaluate the performance of binary classifiers, which can demonstrate the trade-off between true-positive rate (TPR) and false-positive rate (FPR). In the ideal situation, the ROC curve will reach the top-left corner, where the TPR is 1 and the FPR is 0. The ROC curves (under the best case in the training epochs) are plotted for all classes involved using the One-vs-Rest strategy (OvR), which is to take one class as the positive class and the remaining classes as the negative class. Also, the micro- and macro-averaged ROC curves are plotted for the entire PR classification task to see the overall classification performance. It can be seen in Figure 4-7 that the macro-averaging ROC curve almost overlaps with the micro-averaging ROC curve, suggesting that our classification task is barely influenced by imbalanced data.

Moreover, from the ROC curves with different α, it can be seen that the best classification performance is reached when α=0.75, with the area under the curve (AUC) reaching 0.97. From Figure 4-7, it is also seen that, among all classes, the best classification performance is achieved for class 0, which is followed by class 1. It is worth noting that the classification accuracy for class 5 is also high in relative, of which the AUC reaches 0.97 when α=0.75 and equals 0.95 when α=0.25. While the classification accuracy for the middle three classes (classes 2, 3, and 4) is relatively weak. These results suggest that it is relatively difficult to distinguish between the three classes with middle PR, as the similarities they share make the determination of the decision boundaries relatively difficult. Nevertheless, the proposed framework can effectively identify and correctly classify the extreme cases that lead to significantly slow or fast PR rates. This finding is important for controlling the production process, as extreme cases generally make large interruptions to the production process, which can be effectively identified by the proposed method.

(2) Effect of sequence length on the performance

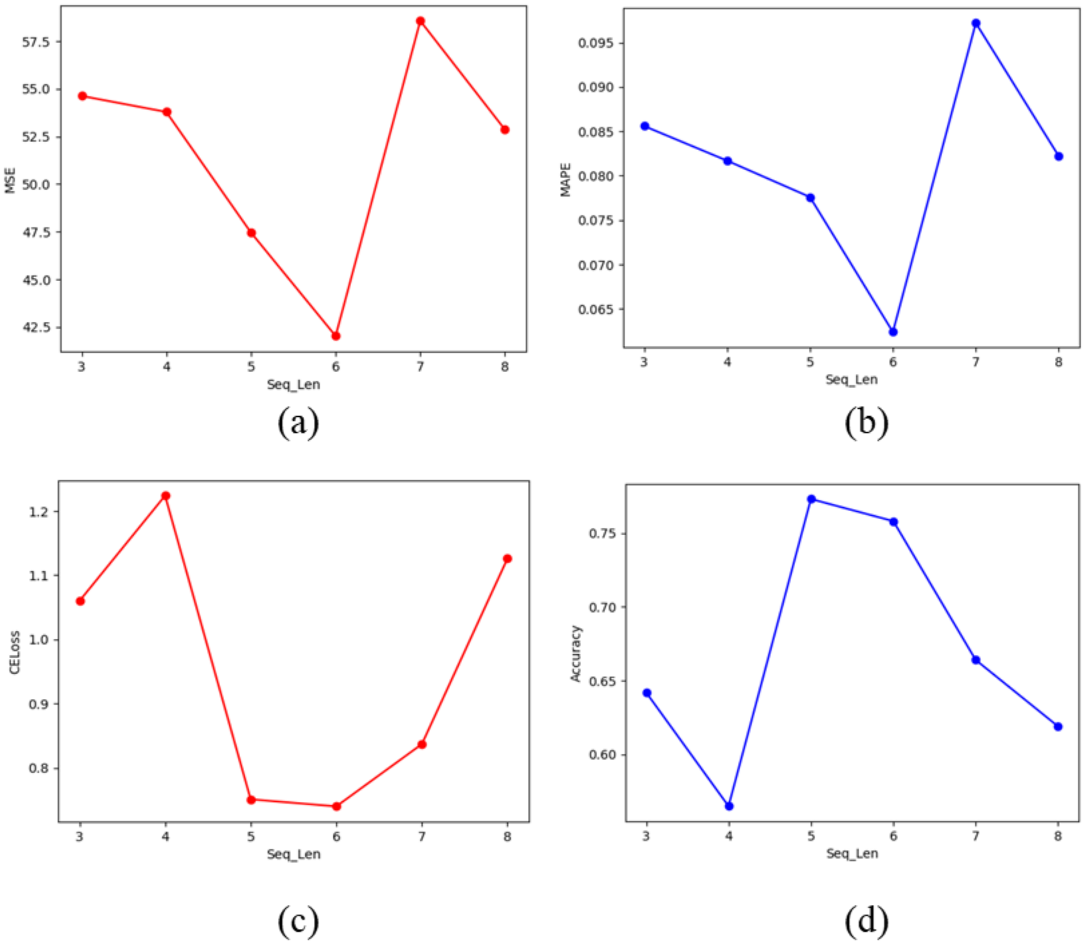
In the last section, the impact of sequence length on the prediction and classification performance is tested. Specifically, the length of involved historical jobs varies from 2 to 7, which means the whole sequence length considered varies from 3 to 8 (including the one under prediction). As involving 5 historical jobs is the preliminary model setting, the results have been shown in the previous sections. As shown in Table 4-13, Table 4-14, and Figure 4-8, it is clear that for the PR classification task, when the sequence length is 5, the model can achieve the best average performance in reducing the CE loss and enhancing the classification accuracy (which can reach 77.29% on average). When the sequence length is too short or too long, the classification performance deteriorates. Then, for JPT prediction, along with the increase of sequence length, the JPT prediction MSE loss and MAPE loss both see a downward trend at first and reach the minimal with the sequence length of 6. With a further increase in the sequence length, the prediction performance worsens. It suggests that the JPT prediction performance and PR classification accuracy are affected by the predecessors. As time goes by, the prior jobs far ahead will have little impact on both indicators.

Table 4-13. Comparisons of different sequence lengths on JPT prediction

	seq_len = 2+1		seq_len = 3+1		seq_len = 4+1		seq_len = 6+1		seq_len = 7+1	
	JPT MSE	JPT MAPE	JPT MSE	JPT MAPE	JPT MSE	JPT MAPE	JPT MSE	JPT MAPE	JPT MSE	JPT MAPE
exp1	73.92	10.23%	62.54	9.48%	45.57	6.36%	53.28	9.20%	44.24	8.00%
exp2	52.49	9.11%	45.81	7.32%	49.34	8.28%	52.48	8.75%	78.13	9.17%
exp3	55.49	8.45%	62.08	8.70%	49.18	9.66%	78.55	11.40%	43.42	7.46%
exp4	37.17	7.69%	50.63	7.95%	50.70	7.40%	45.81	7.86%	45.15	7.86%
exp5	58.65	9.18%	86.93	10.05%	41.51	7.58%	55.58	9.43%	57.33	8.89%
exp6	89.59	9.78%	39.86	7.38%	43.26	8.95%	62.53	9.98%	49.04	8.23%
exp7	43.31	7.98%	55.26	8.12%	43.93	7.90%	56.03	9.25%	55.79	8.24%
exp8	42.60	7.20%	44.57	7.31%	32.53	5.60%	57.59	10.35%	39.09	7.14%
exp9	38.47	7.39%	36.40	7.21%	71.11	8.09%	65.27	11.21%	63.52	9.00%
AVG	54.63	8.56%	53.79	8.17%	**47.46**	**7.76%**	58.57	9.72%	52.86	8.22%

Table 4-14. Comparisons of different sequence lengths on PR classification

	seq_len = 2+1		seq_len = 3+1		seq_len = 4+1		seq_len = 6+1		seq_len = 7+1	
	PR loss	PR accuracy	PR loss	PR accuracy	PR loss	PR accuracy	PR loss	PR accuracy	PR loss	PR accuracy
exp1	0.87	68.04%	1.22	57.61%	0.74	78.04%	0.78	71.30%	1.01	65.22%
exp2	1.03	63.04%	1.28	57.83%	0.77	75.65%	0.82	66.30%	1.13	60.22%
exp3	1.09	65.22%	1.33	48.91%	0.71	78.04%	0.90	62.61%	1.02	65.65%
exp4	1.05	64.13%	1.19	57.17%	0.79	75.22%	0.82	66.52%	1.05	62.83%
exp5	1.02	64.13%	1.17	55.00%	0.69	79.13%	0.89	62.61%	1.20	59.57%
exp6	1.23	61.96%	1.16	58.91%	0.71	78.70%	0.83	64.57%	1.28	58.70%
exp7	1.06	62.17%	1.20	58.26%	0.86	73.70%	0.85	67.17%	1.12	62.17%
exp8	1.14	63.48%	1.23	59.35%	0.76	78.26%	0.79	71.09%	1.13	63.26%
exp9	1.06	65.43%	1.23	55.65%	0.72	78.91%	0.85	65.87%	1.20	59.78%
AVG	1.06	64.18%	1.22	56.52%	**0.75**	**77.29%**	0.84	66.45%	1.13	61.93%



(a)

(b)

(c)

(d)

Figure 4-8. MSE, MAPE, CE loss, Accuracy changes with varying sequence lengths

4.5 Summary

This study is motivated by the operating issues of a real-world production system. The effect of various real-world factors causes discrepancies in the production time of similar tasks. To capture such influences and promote scheduling efficiency, the study proposes to simultaneously predict the job processing time (JPT) and processing rate (PR) level to better understand and capture production status. A multi-module supported dual-task learning model (MMDT) is thus proposed. Extensive computational experiments show that the proposed model can significantly enhance the accuracy of PR classification and JPT prediction compared with other benchmarks. Further ablation studies show that the information captured by different input modules from the three levels (i.e., direct influence, adjacent influence, and sequential influence) is useful in correctly understanding PR variations. Also, both tasks benefit from learning from the representation shared by each other.

Managerial Implications

By implementing the proposed method, a timely alarm/indication of machine status or task-resource alignment degree can be provided to operators. Based on the most recent system information and the predictive indicators, operators can get an idea of when there is an inefficient alignment between the task to be executed and resources allocated to it, (e.g., materials, machines, operators) and even whether environmental factors benefit the execution or not. Therefore, resources can be adjusted promptly ahead of implementation (e.g., changing the task to another machine/operator, or adjusting the environmental elements). Besides, the predictive model also may help to identify useful patterns from historical data. More specifically, it may derive which combinations of parameters can better support a specific type of job. By utilizing this knowledge, wiser resource planning and scheduling decisions can be further developed.

The proposed learning model adopts the co-learning mechanism, which can leverage the synergy between two learning tasks. It is therefore important to identify learning tasks with inherent linkage or knowledge sharing. We conduct sensitivity analyses, which show that the weight that controls the proportion or contributions of the two loss functions in the joint loss function plays a vital role in model performance, which should be carefully chosen. The experiments conducted suggest 0.75 or 1 to be a good choice.

Moreover, it is shown that the sequence length involved will affect the prediction accuracy for both learning tasks. However, the results also suggest that even though the sequential effects should be considered, they can have negative effects if the sequence of prior work involved is too long. Experiments of both tasks suggest that involving too many prior jobs (over 5) will increase training difficulties or even impair the prediction and classification accuracy.

Through ROC curves of separate PR levels, it is discovered that middle-class tasks can be regarded as fluctuating around the average value. Determining decision boundaries for these classes is relatively difficult. However, the proposed MMDT can very effectively identify extreme cases when PR is very high or low. Since extreme cases with significant deviations from the average level have a more significant disturbance or impact on normal production, it will be very helpful to apply the proposed methods to achieve a timely detection of abnormal situations.

Chapter 5. Context-based PR Guided Scheduling[17]

The investigated job scheduling problem is based on the real demand of a major printing company in China. Historical data were collected from different departments of the company, which can be categorized into information related to engineering, order, material inspection, technician, production, and processing environment (Details are presented in Chapter 4). As concerned in Chapter 1, the JPR of the same job may vary under different contexts that are defined by multiple job-specified factors and the position in the job sequence. Thus, a major aim of this study is to derive practical and efficient scheduling approaches able to flexibly deal with variabilities in the processing context.

5.1 Problem Explanation and Preliminary Model[18]

The scheduling of jobs can be extracted as a parallel machine scheduling problem with a set of jobs (printing jobs) $J = \{1, 2, \dots |J|\}$ to be assigned and sequenced on a set of identical machines $M = \{1, 2, \dots |M|\}$. Each machine m is operated by an operator o belonging to $O = \{1, 2, \dots |O|\}$. Operators work either a daily shift or a night shift, with each shift being 8 hours (any job that spans two shifts is split into two jobs performed by different operators). Our task is to assign jobs to operators in a proper sequence to minimize the total completion time of processing all jobs. As it is a common assignment problem, the preliminary model can be easily formulated using a position-based modelling approach.

[17] Most part of this chapter is included in Sun, Y., Chung, S.H., Choi, T.M., & Wang, Y. (2024). Feature-Driven Production Scheduling Systems: Unveiling and Exploiting Job Processing Rate Dependencies. *IEEE Transactions on Systems, Man and Cybernetics: Systems*, under review.

[18] As a remark, meanings of the notations appearing in Chapter 5 below are only applicable to this chapter.

(PM)

$$Minimize \sum_{k\in K}\sum_{o\in O}\sum_{i\in J} p_{i,o,k}\cdot x_{i,o,k} \tag{1}$$

$$\sum_{o\in O}\sum_{k\in K} x_{i,o,k} = 1,\ \ \forall i\in J \tag{2}$$

$$\sum_{i\in J} x_{i,o,k} \leq 1,\ \ \forall k\in K, o\in O \tag{3}$$

$$\sum_{i\in J}\sum_{k\in K} p_{i,o,k}\cdot x_{i,o,k} \leq UBL,\quad \forall j\in O \tag{4}$$

$$x_{i,o,k}\in\{0,1\} \tag{5}$$

The parameter $p_{i,o,k}$ is the processing time of job i to be performed by operator o (on the corresponding machine) in position k of the processing sequence. The decision variable $x_{i,o,k}$ denotes whether a job i is allocated to operator o at the execution position k. Constraints (2) ensure that each job is to be allocated to one operator in one position. Constraints (3) force that at most one job could be allocated to a position for each operator. Constraints (4) ensure that the upper bound (UBL) of a working period for an operator is not violated. Constraints (5) force $x_{i,o,k}$ to be binary.

Model ***PM*** provides a deterministic assignment model that allocates jobs to machines. However, as the JPR (and processing time) of job i varies according to the specific execution contexts this basic model cannot capture such variations unless all possible scenarios are enumerated to obtain parameters $p_{i,o,k}$ under each circumstance. It is obviously unrealistic and inefficient due to the high computational overhead.

5.2 Proposed Solution Architecture

To effectively capture the influences of multiple variants under varying execution contexts and avoid the above computational challenge, this study proposes a context-based branch-and-price heuristic approach with a four-tier solution architecture (as shown in Figure 5-1). Such a framework can incorporate the multi-factor effect in the scheduling process (*prediction layer*) and enable the use of context-based JPR to guide the schedule generation process (*optimization layers*). It should be noted that different from conventional methods that treat parameter prediction and optimization as two independent steps (i.e., predict parameters first and feed them into optimization models), the proposed framework novelly integrates both. More specifically, the optimization layers provide the prediction layer with promising context information, and meanwhile, the CBPR derived by the prediction layer guides the optimization algorithm to further explore promising schedule solutions. The synergy between these two components enables both the effectiveness and efficiency of the proposed solution architecture.

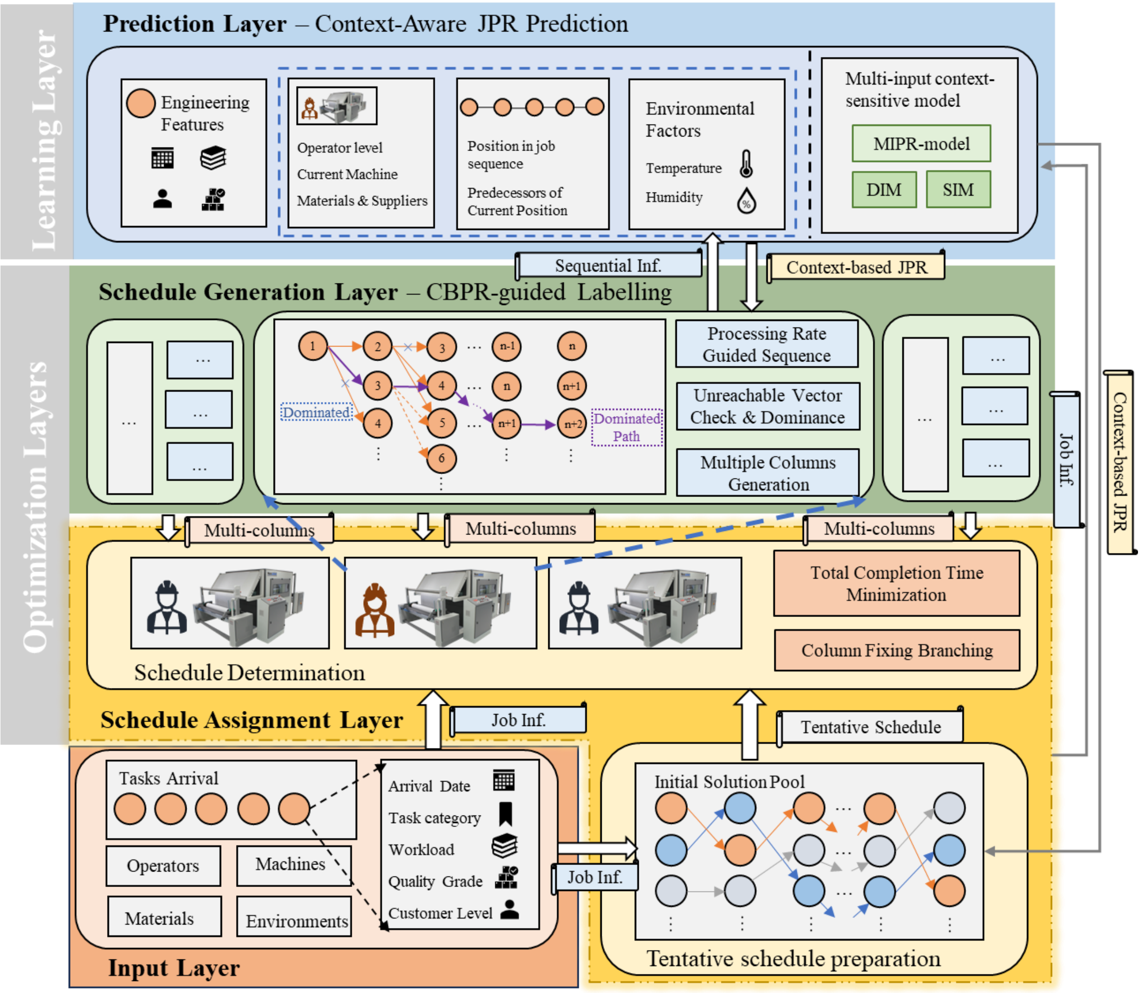


Figure 5-1. The four-tier solution architecture

More specifically,

- ***Input layer*** (Detailed in *Section 5.2.1*) – receives and preprocesses a set of jobs (with job-related features) to be scheduled and passes the job information to *Schedule Assignment layer*.
- ***Prediction layer*** (Detailed in *Section 5.2.2*) - captures CBPR with a proposed DeepPR model for *Schedule Generation layer*. After receiving the features passed from *Schedule Generation layer*, it supplements job-specified features (e.g., engineering requirements, materials) to form the complete execution context.
- ***Optimization layers*** (Detailed in *Section 5.2.3-5.2.5*) - The optimization layers constitute a *Schedule Assignment layer* and a *Schedule Generation layer*, which are created based on the logic of column generation.
    - ***Schedule Assignment layer*** - determines which schedule received from the *generation layer* (including initial tentative schedules) should be adopted for each operator-machine pair. The strategy of creating tentative schedules (initial solution pool) both ensures feasibility and accelerates the computation.
    - ***Schedule Generation layer*** - iteratively produces promising schedules for individual operators with a proposed JPR-guided labelling algorithm enabled by the interaction with the *prediction layer*. More specifically, each time to decide whether a job should be included in the current schedule (i.e., the extension process), it provides information about the current operator/machine- and sequence-related information to the *prediction layer* for CBPR prediction.

5.2.1 Input layer: Feature extraction for CBPR prediction

From the historical records from sensors and ERP systems, enriched context-related information can be obtained, including detailed information for processing starting/ending time, engineering requirements, orders, machine processing parameters, materials, technical operator in charge, environments, and preceding jobs on the same machine (operator). Such abundant information provides a solid foundation for JPR prediction, while challenges arise in identifying available and critical features for scheduling and handling real-world data of high dimensions and multiple types (Gao et al., 2021).

First, data cleaning was performed by visualizing outliers and deleting jobs with maximum or minimum values for each float-type area. Forms from heterogeneous sources, like production data (recorded by machines for each job) and environmental data (by sensors for each time step) were merged. One-hot encoding and ordinal encoding were applied for categorical and string-type columns, which unfortunately further increases the data dimension.

Although feature reduction techniques like PCA-related methods are useful for dimension reduction, they may damage the internal structure of features (Gao et al., 2021). Besides, considering our purpose to combine JPR prediction into the scheduling process, identifying representative and compact features benefits prediction efficiency. Thus, several feature engineering strategies are applied for feature extraction:

- (*Manual aggregation*): using aggregated features to represent several pieces of information. For instance, the *material batch number* is used to involve quality-related specifications; the *number of setup works* is used to simplify encoding strings that record setup works.
- (*Statistical correlation analysis*): statistically calculating the correlation between features and labels and eliminating those with poor correlations.
- (*Noise reduction*): eliminating columns showing poor distribution consistency between training samples and test samples.

After applying the above strategies, the following informative features are extracted to describe the context of a job from the job-specified aspect (*Section B* will further introduce the sequence-based aspect). Furthermore, we distinguish these features into two categories: static features and dynamic features. The static features are those predetermined with the printing jobs, while the dynamic features are obtained during the schedule generation process.

Context Description from Job-specified Aspect	
Internal features	**External features**
Engineering: Job Category (*tc*); Quantity/Workload (*w*); Customer Category (*cc*); Quality Grade (*qg*) **Materials**: Material Batch Number (*mbn*); Supplier (*ms*); Quantity (*mq*); Weight (*mw*); Length (*sl*); Width (*sw*)	**Operating:** Operator Grade (*opg*); Machine speed (*mss*); Daily/Night shift (*s*); No. Setup jobs **Environment**: Temperature (*et*); Humidity (*eh*)

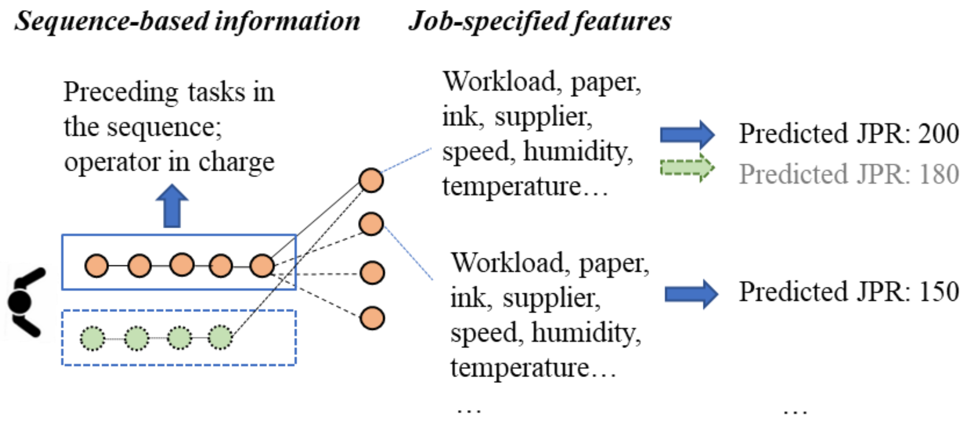

Sequence-based information
Job-specified features
Preceding tasks in the sequence; operator in charge
Workload, paper, ink, supplier, speed, humidity, temperature…
Predicted JPR: 200
Predicted JPR: 180
Workload, paper, ink, supplier, speed, humidity, temperature…
Predicted JPR: 150
…
…

Figure 5-2. An illustrative example of JPR prediction under different contexts

5.2.2 Prediction layer: DeepPR model for CBPR prediction

In practice, prior jobs can have a significant effect on the processing time of the succeeding job(s) in terms of setups and changeover time (e.g., material/ink swapping), operator status, and even machine performance and downtime. Thus, besides the job-specified features (specified in *Section 5.2.1*), JPR may also depend on its preceding jobs. Thereby, we consider the CBPR determined by two aspects as shown in Figure 5-2: *job-specified features* and *sequential-based information* that embeds a series of preceding jobs executed by the same operator on the same machine.

To capture variabilities in JPR under dynamic processing contexts, a learning model capable of extracting the dependencies between both aspects is needed. A DeepPR model is proposed to perform a regression task that predicts the JPR with context-based features (i.e., CBPR). As shown in Figure 5-3, the model contains two input modules, which are designed for extracting features from both job-specified and sequence-based levels. The *direct influence module* (DIM) takes job-specified features of job *k* with the size of *1×Dim_input* (where *Dim_input* is the dimension of the features of job *k*) as input. It comprises several densely connected layers with ReLU activation function and dropout operations so that the direct influences from multiple input features can be extracted. The *sequential influence module* (SIM) extracts the inner pattern from the performed job sequence. It adopts a transformer encoder layer to embed a set of *N* preceding jobs into a *1×N×Dim_input* matrix.

To uncover information from the embedded matrix, two attention methods are applied in parallel, i.e., channel-wise attention and job-wise attention, to extract influences from the preceding jobs by weighing both the importance of features and the importance of each preceding job. The channel-wise attention component first uses a 1D-convolution layer at the feature dimension to transform the influence of all features into a vector with *1×Dim_input* dimension. A sigmoid function is then applied to transform the obtained vector into attention scores, which correspond to the weight of features implied by the predecessors involved. The scores are then applied to the original features. The obtained weighted feature vector is added to the original feature vector of the DIM so that the obtained feature map contains the sequence-based featurewise effect. On the other hand, a simplified transformer decoder with attention and feedforward operations is applied to capture job-wise attention. By comparing the features of job *k* with the embedded *N* preceding jobs, the effect of each single prior job on job *k* can be acquired.

The latent features derived by the dual input modules DIM and SIM are then fused by concatenation. In Figure 5-3, the output features of both modules $\in\mathbb{R}^8$, while it is worth noting that the contribution of the dual input modules can be adjusted by changing the size of their output features. Finally, the CBPR is derived after several dense layers and dropout operations.

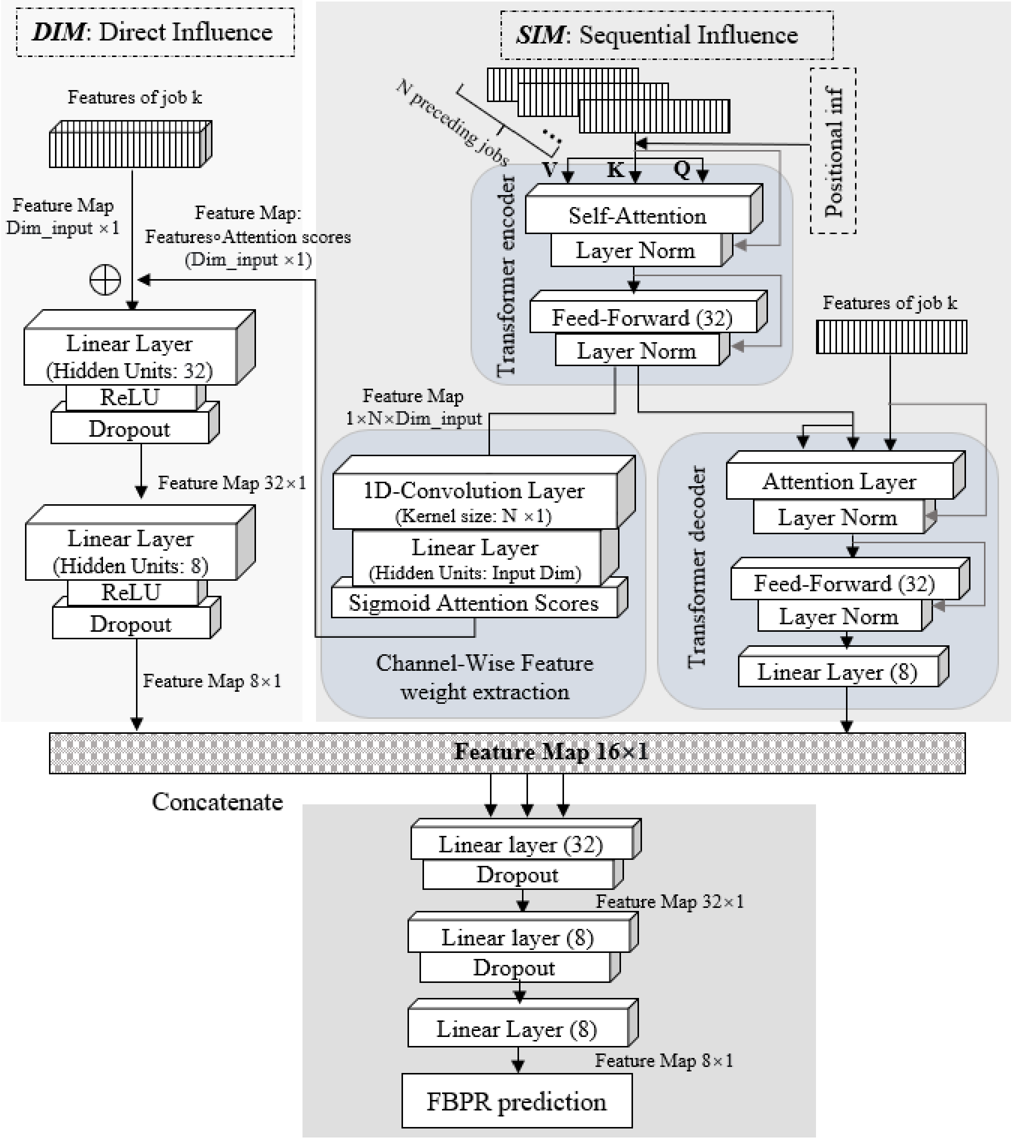


Figure 5-3. DeepPR architecture

Section 5.3.1 compares the model performance with other state-of-the-art deep learning models, namely, transformer, LSTM, TCN, and DNN. Through extensive experiments, it is found that DNN models cannot well capture JPR variabilities without knowing positioning information. However, the JPR prediction accuracy can be largely improved when knowledge of preceding jobs is available. Therefore, sensitivity analysis regarding the length of time sequence is further carried out. Results show that the JPR prediction performance can be largely improved when the prediction window includes only two adjacent jobs, indicating that the JPR of one job is highly related to its immediate predecessor due to the involved changeover and status transition. Even though involving more predecessors in prediction may slightly improve the accuracy, it will significantly increase the computational burden. To balance the computational consumption and model performance, the processing context is defined by the job-related features and features of the immediate predecessor and the DeepPR is used to implement further forecasting-embedded pricing.

5.2.3 Optimization layers – Model reformulation and decomposition

Following the problem setting in Section 5.1, the investigated problem aims to schedule a set of $|I|$ jobs to $|O|$ operators and each operator is in charge of one machine. The notations involved are presented in Table 5-1.

Table 5-1. Notations

Symbol	Meaning
I	Set of printing jobs to be processed
$\|I\|$	Number of printing jobs
S	Set of schedules added in the restricted master problem
s	Index of a schedule

S_o	Set of schedules belonging to operator o
i, j	Index of printing jobs
O	Set of operators
$\|O\|$	Number of operators
w_i	Workload/output quantity of job i
$a_{i,s}$	Binary, which takes value 1 if job i is included in schedule s, 0
$PR(s(i))$	Processing rate derived under the certain processing context of job i
c_s	Cost (completion time) of the schedule s
$C_i(s)$	Completion time of job i in schedule s
y_s^o	Binary variable, 1 if schedule s of operator o is adopted in the

As the operations of operators are independent of each other, it is natural to form job sequences for individual operators and later select the best solution that minimizes the total completion time required to complete the entire job set, i.e., one schedule for an operator/machine. Accordingly, the schedule-based model is formulated as below.

(RM)

$$Minimize \sum_{s \in S} c_s \cdot y_s^o \qquad (6)$$

$$s.t. \quad \sum_{o \in O} \sum_{s \in S_o} a_{i,s} \cdot y_s^o \geq 1, \;\; \forall i \in J \qquad (7)$$

$$\sum_{s \in S_o} y_s^o \leq 1, \;\; \forall \mathrm{o} \in O \qquad (8)$$

$$\sum_{i \in I} a_{i,s} \cdot \frac{w_i}{PR(s(i))} \leq UBL \qquad (9)$$

$$c_s = \sum_{i=1}^{|I|} C_i(s) = \sum_{i=1}^{|I|} a_{i,s} \cdot \frac{w_i}{PR(s(i))} \qquad (10)$$

$$y_s^o \in \{0, 1\}. \qquad (11)$$

Constraints (7) ensure that every job should be covered. Constraints (8) require each operator to be allocated with at most one schedule. Constraints (9) is transformed from Constraints (4), indicating that the sum of processing time (determined by CBPR) of all jobs involved in a schedule should be less than the upper bound limit. Constraints (10) obtain the completion time or cost of a schedule by summing up the predicted processing time (at the corresponding positions in the schedule) of the jobs covered by the schedule. Constraints (11) are variable-type restrictions.

To solve the (***RM***), it is decomposed into a restricted master problem (***RMP***) and a pricing subproblem (***PS***) following the logic of column generation. The decomposed models are solved at two layers respectively:

(*Schedule Assignment layer*): iteratively solves a restricted problem that determines the allocation of schedules to operators. The schedules involved are continuously updated by solving problems at *Schedule Generation layer*.

The ***RMP*** is stated as:

$$(\boldsymbol{RMP}) \quad Objective\ (6)$$

$$\boldsymbol{s.t.} \quad Constraints\ (7), (8), (10)$$

$$y_s^o \geq 0, \ \ \forall \mathrm{o} \in O, s \in S \qquad (12)$$

Let u_i and v_o denote dual values associated with Constraints (7) and (8) respectively. The RMP reaches optimal if the reduced cost of any additional schedule s satisfies:

$$c_s - \sum_{s \in S_o} a_{i,s} \cdot u_i - v_o \geq 0, \forall \mathrm{s} \in S_o, o \in O$$

(*Schedule Generation layer*): solves a set of ($\boldsymbol{PS(o)}$) to generate promising schedules with negative reduced costs for each operator o, which are added to the master problem for improving the master solution. When no columns with negative reduced cost can be found, the master problem reaches optimality. The pricing subproblem is formulated as:

$$(\boldsymbol{PS}(\boldsymbol{o}))\quad Minimize\;\; c_s - \sum_{s\in S_o} a_{i,s}\cdot u_i - v_o$$

$$s.t.\quad Constraints\ (9)$$

Constraints (9) can be satisfied in pricing at the generation layer (detailed in *Section D*). If the optimal objective value is less than zero, the corresponding schedule s derived for operator o can be added to the schedule pool S_o of the operator o. The pricing subproblem terminates when no columns with negative reduced cost can be found for all operators.

5.2.4 Schedule generation layer – CBPR-guided pricing

The schedules are generated with the synergy between *Schedule generation layer* and *the prediction layer*. CBPR is dynamically predicted along with schedule generation to identify promising schedules for individual operators. Note that the subproblem PS(o) can be viewed as a 0-1 knapsack model, but the traditional dynamic programming method is computationally challenging because of the non-deterministic nature of job processing time and the demand for scheduling accuracy. We thus transform the subproblem PS(o) into an elementary shortest path problem with resource constraint (ESPPRC). A tailored CBPR-labelling algorithm is proposed to generate columns with negative reduced costs using dual information (u_i and v_o). Those columns are added to the master problem to improve the objective.

By assigning printing jobs with increasing ID (according to the received timeline) and taking them as nodes within a network (connected in an ID-increasing direction), the labelling algorithm explores possible extensions to the next eligible jobs (rendering the total processing time to stay within the time limit/upper bound of work length). Moreover, we define the following label for a partial path α currently extended to node i (i is the latest extended node in the path α) as: L_i^α = $[i, \alpha,\ r^\alpha,\ v^\alpha,\ D^\alpha = \{d_1^\alpha, d_2^\alpha, \dots\ d_{|I|}^\alpha\}, p(\alpha)]$, which is used to track *current node* i; *ID of current path* α; *accumulated resource consumption* r^α by adding up all predicted processing time of selected jobs in α (i.e., $\sum_{j \in I_\alpha} \frac{w_j}{PR(\alpha \to j)}$, where I_α is the set of selected jobs in α, $PR(\alpha \to j)$ denotes the predicted JPR by placing job j as the next processing job in path α and w_j denotes the workload of job *j*); *accumulated value* v^α (calculated by adding up all the reduced arc costs $\sum_{j \in I_\alpha} (\frac{w_j}{PR(\alpha \to j)} - u_j)$); *vector recording node eligibility* D^α (consisting of elements d_k^α); and *ID of prior path* $P(\alpha)$ from which the α is extended (to derive the entire schedule in a backward direction).

Before presenting the CBPR-labelling algorithm, we introduce the proposed *node-checking step* and *dominance rule*, which play significant roles in accelerating the pricing process. First, after each node extension, the following *node-checking step* is performed to update the reserved D^α, J_{VN} is the set of nodes that have been visited.

$$d_j = \begin{cases} 1, & for\ r^\alpha + \dfrac{w_j}{PR(\alpha \to j)} \le UBL \\ 0, & for\ j \in J_{VN}\ \boldsymbol{OR}\ r^\alpha + \dfrac{w_j}{PR(\alpha \to j)} > UBL \end{cases}$$

After checking for each node, the vector D^α records the remaining nodes (of value 1) that are eligible to be further visited by path α. Different from the generic ESPPRC problem, we consider generating schedules in a node ID-increasing direction. Besides, even though JPR fluctuates with changing processing contexts, it mainly depends on the immediate predecessor and job-specified features. Consequently, tailored dominance operations can be developed and implemented between partial paths ending at the same node as the succeeding jobs only depend on the effect of the same preceding node. Proposition 1 provides the dominance rule. We prove it holds under different possible circumstances so as to ensure a safe elimination of unpromising paths/partial schedules, which enhances the solution efficiency of the CBPR-labelling algorithm.

Proposition 5.1. *For two paths (partial schedules)* α *and* β *ended at node* i*,* β *is dominated by* α *if: (i)* $v^{\alpha} \leq v^{\beta}$ *AND* $r^{\alpha} \leq r^{\beta}$ *with at least one strict inequality, or (ii)* $v^{\alpha} < v^{\beta}$*,* $D^{\alpha} = D^{\beta}$*, AND* $r^{\alpha} + \sum_{j \in D^{\beta}} \frac{w_j}{PR(\alpha \to N^j \to j)} < UBL$*, where* D^{β} *is the set of eligible nodes that can be extended after node* i *and* $N^j (\subseteq D^{\beta})$ *denotes the set of eligible nodes between nodes* i *and* j*.*

Proof. Conditions (i) are obtained by Pareto optimality. As $r^{\alpha} \leq r^{\beta}$ implies $D^{\alpha} \leq D^{\beta}$, comparisons of D are not needed. Conditions (ii) enhance dominance by removing more paths when $v^{\alpha} < v^{\beta}$ but $r^{\alpha} > r^{\beta}$ ($D^{\alpha} = D^{\beta}$ implies the number of remaining unexplored nodes is equal for α and β). We demonstrate that conditions (ii) hold.

Case I. Extending all $j \in D^{\beta}$ benefits the solution.

The accumulated value of the partial path α increases $(\frac{w_j}{PR(\alpha \to j)} - u_j)$ by extending node j. As $r^{\alpha} + \sum_{j \in D^{\beta}} \frac{w_j}{PR(\alpha \to N^j \to j)} < UBL$ guarantees that path α can legally incorporate all remaining nodes in path β, the final accumulated value for path β follow $v^{\alpha} + \sum_{j \in D^{\beta}} (\frac{w_j}{PR(\alpha \to j)} - u_j) < v^{\beta} + \sum_{j \in D^{\beta}} (\frac{w_j}{PR(\alpha \to j)} - u_j)$, and thus β can be dominated by α.

Case II. Extending only partial $j \in D_1^{\beta}$ ($D_1^{\beta} \subset D^{\beta}$) benefits the solution, which may result in JPR changes due to positioning variability (we only prove cases where JPR uniformly increasing or decreasing, while the mixed scenario can be easily extended).

<u>Case II(a)</u>. JPRs are enlarged due to positioning variability. The enlarged JPR results in decreased processing time. This guarantees $r^{\alpha} + \sum_{j \in D_1^{\beta}} \frac{w_j}{PR(\alpha \to N^j \to j)} < UBL$, which implies the extension is feasible for path α. The dominance rule can cover this situation and the proof is easily derived similar to case I by replacing $j \in D^{\beta}$ with $j \in D_1^{\beta}$.

Thus, β can be dominated by α.

<u>Case II(b)</u>. JPR decreases due to positioning variability, which may cause the infeasibility of path α to achieve such an extension. However, this case requires $\sum_{j\in D_1^{\beta}}\frac{w_j}{PR(\alpha\to N^j\to j)} > \sum_{j\in D^{\beta}}\frac{w_j}{PR(\alpha\to N^j\to j)}$. Also, from the duality theory, $u_j \geq 0\ (j\in D^{\beta})$. Thereby, the accumulated value of such an extension of path β is $v^{\beta} + \sum_{j\in D_1^{\beta}}(\frac{w_j}{PR(\alpha\to j)} - u_j) > v^{\beta} + \sum_{j\in D^{\beta}}\left(\frac{w_j}{PR(\alpha\to j)}\right) - \sum_{j\in D_1^{\beta}} u_j \geq v^{\beta} + \sum_{j\in D^{\beta}}(\frac{w_j}{PR(\alpha\to j)} - u_j)$. Thus, the extended β can be dominated following Case I. (Q.E.D)

Remark. The above conditions imply that any extension of path β after node i until the final node can be replicated for path α and path α can achieve a smaller reduced cost. Moreover, the algorithm prefers positioning jobs in a JPR-increasing direction. However, as comparisons of the last condition bring extra computational burden and are less effective when exploring nodes far ahead of the destination, we can adjust when to involve Conditions (ii) in implementation.

By integrating the node-checking step and dominance checking for acceleration (Proposition 1), the CBPR-labelling algorithm is as follows.

Proposed CBPR-Labelling Algorithm

```
Input: Set of jobs I, Dual values U, transformer model, UBL
Initialization: O-label=[s, 0, 0, 0, [1, 1, …1],    NONE]
SET T_LIST = [I]    # the array of set of jobs to schedule
WHILE T_LIST ≠ Ø DO
    CHOOSE the first node i in T_LIST and GET all non-dominated paths at the node $P_i$.
    FOR path $\alpha$ in $P_i$ DO
        {* Path extension step *}
        GET new label $\beta$ extended to a new node (suppose j)
        {* Node checking step *}
        FORALL unvisited nodes $k\in J_{UN}$ GET $w_j/PR(\alpha\to k)$ and UPDATE $D^{\beta}$
        GET all non-dominated paths at node j, $P_j$ and ADD $\beta$ to $P_j$
```

```
{* Dominance checking step *}
FOR path $\sigma$ in $P_j$ APPLY dominance rules
  IF $\sigma$ is dominated by $\beta$, THEN REMOVE $\sigma$
  ELSE IF $\beta$ is dominated by $\sigma$, THEN REMOVE $\beta$
```

5.2.5 Schedule assignment layer - Acceleration strategies

Several acceleration strategies are employed to enhance algorithm implementation. First, following the literature, multiple columns (obtained from pricing) are returned to the master problem instead of only the one with the most negative reduced cost. Second, due to the shortage of columns in the first iterations, the dual values provided can barely reflect the real situation. An initial solution pool is established with a set of feasible schedules (*Preparation layer* in Fig. 1), in which randomly a considerable number of job sequences are generated by using the pre-trained transformer to predict the JPR within the fixed initial schedule patterns. In this way, the algorithm can be started with relatively accurate dual values. Thus, a large portion of initial iterations approximating dual values are skipped. In addition, as the constraint of forcing the number of schedules that an operator can severely restrict the feasibility of the initial master problem, we remove Constraints (8) first to obtain relatively accurate dual values at the beginning and then re-introduce these constraints to derive feasible solutions for the next iterations.

To achieve a balance between solution quality and CPU time, a heuristic rounding strategy is adopted for branching. After solving the master model to optimality, the column with the largest value that approximates one is fixed and all settled jobs are removed from the following subproblems. This method is repeated until the schedules for all operators are determined.

5.3 Computational Experiments

The experiments are conducted in two main aspects. The performance of the proposed DeepPR is first compared with other benchmark models on the previously introduced dataset. Sensitivity analysis for *the length of preceding jobs involved* based on the DeepPR is carried out to uncover the sequence-based influence. Then, the performance of the proposed *CBPR-guided B&P heuristic scheduling approach* (*CBPR-guided approach* for short in the following) is presented by comparing its performance with the scheduling results based on the printing company's practice.

The deep learning models are coded with PyTorch 1.11. The optimization framework is coded in Python. The Python API of the commercial solver CPLEX, DOcplex (version 2.23) is used to solve the relaxed linear programming model of the master problem. The models are run on a computer with a 1.9GHz i9 CPU, 64G RAM, and Windows 11 system.

After data processing, we get N = 2620 pieces of data in total. The data are arranged according to the executed order, with a single operator working for a shift (eight hours) and then turning to another operator for the next shift. Printing jobs across two shifts are split into two jobs with corresponding operators and workload (calculated by actual time proportion). Five categories of printing jobs are involved, namely, *Hardcover Bound* (HCB), *Paperback* (PB), *Loose-leaf Bound* (LLB), *Folding Box* (BXF), and *Saddle Stitch* (SS). The data are divided into a training dataset and a test dataset with a ratio of 8:2. In this way, 2096 pieces of job records are obtained for training, and 524 job data records for testing.

5.3.1 Comparison with other benchmarks

To see the performance of the proposed DeepPR, the following benchmarking models are applied: TCN-based model, transformer-baed model, LSTM, and DNN. It should be noted that the former three models are promising DL models in handling time-series data. Those models take both a set of ordered preceding jobs and the job-specified features as inputs, and thus we name them *sequence-based models*. On the contrary, for the DNN model, jobs are treated as independent, and thus we call it *non-sequence model*. Adam is used as the optimizer. For the JPR regression task, the label *JPR* is derived with *JPR = output quantity/(setup time + machinery processing time + human operating time)*. The criterion selected to train the models is the mean square error (*MSE*). The mean absolute percentage error (*MAPE*) is calculated as another measure for comparison.

To compare the performances of the models, a grid search is performed for model hyperparameter tuning, as shown in Table 5-2. After 150 epochs of training, the performances of all models converge. Table 5-3 records the most promising parameters for each model and the results of the average values for 10 trials. Figure 5-4 plots the MSE loss curve on the test dataset for the involved models. Noticing that the average time for processing a job is around one hour and thus the average number of jobs that can be processed in one shift is around eight. Thus, it is reasonable to take the average value of four as the sequence length for comparing the prediction performance of *sequence-based models*. Later, a sensitivity analysis is conducted to explore the effect of sequence length on the prediction performance.

From Table 5-3, From Table III, it can be seen that the DeepPR can achieve better results in comparison with other benchmarks by reducing the MSE to 42.6 and MAPE to 5.53%, which shows the effectiveness of the proposed model. This result is followed by the Transformer-based model, which is well-known for its powerful attention allocation mechanism. The other two sequence-based models (i.e., TCN-based model and LSTM) are less competitive in achieving satisfiable results. Additionally, compared with these time-series models, the structure of the multiple inputs of DeepPR brings the benefits of adjusting the contribution of different input modules by controlling the number of hidden cells when combining the latent features extracted from both modules, which can bring more flexibility.

Table 5-2. Grid search for model hyperparameters

Model	Hyperparameters	Range
Transformer-based model	Number of encoder layers	[1, 2]
	Number of decoder layers	[1, 2]
	Feedforward dimensions	[128, 64]
	Number of heads	[1, 3]
	Learning rate	[1e-2, 1e-3]
TCN-based model	Number of channels	[16, 8, 4], [32, 16, 8], [32, 8, 4]
	1D-Kernel size	2
	Dilation	[No dilation, [layer_index+1]]
	Number of layers	3
	Learning rate	[1e-2, 1e-3]
LSTM	Number of hidden units	[16, 32]
	Number of LSTM layers	[1, 2]
	Dropout rate	[0.2, 0.3]
	Learning rate	[1e-2, 1e-3]
DNN	Number of layers	[2, 3, 4, 5]
	Number of neurons on each layer	[64, 32], [32, 62, 32], [128, 64, 32, 16], [128, 64, 32, 16, 8]
	Dropout rate	[0.2, 0.3]
	Learning rate	[1e-2, 1e-3]
DeepPR	Concatenating units of DIM	[4, 8, 16]
	Concatenating units of SIM	[4, 8, 16]
	Number of encoder layers	[1, 2]
	Number of decoder layer	[1, 2]
	Number of multi-heads	3
	1D-kernel size	[3, 1]

Then, for DNN models which predict only using job-specific features, the best performance they can achieve is 243.3 of MSE and 13.34% of MAPE, which are significantly worse than the sequence-based models. Besides, the MSE loss curve shows that the DNN model is less stable than other models due to the lack of sequence-related information. This justifies that the context should be built based on both the job-related features and its sequentially positioning information.

Table 5-3. Prediction results by different learning models

Models	**Parameters combination**	**Avg. MSE**	**Avg. MAPE**
TCN-based model	#channels [32, 8, 4], dilation size [1, 2, 3], dropout 0.2, learning rate1e-3	126.8	9.59%
Transformer-based model	# encoder layers 1, # decoder layers 1, # feedforward 128, #heads 3, learning rate 1e-3	57.4	6.61%
LSTM	#layers 1, #hidden units 32, dropout rate 0.2, learning rate 1e-2	92.2	8.45%
DeepPR	#DIM units 8, #SIM units 8, #encoder layer 1, #decoder layer 1, #feedforward 32, kernel size [3, 1], learning rate 1e-3	42.6	5.53%
DNN	#layers 4 respectively with [128, 64, 32, 16] neurons, dropout rate 0.2, learning rate 1e-2	243.3	13.34%

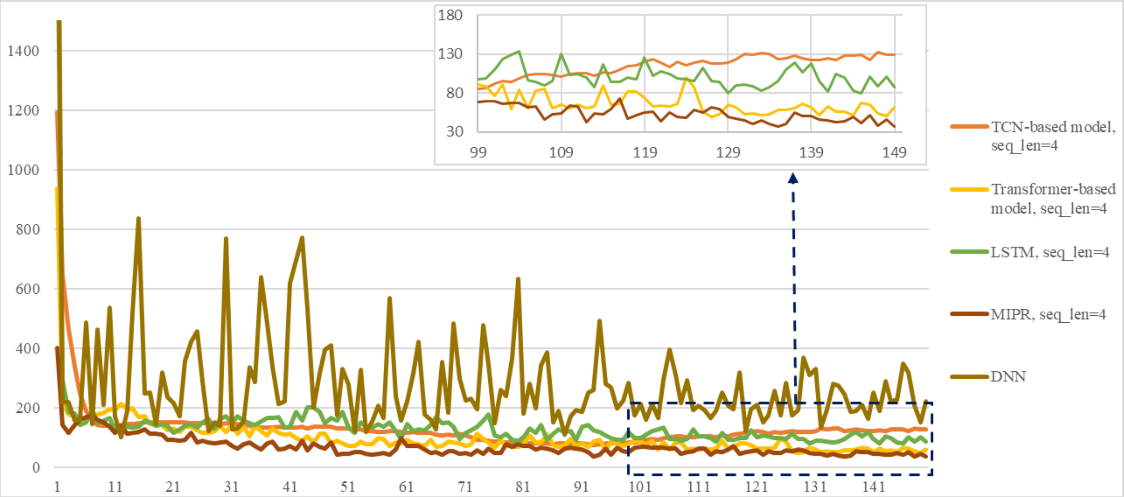


Figure 5-4. MSE loss curve of representative deep learning models

Table 5-4. Prediction results under varying sequence length

Sequence length	Avg. MSE	Avg. MAPE
2	59.6	6.44%
3	58.4	6.42%
4	42.6	5.53%
5	54.1	6.10%
6	66.2	6.47%

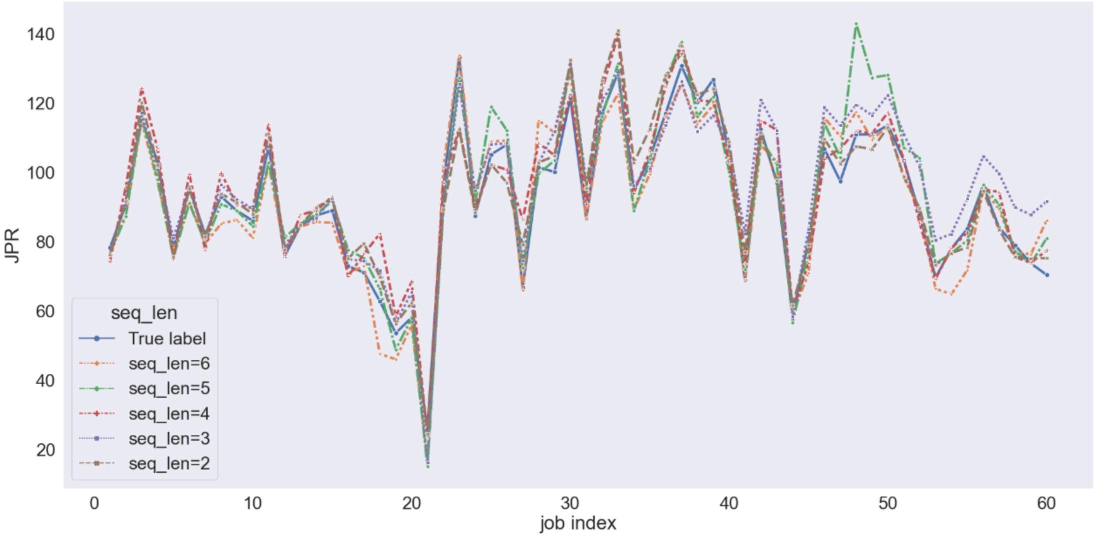


Figure 5-5. Prediction performance visualization for different sequence lengths

Sensitivity analysis is conducted for sequence-length variation. The sequence length (Seq_len) varies from two (only one immediate predecessor and the job to be predicted) to six based on the proposed DeepPR. The results of the average MSE and MAPE of 10 trials are summarized in Table 5-4. Figure 5-5 visualizes the predicted values for the first 60 jobs extracted from the test dataset by DeepPR with different sequence lengths and the ground truth.

The results imply that along with the increase in sequence length, the prediction error first goes down and then goes up. It implies that properly incorporating a few predecessors is helpful for JPR prediction. The performance, however, tends to deteriorate when involving excessive predecessors (the relationships are very weak and interfere with the training). Furthermore, it is noticeable that by adopting two as the sequence length, the prediction performance is already largely improved compared with *non-sequence model* (i.e., DNN) with MSE 59.6 and MAPE 6.44%. Therefore, the execution of the immediately preceding job has a high effect on its immediate successor due to possible changeover operations and extra setups.

Since the slight discrepancy in predicting accuracy will have a small influence on the schedule quality but significantly benefit computational efficiency, the sequence length of two is taken as the learning pattern to implement CBPR-guided scheduling.

5.3.2 Comparisons of scheduling performance

Then, the performance of the proposed CBPR-guided approach is verified by comparing its performance with the scheduling results based on the company's practice. By the company's current practice, a simple linear relationship between the actual processing time of a job and the workload (printing quantity) of the job is considered. Thus, the processing time for a job is estimated by timing the output quantity of the job with a ratio obtained according to experience (i.e., historical total workload/historical total time consumption). We call this scheduling approach *the traditional method*. As a comparison, by using the proposed CBPR-guided approach, the JPR will be predicted under contexts dynamically produced during pricing, and the scheduling algorithm will generate schedules by allocating jobs to operators and placing them in suitable positions in the performing sequence as well.

We test instances obtained by randomly sampling from the test dataset. Nine instances of different scales were generated, from 20 jobs to 100 jobs (the instance code contains information about the number of jobs, the number of shifts, and the number of machines to be scheduled). Acceleration strategies are applied. The result comparison between the CBPR-guided approach and the traditional method is shown in Table 5-5. From the results, it can be seen that after embedding the CBPR prediction into the scheduling process, the model still can be efficiently solved within acceptable time limits. For small instances of 20 jobs in one shift and processed on three machines, the CPU time needed is around 10 seconds. The instance of 100 jobs can be solved at around 1200s, which represents the workload for two days in four shifts. Moreover, schedules derived by the CBPR-guided method significantly reduced the total completion time by an average of 12.84% across all instances due to the increase in JPR under better schedule solutions that have considered the sequence-dependent relationship between adjacent jobs and the combined effect of a variety of job-specified factors related to engineering, operating, and environmental elements.

Table 5-5. Comparison between prediction-based scheduling and current practice

Instance	CBPR-guided-TOT	Estimation-TOT	Improving	CPU time (s)
20j 1s 3m	1277	1438	11.20%	13
30j 1s 3m	1726	2006	13.96%	55
40j 2s 3m	2517	2936	14.27%	136
50j 2s 3m	3053	3488	12.47%	295
60j 2s 3m	3564	4134	13.79%	245
70j 3s 3m	4251	4909	13.40%	338
80j 3s 3m	4784	5466	12.48%	587
90j 4s 3m	5487	6167	11.03%	415
100j 4s 3m	6284	7220	12.96%	1273

CBPR-guided-TOT: Total operation time obtained by the CBPR-guided approach; **Estimation-TOT:** Total operation time obtained by the empirical estimation-based scheduling method.

To further verify the effect of the proposed scheduling method on the scheduling solution, computational experiments are conducted to uncover the effect of the CBPR-guided approach on scheduling solutions. To be specific, we compare the JPR changes of printing jobs belonging to different categories. From the historical data, five categories of printing jobs are involved, i.e.,$CAT = \{HCB, PB, BXF, LLB, SS\}$. Due to different technical requirements, the JPR of these categories shows some discrepancies. For example, the production of BXF is relatively complicated and thus induces lower JPRs, while the JPRs of SS are relatively high. The average JPR of each category of jobs are: 99 (HCB), 93(PB), 64(BXF), 75(LLB), and 115(SS). Then, we constructed 10 instances ($INS = \{INS_1, INS_2, \dots, INS_{10}\}$) by random sampling, and each instance contains 100 jobs, which are a mixture of five job categories (the number of jobs $|INS_{n,c}|, (n \in \{1, 2, \dots, 10\})$ that were randomly sampled from the category $c, (c \in CAT)$, waved around 20). We compare the JPR derived from our method and the original schedule to illustrate the JPR improvement. Specifically, the following performance indicators are evaluated:

- *RNEA (Beyond-sample comparison)*: Ratio of jobs getting a JPR exceeding the average JPR of its category, that is $\frac{\sum_{i \in INS_{n,c}} \mathbb{I}(PR(i) > Avg(c))}{Count(INS_{n,c})}, \forall n, c.$ $\mathbb{I}(\cdot)$ is an indicator function; $PR(i)$ is the JPR obtained by job i.
- *In-sample comparison*: Statistics including median, quartile, minimum/ maximum score of each category for all sampled instances, $INS_c = \cup INS_{n,c}, n \in \{1, 2, \dots, 10\}$

The RENA is called a beyond-sample comparison because it compares the JPR results derived from the CBPR-guided approach with the average JPR of the entire job category. Correspondingly, to ensure a fair comparison, in-sample comparisons are included to unveil changes in JPR distributions between original schedules and results from the CBPR-guided approach. Thus, the former indicator focuses on overall improvement, while the latter stresses individual jobs. As shown in Table 5-6, compared with the average JPR of each category, the mean RNEA of all instances is 66% (across five categories and ten tested instances), which represents a significant improvement for the entire job set since more jobs now obtain a JPR over the previous average level. Besides, through the proposed CBPR-guided scheduling method, the relatively less efficient jobs belonging to BXF and LLB (suffering low JPRs) have seen the largest JPR enhancement, with 72% of jobs exceeding the average JPR level, while the JPR increments of the other three categories are relatively small.

Table 5-6. RNEA of each instance under all categories

	HCB	PB	SS	BXF	LLB	Ins.AVG
INS1	65%	58%	60%	58%	75%	63%
INS2	60%	47%	65%	76%	71%	64%
INS3	70%	75%	60%	63%	78%	69%
INS4	70%	75%	70%	47%	74%	67%
INS5	59%	80%	55%	67%	83%	69%
INS6	80%	65%	65%	78%	68%	71%
INS7	47%	65%	60%	82%	76%	66%
INS8	55%	53%	60%	78%	65%	62%
INS9	50%	55%	75%	94%	61%	67%
INS10	55%	56%	75%	80%	65%	66%
c.AVG	61%	63%	65%	72%	72%	66%

c.AVG: the average RNEA of the corresponding category.

Ins.AVG: the average RNEA of the corresponding instance.

Then, the boxplot of Figure 5-6 is used to show the in-sample comparison result, which indicates JPR changes compared with the original JPR within samples. It can be seen that the CBPR-guided method can enhance the scheduling solutions by improving the JPR of all categories on the whole. Specifically, the lower limit and upper limit of JPR for most categories are enhanced, and for each category, more than half of jobs now obtain JPRs falling into intervals with larger JPRs compared with the median of original sample records. The magnitude of improvement for categories BXF and SS is more significant. These results verify that by capturing variabilities in the scheduling context, the CBPR-guided approach can position jobs in beneficial places and thus lead to improvement in the overall processing rate and production efficiency.

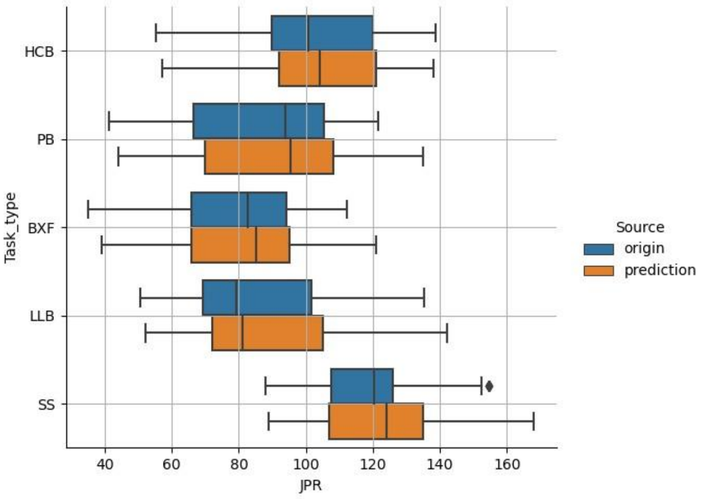


Figure 5-6. JPR changes between predicted results and original records

5.4 Summary

This study is based on the real demand of an industrial printing company. As the JPR is affected by the joint impact of multiple factors (e.g., order, material, operating, and environment) as well as positioning in a sequence, a practical learning-embedded branch-and-price heuristic scheduling approach is proposed to capture the influences of multiple factors on JPR and enable scheduling with more accurate JPR prediction. The CBPR-labelling algorithm proposed in this study embeds the prediction of JPR under varying contexts constructed by positioning a job at different execution places in schedules. Such predictions enable the generation of schedules with minimum completion time with the consideration of multiple realistic factors. Experiments unveil the dependency of JPR on CBPR and show that the proposed deep learning model outperforms other models in prediction accuracy. By employing JPR prediction into the scheduling framework, our method can achieve an improvement of 12.84% on average in terms of completion time compared with the traditional method based on estimation. Furthermore, the CBPR-guided approach can derive better schedules with enhanced JPR.

Managerial Insights

As shown by the real production processes, uncertain combinations of operational factors often have a great influence on the actual job execution process (e.g., affecting the production rate/duration). In smart manufacturing environments, the installation of IoT devices and sensors facilitates timely data collection from multiple sources, which enhances the tracking of machine status, material usage, and operational conditions. Using deep learning technologies can facilitate capturing the interplay of different variations and thus enable a better understanding of how production performance is influenced by multiple factors.

Furthermore, it is more important to transform the knowledge learned from historical production data into decision-making, thereby achieving more efficient resource utilization, reducing machine downtime and improving operator performance. The developed scheduling approach integrates a deep learning engine with an optimization algorithm. Therefore, when implementing our method in an actual production scenario, a set of jobs to be scheduled and their corresponding features can be encoded and input into the algorithm. Then, instead of solving static deterministic scheduling problems, the optimization process will explore a huge number of different circumstances or combinations of production settings to find efficient solutions (i.e., schedules) that place jobs into advantageous performing positions. Besides, guided by

AI-driven insights, the derived solutions can be more practical and suitable for the specific production system and meanwhile, be efficient due to the wide consideration of a large number of potential placements.

Additionally, while the investigated problem is based on a printing process, the proposed scheduling method can be applied to many other industries and production scenarios, such as injection moulding and dyeing. This is because the operations and schedules of these industries are also affected by various factors in a similar manner. Efforts can be made to use deep learning models to identify the key operational factors that influence the crucial performance indicators (e.g., the production efficiency indexes or production quality indicators) in these systems and then leverage the influencing patterns to optimize their decisions of scheduling or operations.

Chapter 6. Concluding Remarks

6.1 Conclusions

Smart manufacturing is demonstrating its significance in various production industries. This dissertation focuses on developing sustainable and efficient scheduling strategies for smart manufacturing systems. On this topic, two significant research aspects are concentrated: (i) from the physical configuration of production cells, mobile robots are increasingly adopted for automating the material handling process, which induces new scheduling challenges and energy concerns for robotic cells, and (ii) production data timely collected from multiple resources enables the predictive analysis of production and AI-enabled scheduling process. Three research studies are described in this dissertation, with the main focuses and key insights summarized in Table 6-1.

The first research study (described in Chapter 3) focuses on a robot-facilitated job shop scheduling problem, which simultaneously plans robot routing and machine operating sequences to reduce energy consumption (promoting sustainable goals) through the collaboration between the two subjects. From the methodology aspect, it focuses on developing efficient modelling techniques (i.e., a network-based model) to tackle the complicated operational restrictions of robot operations and machine processing. Then, the second study (described in Chapter 4) focuses on predicting two important production performance indicators based on a real-world production dataset. An effective deep learning model with multiple input modules and dual output layers is proposed, which can extract useful patterns from different levels of the performing sequence. The third study (described in Chapter 5) further integrates the deep learning-supported processing rate prediction into the scheduling optimization process so that the processing rate can be forecasted for varying processing circumstances to guide the allocation of jobs to beneficial execution positions and proper operators.

Table 6-1. A summary of main focuses, methods, key findings, and managerial implications

Chapters	Main Focus	Proposed Methods	Key Findings	Managerial Implications
Chapter 3	Energy reduction by enhancing collaborations between machine-robot operations	Two MILP models are developed. RJSP-E minimizes the overall energy consumption, while RJSP-EM simultaneously considers makespan and energy consumption.	The RJSP-E can reduce overall energy consumption by an average of 15%, but at a loss of makespan; while RJSP-EM reduces energy consumption by a mean of 10% with no compromise in makespan.	RJSP approaches developed in this work can enhance the energy efficiency of modern robotic cells, thus promoting the healthy and sustainable development of smart manufacturing.
Chapter 4	To capture the effect of various real-world factors on the actual production process	A multi-module supported dual-task learning model (MMDT) to simultaneously predict the job processing time (JPT) and processing rate (PR) level for a better understanding and capture of production status.	(i) The proposed model can enhance the accuracy of PR classification and JPT prediction compared with other benchmarks. (ii) Patterns captured by different input modules from three levels are helpful for predicting PR variations. (iii) Both tasks benefit from learning from the representation shared by each other.	(i) The proposed method can provide timely alarm of inefficient alignment between task and production resources. (ii) To increase the JPR prediction ability, it will benefit if a suitable number of preceding jobs is involved. (ii) By applying the proposed model, extreme cases (when JPR is significantly low or high) can be effectively identified.
Chapter 5	Incorporating multiple factors to develop efficient AI-empowered optimization solution approaches	A CBPR-labelling algorithm is proposed, which utilizes the prediction of JPR under varying contexts to drive the generation of beneficial schedules with enhanced efficiency.	(i) By employing JPR prediction into the scheduling framework, the production completion time can be reduced by 12.84% on average. (ii) The CBPR-guided approach can derive better schedules with enhanced JPR.	(i) Integration of AI engines with optimization algorithms enables the latter to derive more practical and efficient decisions; (ii) Our method can be applied to many other industries, such as injection moulding and dyeing systems, as their operations also suffer uncertain influencing factors in a similar manner.

To conclude, this dissertation focuses on improving scheduling decisions of smart manufacturing systems from two aspects (i.e., physical-level operations collaboration and AI-empowered decision-making) with advances in modelling, deep learning, and optimization algorithms. However, it is worth mentioning that the studies reported also have limitations, which highlight many potential directions for future studies.

6.2 Future Studies

(i) ***Scheduling for robotic cell with multi-robots***. The study in Chapter 3 only considered one mobile robot to carry out the movement of all materials, semi-products and also finished goods. Future research attention thus can be paid to investigating the involvement of additional robots in a robotic cell to increase efficiency, reduce deadlock situations, and further achieve sustainability goals. However, it can be foreseen that the adoption of multiple robots will greatly complicate the interactions between individual robots (e.g., task prioritization and allocation to multiple robots as well as multi-robots route planning) and the interactions between robots and machines (e.g., task allocation and sequencing on machines involving the movement of material/semi-product operations by multiple robots). These elements will significantly increase the problem complexity. Therefore, it is worth further investigating more efficient modelling methods and solution algorithms for such systems.

(ii) ***Scheduling for human-robot collaboration.*** The study in Chapter 3 does not consider human interventions or operations. However, in many production systems, an efficient manufacturing process requires collaboration between human operators and mobile robots. For example, mobile robots are responsible for the delivery of materials or tools, while human operators will carry out specific processing operations on machines. In this mode, the delivery efficiency of robots will have a large influence on overall production efficiency. Therefore, research can be drilled into developing efficient scheduling approaches to better coordinate mobile robots, human operators, and machines to ensure a harmonious production environment.

(iii) ***Investigation in other real-world manufacturing systems.*** The study in Chapter 4 is motivated by the scheduling challenge of a real-world production company in the printing industry. It is worth further investigating the characteristics of other real-world manufacturing systems to figure out the critical factors that will make an impact on their operational efficiency through AI methods, and furthermore to incorporate the data-driven insights in their decision-making.

(iv) ***Developing prediction-enabled scheduling approaches for other production settings.*** In studies of Chapters 4 and 5, we explored the incorporation of precise job processing rate prediction into deriving efficient production schedules. It is worth noting that the nature of printing jobs determines the scheduling mode in the study of Chapter 5, which is to plan each printing task as a holistic job. Future research can be devoted to generalizing the investigated scheduling method to other manufacturing systems with different production modes, such as the flowshop and job shop scheduling problems so that the performance of these systems can greatly benefit from precise scheduling solutions.

(v) ***Incorporation of AMR and energy consideration for production systems under multiple influencing factors.*** The study in Chapter 3 has preliminarily explored the integration of AMR operations and analyzed the impact of speed adjustment on the energy consumption of production systems. Further investigation is warranted to incorporate AMR operations and energy considerations into production scheduling problems that take the effect of various operational factors into account so as to enhance the efficiency and sustainability objectives of real-world manufacturing systems.

(vi) ***Development of online scheduling methods.*** Another promising future research direction is to develop online scheduling solutions for systems that require a rapid response to real-time fluctuations, such as order cancellation or demand surging, task requirement change, and machine breakdown. It is therefore promising to develop dynamic scheduling algorithms by leveraging the advantages of both machine learning and optimization methods to accommodate various production scenarios and unexpected circumstances in an online manner.

References

Abedi, M., Chiong, R., Noman, N., & Zhang, R. (2020). A multi-population, multi-objective memetic algorithm for energy-efficient job-shop scheduling with deteriorating machines. *Expert Systems with Applications*, *157*, 113348.

Alemão, D., Rocha, A. D., & Barata, J. (2021). Smart manufacturing scheduling approaches—Systematic review and future directions. *Applied Sciences*, *11*(5), 2186.

Alidaee, B., & Womer, N. K. (1999). Scheduling with time dependent processing times: Review and extensions. *Journal of the Operational Research Society*, *50*, 711-720.

Avci, M., Avci, M. G., & Hamzadayı, A. (2022). A branch-and-cut approach for the distributed no-wait flowshop scheduling problem. *Computers & Operations Research*, *148*, 106009.

Azab, E., Nafea, M., Shihata, L. A., & Mashaly, M. (2021). A machine-learning-assisted simulation approach for incorporating predictive maintenance in dynamic flow-shop scheduling. *Applied Sciences*, *11*(24), 11725.

Baldea, M., & Harjunkoski, I. (2014). Integrated production scheduling and process control: A systematic review. *Computers & Chemical Engineering*, *71*, 377-390.

Barak, S., Moghdani, R., & Maghsoudlou, H. (2021). Energy-efficient multi-objective flexible manufacturing scheduling. *Journal of Cleaner Production*, *283*, 124610.

Baty, L., Jungel, K., Klein, P. S., Parmentier, A., & Schiffer, M. (2024). Combinatorial optimization-enriched machine learning to solve the dynamic vehicle routing Problem with Time Windows. *Transportation Science*. In press.

Bencheikh, G., Letouzey, A., & Desforges, X. (2022). An approach for joint scheduling of production and predictive maintenance activities. *Journal of Manufacturing Systems*, *64*, 546-560.

Birge, J., Frenk, J., Mittenthal, J., & Kan, A. R. (1990). Single-machine scheduling subject to stochastic breakdowns. *Naval Research Logistics*, *37*(5), 661-677.

Brucker, P., Burke, E. K., & Groenemeyer, S. (2012). A mixed integer programming model for the cyclic job-shop problem with transportation. *Discrete Applied Mathematics*, *160*(13-14), 1924-1935.

Bueno, A., Godinho Filho, M., & Frank, A. G. (2020). Smart production planning and control in the Industry 4.0 context: A systematic literature review. *Computers & Industrial Engineering*, *149*, 106774.

Bukata, L., Šůcha, P., & Hanzálek, Z. (2019). Optimizing energy consumption of robotic cells by a Branch & Bound algorithm. *Computers & Operations Research*, *102*, 52-66.

Buxey, G. (1989). Production scheduling: Practice and theory. *European Journal of*

Operational Research, *39*(1), 17-31.

Cai, J., Li, X., Liang, Y., & Ouyang, S. (2021). Collaborative optimization of storage location assignment and path planning in robotic mobile fulfillment systems. *Sustainability*, *13*(10), 5644.

Çaliş, B., & Bulkan, S. (2015). A research survey: Review of AI solution strategies of job shop scheduling problem. *Journal of Intelligent Manufacturing*, *26*, 961-973.

Cardin, O., Trentesaux, D., Thomas, A., Castagna, P., Berger, T., & Bril El-Haouzi, H. (2017). Coupling predictive scheduling and reactive control in manufacturing hybrid control architectures: State of the art and future challenges. *Journal of Intelligent Manufacturing*, *28*, 1503-1517.

Caumond, A., Lacomme, P., Moukrim, A., & Tchernev, N. (2009). An MILP for scheduling problems in an FMS with one vehicle. *European Journal of Operational Research*, *199*(3), 706-722.

Chen, H., Jeremiah, S. R., Lee, C., & Park, J. H. (2023). A digital twin-based heuristic multi-cooperation scheduling framework for smart manufacturing in IIoT environment. *Applied Sciences*, *13*(3), 1440.

Cheng, C.-Y., Ying, K.-C., Li, S.-F., & Hsieh, Y.-C. (2019). Minimizing makespan in mixed no-wait flowshops with sequence-dependent setup times. *Computers & Industrial Engineering*, *130*, 338-347.

Chui, K. T., Gupta, B. B., & Vasant, P. (2021). A genetic algorithm optimized RNN-LSTM model for remaining useful life prediction of turbofan engine. *Electronics*, *10*(3), 285.

Cornwell, C., Schmutte, I. M., & Scur, D. (2021). Building a productive workforce: The role of structured management practices. *Management Science*, *67*(12), 7308-7321.

Dai, M., Tang, D., Giret, A., & Salido, M. A. (2019). Multi-objective optimization for energy-efficient flexible job shop scheduling problem with transportation constraints. *Robotics and Computer-Integrated Manufacturing*, *59*, 143-157.

Dawande, M., Geismar, H. N., Sethi, S. P., & Sriskandarajah, C. (2005). Sequencing and scheduling in robotic cells: Recent developments. *Journal of Scheduling*, *8*, 387-426.

Del Gallo, M., Mazzuto, G., Ciarapica, F. E., & Bevilacqua, M. (2023). Artificial intelligence to solve production scheduling problems in real industrial settings: Systematic literature review. *Electronics*, *12*(23), 4732.

Demir, Y., & İşleyen, S. K. (2013). Evaluation of mathematical models for flexible job-shop scheduling problems. *Applied Mathematical Modelling*, *37*(3), 977-988.

Deng, S., & Yeh, T.-H. (2011). Using least squares support vector machines for the airframe structures manufacturing cost estimation. *International Journal of*

Production Economics, *131*(2), 701-708.

Deutsch, J., He, M., & He, D. (2017). Remaining useful life prediction of hybrid ceramic bearings using an integrated deep learning and particle filter approach. *Applied Sciences*, *7*(7), 649.

Djenouri, Y., Belhadi, A., Srivastava, G., & Lin, J. C.-W. (2023). Hybrid graph convolution neural network and branch-and-bound optimization for traffic flow forecasting. *Future Generation Computer Systems*, *139*, 100-108.

Drobouchevitch, I. G., Geismar, H. N., & Sriskandarajah, C. (2010). Throughput optimization in robotic cells with input and output machine buffers: A comparative study of two key models. *European Journal of Operational Research*, *206*(3), 623-633.

Du, H., Qiao, F., Wang, J., & Lu, H. (2021). A hybrid metaheuristic algorithm with novel decoding methods for flexible flow shop scheduling considering human fatigue. In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC).*

Ecer, F. (2022). Multi-criteria decision making for green supplier selection using interval type-2 fuzzy AHP: A case study of a home appliance manufacturer. *Operational Research*, *22*(1), 199-233.

Enginarlar, E., Li, J., Meerkov, S. M., & Zhang, R. Q. (2002). Buffer capacity for accommodating machine downtime in serial production lines. *International Journal of Production Research*, *40*(3), 601-624.

Essien, A., & Giannetti, C. (2020). A deep learning model for smart manufacturing using convolutional LSTM neural network autoencoders. *IEEE Transactions on Industrial Informatics*, *16*(9), 6069-6078.

Fan, H., Xiong, H., & Goh, M. (2021). Genetic programming-based hyper-heuristic approach for solving dynamic job shop scheduling problem with extended technical precedence constraints. *Computers & Operations Research*, *134*, 105401.

Fang, Y., Peng, C., Lou, P., Zhou, Z., Hu, J., & Yan, J. (2019). Digital-twin-based job shop scheduling toward smart manufacturing. *IEEE Transactions on Industrial Informatics*, *15*(12), 6425-6435.

Fontes, D. B., Homayouni, S. M., & Gonçalves, J. F. (2023). A hybrid particle swarm optimization and simulated annealing algorithm for the job shop scheduling problem with transport resources. *European Journal of Operational Research*, *306*(3), 1140-1157.

Fragapane, G., de Koster, R., Sgarbossa, F., & Strandhagen, J. O. (2021). Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda. *European Journal of Operational Research*, *294*(2), 405-426.

Gao, K., Cao, Z., Zhang, L., Chen, Z., Han, Y., & Pan, Q. (2019). A review on swarm intelligence and evolutionary algorithms for solving flexible job shop scheduling problems. *IEEE/CAA Journal of Automatica Sinica*, *6*(4), 904-916.

Gao, Y., Zhang, G., Zhang, C., Wang, J., Yang, L. T., & Zhao, Y. (2021). Federated tensor decomposition-based feature extraction approach for industrial IoT. *IEEE Transactions on Industrial Informatics*, *17*(12), 8541-8549.

Ghaleb, M., Zolfagharinia, H., & Taghipour, S. (2020). Real-time production scheduling in the Industry-4.0 context: Addressing uncertainties in job arrivals and machine breakdowns. *Computers & Operations Research*, *123*, 105031.

Glock, C. H., & Grosse, E. H. (2021). The impact of controllable production rates on the performance of inventory systems: A systematic review of the literature. *European Journal of Operational Research*, *288*(3), 703-720.

Gmys, J., Mezmaz, M., Melab, N., & Tuyttens, D. (2020). A computationally efficient Branch-and-Bound algorithm for the permutation flow-shop scheduling problem. *European Journal of Operational Research*, *284*(3), 814-833.

Graves, S. C. (1981). A review of production scheduling. *Operations Research*, *29*(4), 646-675.

Grigsby, J., Wang, Z., Nguyen, N., & Qi, Y. (2021). Long-range transformers for dynamic spatiotemporal forecasting. *arXiv preprint arXiv:2109.12218*.

Gu, W., Li, Y., Tang, D., Wang, X., & Yuan, M. (2022). Using real-time manufacturing data to schedule a smart factory via reinforcement learning. *Computers & Industrial Engineering*, *171*, 108406.

Guo, W., Jiang, P., & Yang, M. (2023). Unequal area facility layout problem-solving: A real case study on an air-conditioner production shop floor. *International Journal of Production Research*, *61*(5), 1479-1496.

Guo, X., Shen, C., & Chen, L. (2016). Deep fault recognizer: An integrated model to denoise and extract features for fault diagnosis in rotating machinery. *Applied Sciences*, *7*(1), 41.

Gürel, S., Gultekin, H., & Akhlaghi, V. E. (2019). Energy conscious scheduling of a material handling robot in a manufacturing cell. *Robotics and Computer-Integrated Manufacturing*, *58*, 97-108.

Ham, A. (2021). Transfer-robot task scheduling in job shop. *International Journal of Production Research*, *59*(3), 813-823.

Han, Z., Zhao, J., Leung, H., Ma, K. F., & Wang, W. (2019). A review of deep learning models for time series prediction. *IEEE Sensors Journal*, *21*(6), 7833-7848.

Hassani, Z. I. M., Barkany, A. E., Abbassi, I. E., Jabri, A., & Darcherif, A. M. (2019). New model of planning and scheduling for job-shop production system with energy consideration. *Management and Production Engineering Review*, *10* (1), 89-97

He, L., Chiong, R., Li, W., Dhakal, S., Cao, Y., & Zhang, Y. (2021). Multiobjective optimization of energy-efficient job-shop scheduling with dynamic reference point-based fuzzy relative entropy. *IEEE Transactions on Industrial Informatics*, *18*(1), 600-610.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735-1780.

Huang, L., Chen, X., Huo, W., Wang, J., Zhang, F., Bai, B., & Shi, L. (2021). Branch and bound in mixed integer linear programming problems: A survey of techniques and trends. *arXiv preprint arXiv:2111.06257*.

Hurink, J., & Knust, S. (2002). A tabu search algorithm for scheduling a single robot in a job-shop environment. *Discrete applied mathematics*, *119*(1-2), 181-203.

Jacso, A., Szalay, T., Sikarwar, B. S., Phanden, R. K., Singh, R. K., & Ramkumar, J. (2023). Investigation of conventional and ANN-based feed rate scheduling methods in trochoidal milling with cutting force and acceleration constraints. *The International Journal of Advanced Manufacturing Technology*, *127*(1), 487-506.

Jamrus, T., Chien, C.-F., Gen, M., & Sethanan, K. (2017). Hybrid particle swarm optimization combined with genetic operators for flexible job-shop scheduling under uncertain processing time for semiconductor manufacturing. *IEEE Transactions on Semiconductor Manufacturing*, *31*(1), 32-41.

Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, *31*(3), 685-695.

Jia, F., Lei, Y., Lin, J., Zhou, X., & Lu, N. (2016). Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. *Mechanical Systems and Signal Processing*, *72*, 303-315.

Jiang, E.-d., & Wang, L. (2019). An improved multi-objective evolutionary algorithm based on decomposition for energy-efficient permutation flow shop scheduling problem with sequence-dependent setup time. *International Journal of Production Research*, *57*(6), 1756-1771.

Jiang, Z., Yuan, S., Ma, J., & Wang, Q. (2022). The evolution of production scheduling from Industry 3.0 through Industry 4.0. *International Journal of Production Research*, *60*(11), 3534-3554.

Juvin, C., Houssin, L., & Lopez, P. (2023). Logic-based Benders decomposition for the preemptive flexible job-shop scheduling problem. *Computers & Operations Research*, *152*, 106156.

Karimi, S., Ardalan, Z., Naderi, B., & Mohammadi, M. (2017). Scheduling flexible job-shops with transportation times: Mathematical models and a hybrid imperialist competitive algorithm. *Applied Mathematical Modelling*, *41*, 667-682.

Karmarkar, U., & Kekre, S. (1987). Manufacturing configuration, capacity and mix decisions considering operational costs. *Journal of Manufacturing Systems*, *6*(4), 315-324.

Keung, K. L., Lee, C. K., & Ji, P. (2021). Data-driven order correlation pattern and storage location assignment in robotic mobile fulfillment and process automation system. *Advanced Engineering Informatics*, *50*, 101369.

Khan, S. H., Hayat, M., Bennamoun, M., Sohel, F. A., & Togneri, R. (2017). Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE transactions on Neural Networks and Learning Systems*, *29*(8), 3573-3587.

Kim, H., Lim, D.-E., & Lee, S. (2020). Deep learning-based dynamic scheduling for semiconductor manufacturing with high uncertainty of automated material handling system capability. *IEEE Transactions on Semiconductor Manufacturing*, *33*(1), 13-22.

Koulamas, C., & Kyparisis, G. J. (2023). A classification of dynamic programming formulations for offline deterministic single-machine scheduling problems. *European Journal of Operational Research*, *305*(3), 999-1017.

Koulamas, C., & Panwalkar, S. (2019). The two-stage no-wait/blocking proportionate super shop scheduling problem. *International Journal of Production Research*, *57*(10), 2956-2965.

Lamotte, R., & Geroliminis, N. (2021). Monotonicity in the trip scheduling problem. *Transportation Research Part B: Methodological*, *146*, 14-25.

Lee, C. Y., & Chen, Z. L. (2001). Machine scheduling with transportation considerations. *Journal of Scheduling*, *4*(1), 3-24.

Lee, H.-Y., & Murray, C. C. (2019). Robotics in order picking: Evaluating warehouse layouts for pick, place, and transport vehicle routing systems. *International Journal of Production Research*, *57*(18), 5821-5841.

Lee, W. J., Wu, H., Yun, H., Kim, H., Jun, M. B., & Sutherland, J. W. (2019). Predictive maintenance of machine tool systems using artificial intelligence techniques applied to machine condition data. *Procedia CIRP*, *80*, 506-511.

Lei, Y., Jia, F., Lin, J., Xing, S., & Ding, S. X. (2016). An intelligent fault diagnosis method using unsupervised feature learning towards mechanical big data. *IEEE Transactions on Industrial Electronics*, *63*(5), 3137-3147.

Li, C., Sanchez, R.-V., Zurita, G., Cerrada, M., Cabrera, D., & Vásquez, R. E. (2015). Multimodal deep support vector classification with homologous features and its application to gearbox fault diagnosis. *Neurocomputing*, *168*, 119-127.

Li, X., Hua, G., Huang, A., Sheu, J.-B., Cheng, T., & Huang, F. (2020a). Storage assignment policy with awareness of energy consumption in the Kiva mobile fulfilment system. *Transportation Research Part E: Logistics and Transportation Review*, *144*, 102158.

Li, Y., Carabelli, S., Fadda, E., Manerba, D., Tadei, R., & Terzo, O. (2020b). Machine learning and optimization for production rescheduling in Industry 4.0. *The International Journal of Advanced Manufacturing Technology*, *110*(9), 2445-2463.

Liu, R., Yang, B., & Hauptmann, A. G. (2019a). Simultaneous bearing fault recognition and remaining useful life prediction using joint-loss convolutional neural network. *IEEE Transactions on Industrial Informatics*, *16*(1), 87-96.

Liu, S. Q., Kozan, E., Masoud, M., Zhang, Y., & Chan, F. T. (2018). Job shop scheduling with a combination of four buffering constraints. *International Journal of Production Research*, *56*(9), 3274-3293.

Liu, Z., Guo, S., & Wang, L. (2019b). Integrated green scheduling optimization of flexible job shop and crane transportation considering comprehensive energy consumption. *Journal of Cleaner Production*, *211*, 765-786.

Lohmer, J., & Lasch, R. (2021). Production planning and scheduling in multi-factory production networks: a systematic literature review. *International Journal of Production Research*, *59*(7), 2028-2054.

Lu, C., Wang, Z.-Y., Qin, W.-L., & Ma, J. (2017). Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification. *Signal Processing*, *130*, 377-388.

Malhotra, P., Vig, L., Shroff, G., & Agarwal, P. (2015). Long short term memory networks for anomaly detection in time series. In *Esann*, *2015*, 89.

Masmoudi, O., Delorme, X., & Gianessi, P. (2019). Job-shop scheduling problem with energy consideration. *International Journal of Production Economics*, *216*, 12-22.

Matsumoto, M., & Komatsu, S. (2015). Demand forecasting for production planning in remanufacturing. *The International Journal of Advanced Manufacturing Technology*, *79*, 161-175.

Mende, H., Frye, M., Vogel, P.-A., Kiroriwal, S., Schmitt, R. H., & Bergs, T. (2023). On the importance of domain expertise in feature engineering for predictive product quality in production. *Procedia CIRP*, *118*, 1096-1101.

Meng, L., Zhang, C., Shao, X., & Ren, Y. (2019). MILP models for energy-aware flexible job shop scheduling problem. *Journal of Cleaner Production*, *210*, 710-723.

Meng, L., Zhang, C., Shao, X., Zhang, B., Ren, Y., & Lin, W. (2020). More MILP models for hybrid flow shop scheduling problem and its extended problems. *International Journal of Production Research*, *58*(13), 3905-3930.

Mittenthal, J., & Raghavachari, M. (1993). Stochastic single machine scheduling with quadratic early-tardy penalties. *Operations Research*, *41*(4), 786-796.

Mokhtari, H., & Hasani, A. (2017). An energy-efficient multi-objective optimization for flexible job-shop scheduling problem. *Computers & Chemical Engineering*, *104*, 339-352.

Morariu, C., Morariu, O., Răileanu, S., & Borangiu, T. (2020). Machine learning for predictive scheduling and resource allocation in large scale manufacturing systems. *Computers in Industry*, *120*, 103244.

Mourtzis, D. (2024). Industry 4.0 and smart manufacturing. In *Manufacturing from Industry 4.0 to Industry 5.0* (pp. 13-61). Elsevier.

Naderi, B., & Roshanaei, V. (2022). Critical-path-search logic-based benders decomposition approaches for flexible job shop scheduling. *INFORMS Journal on Optimization*, *4*(1), 1-28.

O'Donovan, P., Leahy, K., Bruton, K., & O'Sullivan, D. T. (2015). An industrial big data pipeline for data-driven analytics maintenance applications in large-scale smart manufacturing facilities. *Journal of Big Data*, *2*(1), 1-26.

Owen, J. H., & Blumenfeld, D. E. (2008). Effects of operating speed on production quality and throughput. *International Journal of Production Research*, *46*(24), 7039-7056.

Özgüven, C., Özbakır, L., & Yavuz, Y. (2010). Mathematical models for job-shop scheduling problems with routing and process plan flexibility. *Applied Mathematical Modelling*, *34*(6), 1539-1548.

Parente, M., Figueira, G., Amorim, P., & Marques, A. (2020). Production scheduling in the context of Industry 4.0: Review and trends. *International Journal of Production Research*, *58*(17), 5401-5431.

Park, I.-B., Huh, J., Kim, J., & Park, J. (2019). A reinforcement learning approach to robust scheduling of semiconductor manufacturing facilities. *IEEE Transactions on Automation Science and Engineering*, *17*(3), 1420-1431.

Paryanto, Brossog, M., Bornschlegl, M., & Franke, J. (2015). Reducing the energy consumption of industrial robots in manufacturing systems. *The International Journal of Advanced Manufacturing Technology*, *78*, 1315-1328.

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. *Proceedings of the 30th International Conference on Machine Learning, 28*(3): 1310-1318.

Petrovic, D., & Duenas, A. (2006). A fuzzy logic based production scheduling/rescheduling in the presence of uncertain disruptions. *Fuzzy Sets and Systems*, *157*(16), 2273-2285.

Petrović, M., Miljković, Z., & Jokić, A. (2019). A novel methodology for optimal single mobile robot scheduling using whale optimization algorithm. *Applied Soft Computing*, *81*, 105520.

Qiao, F., Liu, J., & Ma, Y. (2021). Industrial big-data-driven and CPS-based adaptive production scheduling for smart manufacturing. *International Journal of*

Production Research, *59*(23), 7139-7159.

Qin, H., Xiao, J., Ge, D., Xin, L., Gao, J., He, S., Hu, H., & Carlsson, J. G. (2022). JD. com: Operations research algorithms drive intelligent warehouse robots to work. *INFORMS Journal on Applied Analytics*, *52*(1), 42-55.

Quinton, F., Hamaz, I., & Houssin, L. (2020). A mixed integer linear programming modelling for the flexible cyclic jobshop problem. *Annals of Operations Research*, *285*, 335-352.

Raheja, A., & Subramaniam, V. (2002). Reactive recovery of job shop schedules–A review. *The International Journal of Advanced Manufacturing Technology*, *19*, 756-763.

Rahmani Hosseinabadi, A. A., Vahidi, J., Saemi, B., Sangaiah, A. K., & Elhoseny, M. (2019). Extended genetic algorithm for solving open-shop scheduling problem. *Soft Computing*, *23*, 5099-5116.

Ramírez-Velarde, R., Tchernykh, A., Barba-Jimenez, C., Hirales-Carbajal, A., & Nolazco-Flores, J. (2017). Adaptive resource allocation with job runtime uncertainty. *Journal of Grid Computing*, *15*, 415-434.

Rathore, M. S., & Harsha, S. (2022). Prognostics analysis of rolling bearing based on bi-directional LSTM and attention mechanism. *Journal of Failure Analysis and Prevention*, *22*(2), 704-723.

Rohaninejad, M., Janota, M., & Hanzálek, Z. (2023). Integrated lot-sizing and scheduling: Mitigation of uncertainty in demand and processing time by machine learning. *Engineering Applications of Artificial Intelligence*, *118*, 105676.

Roshanaei, V., Azab, A., & ElMaraghy, H. (2013). Mathematical modelling and a meta-heuristic for flexible job shop scheduling. *International Journal of Production Research*, *51*(20), 6247-6274.

Rossit, D. A., Tohmé, F., & Frutos, M. (2018). The non-permutation flow-shop scheduling problem: A literature review. *Omega*, *77*, 143-153.

Rossit, D. A., Tohmé, F., & Frutos, M. (2019a). A data-driven scheduling approach to smart manufacturing. *Journal of Industrial Information Integration*, *15*, 69-79.

Rossit, D. A., Tohmé, F., & Frutos, M. (2019b). Industry 4.0: Smart scheduling. *International Journal of Production Research*, *57*(12), 3802-3813.

Ruder, S. (2017). An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Schlemitz, A., & Mezhuyev, V. (2024). Approaches for data collection and process standardization in smart manufacturing: Systematic literature review. *Journal of Industrial Information Integration*, *38*, 100578.

Schweitzer, P. J., & Seidmann, A. (1991). Optimizing processing rates for flexible manufacturing systems. *Management Science*, *37*(4), 454-466.

Serrano-Ruiz, J. C., Mula, J., & Poler, R. (2021). Smart manufacturing scheduling: A literature review. *Journal of Manufacturing Systems*, *61*, 265-287.

Serrano-Ruiz, J. C., Mula, J., & Poler, R. (2024). Job shop smart manufacturing scheduling by deep reinforcement learning. *Journal of Industrial Information Integration*, 100582.

Sharp, M., Ak, R., & Hedberg Jr, T. (2018). A survey of the advancing use and development of machine learning in smart manufacturing. *Journal of Manufacturing Systems*, *48*, 170-179.

Shen, X.-N., & Yao, X. (2015). Mathematical modeling and multi-objective evolutionary algorithms applied to dynamic flexible job shop scheduling problems. *Information Sciences*, *298*, 198-224.

Shi, X., Deng, F., Fan, Y., Ma, L., Wang, Y., & Chen, J. (2021). A two-stage hybrid heuristic algorithm for simultaneous order and rack assignment problems. *IEEE Transactions on Automation Science and Engineering*, *19*(4), 2955-2967.

Singh, A., Madaan, G., Hr, S., & Kumar, A. (2023). Smart manufacturing systems: A futuristics roadmap towards application of industry 4.0 technologies. *International Journal of Computer Integrated Manufacturing*, *36*(3), 411-428.

Singh, N., Panigrahi, P. K., Zhang, Z., & Jasimuddin, S. M. (2024). Cyber-physical systems: A bibliometric analysis of literature. *Journal of Intelligent Manufacturing*, 1-37.

Sotskov, Y. N. (2020). Optimality region for job permutation in single-machine scheduling with uncertain processing times. *Automation and Remote Control*, *81*, 819-842.

Sotskov, Y. N., & Werner, F. (2014). Sequencing and scheduling with inaccurate data. In *Sequencing and Scheduling with Inaccurate Data* (pp. 1-432).

Sun, Y., Chung, S.-H., Wen, X., & Ma, H.-L. (2021). Novel robotic job-shop scheduling models with deadlock and robot movement considerations. *Transportation Research Part E: Logistics and Transportation Review*, *149*, 102273.

Tamssaouet, K., & Dauzère-Pérès, S. (2023). A general efficient neighborhood structure framework for the job-shop and flexible job-shop scheduling problems. *European Journal of Operational Research*, *311*(2), 455-471.

Tang, O., & Grubbström, R. W. (2002). Planning and replanning the master production schedule under demand uncertainty. *International Journal of Production Economics*, *78*(3), 323-334.

Teck, S., & Dewil, R. (2022). A bi-level memetic algorithm for the integrated order and vehicle scheduling in a RMFS. *Applied Soft Computing*, *121*, 108770.

Tirkolaee, E. B., Goli, A., & Weber, G.-W. (2020). Fuzzy mathematical programming and self-adaptive artificial fish swarm algorithm for just-in-time energy-aware flow shop scheduling problem with outsourcing option. *IEEE Transactions on Fuzzy Systems*, *28*(11), 2772-2783.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.

Wang, B., Yang, X., & Qi, M. (2022a). Order and rack sequencing in a robotic mobile fulfillment system with multiple picking stations. *Flexible Services and Manufacturing Journal*, 1-39.

Wang, J.-j., & Wang, L. (2019). Decoding methods for the flow shop scheduling with peak power consumption constraints. *International Journal of Production Research*, *57*(10), 3200-3218.

Wang, J., Ma, Y., Zhang, L., Gao, R. X., & Wu, D. (2018a). Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems*, *48*, 144-156.

Wang, P., Gao, R. X., & Yan, R. (2017). A deep learning-based approach to material removal rate prediction in polishing. *CIRP annals*, *66*(1), 429-432.

Wang, S., Liang, Y., Li, W., & Cai, X. (2018b). Big Data enabled Intelligent Immune System for energy efficient manufacturing management. *Journal of Cleaner Production*, *195*, 507-520.

Wang, Z., Sheu, J. B., Teo, C. P., & Xue, G. (2022b). Robot scheduling for mobile-rack warehouses: Human–robot coordinated order picking systems. *Production and Operations Management*, *31*(1), 98-116.

Workneh, A. D., & Gmira, M. (2022). Scheduling Algorithms: Challenges Towards Smart Manufacturing. *International Journal of Electrical and Computer Engineering Systems*, *13*(7), 587-600.

Wu, C.-C., Gupta, J. N., Cheng, S.-R., Lin, B. M., Yip, S.-H., & Lin, W.-C. (2021a). Robust scheduling for a two-stage assembly shop with scenario-dependent processing times. *International Journal of Production Research*, *59*(17), 5372-5387.

Wu, S., Chi, C., Wang, W., & Wu, Y. (2020). Research of the layout optimization in robotic mobile fulfillment systems. *International Journal of Advanced Robotic Systems*, *17*(6), 1729881420978543.

Wu, X., Cao, Z., & Wu, S. (2021b). Real-time hybrid flow shop scheduling approach in smart manufacturing environment. *Complex System Modeling and Simulation*, *1*(4), 335-350.

Wu, X., & Che, A. (2019). A memetic differential evolution algorithm for energy-efficient parallel machine scheduling. *Omega*, *82*, 155-165.

Wu, Y., Yuan, M., Dong, S., Lin, L., & Liu, Y. (2018). Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. *Neurocomputing*, *275*, 167-179.

Xu, D., Shi, Y., Tsang, I. W., Ong, Y.-S., Gong, C., & Shen, X. (2019). Survey on multi-output learning. *IEEE Transactions on Neural Networks and Learning Systems*, *31*(7), 2409-2429.

Xu, S., & Hall, N. G. (2021). Fatigue, personnel scheduling and operations: Review and research opportunities. *European Journal of Operational Research*, *295*(3), 807-822.

Xu, X. (2012). From cloud computing to cloud manufacturing. *Robotics and computer-integrated manufacturing*, *28*(1), 75-86.

Yan, P., Liu, S. Q., Sun, T., & Ma, K. (2018). A dynamic scheduling approach for optimizing the material handling operations in a robotic cell. *Computers & Operations Research*, *99*, 166-177.

Yang, H., Kumara, S., Bukkapatnam, S. T., & Tsung, F. (2019). The internet of things for smart manufacturing: A review. *IISE transactions*, *51*(11), 1190-1216.

Yang, N. (2022). Evaluation of the joint impact of the storage assignment and order batching in mobile-pod warehouse systems. *Mathematical Problems in Engineering*, *2022*(1), 9148001.

Yang, X., Hua, G., Hu, L., Cheng, T., & Huang, A. (2021). Joint optimization of order sequencing and rack scheduling in the robotic mobile fulfilment system. *Computers & Operations Research*, *135*, 105467.

Yanıkoğlu, İ., & Yavuz, T. (2022). Branch-and-price approach for robust parallel machine scheduling with sequence-dependent setup times. *European Journal of Operational Research*, *301*(3), 875-895.

Ye, H., Wang, X., & Liu, K. (2020). Adaptive preventive maintenance for flow shop scheduling with resumable processing. *IEEE Transactions on Automation Science and Engineering*, *18*(1), 106-113.

Yeh, T.-H., & Deng, S. (2012). Application of machine learning methods to cost estimation of product life cycle. *International Journal of Computer Integrated Manufacturing*, *25*(4-5), 340-352.

Yuan, R., Li, J., Wang, X., & He, L. (2021). Multirobot task allocation in e-commerce robotic mobile fulfillment systems. *Mathematical Problems in Engineering*, *2021*, 1-10.

Yue, F., Song, S., Jia, P., Wu, G., & Zhao, H. (2020). Robust single machine scheduling problem with uncertain job due dates for industrial mass production. *Journal of Systems Engineering and Electronics*, *31*(2), 350-358.

Zeng, C., Tang, J., & Yan, C. (2014). Scheduling of no buffer job shop cells with blocking constraints and automated guided vehicles. *Applied Soft Computing*, *24*, 1033-1046.

Zhang, J., Ding, G., Zou, Y., Qin, S., & Fu, J. (2019a). Review of job shop scheduling research and its new perspectives under Industry 4.0. *Journal of Intelligent Manufacturing*, *30*, 1809-1830.

Zhang, J., Liu, C., Li, X., Zhen, H.-L., Yuan, M., Li, Y., & Yan, J. (2023). A survey for solving mixed integer programming via machine learning. *Neurocomputing*, *519*, 205-217.

Zhang, M., & Yan, J. (2021). A data-driven method for optimizing the energy consumption of industrial robots. *Journal of Cleaner Production*, *285*, 124862.

Zhang, Q., Zhang, M., Chen, T., Sun, Z., Ma, Y., & Yu, B. (2019b). Recent advances in convolutional neural network acceleration. *Neurocomputing*, *323*, 37-51.

Zhang, R., & Chiong, R. (2016). Solving the energy-efficient job shop scheduling problem: a multi-objective genetic algorithm with enhanced local search for minimizing the total weighted tardiness and total energy consumption. *Journal of Cleaner Production*, *112*, 3361-3375.

Zhao, R., Wang, D., Yan, R., Mao, K., Shen, F., & Wang, J. (2017a). Machine health monitoring using local feature-based gated recurrent unit networks. *IEEE Transactions on Industrial Electronics*, *65*(2), 1539-1548.

Zhao, R., Yan, R., Wang, J., & Mao, K. (2017b). Learning to monitor machine health with convolutional bi-directional LSTM networks. *Sensors*, *17*(2), 273.

Zheng, P., Wang, H., Sang, Z., Zhong, R. Y., Liu, Y., Liu, C., Mubarok, K., Yu, S., & Xu, X. (2018). Smart manufacturing systems for Industry 4.0: Conceptual framework, scenarios, and future perspectives. *Frontiers of Mechanical Engineering*, *13*, 137-150.

Zhou, T., Tang, D., Zhu, H., & Zhang, Z. (2021). Multi-agent reinforcement learning for online scheduling in smart factories. *Robotics and Computer-Integrated Manufacturing*, *72*, 102202.

Zhuang, Z., Huang, Z., Sun, Y., & Qin, W. (2021). Optimization for cooperative task planning of heterogeneous multi-robot systems in an order picking warehouse. *Engineering Optimization*, *53*(10), 1715-1732.

Zonta, T., da Costa, C. A., Zeiser, F. A., de Oliveira Ramos, G., Kunst, R., & da Rosa Righi, R. (2022). A predictive maintenance model for optimizing production schedule using deep neural networks. *Journal of Manufacturing Systems*, *62*, 450-462.

Appendix A - Traditional RJSP model

The traditional RJSP model without energy considerations is created following Sun et al. (2021). Still using the notations in Table 3.1, the traditional RJSP model is formulated as below.

$$\textit{Obj. Min } \ C_{max} \tag{A.0}$$

s.t.

$$C_{max} \geq SM_F, \tag{A.1}$$

$$SM_F \geq SM_{ij} \qquad \forall i, j \in \{1,2, \dots, |J_{i'}|\}, \tag{A.2}$$

$$SM_{11} = |PM_{11} - PM_D| / v_R, \tag{A.3}$$

$$SM_{ij} + SPT_{ij} + tl_{i(j+1)} \leq \ SM_{i(j+1)}, \qquad \forall i, \forall j \in \{1,2, \dots, |J_i|\} \tag{A.4}$$

$$\sum_m \sum_n X_{ijmn} + Y_{ij(j+1)} = 1, \qquad \forall i, \forall m, i! = m, j \in \{1,2, \dots, |J_i|\}, n \in \{1,2, \dots, |J_{m'}|\}, \tag{A.5}$$

$$\sum_m \sum_n X_{mnij} + Y_{i(j-1)j} = 1, \qquad \forall i, \forall m, i! = m, j \in \{2,3, \dots, |J_{i'}|\}, n \in \{1,2, \dots, |J_{m'}|\}, \tag{A.6}$$

$$\sum_m \sum_n X_{i(|J_i|+1)mn} + X_{i(|J_i|+1)F} = 1, \qquad \forall i, \forall m, i! = m, n \in \{1,2, \dots, |J_{m'}|\} \tag{A.7}$$

$$\sum_m \sum_n X_{mni1} = 1, \qquad \forall i, \forall m, i! = m, i\,! = 1, n \in \{1,2, \dots, |J_{m'}|\}, \tag{A.8}$$

$$SM_{ij} + SPT_{ij} + tl_{i(j+1)} \leq \ SM_{i(j+1)} + \left(1 - Y_{ij(j+1)}\right) \times \beta, \qquad \forall i, \forall j \in \{1,2, \dots, |J_i|\}, \tag{A.9}$$

$$SM_{ij} \geq \ SM_{mn} + tu_{mni(j-1)} + tl_{ij} - \left(1 - X_{mnij}\right) \times \beta, \qquad \forall i, \forall m, i! = m, j \in \{2,3, \dots, |J_{i'}|\}, n \in \{1,2, \dots, |J_{m'}|\}, \tag{A.10}$$

$$SM_{m(n+1)} \geq \ SM_{ij} + tu_{ijmn} + tl_{m(n+1)} - \left(1 - X_{mnij}\right) \times \beta, \qquad \forall i, \forall m, i! = m, j \in (1, |J_i| + 1), n \in (1, |J_m|), \tag{A.11}$$

$$SM_{i1} \geq \ SM_{mn} + tu_{mnD} + tl_{i1} - (1 - X_{mni1}) * \beta, \qquad \forall i, \forall m, i! = m, n \in \{1,2, \dots, |J_{m'}|\}, \tag{A.12}$$

$$Z_{ijhg} + Z_{hgij} = 1, \qquad \forall i, \forall h, \forall j \in \{1,2, \dots, |J_i|\}, g \in \{1,2, \dots, |J_h|\}, M_{ij} = M_{hg}, \tag{A.13}$$

$$SM_{hg} \geq SM_{ij} \ + SPT_{ij} - (1 - Z_{ijhg}) * \beta, \qquad \forall i, \forall h, \forall j \in \{1,2, \dots, |J_i|\}, g \in \{1,2, \dots, |J_h|\}, M_{ij} = M_{hg}, \tag{A.14}$$

$$SM_{ij} \geq \ SM_{hg} \ + SPT_{hg} - Z_{ijhg} * \beta, \qquad \forall i, \forall h, \forall j \in \{1,2, \dots, |J_i|\}, g \in \{1,2, \dots, |J_h|\}, M_{ij} = M_{hg}, \tag{A.15}$$

$$SM_{hg} \geq SM_{i(j+1)} \ + tu_{i(j+1)h(g-1)} + tl_{hg} - (1 - Z_{ijhg}) \times \beta, \qquad \forall i, \forall h, \forall j \in \{1,2, \dots, |J_i|\}, g \in \{2, \dots, |J_h|\}, M_{ij} = M_{hg,}, \tag{A.16}$$

$$SM_{ij} \geq SM_{h(g+1)} \ + tu_{h(g+1)i(j-1)} + \ tl_{ij} - Z_{ijhg} \times \beta, \qquad \forall i, \forall h, \forall j \in (2,3, \dots |J_i|), g \in \{1,2, \dots, |J_h|\}, M_{ij} = M_{hg,}, \tag{A.17}$$

$$SM_{h1} \geq SM_{i(j+1)} \ + tu_{i(j+1)D} + \ tl_{h1} - (1 - Z_{ijh1}) \times \beta, \qquad \forall i, \forall h, \forall j \in \{1,2, \dots, |J_i|\}, M_{ij} = M_{h1}, \tag{A.18}$$

$$SM_{i1} \geq SM_{h(g+1)} \ + tu_{h(g+1)D} + \ tl_{i1} - Z_{i1hg} \times \beta, \qquad \forall i, \forall h, \forall g \in \{1,2, \dots, |J_h|\}, M_{i1} = M_{hg}, \tag{A.19}$$

$$X_{iJmn} \in (0,1), \qquad \forall i, \forall m \in \{1,2, \dots |I| + 1\}, i! = m, j \in \{1,2, \dots, |J_{i'}|\}, \mathrm{n} \in \{1,2, \dots, |J_{m'}|\}, \tag{A.20}$$

$$Y_{ij(j+1)} \in (0,1), \qquad \forall i, \forall j \in \{1,2, \dots, |J_i|\}, \tag{A.21}$$

$$Z_{iJhg} \in (0,1), \qquad \forall i, \forall h, i! = h, \forall j \in \{1,2, \dots, |J_i|\}, g \in (1, |J_h|), \tag{A.22}$$

$$SM_{ij} > 0, \qquad \forall i, \forall j \in \{1,2, \dots, |J_{i'}|\}, \tag{A.23}$$

$$tu_{ijmn} = |PM_{ij} - PM_{mn}| / v_R, \qquad \forall i, \forall m, j \in \{2,3, \dots, |J_{i'}|\}, n \in \{1,2, \dots, |J_{m'}|\}, \tag{A.24}$$

$$tl_{ij} = |PM_{ij} - PM_{i(j-1)}| / v_R, \qquad \forall i, \forall j \in \{2,3, \dots, |J_{i'}|\}, \tag{A.25}$$

$$tl_{i1} = |PM_{i1} - PM_D| / v_R, \qquad \forall i, \tag{A.26}$$

$$tu_{ijD} = |PM_{ij} - PM_D| / v_R \qquad \forall i, \forall j \in \{1,2, \dots, |J_{i'}|\}. \tag{A.27}$$

Constraints (A.9-A.11) play a similar role as constraints (9-12) in the proposed RJSP-E. While these constraints are a bit different as the RJSP-EM involves a variable RM to capture the removing time of jobs. Moreover, it is worth noting that by implementing the traditional model without energy consideration, neither the machine nor the robot can change the operating speed. Therefore, in constraints (A.3), (A.9), (A.13), and (A.14), the processing time under the normal speed is used to replace the processing time under a specific operating speed scale (i.e., SPT_{ij} is used to replace PT_{ij}). As other constraints in (A.0) - (A.27) are covered by model RJSP-E, they are not repeated here.