Vaasan yliopisto
UNIVERSITY OF VAASA

Timo Stenvall

Production scheduling in electrical switch manufacturing

School of Technology and Innovations
Master’s thesis in Industrial Management
Master's Programme in Industrial Management

Vaasa 2025

UNIVERSITY OF VAASA
School of Technology and Innovations
Author: Timo Stenvall

Title of the thesis: Production scheduling in electrical switch manufacturing
Degree: Master of Science, Economics and Business Administration
Discipline: Industrial Management
Supervisor: Petri Helo
Year: 2025 **Pages:** 131

ABSTRACT:

The optimisation of production schedules is a widely researched topic with several approaches attempting to find optimal or near optimal schedules. The practical task of production scheduling however appears less researched in comparison to the optimisation of the algorithms.

This thesis aims to find a practical production scheduling system for a case company that currently has only informal, unorganised production scheduling systems in place. The literature is reviewed concerning a wide array of topics related to the practical production scheduling task to find the assumptions and practical limitations of the different production scheduling approaches.

The dynamicity and the due date complexity of the scheduling problem in the case company is analysed using historical data gathered from the ERP system of the company and analysed both as one averaged set of data and as a collection of individual weeks. These are used to both understand effects and dynamicity of the production environment given by the due dates and a measure of the frequency of disruptions that are likely to cause a reschedule for predictive scheduling.

The work proposes critical considerations for the case company in their implementation of a production scheduling system along the needs of practical production scheduling identified from the literature. Three approaches to perform production scheduling that are aligned with the practical scheduling task are proposed, since there is no clear single best option for production scheduling for the current production environment.

KEYWORDS: Production scheduling, Practical production scheduling, Finite scheduling, Practical production scheduling task, Scheduling, Scheduling systems

UNIVERSITY OF VAASA
School of Technology and Innovations
Author: Timo Stenvall
Title of the thesis: Production scheduling in electrical switch manufacturing
Degree: Master of Science, Economics and Business Administration
Discipline: Industrial Management
Supervisor: Petri Helo
Year: 2025 **Pages:** 131

TIIVISTELMÄ:

Hienokuormitus ja siihen liittyvä tuotannon aikataulutuksen optimointi on laajalti tutkittu aihe. Siihen liittyy useita erilaisia algoritmisia ja matemaattisia lähestymistapoja, jotka pyrkivät tuottamaan optimaalisia tai lähes optimaalisia tuotantosuunnitelmia. Sen sijaan taas hienokuormitus käytännön tehtävänä on kirjallisuudessa vähemmän edustettu aihe.

Tämä opinnäytetyö pyrkii muodostamaan järjestelmän hienokuormituksen käytännön tehtävään tapausyritykselle, jonka hienokuormitus tapahtuu tällä hetkellä suunnittelemattomalla ja yritystason näkökulmasta organisoimattomalla tavalla. Kirjallisuuskatsauksessa käsiteltiin laajalti erilaisia käytännön hienokuormituksen tehtävään liittyviä aiheita. Näistä pyrittiin löytämään hienokuormituksen lähestymistapojen erilaisia perusolettamuksia ja rajoitteita, jotta yrityksen hienokuormitusjärjestelmän valinnassa voidaan varmentaa lähestymistavan soveltuvuus.

Tuotantoympäristön dynaamisuutta ja monimutkaisuutta tutkittiin hienokuormituksen näkökulmasta käyttäen dataa yrityksen ERP-järjestelmästä. Dataa analysoitiin sekä keskiarvoilla koko ajanjaksolta, että verraten eri viikkoja keskenään. Näiden perusteella pyritään selvittämään sekä töiden kiireellisyyden aiheuttama hienokuormituksen monimutkaisuus, että tuotantoympäristöstä aiheutuvien tuotantosuunnitelmaa häiritsevien tapahtumien tiheys, joiden johdosta kuormitussuunnitelmaa pitää suunnitella uudelleen.

Opinnäytetyö tuloksenaan koostaa ehdotuksen, jossa on nostettu esiin keskeisiä tekijöitä, joita käytännön hienokuormitussysteemissä tulisi huomioida kirjallisuuden mukaan. Tämän lisäksi työssä ehdotetaan kolmea erilaista lähestymistapaa hienokuormitukseen yrityksen tuotantojärjestelmä huomioon ottaen, sillä kirjallisuuden ja järjestelmän analyysin perusteella ei pystytä suosittelemaan yhtä selkeää parasta mallia.

KEYWORDS: Production scheduling, Practical production scheduling, Finite scheduling, Practical produc-tion scheduling task, Scheduling, Scheduling systems

Contents

Pictures

Figures

Tables

Abbreviations

1 Introduction

Production scheduling proposes benefits attainable with little investment. The proposal is that production scheduling can eliminate some of the wastes in a company – unnecessary idle time, production queues between workstations and setup times. The solution proposed by production scheduling does not require change in the processes to reduce these wastes, rather it argues that the optimisation of the job sequence and schedule can handle this reduction.

Given how production scheduling appears to be a low-cost waste reduction strategy, the interest of production companies is apparent. This thesis approaches the production scheduling subject through seeking to understand how production scheduling could be performed in a case company manufacturing electrical switches. There are three research questions in the thesis:

1. What options does the theory present for the implementation of production scheduling, and how do these approaches fit the case company's production environment?
2. How could production scheduling be used in the case company to ensure the best possible material flow?
3. Can the existing ERP system of the case company be used as a tool for production scheduling?

1.1 Production scheduling in the case company

The case company does not have a defined system for production scheduling. This, however, does not mean that production scheduling is not done; decisions about what to make and when are performed daily on several levels by several participants. The functionalities of production scheduling are performed on some level in manufacturing businesses, even if the tasks are implicit and unformalized.

Literature focusing on the theoretical research of scheduling often does not recognize that in the practice a lack of a production scheduling system cannot be represented by complete randomness or a dispatching rule proven to be suboptimal. In practice, a functional company without a formal production scheduling system will have an informal and emergent production scheduling system that could even be very evolved and fitting for the environment.

The emergent production scheduling system inside a company can be expected to be well-adapted to the daily needs of the company with a proven performance – if the system was not acceptable on some level, the existing system would have evolved further. This has a noticeable effect on the requirements of a defined production scheduling system. On the performance metrics it must offer improvements over the existing system, a system that is not defined in any way and the only really known quality of which is the general overall outcome of production. On another side, the defined system will likely need to be perceived as an improvement to the stakeholders of the older production scheduling system to achieve long-term adoption.

The measurement of the existing system would be complicated and is not attempted in this study. As the system is spread out over several actors without any standardisation in the processes, the analysis would have to be done for each scheduling unit, likely a production line in the case company, separately. In addition to this, separation of the emergent system to analyse it would very likely misrepresent the current informal production scheduling system.

This is not, however, an argument that a production scheduling system should not be formalised. The act of designing a production scheduling system will, even if no other performance measures were to be attained, provide organisational benefits. One of the benefits of defining an organised scheduling approach in a production company would be the increased predictability of the system and the ability to tune the system, as the components are known. There also exists an inherent risk in having an informal production scheduling system: the system mainly exists in implicit knowledge of the actors.

1.2 The approach to the scheduling system problem

In this thesis the applicability of production scheduling approaches is investigated from two viewpoints and in two directions. First, literature is used to build a view of production scheduling task functions and the different practical possibilities. This view is further enriched by analysing the environment for the scheduling task. In combination this builds a description of the role of the production scheduler. The production scheduler interacts with a production scheduling tool to fulfil their role. While the needs of the scheduler inform the requirements for the tool, the capabilities of the tool define the practical possibilities of the production scheduling task.

The scheduling task is viewed not just as a task that takes the production situation as an input and provides schedules as an output. It is considered an extended role that interfaces with other functions, namely manufacturing operations, sales, procurement and the wider business environment.

The practical task of production scheduling is context dependent (Romero-Silva et al., 2024). Due to the context dependency, the environment the production scheduling system exists in must be determined to find a correct approach for production scheduling. The application of scheduling tools to practical production systems is a critical issue in practical scheduling that has not been addressed widely in the literature (Framinan & Ruiz, 2012).

A relevant question for companies seeking to implement production scheduling is: what are the benefits over our current state? This question is complicated from the view where the performance of production scheduling systems' outcome is dependent on the environment the system is in, as a system from one company may not provide benefits in another. The outcomes from production scheduling tools may not just be dependent on the production environment, but the organisation around the production scheduling task as well.

Cegarra and van Wezel (2011) argue that the analysis of a production planning and scheduling system should be device independent, meaning that the people and tools should not inform the analysis of the organisation of the planning function. They argue that device dependency limits the possibilities of organisational design. This view is partly used in this thesis. The literature review portion attempts to consider the subject as device independently as possible. However, since the business can function without a formal production scheduling process, the interest in the practice of performing formal production scheduling arises largely from the capabilities provided by the scheduling tools. From this view the design of a scheduling task, in this context, is inherently device dependent.

1.3 Deterministic schedules in a stochastic reality

The approach to production scheduling where a predictive schedule is made assumes that the future state of the production floor can be represented by a schedule. The usefulness of this assumption is not without debate. If a view is taken where deterministic models cannot be used to make decisions with inherent variability, this assumption is broken.

Thus, there is a need to consider the different production scheduling approaches created for different levels of dynamicity, the scheduling outcome capabilities of the approach outcomes. In addition, the actual outcomes of the schedule execution are crucial, as the scheduled performance may not always correspond to the practical scheduling task performance.

1.4 Systematising schedules

Moscoso et al. (2010) found evidence in a case study that even when companies do not formally consider schedules and planning at a detailed level, there are still rescheduling decisions made, and that these decisions can be detrimental to the overall performance. The case company they Moscoso et al. analysed employed order chasers whose work consisted of expediting rush orders with higher priority. These order chasers did not organise their work with the rest of the planning and were suspected to cause notable amounts of order backlog and decreased service level. Moscoso et al. note that this caused a desynchronisation between planning and the actual shop floor. While the production was planned as equal orders with static lead times, a two-tier system of production orders existed in reality with highly variable lead times, the expedited orders and the rest, which were constantly postponed.

It is worth noting that even if a company does not directly employ order chasers, the functional role of order chasers can be covered by several personnel in a factory. The problems induced in the system by distributed order chaser roles may be even harder to quantify. While organised production planning and scheduling will practically typically involve prioritisation of a set of jobs over the other, the organised system is more likely to account the variability induced to the overall planning and thus avoid instability, the planning bullwhip, caused by the prioritisation (Moscoso et al., 2010).

2 Literature review

The literature review was performed with a scoping approach, since the research question cannot be answered with a limited focus on any specific area of production scheduling research. The initial search for literature was made for production scheduling in general and production scheduling articles containing keywords relating to practical cases and implementations. The reference lists and keywords of these seed articles were used to build a comprehensive list of practical production scheduling topics.

Due to the wideness of the topics inside the scope, two further scope limitations were made. Since the outcome of the research is not to propose a scheduling algorithm but to support a selection among ready tools, purely theoretical research in scheduling algorithms is handled only at a level that is relevant to practical use from commercially available solutions. Another limitation is that the discussion of scheduling is performed from the point of view of the case company's production environment.

2.1 Hierarchy of production planning

Production scheduling is a function at the lowest, greatest detail level of the production planning hierarchy. According to Martinsuo et al. (2016) production planning is done at several levels. The highest level, strategic planning, makes long timeframe decisions about the overall production strategy, the available capacity, positioning, and outsourcing. The highest level uses demand estimates, as precise information is not available for such a timeframe and the decisions cannot be made in time with only the available information.

When moving to a lower level of planning, the estimate precision increases as the planned timeframe decreases. The middle level of production planning, rough cut capacity planning, plans the usage of the existing capacity for expected work (Martinsuo et al., 2016). At this level of planning, the capacity often is not modelled exactly, but rather with approximate methods considering larger groupings of workstations (Haverila et al., 2009). This level of planning is often done with master production scheduling performed with an ERP software (Martinsuo et al., 2016).

Production scheduling is a precise production plan that explicitly considers the workstations and how the work interacts with the workstations, often created by line supervisors (Martinsuo et al., 2016). It is based on the existing production orders.

Martinsuo et al. (2016) notes that the information flow in planning hierarchy is not purely top-down, the higher levels may need to update the plans according to information from lower-level plans. Pinedo, p. (2009, p. 9) notes that while the decisions made in the higher level of planning create the constraints for the scheduling, the scheduling process creates and verifies in practice the information about the practical capability of the production line.

The traditional division into separate hierarchies has limitations, as the higher levels of planning typically do not consider the rough schedule level of the production along with the possible levers of operations, possibly creating infeasible plans for the lower level of planning (Yao et al., 2022).

The interface between production scheduling and planning can be argued to be context dependent. Morandini et al. (2024) discuss production planning and scheduling in aerospace manufacturing, where production stages for products with long manufacturing times, such as planes, are planned in detail by production planners, while scheduling tracks short tasks within these phases. In such products production planning can be similar to project planning and scheduling. In contrast in companies manufacturing a wide mix of products with short production times, the interface and distinction between production scheduling and production planning is different.

The hierarchy of production planning creates an important relationship between the higher levels of production planning and production scheduling. The outputs of the strategical planning and production planning functions have an effect on the outcomes of the production scheduling task, and thus attempting to optimise the production scheduling task without considering production planning may lead to diminishing returns.

2.2 Flow patterns

A key category describing the scheduling problem to be solved is the flow pattern, also called the machine configuration (Abedinnia et al., 2017; Pinedo, 2009). There exist numerous adaptations of flow patterns in the literature, but the patterns relevant to multiple machine manufacturing can be divided into three main categories: flow shops, job shops and open shops. The scheduling approach and outcomes are dependent on the layout (Kamaruddin et al., 2013).

Flow shops have identical job routings that are predetermined (Abedinnia et al., 2017). Practically this describes most production lines with a set order of operations for a product, where every product in the production line traverses through the production line passing all the workstations in a set order. This reduces the decision points necessary in the problem, as the items from one step of the process will all move to a certain next step in the process. In the general model of the flow shop, the work order between jobs may be allowed to be rearranged between the steps, or restricted to maintain the same sequence (Pinedo, 2009, p. 23). The restriction that does not allow jobs to change order when inside the flow shop is called the permutation flow shop, and is a typical restriction added to simplify the computational complexity of the scheduling, as the sequence needs to only be constructed once for each starting sequence of jobs (Perez-Gonzalez & Framinan, 2015; Pinedo, 2012).

Job shops have predetermined routings for the jobs, but the routings may be individual to each job (Abedinnia et al., 2017). A practical example of such a problem is a machine shop, where each type of product has a set order of operations. Another product might have a different order of operations or skip some of the operations the other products have. This increases the number of decisions to be made in the schedule, as the products must compete for the entry order for each workstation in the system, and the inputs for one workstation can arrive from several other workstations.

Finally open shops are a flow pattern where there are no limitations on the routing of the products (Abedinnia et al., 2017). The operations in open shops do not need to be performed in any specific order, and thus there is a greater amount of options for scheduling compared to job shops and flow shops.

2.3 Online and offline scheduling problems

There is a division in scheduling approaches between offline and online approaches. Offline scheduling covers the predictive, or future planning approach, where a production schedule is created using a pool of jobs waiting to be processed. Online scheduling does not require a pool of work as input as it schedules the work as the orders appear (Raheja & Subramaniam, 2002). This also means that online scheduling can be performed even if a pool of work, a requirement for offline scheduling, is not available. While online scheduling tends to arrive at solutions that are not strictly optimal from the offline perspective, this is balanced by the notion that information about orders that have not arrived yet is unavailable to offline scheduling as well (Tokola et al., 2014). While online scheduling may not be able to find the global optimum schedule, it cannot be guaranteed that there is always enough time for the offline solution to pool jobs enough to guarantee the true optimal schedule either.

According to Tokola et al. (2014) where the optimal makespan of an offline scheduled flow shop is found with a permutation flow shop where the jobs maintain their order through the system, in online scheduling this is not necessarily true. They note that when new jobs arrive, the optimal online schedule might have jobs overtaking others during the production flow.

The applicability of either the offline or online scheduling approach appears to depend on whether a set pool of jobs is practical to create and the stability of the production environment. If jobs have long enough lead time in the system, then the jobs can be collected towards a pool for periodic offline scheduling. If the lead time for jobs is very short, then the online approach of scheduling them immediately is the only feasible option. Online scheduling, however, has less stability provided to the shop floor, which can be of critical importance if the scheduling is used to guide material flow.

2.4 Objectives and constraints

Framinan & Ruiz (2012) argue that constraints and objectives are closely associated in practice. A feature of the problem might be a constraint in one model and an objective in another. In heuristic and metaheuristic solution approaches the constraints often are modelled through objectives by assigning high costs to unwanted areas (Méndez et al., 2006). While exact mathematical models can have strict constraints, such as not allowing any jobs to be late, this same is often implemented in approximate methods by making late work be very expensive for the schedule cost estimation.

Framinan and Ruiz (2012) note that objectives are difficult to accurately define in the planning phase before the actual implementation of a scheduling system. They argue that production schedulers can define the objectives with greater precision if the objectives are developed along with the scheduling system incrementally. While at the beginning they can only state general goals, over an incremental implementation these transform into practical and actionable scheduling objectives.

Framinan and Ruiz (2012) argue that production scheduling objectives are for short-term goals and thus might not necessarily need to be directly aligned with strategic or long-term objectives. They note that while the long-term goals are important, a short term operationally important goal should be preferred instead for the scheduling algorithm.

There is a design concern with production scheduling algorithms when there are several objectives to be considered simultaneously. According to Jarboui et al. (2013) there are three approaches a scheduling algorithm can use to adjust to several objectives.

- The a priori approach
- The a posteriori approach
- The interactive approach

In the a priori approach where the objective functions are combined or prioritised before the optimisation task to find a single criterion and thus solution. The a posteriori approach is where a set of efficient solutions, called the pareto front, is presented to the scheduler to select from. The pareto front consists of solutions where an increase in any objective requires the decrease of another objective (Pinedo, 2012, p. 79). Finally in the interactive approach the algorithm asks the scheduler for input during the run of the algorithm.

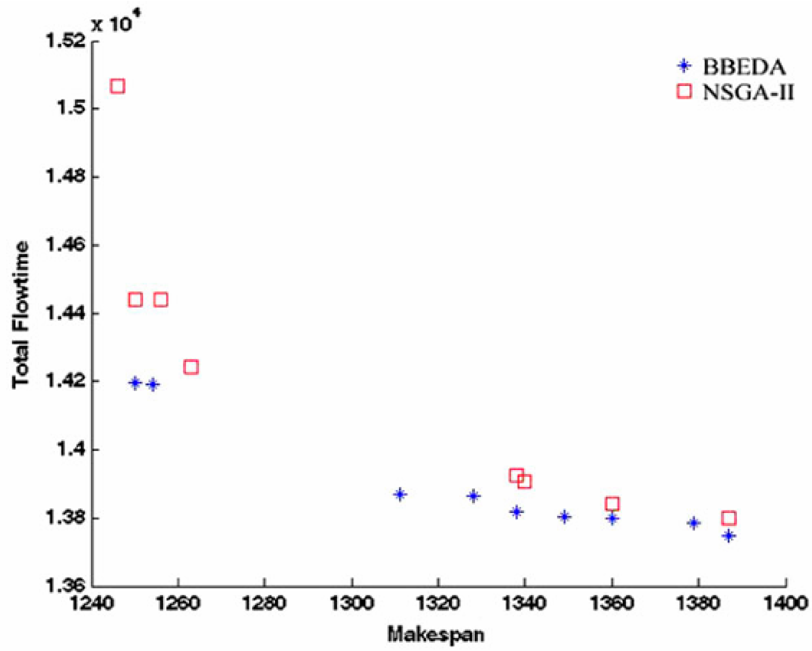


Figure 1 An example of representation of pareto-optimal solutions (Tiwari et al., 2015).

A matter that does not appear to be discussed in literature related to pareto fronts is the challenge of visualising the outcome. For two objectives there is a clear and intuitive single representation as in figure 1, the bivariate chart. As the number of objectives in the front increases, the representation becomes more challenging, and thus there can be expected to be more challenges for the decision maker comparing the different pareto optimal solutions.

2.4.1 Makespan

Minimizing the makespan, the time taken to process all the jobs, is the oldest objective function for the flow shop scheduling problem (Dudek et al., 1992). This goal results in the maximum overall utilization of resources when the resources are static. Dudek et al. (1992) argue that while the typical makespan minimization problem in the academic literature uses traditional manufacturing to present the scheduling problem to be solved, makespan minimization is typically not found interesting in practice in such environments.

Miyata & Nagano (2019) found in their review of blocking flow shop literature that completion time based objectives are the most represented among objectives. Within the category of completion time objectives, makespan considered as a single objective is the most typical objective function. They however note that this dominance of the makespan objective in research has been diminishing in recent years. Cohn et al. (2010) note that results with nearly identical makespan measures can vary significantly when measured with other objectives. This would indicate that makespan as an only measure is not sufficient to distinguish between practically good and bad schedules.

Makespan is used to increase the utilisation of the production line resources. There is, however, a drawback in allowing the scheduling algorithm to completely control the utilisation level. Hopp and Spearman (2011, p. 317) note that due to the inherent variation in the production system, as utilisation increases, both the cycle time and the WIP inventory in the process increase exponentially. This view causes a problem for the reduction of the makespan and all utilisation increasing objectives, as they would need to be countered by another objective attempting to control the WIP and the cycle time.

This problem is further complicated by the nature of the effect, being born out of variability. A manufacturing system representation that is based on a deterministic model of the manufacturing system typically does not generate the information needed to perform this balancing. While a schedule performing exceptionally badly with the makespan criteria can be guaranteed to be non-optimal, this does not prove that a schedule with the minimal makespan is truly optimal. There appears to be an open question: is production scheduling based on deterministic modelling really a suitable tool for optimising utilisation?

2.4.2 Flowtime

Flowtime, the time a product spends in the manufacturing system, is another time-based objective. According to Rajendran (1994) flowtime minimisation results in schedules that have more even utilisation on resources, lower idle time, lower WIP and overall lower scheduling cost compared to makespan minimisation.

2.4.3 Earliness and tardiness scheduling problem

When the utilisation of production is low enough, a very simple and intuitive dispatching rule known as the earliest due date rule can assure that all the jobs are performed in time. This rule however has a problem if the workload has spikes that approach or exceed the capacity, it does not make compromises to limit the amount of late work.

According to (Pinedo, 2009, pp. 29–31), there are several typical objectives that concentrate on different types of due date lateness. The minimisation of maximum lateness attempts to minimise the lateness of the job with the greatest lateness. The minimisation of the number of late jobs attempts to keep the number of jobs that are late as low as possible even if the few late jobs are very late. Minimisation of weighted tardiness attempts to minimise the total tardiness of all the jobs with a weight factor performing prioritisation of the jobs, effectively combining the minimisation of the number of jobs and the lateness of jobs, with a weight factor deciding when the capacity necessitates some late jobs.

The combination of earliness and tardiness objectives creates a scheduling problem that penalizes both early and late finishing of each job (Beck & Refalo, 2003). The extension of this problem that considers each work operation within each job in a job shop requires a specific algorithm to solve and is called the Just-in-time job-shop scheduling problem (Ahmadian & Salehipour, 2021).

This approach, however, has the potential to be very sensitive to variation, as a deterministic model is made where each operation has an exact time window, and the points of failure for the schedule multiply, as every job is performed as late as possible assuming no variation. In addition, the problem of production scheduling between flow shops and job shops differ in this approach. While there is a greater need for coordination in a job shop environment and thus a greater need for more detailed modelling due to the competition of jobs for workstations from several sources, flow shops might perform better with a less detailed approach. Flow shops can be argued to perform better in practice with an input sequence and the inter-operation flow handled by the design of the flow line (Hopp & Spearman, 2011, p. 547). If the production line already handles jobs just-in-time with buffering that is sensible when accounting for the variability, production scheduling might struggle improving this.

2.4.4 Sequence dependent setup times

Setup times are highly meaningful for most practical scheduling problems because setups are non-value adding tasks that can consume significant amounts of time, yet more than 90 percent of all production scheduling literature does not account for setup times (Allahverdi, 2015). When the time taken to setup for a production change is dependent on the previous product, this is called the sequence dependent setup time (Pinedo, 2009, p. 26). There is also the term of sequence dependent cost, which is equivalent to the setup time if the two are proportional (Allahverdi, 2015).

A typical representation of sequence dependent setup times is the setup matrix that contains the products both as the rows and columns. The setup time when the sequence changes from one product to another can be read on the corresponding cell of the matrix.

	Product A	Product B	Product C	Product D	Product E
Product A	X	42 min	15 min	5 min	13 min
Product B	5 min	X	7 min	9 min	14 min
Product C	12 min	16 min	X	16 min	17 min
Product D	30 min	43 min	16 min	X	26 min
Product E	55 min	18 min	24 min	44 min	X

Table 1 A visual example of a sequence dependent setup matrix.

2.4.5 Blocking lines and buffers

While production scheduling is performed with a finite capacity view, there still often is assumed to be infinite capacity between workstations. Blocking models are at the other end of the spectrum where there is no capacity between workstations. If an upstream workstation finishes its' work before a downstream workstation is free in a blocking system, it must wait (Miyata & Nagano, 2019). Often neither of these is an accurate model representing material flows in manufacturing.

Pinedo (2012) states that in production scheduling, systems with limited buffers are modelled as systems without any buffer space, with the buffers being modelled as production machines with no processing time or processing cost. According to him, this significantly simplifies the mathematic modelling of such systems. The same can be assumed to work for the implementation of buffers into the manufacturing environment models of production scheduling software.

2.5 Cyclic scheduling

Cyclic scheduling is a special case of scheduling possible under certain conditions. According to Pinedo (2012, pp. 435–439) cyclic scheduling attempts to build a schedule that provides the product mix in line with the demand that can be repeated over and over. According to him this means building a minimum part set, the smallest set of products with the same proportions as the production mix. This set can then be repeated over and over with minor variations to cover the demand.

In practice, this approach seems to have benefits if the product mix is representable by a minimum part set. This schedule is very predictable, as the schedule for the most part is just the same schedule over and over. Material flows would also be very predictable, as the cyclic schedule is very similar to flow levelling approaches in JIT manufacturing as described by Hopp & Spearman (2011, p. 161). Empirical fine-tuning of the cyclic schedule to get production floor benefits should be feasible due to the repeated nature.

Cyclic scheduling approaches typically assume product setup times to be irrelevant for scheduling (Pinedo, 2012, p. 436). Here another parallel to the JIT flow levelling can be found with the target of single minute exchange of dies. In pull environments this levelling is performed to create a smooth demand for components. Compared to other offline scheduling approaches, which only create information in advance of the variation to material logistics, cyclic scheduling approaches directly reduce the variability of the material flow.

A cyclic schedule however does not directly consider due-dates – the cyclic schedule must be able to intrinsically meet due dates, and it is hard to fit together with production where a set of products are packed for a customer delivery directly. Cyclic schedules require a relatively predictable demand and product mix within the cycle time to be beneficial over other periodic scheduling methods.

2.6 Classifying scheduling solution methods

When considering general approaches, there can be said to be three approaches to building schedules. The first approach is using heuristics and dispatching rules that consider a set of tasks one production job at a time. The other two approaches, metaheuristics and exact methods attempt to consider all the jobs at the same time and perform optimisation to the production schedule.

2.6.1 Heuristics and dispatching rules

Due to the computational difficulty in solving production scheduling problems with exact methods, heuristics are often applied instead (Hopp & Spearman, 2011, p. 521). Heuristics can be considered as rulesets that can be applied to the problem as an algorithm to arrive at a schedule. A defining property of heuristics is the lack of case-by-case optimisation, but rather the application of a predetermined ruleset built into the heuristic to arrive at a solution. These rules are built to arrive at a good solution when the underlying assumptions of the rule hold.

According to Ruiz and Maroto (2005) heuristics can be categorised into two approaches: constructive heuristics that attempt to create a schedule, and improvement heuristics used to improve an existing schedule. In this subchapter only constructive heuristics will be discussed, and improvement heuristics will be relevant later when discussing metaheuristics.

Dispatching rules are among the simplest scheduling approaches. They are a typical approach to completely reactive scheduling (Ouelhadj & Petrovic, 2009). In comparison to offline optimising heuristic methods, common dispatching rules, such as FCFS, SPT and LPT, have been found to result in significantly lower solution performance in some environments (Ruiz & Maroto, 2005).

However dispatching rules can be used to make predictable decisions for the production schedule. They can be used to build offline schedules, but also to implement completely dynamic online scheduling, where scheduling is performed whenever a new production order appears. Dispatching rules can be optimal for a certain machine environment regarding a single parameter, given that the assumptions of the rule hold (Pinedo, 2012, pp. 376–378).

Single dispatching rules use a measure to select among candidates for work, such as the earliest due date -rule (Pinedo, 2012, p. 376). If there is a clear single parameter that can define the effective sequence of production, even these simple rules can build good schedules.

The single dispatching rules can be combined to make composite dispatching rules, as described by Pinedo (2012, pp. 377 - 382). According to Pinedo, composite dispatching rules combine several dispatching rules with weight factors to each rule to generate a decision according to several parameters. An example of such a rule outlined by him is the apparent tardiness cost, which selects work by a weight factor divided by processing time, called the shortest processing time rule and combines this with an evaluation of the minimum slack rule, which selects jobs with the smallest slack time until the production must be started to meet the due date. The weight factor of each dispatching rule in a composite rule must be selected, and for the apparent tardiness cost rule, the overall due date tightness and the range of due dates should be considered (Pinedo, 2012, p. 379). Another typical extension of the apparent tardiness cost rule is to include setup times, where the cost of switching to the next product is used in the consideration.

In their simulation analysis of several dispatching rules in a flow shop Oukil and El-Bouri (2021) found that the most effective dispatching rule depends on the tightness of due dates and the congestion level. Their approach suggests that for practical reactive scheduling or dispatching scheduling, a matrix of rules could be selected according to the congestion and the current due-date tightness of the product mix waiting to be manufactured. This approach attempts to overcome the limitation posed by the assumptions and limitations of each individual rule.

Another theoretical shortcoming of the dispatching rules lies in how the rules are used; the apparent best first step is chosen using the rule, and from there on the rule selects the next best option. The approach locks into a particular sequence of products after the first item is selected. The sequences created by dispatching rules are not able to select a slightly worse option for the current sequence step to avoid a significantly worse outcome later.

2.6.2 Exact and approximate optimisation methods of optimisation

When production schedules are built with optimisation, there are two approaches: exact mathematical methods and approximate methods. The exact methods formulate the optimisation problem to be solved exactly. In these models a global optimum point, the guaranteed best value of the model, can be found. The computational requirements for exact approaches increase when the problems have any complexity in a highly nonlinear fashion, where increasing pure processing power quickly meets diminishing returns (Hopp & Spearman, 2011, pp. 522–523). Both the size of the problem, being number of machines and jobs, and the number of constraints modelled increase the complexity of the problem (Klemmt et al., 2009).

The approximate methods use approaches that attempt to approach the optimum points. According to Méndez et al. (2006) These approaches do not know whether they have reached the true global optimum point, as this would require knowledge only given by the exact methods. Furthermore, they note that while an exact method strictly considers all the constraints, the constraints in approximate methods are modelled by adding penalties for violations of the constraints. This can cause approximate methods to struggle finding solutions that do not violate the constraints for complex problems with many constraints.

The key benefit of approximate methods is that the algorithm can be given a limited time to attempt to solve the problem, and the output is the best result found within that time. Klemmt et al. (2009) found evidence suggesting that after a certain problem size, the approximate methods are better suited. They proposed a system where an approximate simulation-based system identifies the bottleneck of the problem, an exact method solves this smaller bottleneck problem, and the approximate simulation method then builds a schedule with this exact bottleneck schedule. Similarly, the metaheuristic algorithm by Ahmadian & Salehipour (2021) uses linear programming to solve a small portion of the problem and heuristic methods to build the schedule using the information from the mathematic model.

Méndez et al. (2006) suggest that in practice the difference between guaranteed optimal exact solutions and approximate solutions might not be relevant. They note that scheduling solutions in practice must be generated within a short time window and that the model only accounts for a limited set of the actual scheduling goals. In addition to these limitations, they note that the implementation of the schedule is rarely perfect in the shop floor and that the dynamicity of the industrial environment weakens the optimality. This reasoning can be seen to be supported by De Snoo et al. (2011) who argue that especially when there are disruptions and uncertainty in the production floor, the optimality of the schedule becomes less important than the overall process of scheduling.

In practice, the optimal solution might not be truly any more optimal than a good, non-optimal solution. A counterpoint to this thinking, however, is that the approximate solution cannot be guaranteed to even be close to the optimal. This gap can however be empirically tested, and has been found to typically be small, for example by Klemmt et al. (2009).

Zupan et al. (2024) note that heuristic and metaheuristic algorithms typically have parameters that are relevant to the performance of the algorithms. They note that this in turn makes theoretical and computational comparison difficult as the algorithms can be tuned to work with specific datasets. Ensuring equal level of parameter tuning for each algorithm in order to perform accurate testing is hard.

Exact methods are typically defined and customised to solve specific scheduling problems. While there are commercial solvers for the mathematical methods used for exact modelling, there do not appear to be any commercial software packages for generic modelling of production and production scheduling. These approaches require formulation by the implementer.

2.6.3 Metaheuristics

Metaheuristic algorithms are approximate optimisation methods that use iterative processes to improve a schedule. They can use local search methods (Jarboui et al., 2013, p. 9) or they can be modelled after nature, such as genetic algorithms, ant colony optimization and particle swarm optimization (Liang et al., 2023). Where directly mathematical solution methods are widely considered and mostly proven NP hard and thus difficult to solve at a reasonable scale, these meta-heuristic approaches approach optimisation by selecting a set of non-optimal solutions and evolving the solutions over iterations to find solutions with greater optimality. This detaches the amount of potential solution appraisals from the problem parameters, as all the possible combinations do not need to be checked.

Zupan et al. (2024) note that metaheuristics make fewer assumptions about the problem than heuristics and as such can be better suited for general usage. In comparison purely heuristic algorithms typically are more based on assumptions about the functionality of the production environment.

The major theoretical drawback of these methods for optimisation, however, is the risk of the algorithm getting stuck on a local optimum point. Due to the genetic algorithm finding the optimum by making small corrections to the existing solution, there is a risk that it finds a solution that is the best among its neighbours but is not the best solution overall (Liang et al., 2023). To be able to escape this local optimum, the algorithm would have to move in a non-optimal direction, as the nearby region points to the local optimum being the best direction. Here an extra challenge is the fact that the global optimum point is unknown, and that the local optimum points do not give any kind of indication about whether they are local or global optimum points. There are often used approaches that attempt to deter the algorithms from getting stuck in local optimum points (Liang et al., 2023; Pinedo, 2009, pp. 452–458).

2.7 Forwards- and backwards scheduling

Production scheduling can be performed in two directions, forwards and backwards. These terms are used both in infinite and finite capacity production planning and scheduling. This term describes which direction production is planned. Forwards planning starts from the first possible start date of production forwards and places the job at the first possible time, backwards scheduling starts from the due date and moves back in time (Haverila et al., 2009).

Even with production schedule optimisation this parameter can have an effect when there is idle time created by slack capacity. Whenever possible, backwards scheduling will attempt to start any job as late as possible and may leave empty space in the schedule. Forward scheduling will pack the front of the schedule, leaving most of the idle time at the back unless forced to wait by release dates. This can influence what happens when new production orders appear with tighter due dates, as a backwards scheduled schedule could be expected to fit the arriving job with less changes. A backwards schedule might also be slightly more fragile to disruptions, as there is, by definition, as little buffer between the scheduled finish and the due date as possible.

Kamaruddin et al. (2013) suggest that job shops might benefit from backwards scheduling, while cellular layouts benefit from forward scheduling. They found that cellular layouts in their simulation had the lowest throughput time, lateness, and highest labour productivity when the scheduling was performed in the forwards direction. It is however worth noting that their study had a relatively simple heuristic scheduling algorithm. It could be expected that linear heuristic algorithms are more sensitive to the scheduling direction than metaheuristic improvement algorithms and exact methods of optimisation.

A production scheduling vendor Simio discusses scheduling direction in their blog (Simio, 2019). They argue that, in their experience, forwards scheduling is always the right choice. The blog post argues that backwards scheduling in production scheduling can only tell what should have been done, not what can be done if the schedule is not feasible. According to them, forwards scheduling on the other hand will schedule over capacity production in the future with some jobs overdue, giving the scheduler a chance to make compromises. They further note that forward scheduling does not necessarily mean a greater finished goods storage, as the scheduler has control over the release dates for each job, allowing the postponement of jobs to avoid finished goods storage. They note that the release date allows the scheduler to try different scenarios of earliness.

2.8 The dynamic scheduling problem

Scheduling problems, or how the problem to be solved is perceived, can be classified with two sets of categories: static and dynamic (Abedinnia et al., 2017; Dudek et al., 1992). Of these approaches, static deterministic approaches make up most of the models presented in the literature (Abedinnia et al., 2017; Dudek et al., 1992; Moratori et al., 2012). Some articles in the literature use the distinction to online and offline scheduling problems independently of this categorisation, while some use them interchangeably with static and dynamic (Abedinnia et al., 2017).

Where the typical scheduling solutions are created to solve a deterministic problem where all the jobs to be scheduled are known and the manufacturing environment is predictable, dynamic scheduling approaches attempt to create methods for dealing with situations where reactions to real-time events are necessary. The primary types of real-time events considered in the literature are resource-related, such as machine failures, capacity shortages and material shortages and job-related, such as rush jobs arriving, job cancellation and changes in due-dates (Ouelhadj & Petrovic, 2009).

Katragjini et al. (2013) note that there is no existing comprehensive body of procedures and practices for dealing with a dynamic production environment. They also note that most of the existing literature into dynamic scheduling only focuses on a single type of disruption, while in the practical dynamic scheduling case often several diverse types of disruptions, each calling for different actions, occur.

According to Ouelhadj and Petrovic (2009) there exist three categories for dynamic scheduling approaches: reactive, predictive-reactive, and robust pro-active. These will be discussed in depth in their own separate sub-chapters below along with the special considerations for each approach. The approach to the practical scheduling task by each of the strategies with some further variations can be seen in figure 2, as represented by Moratori et al. (2012). The reactive and predictive-reactive strategies have a set approach to schedule repair and rescheduling approaches, but robust pro-active approaches can either attempt to create plans that do not require rescheduling at all, or reserve rescheduling for extreme disruptions.

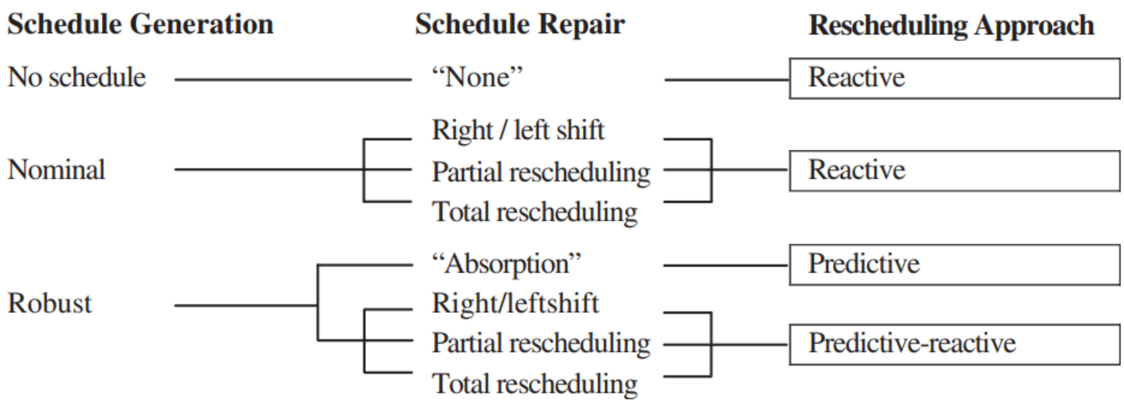


Figure 2 Schedule generation, repair, and rescheduling approaches (Moratori et al., 2012)

Dynamic scheduling, and rescheduling especially, has an interesting interaction with the scope of the scheduled system. Ouelhadj and Petrovic (2009) discuss that the sensitivity and speed of reaction to disruptions, along with the impact of rescheduling, is linked to whether a global schedule containing all the production lines, or individual schedules for each line are maintained. They argue, that as the distance between the disruption and the scheduler increases, the time taken to react to the disruption increases. According to them, local scheduling for smaller units, such as individual production lines, is not as likely to reach the global optimal schedule as a global scheduling scheme containing all the lines but is more responsive to disruptions. This could potentially mean better results in practice overall, as in some dynamic systems capturing slightly smaller benefits reliably could be more beneficial than intermittently captured larger benefits.

Ouelhadj & Petrovic (2009) note multi-agent based dynamic scheduling systems as a possible solution to the scale problem in the literature. This approach uses agent-based simulation where the production lines and jobs are represented by agents that create separate schedules for each of the machines by the jobs auctioning, contracting or negotiating with each other for the resources. This system is seen in the literature as an effective way to build robust and cost-effective schedules that are not dependent on a central top-down hierarchy. However, to my knowledge commercial scheduling applications using this approach are rare.

2.8.1 Reactive scheduling

Reactive scheduling refers to practices where a predictive schedule is not made, and dispatching rules are used instead to select among the available jobs (Ouelhadj & Petrovic, 2009). While this approach does not fulfil all the goals of a scheduling system, it requires little knowledge about the future as input information. Romero-Silva et al. (2024) suggest in their framework that production environments with little similarity between jobs, and thus low predictability, and high dynamism fit this approach the best. The reason behind the fitness is that these approaches require very little information to be performed.

2.8.2 Predictive-reactive scheduling

According to Ouelhadj and Petrovic (2009), predictive-reactive scheduling is an approach where typically static schedules are made to predict the optimal schedule, and then the plans are adjusted according to a rescheduling strategy. Typically the rescheduling strategies are either to reschedule when a critical disruption is detected or rescheduling periodically. There is also the hybrid of the two approached with periodic rescheduling with additional rescheduling when a notable disruption occurs.

A further distinction between approaches is whether the schedule is rescheduled completely or if the past schedule is considered or repaired (Ouelhadj & Petrovic, 2009).

A key issue in rescheduling is an effect called shop floor nervousness (Ouelhadj & Petrovic, 2009). This effect happens when a production schedule is rescheduled due to a change, and the new overall schedule is significantly different from the last schedule. The practical downsides of this effect can be intuitively deducted: If material is picked or otherwise preparation is performed based on the schedule, the change of near-term schedule causes problems. Additionally, when a global schedule is made for several workstations, the schedule in one workstation may change even if it does not directly interact with the workstation where the dynamic event happens due to the rescheduling.

The shop floor nervousness can be seen as a key practical problem limiting complete rescheduling of the whole factory in reaction to a disruption out in environments where the disruptions in the schedule happen with a certain frequency. The question of whether to schedule centrally or distributed becomes relevant here again, as local fixes to the schedule concerning just one line are likely easier to perform than repairing a schedule containing all the production lines.

Dong and Jang (2012) note that rescheduling procedures have goals typically divisible to two categories: minimisation of the degradation of efficiency, used when the production delay cost is high, and minimisation of the impact on shop floor stability, used when the costs caused by resource re-allocation are high.

An approach to overcome the problem of shop floor nervousness is to attempt to repair the original schedule in some way to maintain as much of the original schedule as possible while reacting to the disruption without performing another scheduling operation (Ouelhadj & Petrovic, 2009).

Abumaizar and Svestka (1997) found in their research on job shop rescheduling that complete rescheduling did not in most cases provide a statistically significant improvement in performance over a method called Affected Operations Rescheduling, where the order of jobs is maintained. This was, however, partly due to the availability of idle time in the original schedule that the algorithm holding the order of operations stable could use to accommodate the disruption. Their research was performed on a simulated job shop environment, the amount of idle time in a flow shop might be different.

Dong and Jang (2012) however proved that their algorithm that changes the production order sequence achieved greater stability than the Affected Operations Rescheduling that maintains the order, when the change in job completion times is taken in account. While the algorithm changes the sequence of work, the stability of which is often solely used to measure overall stability, it managed to keep the new jobs closer to their original completion times. Effectively their approach exchanged the stability objective of the maintenance of the production sequence to the maintenance of the production schedule.

A relatively simple approach of schedule repair maintaining the scheduled order while consuming as much of the idle time available as possible is a valid approach and is likely appropriate if a change of the production sequence would create disruptions in the material flow or cause other setup costs. At its simplest, just performing an extra, non-scheduled job, could even be performed on the production floor directly, as it does not require significant computation. If, however, the costs caused by late jobs are significant in comparison to the costs incurred by changing the production sequence, algorithmic approaches to rescheduling can provide improvement in a critical metric – due dates.

Perez-Gonzalez and Framinan (2015) studied two approaches to schedule updating upon arriving jobs while there are scheduled jobs waiting to be processed. The first approach is a frozen zone -policy, where the jobs scheduled for a near-period are considered frozen and the added items are added wherever slack in the schedule allows. The second approach they considered allows the rescheduling of the production including the set of already scheduled jobs.

The study by Perez-Gonzalez and Framinan (2015) found that the approach allowing the rescheduling of existing jobs gives better performance for the makespan of the schedules. They however give support to the practical benefits of the frozen zone -policy on rescheduling by noting that the rescheduling approach can be computationally restrictively difficult, the maintenance of service level being difficult for the existing scheduled jobs. This is in addition to the other practical benefits of the frozen zone that are derived from the lower shop floor nervousness.

Another possible approach to reschedule while attempting to control the shop floor nervousness is the generation of a new schedule while considering closeness to the original schedule as a goal in the schedule generation or rescheduling only a part of the original schedule (Moratori et al., 2012; Ouelhadj & Petrovic, 2009). The property describing the similarity of the new schedule the old is called stability (Moratori et al., 2012).

Moratori et al. (2012) proposed a set of match-up algorithm strategies for a dynamic job shop environments and conducted simulated experimentation to compare the match-up strategy to the optimality of a complete reschedule and the stability of a simple right-shift and insert-at-end strategies when a new job arrives to the system. The match-up strategy counts the available slack time inside a schedule to find a window inside the production schedule where rescheduling can be performed to accommodate the new job, while the rest of the schedule remains unchanged. Effectively they find a smaller limited rescheduling problem inside an existing schedule. Their experimentation found that the match-up strategy could fit the new arriving job to the existing schedule reliably with a performance that is statistically indistinguishable from complete rescheduling and a stability that is close to the simple strategies that have near-perfect stability.

A practical consideration identified by Moratori et al. (2012) regarding make-up scheduling is that the local part of the schedule that is to be rescheduled should have the minimization of makespan as the primary goal. They note that if the new schedule cannot be constructed inside the rescheduling window, the rest of the schedule will have to be delayed to fix the overlap of the schedules, and that this delay is negatively correlated with the overall schedule performance. This causes every other objective for the specific make-up scheduling problem to be secondary to the length of the make-up schedule.

The make-up approach, however, might have some additional limitations in practical use. The performance of the approach interacts with the available slack time, as the window to be rescheduled is dependent on the available time. Another consideration is whether the match-up approach suits better to job shop -environments, where the jobs' flows are more complicated, and more options are available to reorganize the flow. In flow shop environments the rescheduled solution might be similar to results obtained by simply right-shifting. Additionally, flow shop environments are likely to have less slack time, since all the jobs have the same flow pattern.

2.8.3 Robust pro-active scheduling

Robust pro-active scheduling, as its' name suggests, takes a proactive approach to mitigating the need to reschedule and thus aids in the reduction of shop floor nervousness. In robust pro-active scheduling, the goal of the scheduling process is to create a schedule that can withstand disruptions predictably (Ouelhadj & Petrovic, 2009). Full proactive scheduling would attempt to absorb all the variability in the schedule, but this goal is difficult to achieve and might compromise the optimality of the schedule too much. In practice, a combination of reactive re-scheduling strategy with some proactive consideration to reduce the rescheduling frequency is a practical approach (Li & Ierapetritou, 2008).

When robustness is not considered as an objective in the schedule optimization process, the fragility of the output is unknown, and the scheduling algorithm will select the solution with the best performance, even if there would be a significantly more robust solution with nearly similar performance. When some form of robustness is considered in the objective function of the schedule, the algorithm should select the robust option among the possibilities, if such an option exists. The measure for robustness, however, is hard to define (Ouelhadj & Petrovic, 2009).

Li and Ierapetritou (2008) discuss the implementation of robustness in scheduling algorithms in their literature review. The implementation of stochasticity, or the inclusion of randomness into the scheduling model, depends on the scheduling algorithm approach. Typically, the implementation of stochasticity consists of several tests for each schedule using different expected values for the steps and using standard deviation to analyse which schedules have a predictable performance. They however note that at least at the time of the review, few applications to full-scale practical environments have been made due to the increased complexity of the approach.

Notably exact methods can support a mathematical definition of robustness better, as these methods tend to have awareness of the type and quality of optimality with methods such as sensitivity analysis.

Lin et al. (2004) noted that schedules optimised without considering the variation can be highly sensitive to even small variations between the parameters used to schedule and the true performance. They proposed a method that accounts for bounded uncertainty for several variables in a mixed linear integer program. They note that the method output practically creates schedules with some additional time for the operations.

Relevant to exact methods of scheduling, Bertsimas and Sim (2004) discuss approaches to robust optimisation and the price of robustness, how much optimality is lost to ensure robustness. Some traditional approaches to robust optimisation perform optimisation with worst case values for all the possible uncertain variables. They note that these approaches are overly conservative and often sacrifice too much optimality to ensure robustness. They propose a method for mathematical optimisation where the level of covered uncertainty can be selected, giving a solution that only covers a set level of uncertainty.

2.9 Rescheduling

Rescheduling is a necessity of dynamic scheduling when the original schedule turns infeasible. The approach to rescheduling is likely not only important to the optimisation found in the schedule, but also the overall trust in the scheduling process: if schedules are often broken and not fixed, the adherence to scheduling and motivation to schedule will likely decrease in the organisation. This could launch a negative feedback loop between the input quality of the schedule, quality of the schedule and the outcomes.

Some approaches to rescheduling are reactive, such as the usage of dispatching rules to select work to available machines if the schedule becomes infeasible (Tokola et al., 2014). With such approaches, the schedule is only followed when possible. Whenever a need to reschedule arises, reactive measures are used until another schedule is made.

Pinedo (2012, p. 491) notes the influence of scheduling robustness and rescheduling in scheduling software design requirements. He notes that despite this he there is little theoretical research on the topic. One critical quality in the schedule recovery process is the speed at which the schedule can be repaired. If a schedule is repaired, the repair must be performable in a reasonable time to avoid the new, repaired schedule being outdated by the time it is finished (Raheja & Subramaniam, 2002)

Raheja and Subramaniam (2002) note that in schedule repair, the involvement of the scheduler's expertise is important since disruptions can vary significantly. This does pose an interesting requirement for the scheduling tool: the tool should provide effective methods for schedule repair while allowing the scheduler to adapt the approach to the specific disruption. All this, as noted earlier, should be doable quickly enough to avoid downtime while waiting for the schedule repair.

De Snoo and van Wezel (2014) argue that rescheduling requires a different organisational approach to initial scheduling. They note that while there typically is time available to perform initial scheduling in a serialized fashion, schedule by schedule, rescheduling is characterized by shorter time available, and thus a greater focus on coordination is required if there are several schedulers. If there are several schedulers, they suggest that coordination between schedulers should be prescribed and facilitated to arrive at higher quality rescheduled schedules.

Li et al. (2020) investigated using machine learning to decide a rescheduling point. They note that the selection of when to reschedule is complicated considering the need to balance rescheduling costs with the degrading performance of a schedule. They built a machine learning model that attempts to identify the correct time to reschedule for highest impact. They noted that the approach performed better compared to periodic rescheduling by selecting high impact rescheduling points with less variation in impact compared to periodic rescheduling policies.

Some of the functionality of their approach can be captured without a machine learning model with a production scheduler and a tool allowing the comparison of schedules. If the scheduling software allows for testing and comparison of a proposed schedule to the existing schedule, the human scheduler can select reasonable reschedule points and reject rescheduling with little benefit.

2.10 Schedule robustness with practical approaches

According to Pinedo (2009, pp. 374–379) the scheduling robustness refers to the ability of a schedule to perform after some assumptions made during the creation of the schedule break, such as a rush-job arriving or machine breakage. Pinedo asserts that robustness goals depend on the system that is scheduled and the frequency and severity of the disruptions. In addition, the level of ability to reschedule and the way that the system responds to rescheduling impacts the need of schedule robustness.

Pinedo (2009, pp. 378–379) outlines several practical rules not directly dependent on the scheduling algorithm that can be used to improve the ability of the schedule to sustain under changing conditions. These rules are as follows:

- Inserting idle time in the schedule
- Scheduling less flexible jobs first
- Limiting postponement when possible

- Maintaining an input buffer in front of bottlenecks

The first rule, inserting idle time in the schedule, outlined by Pinedo (2009, p. 378) is a scheme where extra time is inserted into the schedule to effectively limit the maximum utilisation of the workstations. He also notes that some schedulers in practice limit the utilisation of the workstations in a rolling fashion the further the plan progresses in time towards future to account for the increasing likelihood of disruption. An example of such a scheme is to only schedule up to a certain utilisation percentage, for example 80% more than a week ahead.

The second rule by Pinedo (2009, p. 378) considers the flexibility of jobs in the scheduling order. According to him inflexible jobs could be for example jobs with highly sequence-dependent setup times or high machine specificity. If there are several machines the job could be processed in, the rule attempts to leave such jobs to be done later. However, the practical examples of such jobs are more typical in the job-shop manufacturing environment or environments with parallel machines.

The third rule proposed by Pinedo (2009, p. 378) is to limit how just-in-time an order is dispatched to the manufacturing environment. Some time is left to spare if a disruption occurs, causing the postponement of all jobs. Pinedo notes that this approach causes a trade-off between inventory holding costs due to the extra inventory of finished products and the robustness of a schedule. When this logic is followed in the other direction, an assertion can be made that inventory cost minimization as the sole objective for a production schedule can decrease the robustness of a production schedule. This rule can however be considered from the point of view of changing customer orders, as discussed by Cohn et al. (2010). From this point of view, earlier production of products exposes the company to risks of overproduction if customers change order contents after the production has started. There is a trade-off in the robustness depending on the proportions of customer order content uncertainty and production stoppage uncertainty.

The last rule, as listed by Pinedo (2009, pp. 378–379) is to maintain a buffer of jobs in front of workstations with high utilization to prevent the bottleneck of the process starving for jobs if an upstream workstation has a fault.

The controls presented by Pinedo (2009, pp. 378–379) are largely analogous to typical manufacturing environment controls for a production line in the face of variability, as presented by Hopp and Spearman (2011) especially the rules number one and four. Hopp and Spearman note that when production lines are modelled with variability, the controls to limit the degradation of performance caused by the variability are to lower the utilisation of the workstations through increasing capacity and to increase the WIP level in the process, especially feeding the capacity constraint process. The second rule can also in part be interpreted as a kind of a queue-sharing dispatching rule, where the items are sorted in order by how suitable jobs are for queue sharing, which acts as a variance pooling strategy (Hopp & Spearman, 2011, p. 300).

Cohn et al. (2010) propose a measure they call responsiveness to balance the packing of jobs inside a production schedule. This measure can be considered a robustness measure. The responsiveness measure attempts to maximise the number of different product types within a portion of the schedule and thus divide the batches where possible. Here an analogue can be seen to minimum part sets. They suggest that production schedules, especially ones created optimizing the makespan, can create schedules that are inherently slow at reacting to customers' order changes by packing all production of one product type in a single large batch. They describe a situation where all the products of one type are made in one packed batch on the production sequence. If customers change the demand mix by reducing the need for that product during the run of the schedule after the large batch has been produced, the product ends up overproduced. They note that the effect of the responsiveness objective varies depending on the setup cost levels, as higher setup costs tend to prefer larger product type batches and lower higher responsiveness.

2.11 Scheduling environment dynamicity

Tubilla and Gershwin (2022) used the frequency of the times a machine cycles through all its' products to the frequency of machine failures to estimate the nature of the production environment in their simulation study on the single machine dynamic scheduling problem. They found that scheduling approaches based on deterministic approaches' capability in comparison to reactive approaches failed fast when the disruptions in the process became less frequent but larger.

Tubilla and Gershwin (2022) offer intuition into the nature of the disruptions by noting that when there are frequent but small disruptions, the disruptions can over time average to be reasonably representable by a deterministic, but lower capacity system. This intuition can also be stated in the form of variability buffering. Low amplitude variability can be buffered though reducing the utilisation of the resources (Hopp & Spearman, 2011, pp. 309–318)

Abumaizar and Svestka (1997) argue that most production floor disruptions are similar enough to machine breakdowns when viewed from the scheduling process, that the machine breakdown research can be utilised to model different kinds of disruptions. A notable exception to this approach would be the arrival of rush orders, which causes new jobs to be added to the schedule.

2.12 The gap between scheduling research and practice

Among articles discussing the practical implementation of production scheduling, there is a tendency to find a gap between the theoretical research and practical applications of production scheduling methods arising from most research's too narrow focus on just scheduling algorithms in isolation instead of the wider task of production scheduling (Abedinnia et al., 2017; Cegarra & van Wezel, 2011; Dudek et al., 1992; Ouelhadj & Petrovic, 2009; Romero-Silva et al., 2015; Tenhiälä, 2011).

Romero-Silva et al. (2024) have identified the gap between the theory and practice of scheduling to be attributable to three main causes: simplification of the scheduling problems, simplification of the practical scheduling task and suitability of the traditional approach to all environments. In addition to this, they have identified some efforts required to cross the gap. The three identified causes are used as a structure to discuss this topic further below.

2.12.1 Simplification of the scheduling problems

The first theory-practice gap arises from the identified problem of theoretical solutions mainly solving the simpler variations of the scheduling problem compared to their real-world application counterparts (Romero-Silva et al., 2024).

Carvalho et al. (2014) note however that even from the practical perspective, there is a balance that should be found in the complexity of the problem. They note that while a more complex model can better represent the production environment consistently, the added complexity inhibits the use and maintenance of the model.

Framinan and Ruiz (2012) argue that a careful consideration should be given to the complexity of the scheduling system to ensure that a scheduling approach is not overly complicated in relation to the goals of the scheduling and the magnitude of the benefits. They note that a pareto or A-B-C analysis might be undertaken to identify what should be accounted for in the scheduling.

The practical-minded interpretation of this compromise is that the scheduling system has some required complexity to ensure a sufficient representation of the production environment to make decisions that are actually beneficial to the production system. After a certain point, however, the marginal benefits of additional representativeness will decrease, especially when the problems arising from deterministic approximations of dynamic real production systems are considered.

The basis for informing this compromise, however, may need theoretical research to identify the components that have relevant interactions with the scheduling approaches.

2.12.2 Simplification of the practical scheduling task

Simplification of the scheduling task, according to Romero-Silva et al. (2024), causes a theory-practice gap since most of the literature in scheduling still considers the scheduling as a combinatorial problem. Such problems are interested in finding an optimal solution to a model of manufacturing or a prebuilt problem set, and not as a tool to be implemented to a practical environment. While the solutions offered by such articles could provide an optimal solution to the model it is tested with, the question of how well the solution is implementable to practical use remains unanswered. The application of scheduling tools to practical production systems is a critical issue that has not been widely addressed in the literature (Framinan & Ruiz, 2012).

Carvalho et al. (2014) state that few studies consider practical implementation of scheduling systems. They note that to solve this problem, academic literature needs practical cases and structured information about the challenges in implementation arising from the complexity of the practical application in actual company implementations.

De Snoo et al. (2011) note that production scheduling literature has overwhelmingly focused on measuring the performance of the product of scheduling: the theoretical performance of the schedule. They argue that in practice this view is too narrow, and the process of scheduling is as important, and often more important, from the viewpoint of applying scheduling to practice. They performed case studies into companies and found that companies find performance of the scheduling process more important than pure schedule performance especially when there is high uncertainty.

Moscoso et al. (2010) note that simulation-based studies about the effects of planning and scheduling do not provide full information. The note that models underrepresent the amount of planning flexibility available. The effects of decision models can be different when modelled with strict rules in simulation, as human actors consider wider amounts of information and typically have a wider selection of tools for the problems.

2.12.3 Suitability of the traditional approach to all environments

This identified gap translates to context dependency of the solutions, and the lack of applicability analysis in the literature to identify the correct contexts for each scheduling solution. Generally, solutions proposed by scheduling literature do not attempt to find and define the limits of where the solutions are applicable. This same problem partly exists in commercial scheduling software, the underlying assumptions of the model and thus the limits of applicability are often not stated.

Romero-Silva et al. (2024) conclude that productions scheduling is context dependent based on a review of the literature concerning the applicability of scheduling methods. The finding practically means that the scheduling approach should be selected based on the environment it is used in.

In their comparison of scheduling heuristics, Ruiz and Maroto (2005) found performance results significantly different from results claimed by past comparison, when using a different set of benchmark problems for scheduling. This could indicate that even in the realm of theoretical test benchmarks, the outcome of approaches is dependent on the specific problem format.

In addition, Ruiz and Maroto (2005) argue that the overall context of most companies' operating environment has changed since the development of the initial scheduling models as companies attempt to create more customer-focused interactions. According to Romero-Silva et al. (2024) this increases the disturbances in the scheduling environment, such as an increasing amount of order arrivals and changes in specifications.

Effectively this change could mean that there are less sources of variability pooling and buffering between production lines and the customers.

2.12.4 Addressing the theory-practice gap in a practical case

Knowledge of the theory-practice gap can be used to make meaningful decisions about the implementation of a production scheduling system considering the gaps as well as informing some of the factors necessary for a productive choice of scheduling software. In this regard, the scheduling software should aid in the practical adaptation of the underlying scheduling model to streamline the practical scheduling task.

Romero-Silva et al. (2024) state that the three issues should be tackled simultaneously in a practical application, as the solutions for each of the problems support each other. They note that generating an accurate characterisation of the system including the constraints, resources, variation and stability informs the solution to the problem of simplified overall task of scheduling. The scheduling as a task requires analysis of the needed monitoring and inputs, which arise from the characterization of the system. In addition to this, they note that neither of the problems can be feasibly solved without ensuring that the solution method is relevant to the environment.

2.13 The practical production scheduling task

Romero-Silva et al. (2015) note that while there is a commonly used and accepted framework for studying the technical implementation of scheduling problems, there is a lack of a solid framework for understanding practical application cases for scheduling. Furthermore, they note that there are indications in the literature that the theoretical research may not be completely applicable due to too many simplifying assumptions. To help alleviate this problem they propose a framework for identifying the process components of a practical production scheduling system by modifying a feedback control system.

This system will be used in this text as a basis for understanding the process of scheduling and the key information flows in the process. The framework contains the schedule construction itself, a function that by itself is a typical scope for theoretical scheduling research. In this framework it is extended by considering how information flows back to the scheduling task. The output of the production schedule creation step, the created schedule, is implemented by the manufacturing environment. Disturbances affect the implementation of the schedule. Finally, a monitoring process attempts to collect meaningful information about the manufacturing environment to generate input to the scheduling task to reschedule or improve the scheduling process.

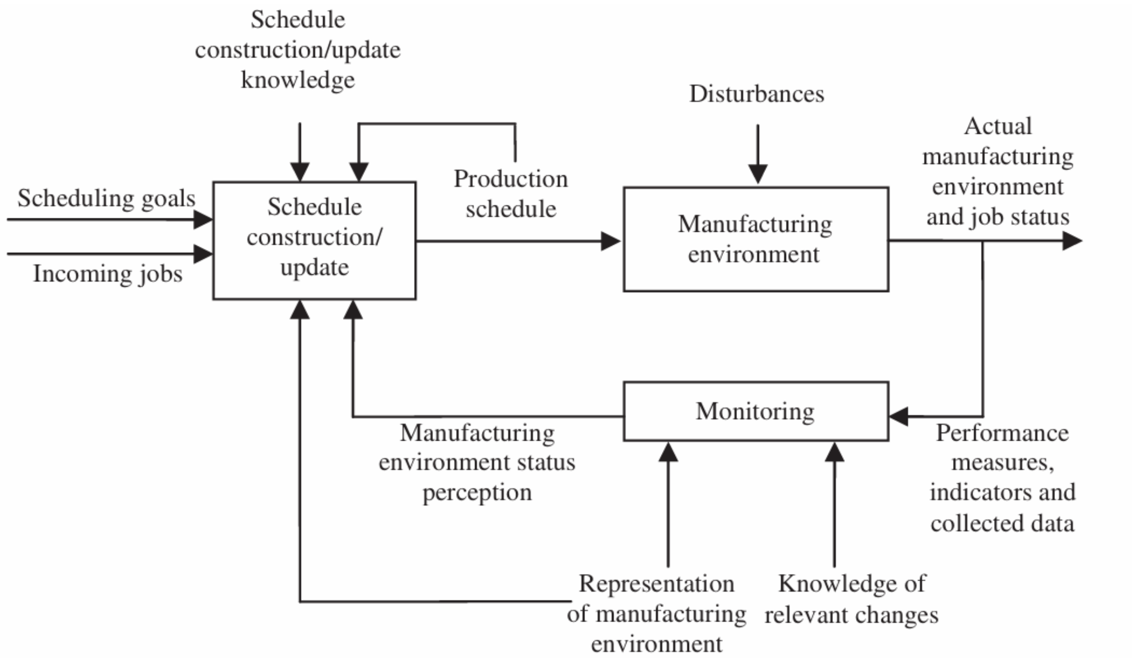


Figure 3 The practical production scheduling task (Romero-Silva et al., 2015).

Considering the relative maturity of the theoretical schedule construction process and the incompleteness of the literature considering the input loop feeding information from the production environment to the scheduler, the monitoring portion of the process can be argued to be of critical interest when implementing a production scheduling system. This is especially true when the production scheduling is performed through a commercially available software solution, and the ready solution used for schedule construction must be provided information about the manufacturing environment in a form that is both usable by the scheduling solution and relevant to the manufacturing environment.

The monitoring process needs to find the correct level of information to feed back into the schedule generation task and the parameters of the scheduling solution must be tuned in reaction to the monitored state. Fransoo and Wiers (2008) found in their case study of a company's scheduling practice a tendency of the production schedulers to prefer manually fixing the schedules rather than changing the parameters of the scheduling tool to improve the schedule generation. They suggested that this could be caused by incomplete training of the schedulers to the working of the scheduling software and the perception that the software does not have enough accuracy to generate truly good schedules.

A consideration for the constant tuning of the monitoring process and scheduling parameters, however, is that a production scheduler might not have the necessary information at hand to productively change the parameters of the scheduling software to account for future uncertainty. An example of such a problem is reaction to some form of a variation, that is bound to naturally regress to the mean (Fransoo & Wiers, 2008). The system's overall problem in this regard is finding a way for the production schedulers to use their experience and expertise to tune the scheduling system while maintaining the statistical validity of the system and avoiding overreacting to different situations and thus increasing the variation in the whole manufacturing system. From this perspective it would be important to change the parameters of the scheduling solution when a deficiency in the overall system configuration is found and refrain from changing the parameters for causes that are not relevant in the general context. Discriminating between the two options, however, is likely not trivial in practice.

2.14 Choosing the right practical scheduling approach

Tenhiälä (2011) suggests that a production scheduling approach should be selected based on the process type that is scheduled with a focus on the availability of information and predictability of the process. As the more detailed loading approaches require more exact parameters for the scheduling system, he notes that job shops do not have the required repetition of products to feasibly find the parameters. Production lines, however, have highly repetitive production runs, and the parameters required to perform finite loading can be determined. He suggests that optimisation in scheduling is only possible when the process to be scheduled can be represented by a static formulation. Tenhiälä suggests that capacity planning and loading has an optimal level of accuracy for the different process types as presented in figure 4.

<table>
<tr><th></th><th>Job shop</th><th>Batch process</th><th>Batch process with bottleneck control</th><th>Production line</th></tr>
<tr><td>Process complexity
■ resources
→ different kinds of routings</td><td>Planning points = 1</td><td>Planning points > 1</td><td>Planning points = 1</td><td>Planning points = 1</td></tr>
<tr><td>Task interdependence</td><td>Pooled</td><td>Reciprocal</td><td>Sequential around the bottleneck</td><td>Sequential</td></tr>
<tr><td>Non-systematic capacity planning</td><td colspan="4">Not recommendable for any environment due to high exposure to human error and variance in planners' personal competences</td></tr>
<tr><td>Rough-cut capacity planning (RCCP)</td><td>Fit</td><td colspan="3">Unfit due to insufficient precision</td></tr>
<tr><td>Capacity requirements planning (CRP)</td><td rowspan="3">Unfit because the high variety of outputs makes the maintenance of planning parameters very difficult</td><td>Fit</td><td colspan="2">Unfit because calculating loads for all resources is not necessary and more precise methods are possible</td></tr>
<tr><td>Finite loading with capacity leveling</td><td rowspan="2">Unfit because the subject of finite loading is not stationary</td><td colspan="2" rowspan="2">Fit</td></tr>
<tr><td>Finite loading with optimization</td></tr>
</table>

Figure 4 The fit of capacity planning and scheduling approaches for process complexity (Tenhiälä, 2011).

Romero-Silva et al. (2024) suggest, that a suitable practical production scheduling approach can be guided by considering two dimensions; the scheduling problem complexity and the dynamicity of the manufacturing environment.

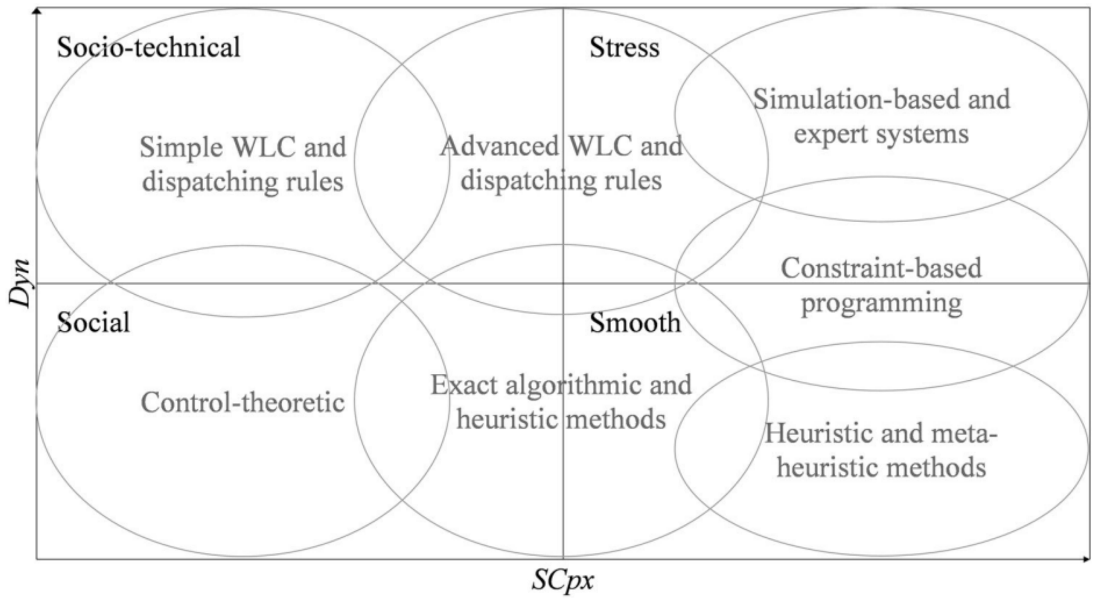


Figure 5 Selection of production scheduling approaches in relation to environment dynamicity (Dyn) and scheduling problem complexity (SCpx) proposed by Romero-Silva et al. (2024).

The two parameters of the model, dynamicity of the environment and scheduling problem complexity suggested by Romero-Silva et al. (2024) contain several characteristics. They suggest that the production environment dynamicity is a function of several factors, which are listed below:

- Demand predictability
- Job routing predictability
- Processing- and setup time randomness
- Machine breakdown probability
- Constraint- and due-date stability
- Job arrival dynamicity

The scheduling problem complexity according to Romero-Silva et al. (2024) is a function of the number of jobs to be scheduled, machine environment complexity, number of constraints and scheduling objective complexity.

In addition to being relevant to the scheduling approach, Romero-Silva et al. (2024) suggest that the two dimensions can suggest the level of monitoring needed to enable and maintain the practical production scheduling task. Practically this likely is related both to the outcome of their first table and the dimensions, as the need for feedback from the manufacturing environment is typically greater in solutions for complex and dynamic environments, and as the more complex and dynamic environments generate information relevant to the scheduling task more often.

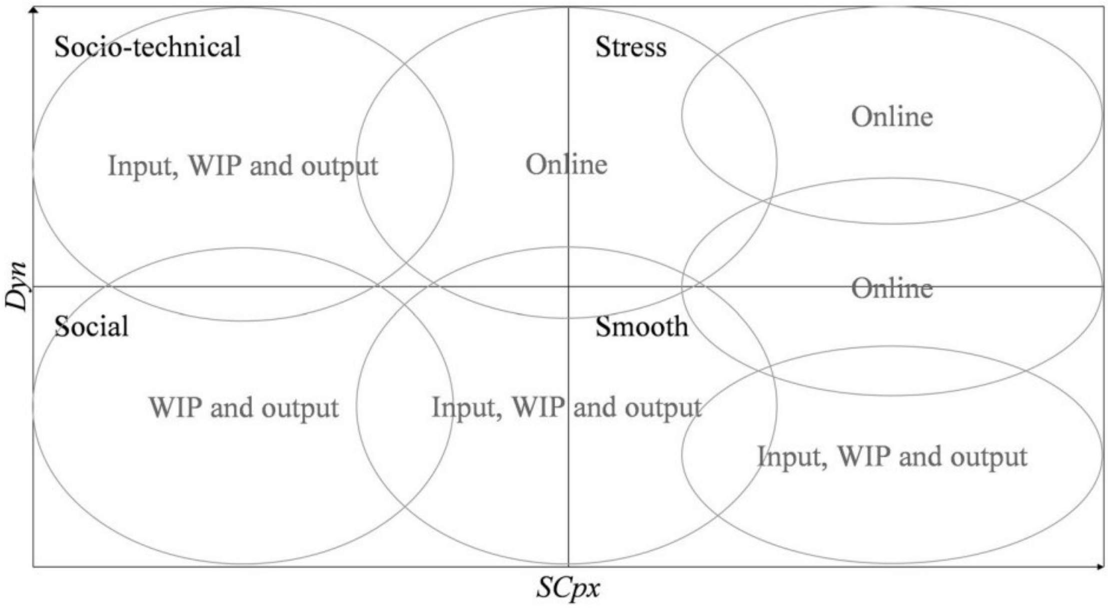


Figure 6 The dependency of information availability on the production dynamicity and scheduling problem complexity (Romero-Silva et al., 2024).

Romero-Silva et al. (2024) state that the utilisation of the production line influences the criticality of a production scheduling system. They suggest that companies with low utilisation of the production capacity do not necessarily benefit from production scheduling as the slack in the capacity allows for sub-optimal schedules.

2.15 The human dimension of production scheduling

In addition to technological concerns, production scheduling has relevant organisational and human dimensions (Carvalho et al., 2014). The production scheduler provides a wide array of inputs to the scheduling process: they balance conflicting goals by different stakeholders, anticipate problems, filter information to the scheduling tool, network to find new relevant information for the scheduling, and interpret situations (Carvalho et al., 2014).

Framinan and Ruiz (2010) argue that production scheduling research should shift focus towards tools that aid production schedulers through the implementation of scheduling methods to user interfaces and providing them options. They argue that automated scheduling algorithms should be aimed to support human schedulers in their tasks instead of replacing them.

Van Wezel et al. (2011) studied production scheduling using the framework of function allocation from the field of cognitive ergonomics. They note that production scheduling as a task can be characterized as having relatively little inherent risk that would necessitate human intervention, as well as there being recovery time available. The tasks are also serializable, meaning that they can typically be performed one after another, and the available time for a decision is typically long compared to other decisions that are often studied in the field of cognitive ergonomics. They note however, that in contrast production scheduling can have high information complexity.

Van Wezel et al. (2011) outline among others risk mitigation, situational awareness, and comprehensibility of the solution as practical considerations in the allocation of tasks between algorithms and the scheduler. They note the higher automation leads to lower situational awareness and lower development of the scheduler's skills, as well as complacency. This effect on complacency is noted to be higher in multitasking environments.

The ability to view alternatives in scheduling, fixing the software schedule output and fixing incompleteness increases the situational awareness and knowledge of the scheduler, as long as these tasks are reasonable considering the cognitive limits of the scheduler (Van Wezel et al., 2011). The positive effects of scheduler participation is likely amplified by the recognized contribution of production schedulers to the overall task in addition to just utilising scheduling tools (Berglund et al., 2011).

Van Wezel et al. (2011) condense the essence of the allocation relevant to manufacturing production scheduling in that when situational awareness is relevant to the task, a human scheduler is required. When highly complex decisions have to be made quick, an algorithm is required.

Cegarra and Hoc (2008) found that the comprehensibility of the schedule representation has a positive effect on the outcome of the scheduling task. They also found indications that in-depth understanding of the algorithm is not necessarily needed in scheduling. This finding would support the view that a scheduling software tool, as long as it provides clear representation of the schedule, would be a beneficial choice to support the practical scheduling task without the user needing to understand the algorithm itself. This is relevant when considering commercial software as scheduling tools, as the algorithms are relatively hidden, but selection among tools by schedule representation comprehensibility can be made.

Framinan and Ruiz (2012) note the importance of manual functions in the manipulation of the schedules, as they note that no mathematical model can account for all the possible disruptions or even objectives. They note that while the requirement of manual manipulation is widely noted that there are several possible approaches in how these manipulations could be performed. They discuss two often used options, manual manipulation in a gantt -chart view and partial rescheduling that can be implemented by freezing parts of the schedule and rescheduling.

2.16 The organisational dimension of production scheduling

The human dimension of production scheduling has been discussed in the earlier section. However, production scheduling can be seen to have an organisational dimension (Carvalho et al., 2014). Carvalho et al. (2014) state, that "The organization must be prepared to face the complexity and dynamics of real-world problems by equipping its planning process with appropriate technology and creating an environment that allows for implementing the plan and addressing operational problems".

De Snoo et al. (2011) suggest that the level of uncertainty should considered when considering the amount and type of support that the production schedulers require. When the uncertainty increases, the flexibility, availability and negotiation tasks of the schedulers become more important than the optimisation of the schedules.

Gasser et al. (2011) conducted a case study into the production planning and scheduling function in a medium sized manufacturing company through the view of Naturalistic Decision Making. According to them, naturalistic decision making is suitable to modelling expert decision making, where the decisions are not just considered through a view of rationality, but can rather be characterised as pattern recognition for cues about the problem state and the matching of these patterns to actions.

The case study performed by Gasser et al. (2011) found that the scheduling and dispatching team used information widely from contexts outside the immediate production scheduling teams, such as knowledge about the state of the suppliers, and the dynamics within the company's wider production system. They found that the schedulers often utilised personal tools and unofficial communication channels to support their production scheduling task. Their study suggests that to enhance the scheduling process quality, more focus should be given to growing the expertise of the schedulers along with providing the schedulers relevant information in a more available format to perform the pattern recognition that is used to ascertain the current situation. For the first objective the study notes, that rapid feedback to how well the scheduling performs is important for the learning of the schedulers. To the latter objective the authors note that while information systems contain a lot of data for the scheduling task, the schedulers in the case found the filtering and collection of the data challenging.

In the design of a production scheduling system, mapping of the information streams relevant to the case might not just need to be performed for the needs of the scheduling solution, but also to support the work of the scheduling personnel, who might require information from wide contexts. Gasser et al. (2011) note, that the information streams for the production planners need to be in a format that is compatible with the knowledge structure of the production planners.

2.17 Guidelines for scheduling system implementation

Carvalho et al. (2014) suggest that building a model for a practical production scheduling task or even understanding the problem in suitable detail was not possible in one implementation. They recommend that production scheduling systems should be implemented in an incremental fashion to both create a well-suited system, as well as to properly introduce and familiarise the production schedulers to the system in an iterative way to aid in the schedulers' understanding of the basis of the tools. This incremental approach would contribute to the capturing of existing scheduling and scheduling environment tacit knowledge from the line supervisors. This view is shared by Framinan and Ruiz (2012)

Decomposition of the scheduling task is another key issue that must be addressed before the implementation of a production scheduling system can begin. The question is whether the scheduling should be performed in one large model or decomposed into several separate models. This decision has consequences both in the implementation of the model, the available methods, and the organisational dimensions of the production scheduling.

Zupan et al. (2024) proposed an online heuristic that considers production orders only locally in the queue of a workstation. This contrasts with a typical offline scheduling approach where the whole shop is managed with a single schedule. When evaluated in simulation studies, they found that their algorithm often reached optimal or near optimal solutions. However, their method was created for job shops and tested with simulated job shops. There are notable differences between approaches for flow shops, as the maintenance of queues and queue orders are more pronounced in the job shop problem. Their findings nonetheless support that local decisions may be performant in comparison to global schedules under some circumstances.

Tokola et al. (2017) found evidence in their study that local control can have similar performance to global schedules when the process time variation is low enough. They note that if processing time variation is high, local control tends to have worse utilisation levels.

Carvalho et al. (2014) found in their implementation that decomposition would have to be performed in their case of a vertically integrated plant to keep the scheduling problem reasonably sized. They note that this decomposition added another level of information flows and consideration. Different production units had different priorities in the scheduling, for example some units having greater setup costs causing larger batch sizes. Their solution scheduled the schedule from a pre-identified bottleneck process according to the target of the final assembly and then used this schedule to build the rest of the schedules. Their research case, however, was highly vertically integrated, and a large part of the concerns in this regard might be relevant only to such environments.

2.18 Due date quoting

There is a notable consideration concerning the performance of production scheduling: an infeasible production plan providing too many jobs to a scheduling algorithm will make any schedule infeasible. There are processes outside the scope of the production scheduling system with impacts on the outcome of the scheduling system.

An example of such a system is due date quoting. There is a complex systematic question regarding the setting of due dates. Intuitively the shortest possible due dates for each production order seems the best for customer responsiveness. The due date decision should, however, consider the congestion of the production line. Each order that must be inserted into the schedule in the middle causes some level of rescheduling (Jodlbauer & Tripathi, 2024) and thus has a chance of causing production tardiness.

Production inserted at the back of the schedule does not require rescheduling, but if the production due-date is quoted to the customer at the first possible time according to the schedule, the order may induce rigidity when a new rush order arrives and must be inserted. Due date quoting considering just the production currently scheduled can be inaccurate due to orders arriving later with higher priority (Jodlbauer & Tripathi, 2024). This outlines some of the difficulties both of due date quoting and the extended processes that may directly make or break production scheduling performance.

Jodlbauer and Tripathi (2024) note two types of customer orders, types where the due date are quoted by the manufacturing company and are thus negotiable and types where the due date is fixed from set customer constraints. Here, effectively, is a relevant question for the practical implementation of a production scheduling system: whether the two different types are both treated the same as absolute constraints in the planning level, adding unnecessary constraints to scheduling.

This problem, and some direction to where it should be solved, is argued by Hopp and Spearman (2011, pp. 544-549). Their argument is that while production scheduling approaches are attempting to control harder problems by creating more complicated formulations, most parameters, such as lead times, lot sizes and WIP, should be optimised on the planning level. This simplifies and reduces the need to optimise execution, for which they argue pull methods are generally better compared to the inherently push method of detailed production scheduling. They note of the due date problem (Hopp & Spearman, 2011, p. 549):

> Because scheduling is difficult, an important insight from our discussion is that it is frequently possible to avoid hard problems by solving different ones. One example is replacing a system of exogenously generated due dates with a systematic means for quoting them. (p. 549)

There might be a need to check the processes defining some of the parameters used as objectives critically – whether for example the due dates provided to the production scheduling algorithms are sensible from the business perspective. For example with a production scheduling approach attempting to balance between due dates and flow time, the solution can already be constrained by overly sensitive due dates.

2.18.1 Capable to promise

Due date quoting can be seen as a part of a frameworks capable- and available to promise. Framinan and Leisten (2010) outline the functions of available to promise systems. According to them, the systems deal with incoming requests for quotation from order acceptance, due date quoting and confirming to new order scheduling. They note three main tasks in the system: order acceptance / selection, due date assignment and order scheduling.

The business environment has experienced a change to the order capture process, where the timeframe and opportunities for negotiation of the due date has decreased, calling for automated solutions in due date setting (Framinan & Leisten, 2010; Jodlbauer & Tripathi, 2024). While the due date setting problem has been discussed here as an external problem that impacts production scheduling, there is an opportunity in production scheduling to make better informed automatic due date quotes by the scheduling solution suggesting scheduled capacity based due dates.

The importance of this task can also be seen through the “fixes that fail” system archetype (Kim, 2000). Rush orders, which can appear due to too high expected lead times, can cause delayed disruption to the production system further feeding the problem further increasing lead times and thus increasing the amount of rush orders.

The effect called the lead time syndrome is a positive feedback loop between lead time increases and work releases, further driving lead time increases (Selçuk et al., 2009). Due date quoting in an MRP environment likely at some level is implemented through lead time changes. Thürer et al. (2023) however argue that lead time syndrome is only relevant in make-to-stock systems, and that the effect does not exist in a meaningful level at make-to-order production as long as a lower bound for lead time is enforced. The lower bound for dynamic lead time is required to avoid overcrowding the production schedule when there is a small number of jobs waiting to be processed, causing an artificial backlog. This artificial backlog then impedes the scheduling of new work once the volume of incoming jobs increases.

3 Method

The first research question is: what options does the theory present for the implementation of production scheduling, and how do these approaches fit the case company's production environment? This topic has been discussed in depth during the literature review, but further analysis is needed for the second part of the research question: how do these approaches fit the company's production environment?

A theme arising from the literature review of the practical production scheduling task indicates that the dynamicity of the production environment has a notable impact on the possibilities of scheduling. Due to this reason, the environment is analysed to find indications for a suitable production scheduling approach.

3.1 Scheduling environment dynamicity

The measurement of manufacturing system dynamicity in the case company is performed following a breakdown into components as in the framework presented by Romero-Silva et al. (2024). They propose that system dynamicity is a function of the predictability of the demand, predictability of job routings, randomness of processing and setup times, probability of machine breakdowns, stability of constraints and due-dates and job arrival dynamism.

In the framework presented by Romero-Silva et al. (2024) the terms of dynamicity are valued with the range 0-1 and contain a weighting term for each of the terms of dynamicity. They discuss that the weighting terms can differ for different applications and environments. This method limits the maximum contribution of a single factor to dynamicity. In this paper, the terms are not weighted, as there are no values to benchmark against, and the terms are already considered in the context of the company case.

In addition the terms should be considered as separate parts without weighing and a constraint to a range, as the nature of the dynamicity in practice is additive – if one factor causes high unpredictability and thus makes schedule building difficult by itself, this would not be compensated even if all the other factors were perfectly predictable and stable.

3.1.1 Mean time between production schedule disruptions

The practical measurement for production scheduling dynamicity is the frequency of schedule changing events. It answers the question of how large or small the typical time period is between events that cause a reschedule. This metric is expected to work much like another typical metric in manufacturing, the mean time between failures on machines.

Because the case company currently has an online reactive system relying mostly on the work performed by production line supervisors, data about whether plans would be stable or not is difficult to collect directly. However, due to the supervisors' method of interaction with the ERP system, whenever a production order is moved to another day due to a disruption that requires postponement, such as load over capacity, material problems, machine malfunctions or other reasons, the changed production order can be identified in the production order list. In the case company such production orders are identifiable by the basic dates of the orders. Orders that have not been moved always have some days between the basic start and basic end date -parameters of the ERP, while on orders that have been postponed these two dates are the same.

This metric is assumed to display a combination of dynamicity measures identified in the literature, such as machine breakdowns, material quality problems affecting the production capability, rush orders and other environment disruptions.

These orders, which have been manually changed, are analysed from historical ERP data and used to estimate larger disruptions in the production order and thus events that would require a rescheduling in the production schedule. Finer data of scheduling stability within the day does not exist, and would not be estimable in the current system, as there is no intra-day schedule that is practically attempted to be followed.

Another major consideration in the analysis is how the disruptions should be counted, as sometimes moved jobs appear in large groups within the production data when ordered by the actual finishing date of the production order. Some groups can be large, up to 66 orders. A decision must be made whether a group should be analysed as a single disruption or several disruptions. Here an assumption is made; moved orders that appear as a group of orders were related to a single disruption. Under a production scheduling approach, these would all be rescheduled in one rescheduling event.

A practical predictive production scheduling is likely done within timeframes that can be represented by a multiplier of working days, such as twice a day or once every two days. A reasonable denominator in the analysis of disruption is the working days of the line within the analysis period.

The analysis of disruptions was done using recent past data from a time window containing in total 126 calendar days. Different lines within this time window had 87 – 109 working days. The measure, disruptions per day was calculated for each of the production lines and overall.

The reliability of this data warrants some discussion. As the data is composed of all the production orders within the timeframe, the data can be considered fully representative of that period given the assumptions outlined. Representativeness is therefore not a matter of statistical testing of representativeness, but rather a matter of business discussion of the relevance of the assumptions for what the input data represents and the representativeness of the selected period between the 3rd January 2025 and 9th May

2025. The business must deem the period to be characteristic for the performance overall for the analysis to be applicable.

3.1.2 Due date tightness

Another relevant compound measure for the production scheduling problem was calculated by measuring the due date tightness weighed with workload. This data provides a basis for estimating some possible effects of the job arrival dynamism, scheduling complexity given by due-date constraints and how new job arrivals should be considered in the scheduling approach.

The data was acquired by calculating the difference between a production order creation date and the requested delivery date for the production order obtained from the ERP using the period between 3rd January 2025 and 9th May 2025. Then the workload, estimated by production order quantity, for each time frame was compared to the overall workload for the whole period. Because the production lines contain work steps and products designed around a nominal takt time, the workload may be estimated with the production order quantity. The statistic therefor gives the estimated workload with a due date at or shorter date from the production order creation in proportion within the whole data set for the production line. Due to the production quantity used as an estimation for the workload, this statistic cannot be calculated for the whole dataset, only for a single production line at a time.

Another assumption made in the analysis of this data is that the period analysed and the future state represents near practical full capacity for the production lines. If production lines are operating at a lower capacity, this acts as a capacity buffer capable of reducing the effects of arriving rush orders, and arrivals can fit into the production schedule without any special consideration even if some tight due date work arrives.

3.1.3 The variance of the due date tightness

The analysis of due date average tightness raises the question of how variable the due dates with workload are. If the due dates are relatively stable, the average workload per due date calculated in the earlier analysis can give a reasonable estimate of how much slack capacity can be left in the production schedule to accommodate most orders with shorter due dates. If the variance is notable, then using the averages might not provide effective results in managing the arriving jobs.

To analyse this problem, the due date weighed with workload is calculated as above. This workload is then calculated separately for different weeks and compared. There remains a necessary selection to be made: what is the statistical unit, the week used in the analysis? The primary available candidates are the weeks the work was performed in or the week the work was created in. The production order creation week was selected as a basis to highlight especially the arriving production order variability.

Another consideration in the analysis is the unit used. The workload, as in the analysis of the due date tightness, is based on the quantity of items on the production order. A suitable denominator must be found to make the units more intuitive for load comparison. To calculate a stable and usable metric, an estimate of full capacity workload must be made to use as a denominator. This was estimated by selecting a value that 75% of the workload for all weeks of the line would be under.

Finally, the quantities of new production orders opened under a certain time for each week were calculated. This was performed with a Python script that found all the production orders created within the analysed week, tabulated the quantities of production with a due date smaller than a certain size, and output the table of work quantities divided by the approximated full capacity for each week under analysis.

3.2 Evaluating the SAP ECC CM27 capacity levelling tool

The intended method for implementing production scheduling inside the SAP ecosystem with SAP ECC is the separate APO application that interacts with the system (SAP, n.d.). This is in contrast to the newer S/4 HANA version of SAP where the available to promise and detailed scheduling functionalities have been integrated directly (Galla, 2020).

Another research question is whether the existing ERP system can provide tools for production scheduling. The case company uses SAP ECC without the APO application. There is limited capability for production scheduling in SAP ECC. A functionality for production levelling under the transaction code CM27 is available in the ERP. This tool is evaluated for its production scheduling capabilities.

This part of the research is performed through constructing test scenarios to test individually different components of the tool outputs. Test scenarios are made with simple known preferred outcomes as hypotheses. Due to the exploratory nature of the tests, further test hypotheses were formed during the run of the tests. The tests were performed in a sandbox environment mirroring the actual ERP system. The tested properties and tests are outlined below.

3.2.1 Sequence dependent setups

The function for sequence dependent setups was tested with a set of jobs that were divided into three setup families with sequence dependent setup times. The sequence dependent setup times between the different setup families were designed so that there would be a clear optimal sequence of setup times. The test checks whether the program correctly identifies and selects this setup sequence.

A second test for sequence dependent setup times is performed by giving the algorithm a problem where the algorithm must select a locally suboptimal setup in order to avoid a global suboptimality. This tests whether the algorithm looks just at a single setup or considers the total setup time. This test uses the same sequence dependent setup families and times as the first test, but a product from a fourth setup family is added with cheaper setup times for products that cause the production sequence to be more expensive overall compared to the globally optimal choice.

3.2.2 Due date handling and sequence generation

Due dates are an integral part of production scheduling in the case company since most of the production is make-to-order production. The scheduling approach should prevent both late and too early production. The tests under this category evolved over the testing process as new problems and considerations were encountered.

The first test in this category was whether the setup time optimisation function can be combined with problems that have due date constraints. As the sequence dependent setup time optimisation feature is separate from the sequence building, the test attempts to identify if it recognises when the optimal setup time outcome violates due dates. This test was performed using the same setup from the sequence dependent setup time test, but this time three jobs with a strict delivery date were added. These jobs were added to the last setup family in the sequence, where their spot in the optimal setup order observed from the earlier test would be around two weeks past the delivery date.

A second test into the sequence generation tests whether the production orders would be sequenced by their basic finish date, a date originally generated by the MRP in a backwards direction from the due date, but also manually manipulatable. The jobs were specified different basic finish dates and scheduled with setup time optimisation.

A third test into the sequence generation tested three different dispatch sequences available for the dispatching of jobs to the schedule. Two default sequences and one custom made sequence was tested to both check whether the sequences behave in a predictable pattern. A set of jobs was prepared for each test with known qualities to allow the formation of expected outcome as a hypothesis.

3.2.3 Sequence generation hierarchy strictness

In the tool the sequence is defined through several levels of parameters used to build the order of the sequence. A question was raised during the sequence tests whether the parameters were strictly hierarchical or whether several parameters could change the overall outcome. A strictly hierarchical generation would mean that the first sequence parameter will decide the overall order and that the lower parameters are only used to decide the order when there is an equality on the higher-level parameter.

For this test, a custom sequence rule was built that first considers the basic finishing date and then considers the delivery date. A set of jobs was selected for sequencing with a large variation of delivery dates and a small variation of basic finishing dates. The basic finishing dates were defined to be the inverse order of the delivery dates. If the lower parameter can override the higher parameter in the hierarchy, the sequence is expected to consider the delivery date. If the hierarchy is strict, then the sequence is expected to be according to the basic finishing date.

3.2.4 Schedule manipulation

There is an identified need to be able to manipulate the schedule due to dynamic events. A set of four basic schedule manipulation tasks are identified and tested.

1. Inserting a production order in the middle of a ready schedule
2. Removing a production order from the middle of a ready schedule
3. Changing the sequence of two orders
4. Right-shifting orders

Some of the tests have subtests. For example, the tests dealing with insertion and removal have subtests while considering sequence dependent setup times and not considering setup times.

4 Results

The results are divided into two separate categories along the research questions. The first section, production environment dynamicity, considers the fit of production scheduling approaches to the case company's production environment dynamicity. The second section focuses on testing the currently available tool that the case company has regarding the production scheduling capabilities and answers the question whether the existing ERP system could be used to perform production scheduling.

4.1 Production environment dynamicity

The production environment dynamicity is a key part of understanding the fit of production scheduling approaches and organisation in the case company. The frequency of disruptions to the schedule and the arrival of rush jobs have been identified in the literature to be relevant to how production scheduling should be performed.

4.1.1 Mean time between production schedule disruptions

The overall expected disruptions for a production line is 0.91 disruptions per day. The dispersion of individual lines' disruptions can be seen in figure 7.

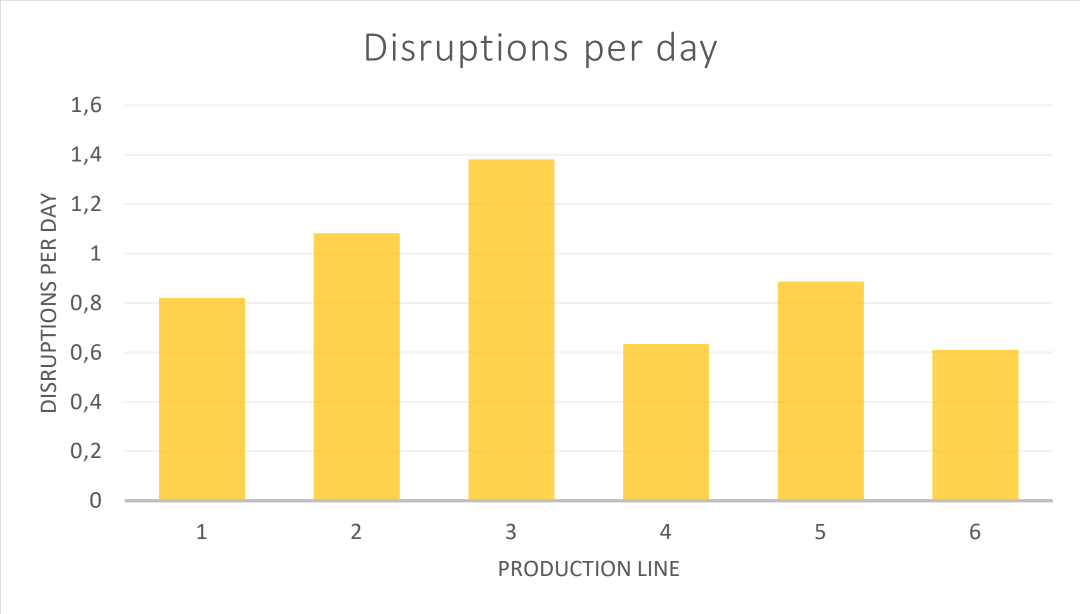


Figure 7 Mean time between schedule disruptions for each analysed line in days.

This statistic has some practically relevant interpretations. When scheduling an individual line, there is likely to be a major disruption, necessitating a rescheduling event every 1.09 days. If the six lines analysed were to be scheduled in a single schedule, the disruptions can be expected to compound, and there would be 5.5 disruptions per working day, assuming that the disruptions for each production line are independent.

4.1.2 Average due date tightness

The data will be considered from two perspectives. First the due-date workload data is considered from the perspective of short-term disruptions to production scheduling. The relevant idea behind this analysis is that for a predictive schedule made for a certain number of days, any new orders appearing with deadlines sooner than the rescheduling time will necessitate a reschedule if the due dates are to be held firm and the production cannot be inserted into the schedule without rescheduling. The relevant data can be seen from figures 8 and 9 below, with each of the five production lines represented by a line. Both charts below are representations of the same data, just with different scaling and scopes of axis. As a reminder, the unit in these graphs is the proportion of workload with a due date equal or under the days compared to the whole combined workload of the whole line. The data is calculated using the whole time period, so the output is effectively averaged over the whole period.

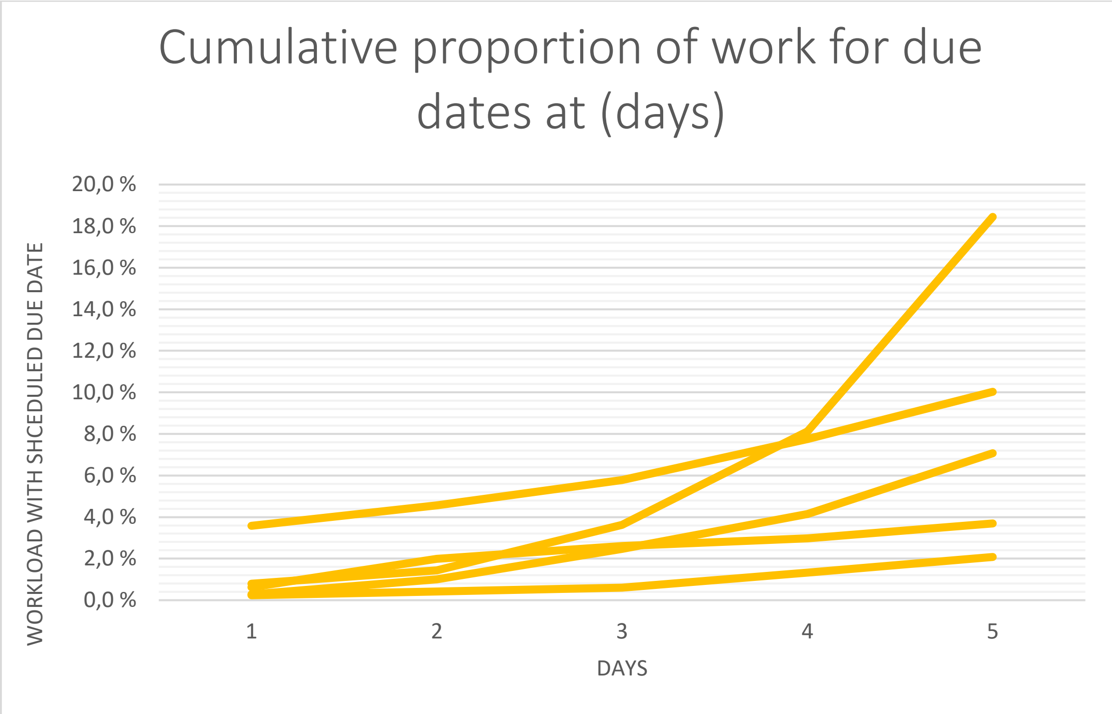


Figure 8 Proportion of workload within 5 days, different production lines displayed with individual lines.

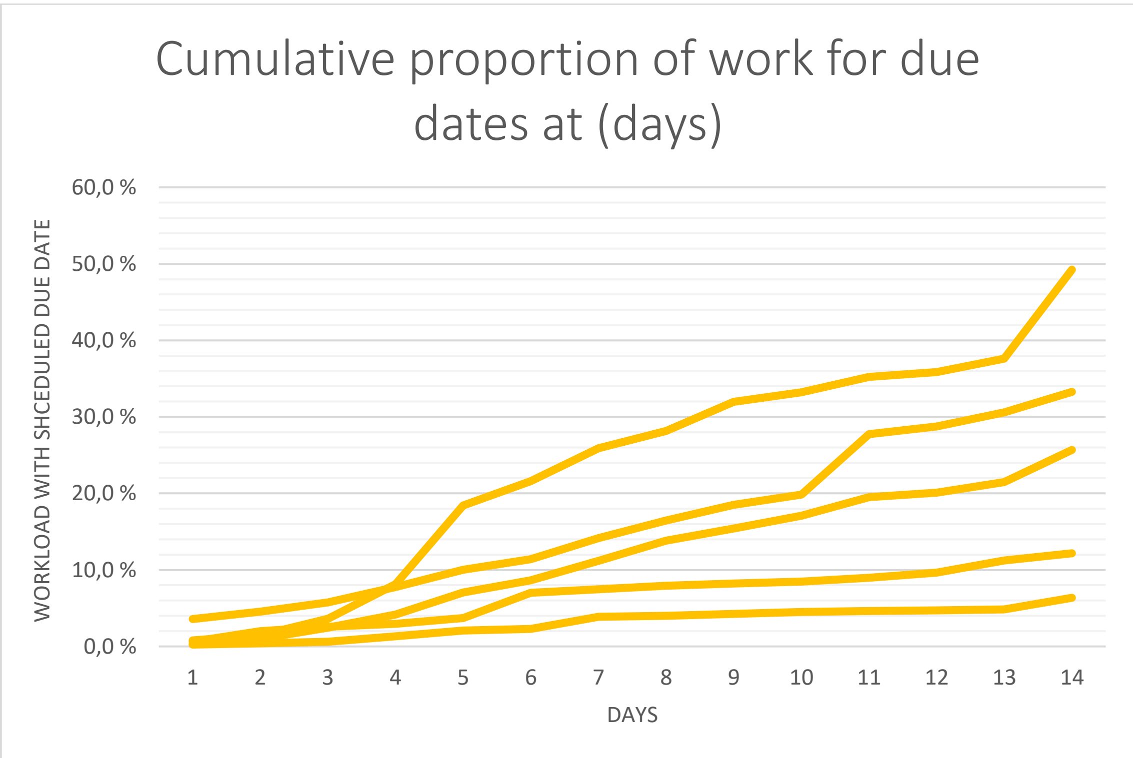


Figure 9 Proportion of workload within 14 days, different production lines displayed with individual lines.

There appears to be a notable difference between the different production lines for the workload for each period. For a period of 3 days for example, the averaged workload arriving within the period varies from 0.6% to 5.8% depending on the production line. At a 14-day period this range is from 6.4% to 49.2%. From this finding it appears that a predictive or predictive-reactive production scheduling approach could benefit from parameters tuned to that specific production line.

The other perspective for this data investigates the possibility of performing production scheduling with a longer timeframe, approaching production planning. Here the same measure, workload with a delivery date under the time is given for weeks. This data can be used to estimate the magnitude of scheduling capacity limit. This means that while scheduling, the capacity limit for scheduling can be rolling so that schedules in the further weeks have a tighter capacity limitation to accommodate later arriving work (Pinedo, 2009, p. 378).

This also communicates the tightness levels of the due dates – an important distinction when considering both the complexity of the scheduling task and the inherent constraints posed by the due dates for the degrees of freedom of the scheduling algorithm to perform optimisation.

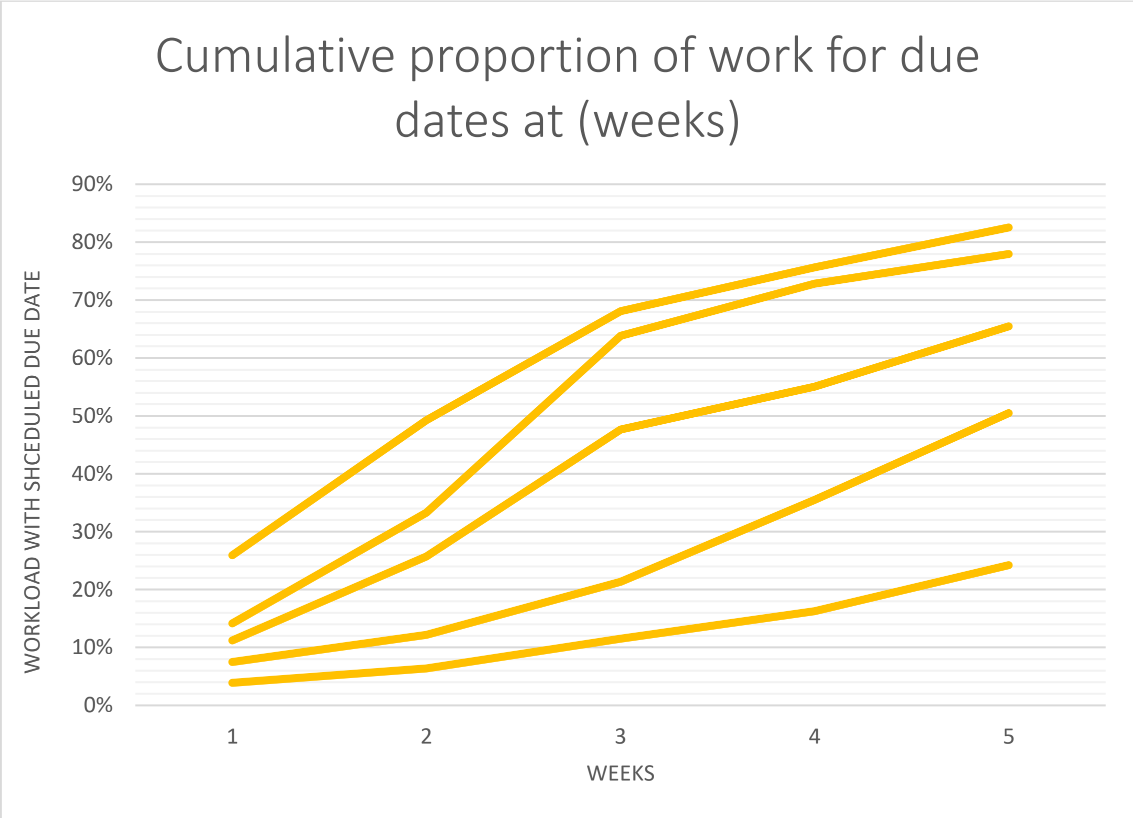


Figure 10 Proportion of workload within 5 weeks, different production lines displayed with individual lines.

There appears to be a clear difference between production lines on this time frame as well. Especially at this time level, these results should give a reasonably good estimate of the magnitude for the approach if rolling capacity limits are used for longer time scheduling.

There is a practical consideration in the direct use of these. The calculation is using information from the whole period. These values for each production line is effectively averaged from the whole analysed time window, meaning that if these values were used to limit the capacity directly, about half of the weeks would finally end up over capacity. This data does not contain information about the variability of the due dates from one week to another week.

4.1.3 Due date tightness variance

Due to the different production lines having different distributions of due date variations, the analysis is performed visually using boxplots. Using these, the variation of workload due dates can be visually identified inside and between the different production lines. First, the overall variation of workload within a certain due date tightness is analysed for all of the inspected product lines at once.

Notable here is that the statistical unit and measure has changed from the average due date tightness. This method calculates the workload under a certain due date tightness for each week, selected by order entering date, separately. The workload is calculated using the assumed full capacity of the production line. The weeks are presented first and then the load for days.

A brief description of boxplots is given here in the context of the presented data. The rectangular body of the boxplot contains the centremost 50% of all the data, in this case weeks with a certain amount of workload with due date as tight or tighter. The so-called whiskers of the boxplot attempt to contain the rest of the data, but if there are any data points considered outliers, these are excluded and drawn as points. It is notable here, that the plot height for workload has been artificially limited for readability, so some outliers are outside of the plot and thus cannot be seen. The horizontal line inside the box visualises the median of the dataset and the cross visualises the mean.

The upper edge of the box can be used to find the value under which 75% of the values fit. The height of the box can be typically used to get a visual estimate of the size of the typical variation. The distance between the median and the mean can give some estimation of the skewness of the distribution.

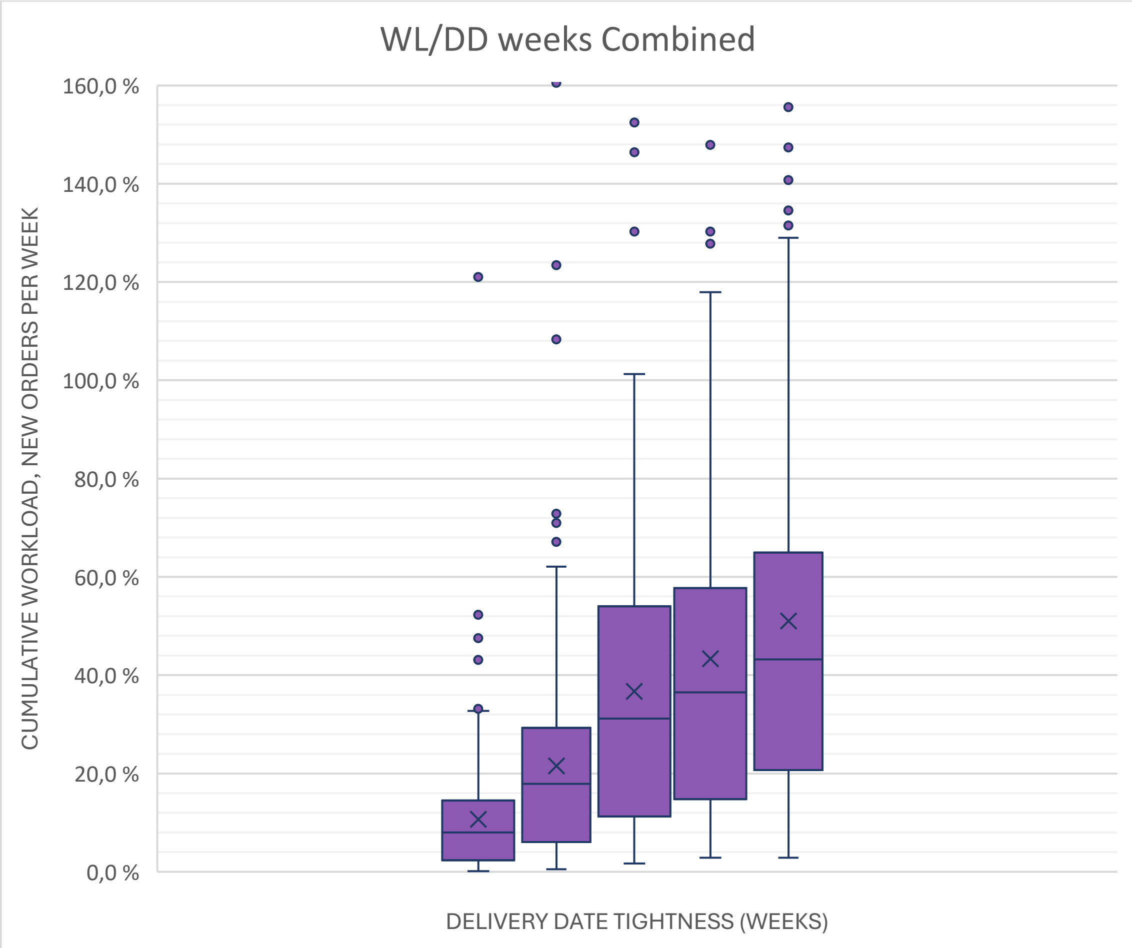


Figure 11 The distribution of workload for all production lines combined within a due date arriving each week. Bars represent weeks 1 to 5 from left to right.

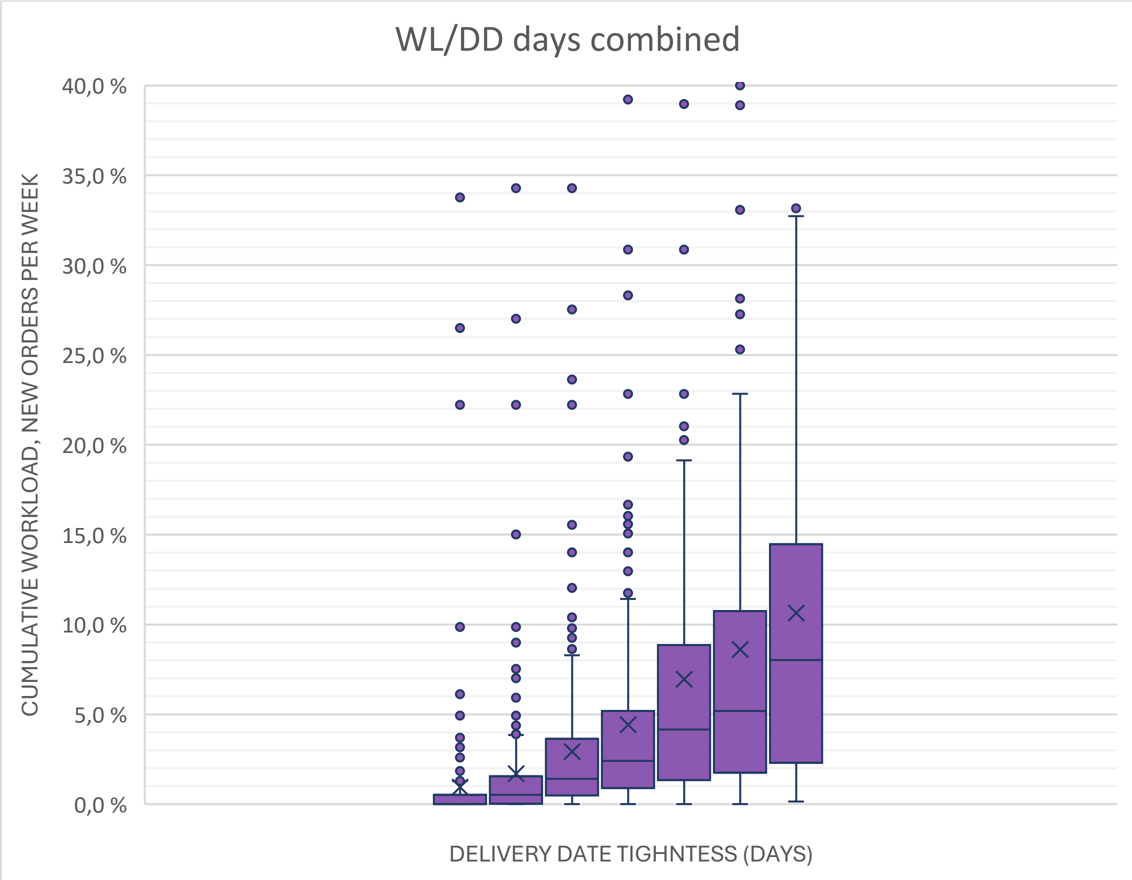


Figure 12 The distribution of workload for all production lines combined within a due date arriving each week. Bars represent days 1 to 7 from left to right.

Some expected patterns can be noticed from the boxplots. The variability of the cumulative workload within a range of delivery dates increases with a seemingly linear relationship within the first week. The skewness of the distribution increases with time, as evidenced by the growing gap between the mean and the median, along with the upper limit stretching.

This data, when presented with boxplots, can now be used to estimate how likely a certain set of buffer capacity on the scheduling system would accommodate an appearing rush order. The drawback of using the combined chart is that if there are significant differences between the variances of two production lines, a single level of capacity buffer will impact the production lines differently. This appears to be relevant in this case. As an example, when comparing two extremes of this variation at day five, the policy where

75% of short due-date orders would be accommodated directly, 9% extra capacity 5 days out, would only cover slightly over half of the orders at one production line, and on the other extreme typically end up mostly as unused capacity on another line. The comparison can be seen below in figure 13.

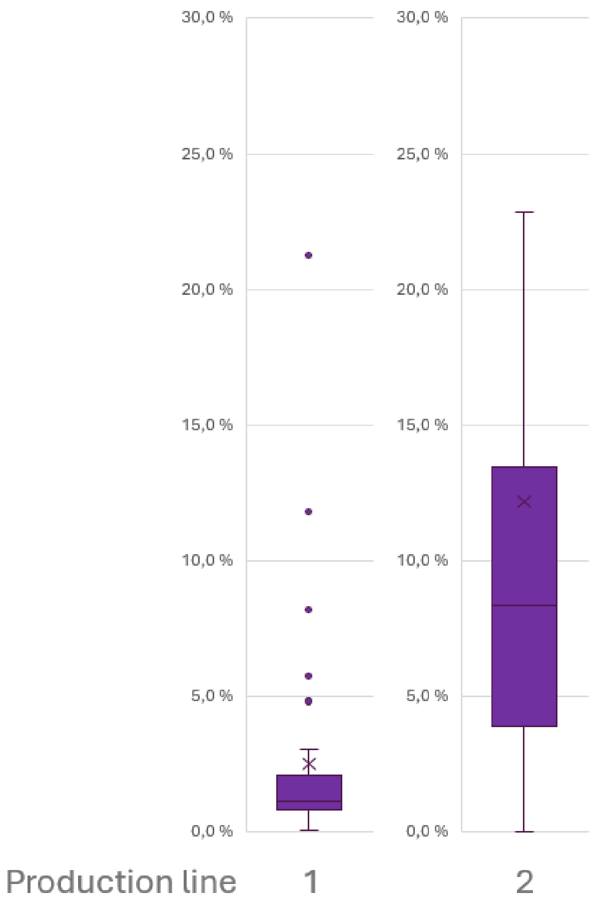


Figure 13 Day 5 difference of due date tightness distribution between two production lines.

To ensure that there is a statistically significant difference between the different lines, the variance is analysed statistically. The data is non-normally distributed, and therefore Kruskal-Wallis tests are used to assess if there are significant differences between the expected workload levels of different production lines. Since the Kruskal-Wallis test can only be done for one day at a time and thus requires repeating seven times to test all the days, a Benjamini-Hochberg procedure is performed with a 5% false discovery rate to control the accumulating risk of type 1 errors.

			Q = 5%
Analysis	**P value**	**Rank**	**(i/m)Q**
7 Days	1,14E-16	1	0,007143
6 Days	3,59E-16	2	0,014286
5 Days	6,12E-16	3	0,021429
4 Days	1,6E-13	4	0,028571
3 Days	4,19E-11	5	0,035714
2 Days	7,47E-06	6	0,042857
1 Day	0,02459	7	0,05

Table 2 The results with Benjamini-Hochberg procedure.

The analysis supports the conclusion that there are statistically significant differences in the expected workload between the different analysed production lines. The practical relevance of these differences has been discussed above. A single strategy to allocating the arriving work with a short due date may not be optimal for all the production lines. However, there is still the need for the case company to consider whether the period the data is collected from, 39 weeks starting from 15th of July 2024, is representative of the typical workload and performance.

In this case it is recommended that the production lines should have separately defined capacity buffers according to the expected tight due-date orders to reduce the amount of significant rescheduling needed to fit new arriving rush orders. While on some lines these can be typically within just a couple of percentage points of the overall capacity for 6 days, on other lines the typical workload arriving with due dates 6 days and under can be up to 20% of the capacity.

If time-bound capacity buffering can be implemented in the scheduling software, these give guidance for the parameters of the implementation. If such a capability is not available, the data can be used to estimate the effects of the scheduling time window in relation to how often a reschedule would be expected.

There is a discussion here about the scope of detailed scheduling against production planning. A narrow view of production scheduling would argue that work arrivals that are over a week in the future are not within the scope of production scheduling but are production planning. However, since in the case company both tasks are performed currently by the production line supervisor, this distinction might not be necessary. If the production scheduling tool can provide meaningful short term production planning outcomes, these can be useful.

4.2 Analysis of the SAP ECC CM27 capacity levelling tool

To answer the research question about whether the existing ERP system can be used as a tool to for production scheduling, a potential tool inside the package was identified and evaluated. The tool selected was a tool for capacity levelling with the transaction code CM27.

While capacity levelling tools are not strictly production scheduling tools, a case can be made in their favour. Due to the production moving in single units within the production line with no sequence changes while in the production, the flow generally manages itself. There can be limited benefits to optimising the production sequence in order to increase utilisation. If the tool can build a feasible sequence, the benefits of having a set sequence of production can be captured, even if the sequence is not strictly optimal. Effectively this approach considers the production line as a single machine scheduling problem.

4.2.1 Sequence dependent setup times

First the capability of considering sequence dependent setup times is confirmed from the production levelling tool. This can be divided into two tests. The first test is very simple, where a set of products are configured with three families each with their own setup times.

For the first test the product family setup times were defined with the following values displaying notable variations. The reasoning for the selection is to create an obviously optimal pattern before the test, so that the test can confirm if the tool can find this pattern. The product families were defined in the ERP using the transactions OP43 and OPDA.

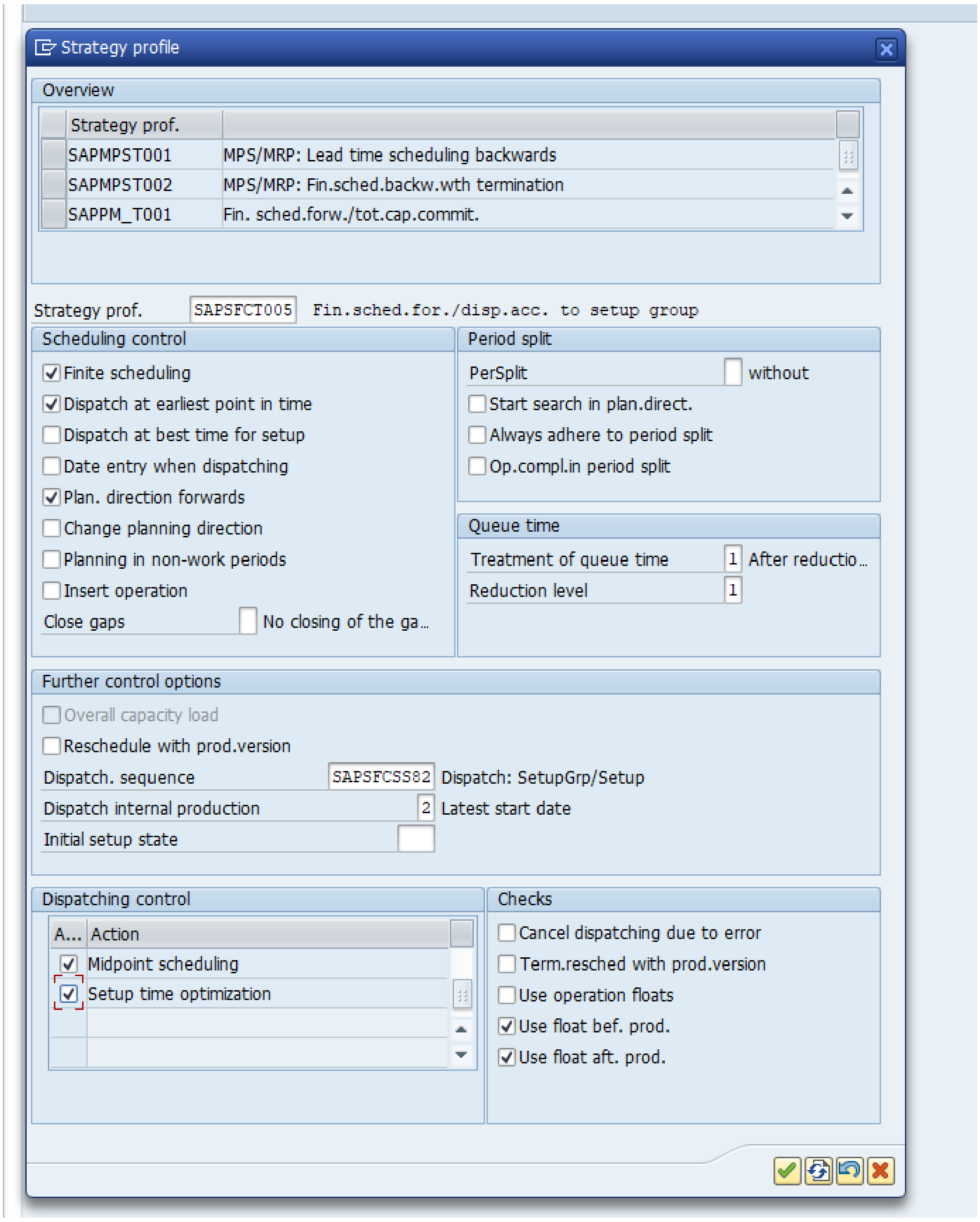
	Prod. Family 1	Prod. Family 2	Prod. Family 3	To
Prod. Family 1	x	8 h	1 h	
Prod. Family 2	6 h	x	8 h	
Prod. Family 3	4 h	1 h	x	
From				

Table 3 Sequence dependent setup times used in the test.

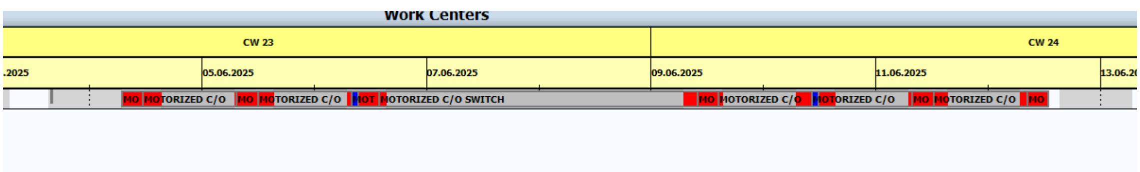
Plnt	Pred.group	PreSubgrp	Succ.group	SuccSubgrp	Standard V...	Unit	S..
0301	SG2	FAM1	SG2	FAM2	8,000	H	
0301	SG2	FAM1	SG2	FAM3	1,000	H	
0301	SG2	FAM2	SG2	FAM1	6,000	H	
0301	SG2	FAM2	SG2	FAM3	8,000	H	
0301	SG2	FAM3	SG2	FAM1	4,000	H	
0301	SG2	FAM3	SG2	FAM2	1,000	H	

Picture 1 The setup times defined in the ERP transaction OPDA.

From the setup matrix it is apparent that the production order of families should be 1 – 3 – 2. Any other pattern would be highly inefficient. While the selection order should be obvious even for very basic sequence dependent scheduling heuristics, the algorithm will need to identify family 1 as the first family in order to build the optimal sequence. In total twelve production orders were created in the sandbox with four products assigned to each setup group randomly. All the production orders were created using a single existing production order as a reference with equal standard dates.



Picture 2 Strategy parameters for the test.



Picture 3 Outcome of the test.

The setup time optimisation appears to have identified the optimal solution. This can be visually confirmed from the picture of the proposed schedule above: there are two setup periods marked in blue, each of which are in total 1 hour.

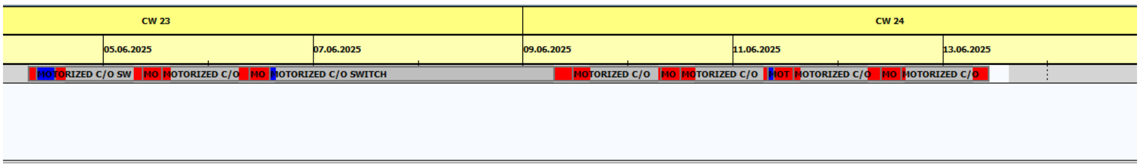
Another test to confirm the nature of the schedule optimisation method available in the current version of the ERP is to create a heuristic trap where the scheduler must make a local suboptimal move to avoid a greater setup problem later on. This test expands the setup used earlier and allows some insight into how much information the setup optimisation algorithm uses. The expected outcome is that the scheduler considers the setup time locally in the forward direction, locking into a non-optimal sequence at the first step.

A fourth production family is added with 3 hours of setup to product family 1, the known optimal second step, and 1 hours of setup for the other families. A product representing the product family 4 is added at the start of the sequence before the rest are scheduled in. If the scheduling algorithm only considers the problem locally, it should select the cheaper setup first and suffer a costly setup later.

Plnt	Pred.group	PreSubgrp	Succ.group	SuccSubgrp	Standard V...	Unit	S..
0301	SG2	FAM4	SG2	FAM1	3,000	H	
0301	SG2	FAM4	SG2	FAM2	1,000	H	
0301	SG2	FAM4	SG2	FAM3	1,000	H	

Picture 4 Added setup times.

The outcome of the test was that the algorithm chose the globally optimal sequence, it confirms that the sequence setup time optimisation performs a more in-depth check than a purely constructive heuristic forwards, as the dispatching correctly identified the situation and made a locally suboptimal selection of a three-hour setup to avoid a worse global outcome later.

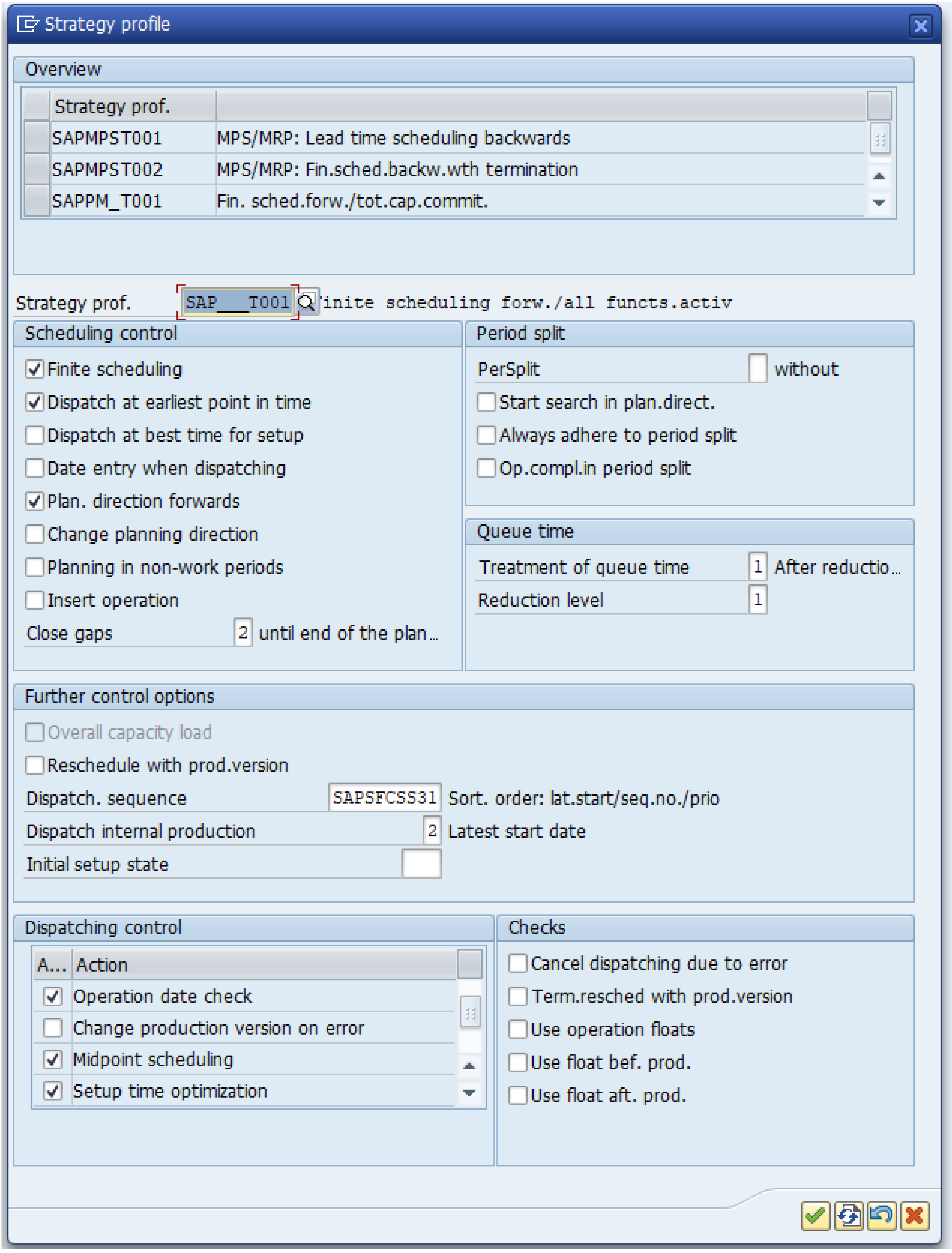


Picture 5 Outcome of the second setup time test.

4.2.2 Due date handling and sequence generation

While the scheduling tool tested, not being strictly intended for detailed optimisation, is not tested for solution optimality, it should consider the due dates to be of realistic use.

The first test is the repeat of the earlier sequence dependent setup times, but this time with two orders that have a delivery date that is a week earlier than the scheduling outcome without them for their respective family, the scheduling family two, would be. The test evaluates how the program reacts to the existence of these two jobs. A third job with the same due date is added for production family 1.



Picture 6 Strategy used for due date test with setup time optimisation.

The program constructs an identical sequence of setup times despite the late due date jobs. The outcome of this test indicates that at least with these dispatching settings, the production ordering violates the requested delivery dates for the two production orders which optimal setup order is in the last group of jobs. It is notable that the two orders were not even the first inside their setup group. The tool, at least with the current configuration, does not appear to consider the delivery date. Whether this outcome is due to the configuration of the strategy or the nature of the algorithm is tested in further tests.

When a production order is created, the MRP assigns it a basic finishing date backwards from the date the production should be finished according to the float time after production finishing. A test was conducted checking if without the function for setup optimisation the orders would be ordered according to the basic finishing date. A set of work was dispatched without the sequence dependent setup time optimisation with an arranged basic finishing date. The orders were dispatched according to the basic finishing date order.

A complicating factor the usability of the levelling tool was noted at this point. Whenever a job is dispatched to the schedule, SAP will adjust the basic dates to align with the schedule, even if the basic dates are in violation of the delivery dates. The basic finishing date in the system is a sensible information to build a due-date relevant schedule as the basic dates are built by backwards scheduling in the MRP. However, the scheduling operation itself modifies the basic dates just by attempting to dispatch the job. This leads to the degradation of the data that carries the due date information into the scheduling, thus effectively committing to the scheduled date when scheduled.

A third test is made to understand whether the different sequencing rules available behave predictably. This test contains three dispatch sequences, two of which are ready sequences from the ERP and one of which is a custom sequence configured in transaction CY39 for this test.

Test number	Dispatch sequence	Sequence long text
1	SAPSFCSS13	Setup grp/lat.start/mat.no.a
2	SAPSFCSS21	Latest start/seq.no./prio a
3	ZTEST1	Custom sequence considering only delivery dates

Table 4 Dispatch sequences to be tested.

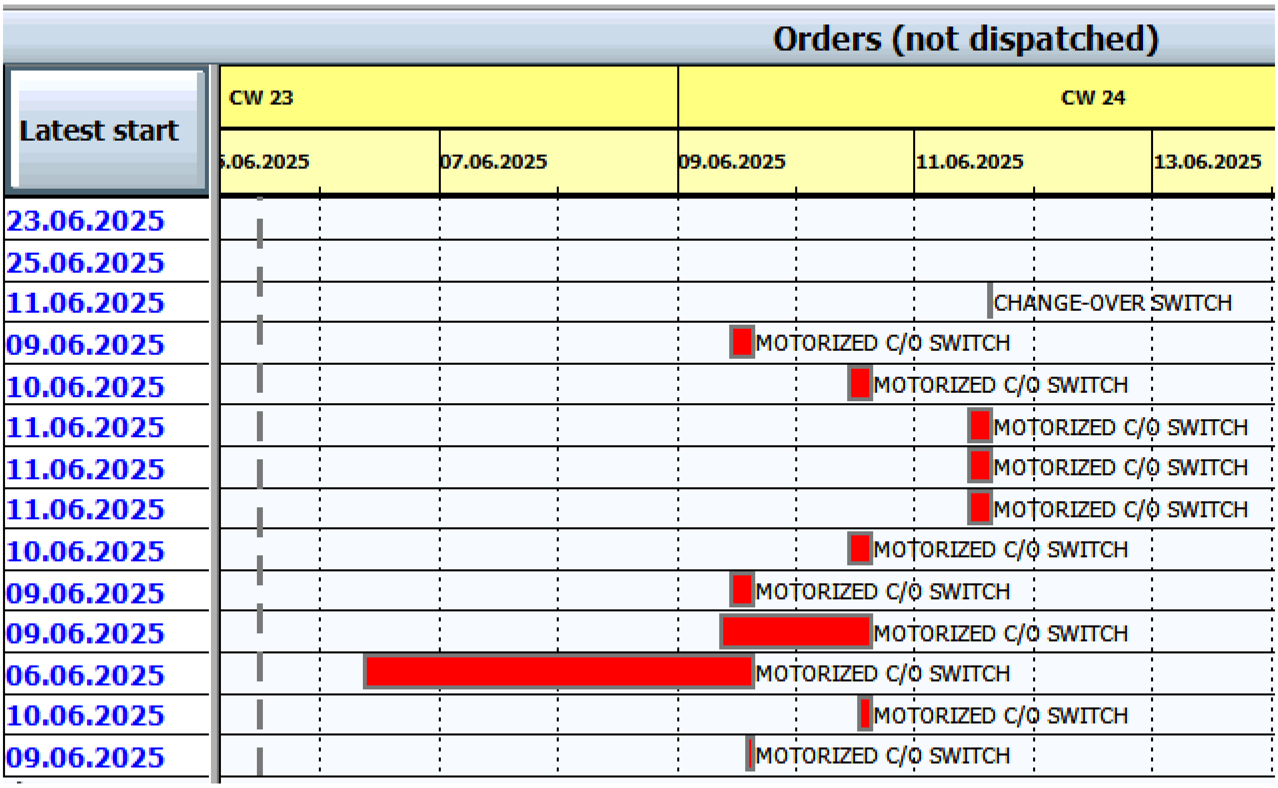
A set of jobs with known parameters are used to model each dispatching rule. The same production orders are set to have the same basic dates and same setup group keys for each test. Priorities are set for each job. The setup group keys are set to be in a different order compared to the order of the basic dates. All of the tests are performed with forward scheduling, dispatching at earliest point in time on, and setup time optimisation off.

The first test is expected to arrive at an order that groups the same setup keys together without considering the optimality of the grouping sequence of setups and then the jobs inside the groups according to the basic end date.

The outcome from the first test confirms the first part of the hypothesis, the orders were grouped according to the setup keys, the setup keys are the same property that is used to define the setup families for the setup time optimisation.

The second part of the hypothesis, that the orders would be dispatched in order of the basic dates was not confirmed. The parameter used to sort does not strictly match with the basic end date. A further hypothesis was made. In earlier tests, the sorting was performed by the basic end dates because the orders had identical latest start date -parameter. This was initially assumed to be equivalent to the basic finish date, but this appears to not be true.

The second test is expected to order the work according to the latest start date. For this, the parameter “latest start date” must be defined. The parameter appears to be controlled by the processing time and the basic end date. This explains the earlier outcome: identical orders were sorted according to their basic end date, the processing time for each job was identical, and thus the sort reduced to just the basic end dates. Since the last test had different lengths for the jobs, this was no longer true.



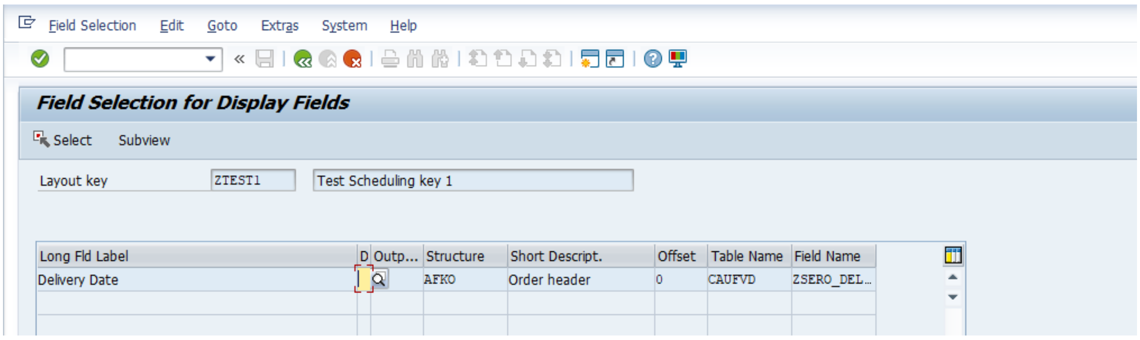
Picture 7 The production orders and the latest start date.

The starting dates and times were noted from earliest to latest and the orders dispatched. The outcome confirmed the hypothesis, the orders were sequenced in order to their latest starting dates and times, represented with the starting points of the red bars in picture 7.

This finding allows to make inferences about the capabilities of the tool for scheduling with due dates. The scheduling capabilities can consider due dates at a very basic sorting level. The due dates are represented by the basic finish dates of the orders and the processing time without directly considering the workload. Because the algorithm only performs sorting according to this date, the actual due date performance is not optimized. If the capacity is not able to handle the due date load, the tool will still order jobs by their latest start date, and thus all of the jobs will be late until there is enough capacity left over to catch up.

Another important finding is the earlier found combination that the act of scheduling changes the basic end date along with that the latest start date, the sequencing variable is defined by the basic end date. What this means is that whenever a job is scheduled, the due date is committed to. If a job has a delivery date two weeks ahead and it is scheduled to be performed in two days, the due date is effectively moved to two days when scheduled, since the basic finishing date is moved and the latest start calculated from it. Now if a work that should be performed in three days appears and the scheduler deallocates the work scheduled and reschedules, the job with a delivery date in two weeks will be higher priority according to the sorting algorithm, despite the other work having a greater criticality in reality. For this reason the basic end date appears a problematic solution for selecting the scheduling order.

A third test is performed whether the configuration of sorting orders functions as expected and whether the delivery date can be used as a sorting basis. A new dispatching order was created in the ERP with the transaction CY39. This order contains just a field called “delivery date”.



Picture 8 A custom sequencing rule for delivery date in CY39.

To test this, a set of five jobs with delivery dates was dispatched. The expected outcome was that the orders would be sorted in a descending order of delivery dates. The outcome was as expected; the jobs were ordered along their delivery dates. This is equivalent to a common dispatching rule, the earliest due date.

4.2.3 Sequence generation hierarchy strictness

The sequences can be defined with several categories, such as sequencing by setup groups or delivery dates, and the parameter highest in the list of the definition transaction CY39 seems to have the greatest impact. Another interesting question is whether the sequences are constructed hierarchically or with weights. Effectively this means whether a category lower in the list can override a higher category if the outcomes are important enough, or if the lower categories can only perform sorting when a higher category has equal values.

To test this, another custom sorting order was made. This sorting order considered the basic finish date as the first parameter and delivery date second. This sorting order is tested with a built test scenario: there are five orders with delivery dates. The orders are placed in such a way that the basic finish dates are in a 6-day span. The basic finish dates are in opposite order of the delivery dates, which are from a month span with some of the work being overdue. One set of delivery dates is placed with the same basic finish times. If the hierarchy of the algorithm is not strict, the orders are expected to be sorted by the delivery dates due to the high variation between the extreme ends. If the orders are sorted by the basic finish dates, then the hierarchy of the categories is strict. Either way the pair with identical basic finishing dates is expected to be ordered by delivery date.

Job number	Delivery date	Basic finish
1	26.6.2025	6.6.2025
2	26.6.2025	9.6.2025
3	23.6.2025	9.6.2025
4	13.6.2025	10.6.2025
5	28.5.2025	11.6.2025
6	28.5.2025	12.6.2025

Table 5 Jobs used in the test.

The outcome of the test was that the products were scheduled strictly in the order of the scheduled finish date without the delivery date having an effect, apart from the two with an identical basic finish date. Since the span of the basic finish dates was very small while the span of the delivery dates was large, it is deemed that the sorting algorithm, while allowing specifying several levels of sorting, is strictly hierarchical. A lower level cannot change the outcome of a higher-level parameter in the sort. The algorithm allows sorting only along a single value with the other values effectively reserved for tie breaking.

4.2.4 Schedule manipulation

Due to unforeseen changes in the production floor there is an expected need for the scheduling tool to support schedule manipulation. The interface usability requirements for a scheduling tool being used among many other tasks by the production line supervisors are likely high. The apparent usability of the ERP solution's production levelling tool is lower compared to commercial specific scheduling solutions. A set of tests are performed to explore the practical scheduling task capabilities of the program.

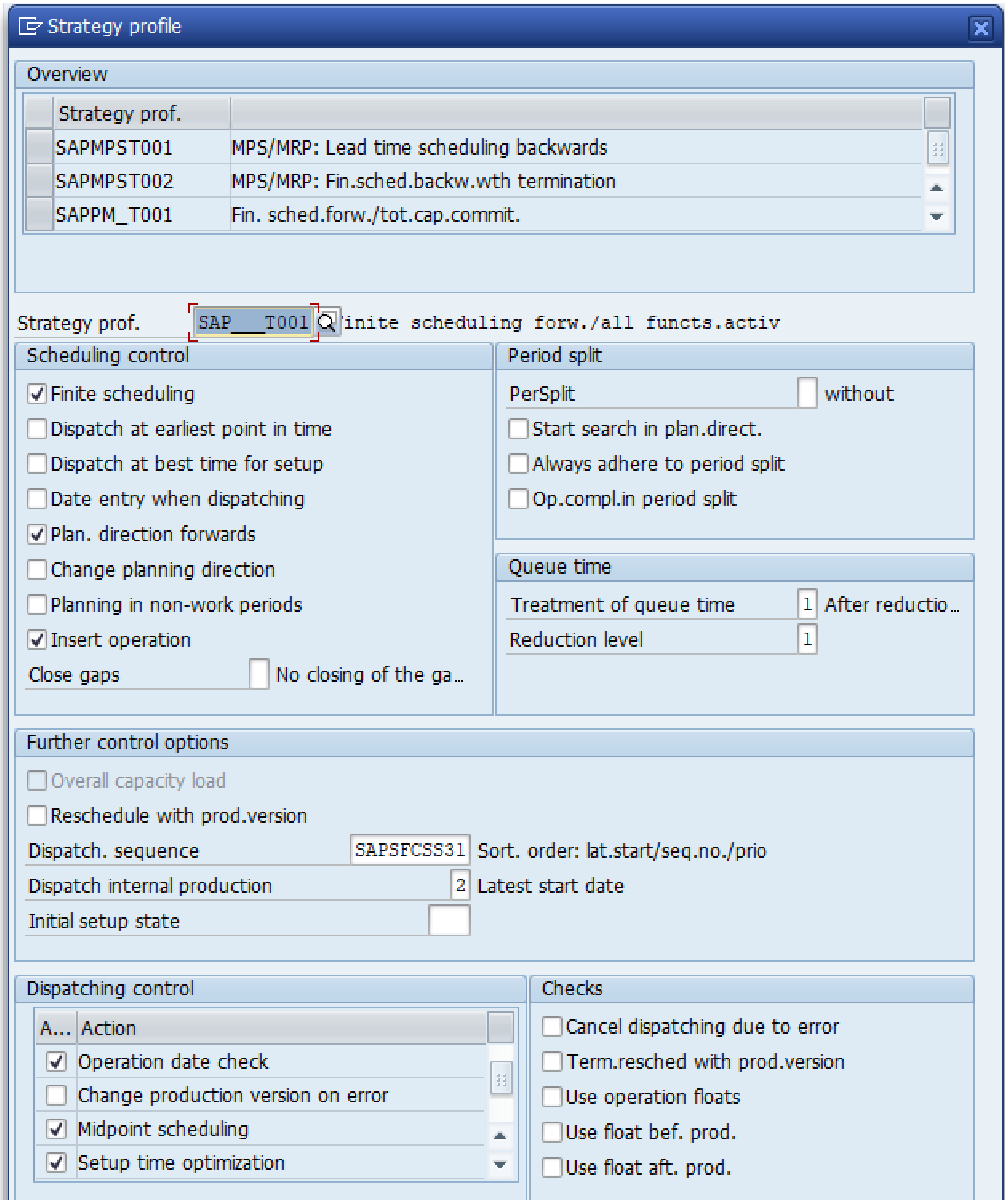
4.2.4.1 Inserting a production order in a schedule

The capability of production order to be inserted into the middle of the schedule is tested. This test is performed with a number of subtests. The first test concerns a simple case where the order is simply inserted without considering the sequence dependent setup time. The second test attempts to get the schedule to react to the sequence dependent setup time. The third test attempts to test if the tool can select an insertion spot considering the setup time.

The first subtest tests whether a production order can be inserted into the schedule without consideration of the sequence dependent setup time. This functionality is noted to work. If the scheduling strategy in CM27 is configured with “insert operation”, the task can either be dispatched to its current location in the order pool or the task can be dragged to any point on the schedule. This action moves all the later operations to the future, right-shifts them by the time it takes to perform the added operation.

The second subtest tests whether a production order can be inserted, and the system recognises the need for both the inserted work and the following work in the sequence to change their setup times. This functionality does not appear to work as would be intuitively expected from above. When inserted into the sequence by dragging, the setup times are not updated, even when the “setup time optimization” -setting is selected. Furthermore, the “setup time optimization” -function appears to disable the “insert operation” -functionality from dispatching, only ever inserting the operation at the end of the schedule. Insertion to a schedule where sequence dependent setup times were considered is deemed not correctly functional in CM27.

This was further tested with a job that has a tight delivery date. In this test, a job is dispatched to the schedule with a tight due date. The existing schedule is scheduled according to setup families. The job with the tight delivery date has a setup family that that belongs to the last processed family. If the work was processed at the first slot of its’ own setup family slot in the current schedule, it will be a week late. If the work is scheduled to the very end, it will be nearly two weeks late. The insertion was tested with the following settings.



Picture 9 Job insertion with “setup time optimization” on.

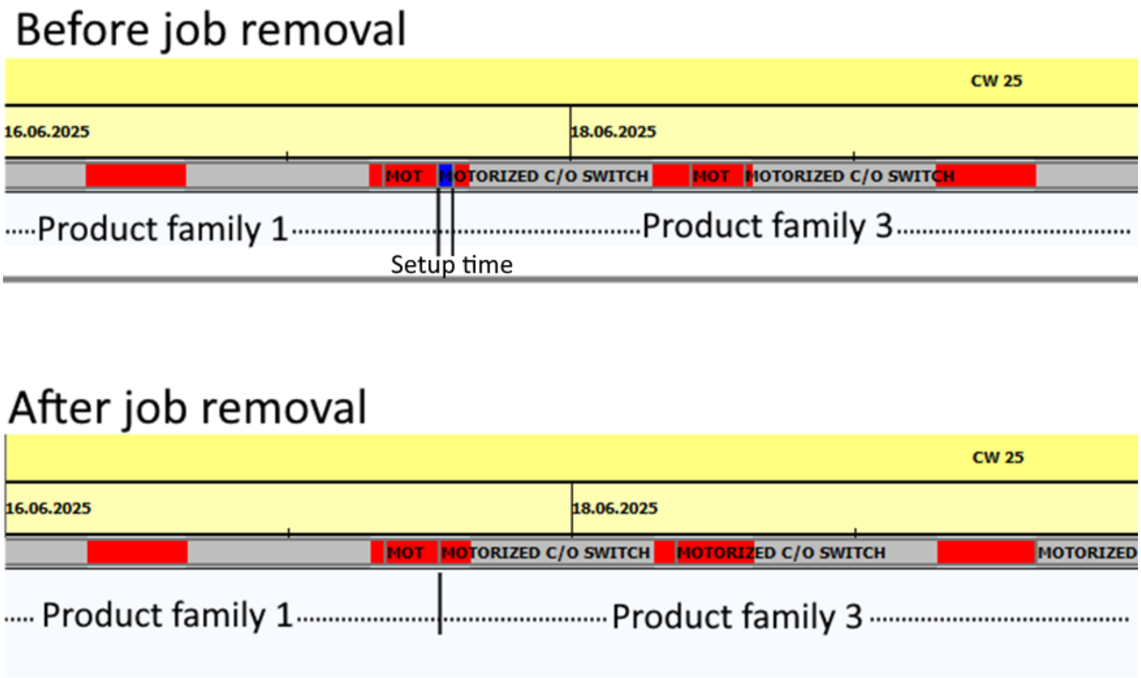
This test resulted in the worst expected result; the work was added to the end of the schedule.

4.2.4.2 Removing a job from the middle of a schedule

A work order can be removed from the schedule to the available job pool. Often this action will be performed with an intention that there should be no gap; any later orders in the schedule should be left-shifted to fill in the gap.

The first test is whether this action can be performed when not considering sequence dependent setup times. This action is possible in the system by entering the strategy profile and selecting an option to the field “close gaps”. Whenever a production order is deallocated, the work scheduled after the operation shifts earlier to fill in the gap.

The second test concerns the removal of a job while there is a product family change between the new neighbours of the resulting schedule. The expected functionality for this action is that the schedule should note the product family setup change between the new product neighbours. The outcome is however that the tool does not recognize such a move, and when the first job of a setup family is removed, the setup time is re-moved along with the work, as visualised in picture 10.



Picture 10 A job is removed, eliminating the setup time from the schedule.

4.2.4.3 Right shifting and job swapping

Some dynamic events on the production floor may require delaying every job at a time. When the schedule is considered a sequence for a single production line, some minor discrepancies between the planned and actual schedule times are not necessarily impactful to the real outcome. There might however be a need to perform right-shifting. Right-shifting was deemed to not be possible in the tool.

Another operation that would possibly be interesting is the ability of swapping the order of two jobs. This can be performed by using the settings "close gaps" and "insert operation" and dragging the production orders to their places. This operation does not adjust the sequence dependent setup times, just as the earlier operations. It is notable that this operation appears to have significantly different outcomes depending on other settings of the strategy profile. If the planning direction is backwards, the gap closing algorithm will reschedule the whole schedule by backwards planning. This can especially a problem because the tool does not have an undo -function apart from the save function.

4.2.4.4 A table of schedule manipulation test outcomes

For clarity, the outcomes of the above tests are charted below.

Test	Can the manipulation be performed?
Inserting a job	Yes
Inserting a job, the setup times are updated	No
Inserting a job automatically according to the setup time	No
Removing a job without creating a gap	Yes
Removing a job without creating a gap, the setup times are updated	No
Right shifting a schedule	No
Swapping two jobs	Yes

Table 6 Outcomes of the CM27 schedule manipulation tests.

5 Conclusions

The thesis has explored the practical production scheduling tasks and the different interfaces the task has with the production environment, business environment and the production scheduling tool found in the literature. Two dynamic factors were measured from the case company to gather information about the interaction in these interfaces. The due date tightness that was measured also acts as a factor in the complexity of the scheduling problem.

As identified from the literature, there are several available approaches to production scheduling. Each scheduling approach can be considered built around different assumptions about the production scheduling problem. The applicability of different production scheduling approaches depends on how well the environment to be scheduled can be modelled under the assumptions of the scheduling model. A vast majority of scheduling literature is based on simulations and there is little information on practical implementations and the fit of the approaches to practical use.

A notable assumption divide is the selection between online and predictive approaches. One group of scheduling approaches uses online and reactive scheduling approaches, and the other group attempts to build schedules for the future with offline, predictive approaches. The major difference in the assumptions of the two groups is that in predictive scheduling the disruptions are assumed to either be completely absent or at least not significant enough to override the benefits of predictive scheduling.

Considering the measured level of dynamism in practice, predictive production scheduling is likely feasible, with the production lines arriving workload being typically several days or even a week out. A combined model of production scheduling for all the production lines together is possible but may need to consider the dynamic events more thoroughly.

The objectives and constraints of the scheduling algorithm are similarly situational. A traditional objective is the minimisation of makespan which attempts to pack the production steps into as short of a time period as possible. This objective has been criticised in the literature and might not sufficiently discriminate between practically good and bad schedules. Flowtime and WIP related objectives have been proposed as an alternative, but the optimal WIP level depends on the variation of the production environment, a measure that production scheduling algorithms generally do not consider.

The notion by Framinan and Ruiz (2012) that production scheduling objectives do not need to be directly aligned with strategical objectives is noted here. Higher levels of production planning and operational strategy planning aim to align the production with the strategy, and their output acts as the constraints and overall objectives of the production system. Production scheduling appears to be of the greatest use in solving specific problems that impede the attainment of these goals. Production scheduling indirectly serves a role in the attainment of these strategical goals, but modelling of all the strategic goals directly into the system may not be necessary. Some of the strategical goals are already built into the constraints in which production scheduling is performed and further modelling as a goal of the production scheduling tool may complicate the task of solving the operational problems.

There is another important decision to be made with the selection of the number and type of objectives. Where too few objectives might lead to the algorithm making solutions that are truly not performant in measures not specified by the objectives, too many objectives might cause an overfit of the algorithm: the truly best performing schedules are constrained out of consideration due to wrong expectations of the production system, as every objective carries with it assumptions.

An important note suggested by Hopp and Spearman (2011, pp. 544–549) is that production scheduling is often fed infeasible inputs as constraints with the expectation that the production schedule would be able to fix the outcome. An example of this are due dates. Since due dates are likely a high priority objective to a production scheduling tool, having unnecessarily tight due dates as in input to the scheduling task will constrain the outcomes possible from the production schedule. When these due dates are tight for valid business reasons this is unavoidable, but unnecessary due date tightness as an input can harm production scheduling processes.

This calls for production scheduling to have an active role in the quotation of due dates. Production scheduling, when performed predictively, creates up to date information about the finite capacity relevant to production planning and due date setting.

The participation of the human scheduler in the scheduling task has been identified in the literature as important in scheduling outcomes. There can be seen to be several interfaces between the production schedule and the production environment where the human scheduler acts.

The scheduler acts as an information collector and filterer to the schedule creation process. A majority of the theoretical scheduling literature consists of simplified simulated cases where all the relevant information is known in the scheduling task. Whenever a disruption is modelled, its importance and effect are known. In the practical production scheduling task this assumption likely does not hold often. The production scheduler has been noted in case studies to collect and filter information from wider contexts and use it to base their scheduling decisions. From this point of view, the scheduling system and scheduling tool should allow the scheduler to incorporate at least the relevant part of this information into the schedule output.

When the dynamicity and specificity of the production environment increases, the information flow becomes more important. When the dynamicity becomes higher, the scheduler will need information faster. In such environments it might be reasonable for production scheduling to be performed by the production line supervisors. With such organisation there is the additional benefit of the schedule execution being naturally close to the scheduler.

A hypothetical completely predictable production system could theoretically be planned by a centralised production planner and scheduler. A centralised scheduling system, when considering the case studies of production scheduling in the literature, appears to be common in companies with a long history of production scheduling. A possibility worth noting here though, is that such companies will likely have very different production environments to companies that have not considered formal production scheduling systems necessary. The companies with long histories of production scheduling likely are those in environments where formal planned production scheduling is absolutely necessary for basic functioning.

When production scheduling is divided among several part-time schedulers, such as production line supervisors, for whom scheduling is just one task among other tasks, the usability and readability of the scheduling tool will have pronounced importance. A positive finding from the literature in this regard is that the algorithm might not need to be understandable by the scheduler for performance, just the output. In a use environment where the user must often context switch between tasks, the tool needs to be usable with a reasonable effort and provide a clear output.

The robustness of production scheduling, reactive rescheduling and statistical process control principles have an interesting area of intersection that is relevant to the production scheduling task.

Statistical process control has the distinction between natural variation of the system and special cause variation (Breyfogle, 2003, pp. 213–215). Natural variation is the typical variation that exists in the system parameters due to a combination of small, mostly uncontrollable causes. Special cause variation on the other hand is variation that is caused by some specific event, and that is not typical of the system. Reacting to a variation caused by the natural causes increases the variation in the system (Hopp & Spearman, 2011, p. 308). Hopp and Spearman (2011, p. 308) also point that Advanced Planning and Control systems are particularly susceptible to reacting to random noise. This causes practical considerations both regarding the rescheduling strategy and the robustness of schedules.

Rescheduling, whether by performing complete scheduling or fixing the schedule, is a reaction to a variation from the original scheduled plan. Here mainly two cases can be imagined. The first case is that there is a true special cause variation in the system, and the correct action is to perform a reschedule. Another case could be that the disruption in the schedule is not due to any specific cause, but rather a disconnect between the natural variation of the production environment and the scheduling approach. In this case while rescheduling might fix the immediate problem caused by the disconnect, the underlying model and parameters of the scheduling might need to be changed as well.

If a schedule fails due to natural variation of the production system, this acts to signal that the model underpinning the scheduling is not able to produce feasible schedules. This is not only relevant when considering the robustness of the scheduling system, but the overall control system and assumptions about the factory floor. However, when the disruption in the schedule was caused by a special cause variation, this does not indicate strongly that the underlying model would need to be changed.

When considering more sophisticated production scheduling tools, especially tools where scheduling parameters are present or that have multiple objectives weighed with an a priori -approach, the timely feedback of performance information is a priority for the scheduling system. The identification of the scheduling tool parameters and when there is a need to adjust them requires accurate information about the actual executed outcomes of the production schedule. This information should be based on not just the performance of the schedule, but the outcome performance of the production system.

5.1 Options for the implementation of production scheduling for the case company

There is a major difference between the product flow inside a production line in the case company and production scheduling approaches in the literature. In all of the scheduling approaches in the literature review where the flow is specified, a flow between workstations in complete batches is assumed. In the case company while production orders are handled as batches, the product flow inside a production line is done either with a single-piece flow, or with very small batches. The move in complete batches only occurs at scopes greater than the production lines, such as the movement from production lines to the dispatching department.

Considering how common lines with a single piece flow are, a question can be asked why the lack of representation in the scheduling literature? Such flow shop mixed production lines with a takt time and a low variation of processing time in the relevant flow unit, a single product, can be considered self-managing for the flow. This same property cannot be found in job shops or batch flow shops, and thus these approaches need production scheduling to manage the flow. Unless there are sequence dependent setup times, there just are not that many benefits for one production sequence over another if the unit of flow has low variance in length. This view can be seen to align with Romero-Silva et al. (2024) framework of scheduling approaches: the complexity of the scheduling task production line by itself is low, suggesting control-theoretic, workload control, or dispatching approaches.

This raises two points relevant to the practical implementation of a production scheduling system in the case company. First, accurate modelling of the product flow inside the production line with production scheduling may not be possible with the current approaches. This does, however, point to further opportunities: the production lines' flow can be considered relatively self-managing, posing the possibility for models where a production line is a modelled as a single workstation.

From this, there are three apparent approaches the case company could take to implement production scheduling, the approaches are listed with an increasing order of work necessary. There is no clear one approach that would be better than others, rather options with different levels of investment and expected benefits. While there are no existing guidelines for the levels of dynamicity in the literature, the dynamicity measured in the company production is considered to be relatively low.

Considering that the production lines have a design that maintains a reasonably good production flow inherently, a very simple approach building any schedule is enough to capture easy wins from production scheduling. Any form of predictive schedule for production will reduce the variability experienced by logistics. The tested tool currently existing in the company's ERP system, CM27 can be used to fulfil this role, although the limitations of the user interface for schedule manipulation operations must be acknowledged. The implementation of this is discussed in a separate subchapter and appendix 1.

The second option is the utilisation of commercially available production scheduling software to perform optimisation on the production lines as separate units. In this solution especially the capable to promise estimation, due date quoting and production planning outcomes are the desired benefits. The production line in this approach may be modelled either as a single machine as above or an attempt may be made to perform optimisation of the existing machine scheduling problem identified in the company in some of the production lines: there are parallel testing machines with different job eligibility.

The tester scheduling problem however is complicated from the viewpoint of production scheduling, as typical scheduling approaches only consider material flows in batches. The selection of tools available for productive solution of this problem is limited, and their capability to model realistic production flow must be specifically confirmed. The testing machine problem has another complication, as the flow from the tester adds another element not typically present in production scheduling problems. There is an automated buffer solution after the testing station that interrupts the flow occasionally to change a pallet used for the testing. This introduces a block occasionally and is also dependent on the mix of the single product flow. Essentially this acts as a frequent random disruption for the model.

The third option uses the earlier mentioned opportunity posed by the self-flow managing property of the production lines. The production lines may be reasonably well modelled as a single workstation. This allows a scope extension of the production scheduling system to include the internal dispatching logistics, reducing especially the need for finished goods inventory warehousing. In the current system, due to the unknown production order and workload effects, the customer service level is secured against the variation caused by an essentially unknown production schedule by a time buffer. If production can be scheduled to be dispatched directly, the need to utilise time buffers for make-to-order production should be reduced. In this model the dynamicity of the production environment is especially important, since any production workstation can trigger a disruption of the schedule. At this level of scheduling, the management of arriving rush work as well will likely be critical, and thus both a good due date quoting system and a management of capacity buffering to accommodate rush orders are likely necessary.

5.2 How could production scheduling be used in the case company to ensure the best possible material flow?

An identified problem regarding the informal online production scheduling practice currently in the company is that there is no predictable and visible production schedule. This poses problems for internal material logistics. Any sequence, whether optimised or not, would significantly reduce the variability experienced by the inbound material logistics. With lower variability, there is a reduced need to buffer either with longer lead times, such as bringing in material early just in case, or the need to have spare capacity to respond to multiple sudden material needs at the same time. This can be considered an easy win for the production scheduling, implementable with little technical difficulty, since any schedule good enough to not compromise other performance parameters will be enough.

As noted earlier, the current tool in the ERP of the case company is likely able to cover the basic needs of a system to build a good production schedule that significantly decreases the variability of the production schedule experienced by the material logistics. A schedule capable of this is any schedule that has a time-bound sequence of jobs.

5.3 Can the existing ERP system be used as a tool for production scheduling?

The capacity levelling tool in the ERP is limited in its production scheduling capabilities, but as noted earlier, even limited scheduling capabilities will allow the capture of some of the benefits. The found key limitations of the applicability of the tool are the single objective limitation and user interface limitations in schedule manipulation.

The major limitation of the CM27 capacity levelling tool for production scheduling was identified: the tool bases the schedules only on a single factor, either by building a sequence with an optimal sequence dependent setup pattern or by selecting a single parameter to sort the orders by. However, the optimisation of the setup pattern does not consider the due dates and thus will violate due dates in the creation of the setup pattern.

Another notable limitation found in the testing is the schedule manipulation capabilities. Whenever sequence dependent setup times are used, the schedule manipulation tools, such as job insertion and switching do not function as would be expected. These interface quirks are especially important when considering that the tool would be used by the production line supervisors for whom the production scheduling task is one among many tasks. They currently have their own individual systems to schedule production, and this tool must compete with that in the limited usage time available.

Despite the limitations, the CM27 production levelling tool has capabilities to act as an interface to implement simple dispatching rules. For this, two approaches are presented. These are presented in an increasing order of work required.

A traditional simple dispatching rule can be implemented directly using the production levelling tool. By configuring the sorting rule in the program to consider the job delivery date first, the output will be equivalent to a dispatching rule called earliest due date. This dispatching rule date can be performant in environments where the jobs are similarly sized, and routings are consistent (Hopp & Spearman, 2011, p. 146). Both assumptions hold in the case of the production lines, as the routings are consistent in a flow shop, and the flow of jobs is level due to single piece flow with predictable process times. This can be paired with a lower parameter of setup key as a secondary goal to group similar setup products together within the day.

A more involved approach to dispatching would be the implementation of dispatching rules not directly available through the sorting. These approaches would need extension programming with the ERP. The dispatching sequences in the production levelling tool reference a value from a table specified in the configuration for each job and sorts either in ascending or descending order from these. If a dispatching rule can be used to calculate a value and this value is written for each job, the tool can then use this value to sort the production order according to the output.

For this approach, composite rules or heuristics are proposed. Two composite dispatching rules, earliest due date with setup times and earliest due date with priority are proposed for this application. In addition, a completely new heuristic is tested. The earliest due date with setup times and the new heuristic is compared to the earliest due date rule with a simulated workload with sequence dependent setup times. A rule combining the earliest due date with setup times is proposed for this task for production lines with notable setup times. A rule combining the earliest due date with priority is proposed for lines without relevant setup times. The details of this can be found in appendix 1.

6 Limitations and suggestions for further research

A large part of the study consists of the literature review. Because the topic of practical production scheduling necessitates a wide scope for the literature review, the scope was managed by considering the literature only through the view of what is interesting for the case company's production environment.

The environment considered is one where the production lines are a mostly make-to-order flow shop with production orders handled as batches, but the production lines flowing with a single piece flow. The single piece flow products are roughly equal in process length within each production line.

Due to this limitation, the research setting and literature review conclusions can only be considered relevant under similar assumptions and production environments.

6.1 Suggestions for further research

The contingency approach to evaluate the fit of different production scheduling approaches to different environments appears to be important to reduce the gap between production scheduling theory and practice. The approaches suggested by Tenhiälä (2011) and Romero-Silva et al. (2024) along with the practical task feedback loop by Romero-Silva et al. (2015) provide a basis for the analysis of scheduling environments.

Still, the model of scheduling approach fit appears to be missing dimensions that are relevant to the selection of scheduling approaches. Some features of the scheduled problem have been found in the literature to be relevant to the outcome, such as due date tightness and congestion level (Oukil & El-Bouri, 2021), dynamicity (De Snoo et al., 2011; Romero-Silva et al., 2024), complexity (Romero-Silva et al., 2024), predictability (Tenhiälä, 2011), scheduling direction (Kamaruddin et al., 2013; Simio, 2019), process time variation (Tokola et al., 2017) and idle-time in the schedule (Abumaizar & Svestka, 1997). When these contingency parameters remain specific interests in individual studies and not systematised, a comprehensive view of the fits between different scheduling approaches and production environments cannot be built. In addition, theoretical research building the scheduling algorithms lacks the tools to position the algorithms since the applicability of different approaches and problem models are not known.

Likely not all of the individual scheduling problem parameters are necessary to be accounted, as some parameters can inherently outline other parameters of the problem. A job shop carries with it different assumed parameters than a flow shop and have a different expected applicability. However there appears to be a reason to further research what all environment parameters are accounted for with the current division of scheduling models, and what parameters are impactful but not currently captured as important to the scope of the application in the literature.

References

Abedinnia, H., Glock, C. H., Grosse, E. H., & Schneider, M. (2017). Machine scheduling problems in production: A tertiary study. *Computers & Industrial Engineering*, *111*, 403–416. https://doi.org/10.1016/j.cie.2017.06.026

Abumaizar, R. J., & Svestka, J. A. (1997). Rescheduling job shops under random disruptions. *International Journal of Production Research*, *35*(7), 2065–2082. https://doi.org/10.1080/002075497195074

Ahmadian, M. M., & Salehipour, A. (2021). The just-in-time job-shop scheduling problem with distinct due-dates for operations. *Journal of Heuristics*, *27*(1–2), 175–204. https://doi.org/10.1007/s10732-020-09458-6

Allahverdi, A. (2015). The third comprehensive survey on scheduling problems with setup times/costs. *European Journal of Operational Research*, *246*(2), 345–378. https://doi.org/10.1016/j.ejor.2015.04.004

Beck, J. C., & Refalo, P. (2003). A Hybrid Approach to Scheduling with Earliness and Tardiness Costs. *Annals of Operations Research*, *118*(1–4), 49–71. https://doi.org/10.1023/A:1021849405707

Berglund, M., Guinery, J., & Karltun, J. (2011). The Unsung Contribution of Production Planners and Schedulers at Production and Sales Interfaces. In J. C. Fransoo, T. Waefler, & J. R. Wilson (Eds.), *Behavioral Operations in Planning and Scheduling* (pp. 47–81). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-13382-4_4

Bertsimas, D., & Sim, M. (2004). The Price of Robustness. *Operations Research*, *52*, 35–53. https://doi.org/10.1287/opre.1030.0065

Breyfogle, F. W. (2003). *Implementing six sigma : smarter solutions using statistical methods* (2nd ed). Wiley.

Carvalho, A. N., Scavarda, L. F., & Lustosa, L. J. (2014). Implementing finite capacity production scheduling: lessons from a practical case. *International Journal of Production Research*, *52*(4), 1215–1230. https://doi.org/10.1080/00207543.2013.848484

Cegarra, J., & Hoc, J.-M. (2008). The role of algorithm and result comprehensibility of automated scheduling on complacency. *Human Factors and Ergonomics in Manufacturing & Service Industries*, *18*(6), 603–620. https://doi.org/10.1002/hfm.20129

Cegarra, J., & van Wezel, W. (2011). A Comparison of Task Analysis Methods for Planning and Scheduling. In J. C. Fransoo, T. Waefler, & J. R. Wilson (Eds.), *Behavioral Operations in Planning and Scheduling* (pp. 323–338). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-13382-4_13

Cohn, A., Kurnaz, S., Guan, Y., & Jiang, Y. (2010). Trading off between makespan and customer responsiveness in flow shop sequencing. *International Journal of Production Research*, *48*(22), 6777–6797. https://doi.org/10.1080/00207540903280531

de Snoo, C., & van Wezel, W. (2014). Coordination and Task Interdependence during Schedule Adaptation. *Human Factors and Ergonomics in Manufacturing & Service Industries*, *24*(2), 139–151. https://doi.org/10.1002/hfm.20363

De Snoo, C., Van Wezel, W., & Jorna, R. J. (2011). An empirical investigation of scheduling performance criteria. *Journal of Operations Management*, *29*(3), 181–193. https://doi.org/10.1016/j.jom.2010.12.006

Dong, Y.-H., & Jang, J. (2012). Production rescheduling for machine breakdown at a job shop. *International Journal of Production Research*, *50*(10), 2681–2691. https://doi.org/10.1080/00207543.2011.579637

Dudek, R. A., Panwalkar, S. S., & Smith, M. L. (1992). The Lessons of Flowshop Scheduling Research. *Operations Research*, *40*(1), 7–13. https://doi.org/10.1287/opre.40.1.7

Framinan, J. M., & Leisten, R. (2010). Available-to-promise (ATP) systems: a classification and framework for analysis. *International Journal of Production Research*, *48*(11), 3079–3103. https://doi.org/10.1080/00207540902810544

Framinan, J. M., & Ruiz, R. (2010). Architecture of manufacturing scheduling systems: Literature review and an integrated proposal. *European Journal of Operational Research*, *205*(2), 237–246. https://doi.org/10.1016/j.ejor.2009.09.026

Framinan, J. M., & Ruiz, R. (2012). Guidelines for the deployment and implementation of manufacturing scheduling systems. *International Journal of Production Research*, *50*(7), 1799–1812. https://doi.org/10.1080/00207543.2011.564670

Fransoo, J. C., & Wiers, V. C. S. (2008). An empirical investigation of the neglect of MRP information by production planners. *Production Planning & Control*, *19*(8), 781–787. https://doi.org/10.1080/09537280802568163

Galla, M. (2020, June). *From SAP APO to SAP S/4HANA and SAP IBP*. NTT Data. https://nttdata-solutions.com/us/blog/from-sap-apo-to-sap-s-4hana-and-sap-ibp/

Gasser, R., Fischer, K., & Wäfler, T. (2011). Decision Making in Planning and Scheduling: A Field Study of Planning Behaviour in Manufacturing. In J. C. Fransoo, T. Waefler, & J. R. Wilson (Eds.), *Behavioral Operations in Planning and Scheduling* (pp. 11–30). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-13382-4_2

Haverila, M., Uusi-Rauva, E., Kouri, I., & Miettinen, A. (2009). *Teollisuustalous* (6. p.). Infacs.

Hopp, W. J., & Spearman, M. L. (2011). *Factory physics* (Third edition). Waveland Press.

Jarboui, B., Siarry, P., & Teghem, J. (2013). *Metaheuristics for Production Scheduling*. Newark: John Wiley & Sons, Incorporated. https://doi.org/10.1002/9781118731598

Jodlbauer, H., & Tripathi, S. (2024). Due date quoting and rescheduling in a fixed production sequence. *International Journal of Production Research*, *62*(4), 1206–1220. https://doi.org/10.1080/00207543.2023.2179342

Kamaruddin, S., A. Khan, Z., Noor Siddiquee, A., & Wong, Y.-S. (2013). The impact of variety of orders and different number of workers on production scheduling performance: A simulation approach. *Journal of Manufacturing Technology Management*, *24*(8), 1123–1142. https://doi.org/10.1108/JMTM-12-2010-0083

Katragjini, K., Vallada, E., & Ruiz, R. (2013). Flow shop rescheduling under different types of disruption. *International Journal of Production Research*, *51*(3), 780–797. https://doi.org/10.1080/00207543.2012.666856

Kim, D. H. (2000). *System Archetypes I : Diagnosing Systemic Issues and Designing High-Leverage Interventions*. Pegasus Communications, Inc.

Klemmt, A., Horn, S., Weigert, G., & Wolter, K.-J. (2009). Simulation-based optimization vs. mathematical programming: A hybrid approach for optimizing scheduling problems. *Robotics and Computer-Integrated Manufacturing*, *25*(6), 917–925. https://doi.org/10.1016/j.rcim.2009.04.012

Li, Y., Carabelli, S., Fadda, E., Manerba, D., Tadei, R., & Terzo, O. (2020). Machine learning and optimization for production rescheduling in Industry 4.0. *International Journal* *of Advanced Manufacturing Technology*, *110*(9–10), 2445–2463. https://doi.org/10.1007/s00170-020-05850-5

Li, Z., & Ierapetritou, M. (2008). Process scheduling under uncertainty: Review and challenges. *Computers & Chemical Engineering*, *32*(4), 715–727. https://doi.org/10.1016/j.compchemeng.2007.03.001

Liang, Z., Zhong, P., Liu, M., & Zhang, C. (2023). A generic enhanced search framework based on genetic algorithm: Case study on job shop scheduling problem. *Journal of Intelligent & Fuzzy Systems*, *45*(4), 7095–7111. https://doi.org/10.3233/JIFS-230076

Lin, X., Janak, S. L., & Floudas, C. A. (2004). A new robust optimization approach for scheduling under uncertainty:: I. Bounded uncertainty. *Computers & Chemical Engineering*, *28*(6), 1069–1085. https://doi.org/10.1016/j.compchemeng.2003.09.020

Martinsuo, M., Mäkinen, S., Suomala, P., & Lyly-Yrjänäinen, J. (2016). *Teollisuustalous kehittyvässä liiketoiminnassa* (1. painos). Edita.

Méndez, C. A., Cerdá, J., Grossmann, I. E., Harjunkoski, I., & Fahl, M. (2006). State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Computers & Chemical Engineering*, *30*(6), 913–946. https://doi.org/10.1016/j.compchemeng.2006.02.008

Miyata, H. H., & Nagano, M. S. (2019). The blocking flow shop scheduling problem: A comprehensive and conceptual review. *Expert Systems with Applications*, *137*, 130–156. https://doi.org/10.1016/j.eswa.2019.06.069

Morandini, S., Fraboni, F., Hall, M., Quintana-Amate, S., & Pietrantoni, L. (2024). Human factors and emerging needs in aerospace manufacturing planning and scheduling. *Cognition, Technology & Work*. https://doi.org/10.1007/s10111-024-00785-3

Moratori, P., Petrovic, S., & Vázquez-Rodríguez, J. A. (2012). Match-up approaches to a dynamic rescheduling problem. *International Journal of Production Research*, *50*(1), 261–276. https://doi.org/10.1080/00207543.2011.571458

Moscoso, P. G., Fransoo, J. C., & Fischer, D. (2010). An empirical study on reducing planning instability in hierarchical planning systems. *Production Planning & Control*, *21*(4), 413–426. https://doi.org/10.1080/09537280903454172

Ouelhadj, D., & Petrovic, S. (2009). A survey of dynamic scheduling in manufacturing systems. *Journal of Scheduling*, *12*(4), 417–431. https://doi.org/10.1007/s10951-008-0090-8

Oukil, A., & El-Bouri, A. (2021). Ranking dispatching rules in multi-objective dynamic flow shop scheduling: a multi-faceted perspective. *International Journal of Production Research*, *59*(2), 388–411. https://doi.org/10.1080/00207543.2019.1696487

Perez-Gonzalez, P., & Framinan, J. M. (2015). Assessing scheduling policies in a permutation flowshop with common due dates. *International Journal of Production Research*, *53*(19), 5742–5754. https://doi.org/10.1080/00207543.2014.994077

Pinedo, M. (2009). *Planning and scheduling in manufacturing and services* (2nd ed). Springer.

Pinedo, M. (2012). *Scheduling : theory, algorithms, and systems* (4th ed). Springer.

Raheja, A. S., & Subramaniam, V. (2002). Reactive Recovery of Job Shop Schedules - A Review. *International Journal of Advanced Manufacturing Technology*, *19*(10), 756–763. https://doi.org/10.1007/s001700200087

Rajendran, C. (1994). A heuristic for scheduling in flowshop and flowline-based manufacturing cell with multi-criteria. *International Journal of Production Research*, *32*(11), 2541–2558. https://doi.org/10.1080/00207549408957083

Romero-Silva, R., Santos, J., & Hurtado, M. (2015). A framework for studying practical production scheduling. *Production Planning & Control*, *26*(6), 438–450. https://doi.org/10.1080/09537287.2014.919413

Romero-Silva, R., Santos, J., & Hurtado-Hernández, M. (2024). A conceptual framework of the applicability of production scheduling from a contingency theory approach: addressing the theory-practice gap. *Production Planning & Control*, *35*(3), 262–282. https://doi.org/10.1080/09537287.2022.2076627

Ruiz, R., & Maroto, C. (2005). A comprehensive review and evaluation of permutation flowshop heuristics. *European Journal of Operational Research*, *165*(2), 479–494. https://doi.org/10.1016/j.ejor.2004.04.017

SAP. (n.d.). *SAP Advanced Planning and Optimization (APO)*. SAP Help Portal. Retrieved June 6, 2025, from https://help.sap.com/docs/SUPPORT_CONTENT/sapo/3354621778.html?locale=en-US

Selçuk, B., Adan, I. J. B. F., de Kok, T. G., & Fransoo, J. C. (2009). An explicit analysis of the lead time syndrome: stability condition and performance evaluation. *International Journal of Production Research*, *47*(9), 2507–2529. https://doi.org/10.1080/00207540701420552

Simio. (2019, December 12). *Forward Scheduling: What is it and how does it differ from backwards scheduling?* Simio Blog. https://www.simio.com/forward-scheduling-what-is-it-and-how-does-it-differ-from-backwards-scheduling/

Tenhiälä, A. (2011). Contingency theory of capacity planning: The link between process types and planning methods. *Journal of Operations Management*, *29*(1), 65–77. https://doi.org/10.1016/j.jom.2010.05.003

Thürer, M., Fernandes, N. O., Haeussler, S., & Stevenson, M. (2023). Dynamic planned lead times in production planning and control systems: does the lead time syndrome matter? *International Journal of Production Research*, *61*(4), 1268–1282. https://doi.org/10.1080/00207543.2022.2034193

Tiwari, A., Chang, P.-C., Tiwari, M. K., & Kollanoor, N. J. (2015). A Pareto block-based estimation and distribution algorithm for multi-objective permutation flow shop scheduling problem. *International Journal of Production Research*, *53*(3), 793–834. https://doi.org/10.1080/00207543.2014.933273

Tokola, H., Ahlroth, L., & Niemi, E. (2014). A comparison of rescheduling policies for online flow shops to minimize tardiness. *Engineering Optimization*, *46*(2), 165–180. https://doi.org/10.1080/0305215X.2012.753438

Tokola, H., Niemi, E., & Kyrenius, P. (2017). How Lean transformation affects scheduling. *Robotics and Computer-Integrated Manufacturing*, *43*, 171–178. https://doi.org/10.1016/j.rcim.2015.09.012

Tubilla, F., & Gershwin, S. B. (2022). Dynamic scheduling in make-to-stock production systems with setup times and random breakdowns: performance analysis and improved policies. *International Journal of Production Research*, *60*(10), 3263–3281. https://doi.org/10.1080/00207543.2021.1917013

van Wezel, W., Cegarra, J., & Hoc, J.-M. (2011). Allocating Functions to Human and Algorithm in Scheduling. In J. C. Fransoo, T. Waefler, & J. R. Wilson (Eds.), *Behavioral Operations in Planning and Scheduling* (pp. 339–370). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-13382-4_14

Yao, X., Almatooq, N., Askin, R. G., & Gruber, G. (2022). Capacity planning and production scheduling integration: improving operational efficiency via detailed modelling. *International Journal of Production Research*, *60*(24), 7239–7261. https://doi.org/10.1080/00207543.2022.2028031

Zupan, H., Herakovič, N., & Žerovnik, J. (2024). A Robust Heuristics for the Online Job Shop Scheduling Problem. *Algorithms*, *17*(12), 568. https://doi.org/10.3390/a17120568

Appendices

Appendix 1. Heuristic algorithm to create sequences with CM27

The dispatching function of CM27 appears to only really support sequencing performed by sorting according to a single parameter. This limits the direct scheduling usability to only simple dispatching rules. A practically usable such rule is the earliest due date rule (EDD).

It was identified however that the rules are defined by specifying the name of a table and a value in the table. If a value can be written through a custom SAP program for each job to indicate the required order, this can be used to perform better suited scheduling approaches while still keeping investment small compared to an integration of an external application. Recommendations for scheduling methods that are implementable through this approach are outlined in this appendix.

The CM27 approach is best suited for sequencing and scheduling single production lines. In these approaches, an assumption is made that the line can be represented essentially as a single machine due to the flow arrangement that is mostly independent from the sequence. For this approach, the appropriate measures are due date performance and setup time reduction.

Three heuristic algorithms were tested with a simulated workload for the task. The first is EDD, the earliest due date rule that selects the job with the earliest due date. This rule is interesting in the test because it is directly implementable in the system without any programming. The second rule to be tested is a novel algorithm designed specifically for this application. The third rule is a simple combined rule of earliest due date and lowest setup time. This rule will be explained first.

The EDD + Setup time dispatch rule in this test was created with a simple approach: the algorithm first selects the earliest due date job as the first job. From this on, the algorithm ranks all the remaining jobs and selects the job with the lowest value using the following formula, dispatches it, and repeats the process until all jobs are dispatched to a sequence. From here on, the implementation would need to assign a rising value for each job in the sequence depending on their position in the output to be read by CM27.

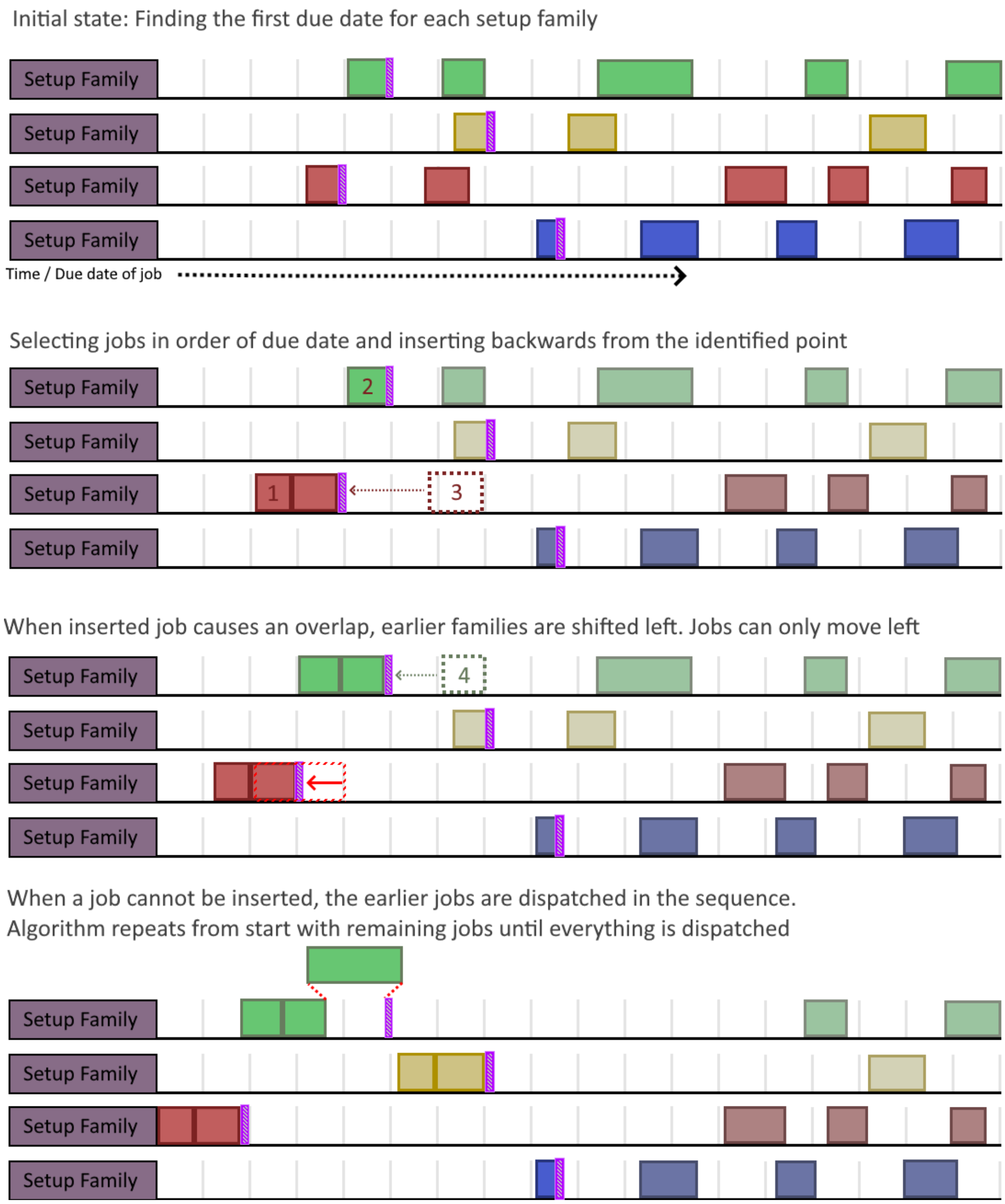
The EDD + Setup time combination rule

$d_j - t_d + t_s * w_s$, where

d_j = due date of the job	t_d = Finishing time of last dispatched job	t_s = Setup time from last dispatched job to this job	w_s = Multiplier of setup time (10 used in testing)

The second heuristic algorithm, a new scheduling algorithm was built specifically for this case. This algorithm attempts to create a schedule that is strictly adherent to the due dates while constructing as long groupings of the same setup families as possible.

The basic loop of the algorithm is that it identifies the earliest job for each setup family and uses these as backwards scheduling points. Then the algorithm selects jobs with the earliest due date with a priority measure as a tie breaker. The jobs are backwards scheduled into their respective setup family queues. If one queue overlaps an earlier setup family queue, the earlier queue is moved left, or earlier in the schedule. The backwards scheduling points can only move left, so the algorithm is incapable of violating due dates. When a job is found that cannot be inserted into the queue without moving the first job into the past, the existing jobs in the queue are dispatched to the production queue forwards in order. The job that did not fit is returned to the job pool. The loop begins again by selecting new earliest jobs for each family and is repeated until all the jobs are dispatched as a continuous sequence. If the first job in pool for any loop is late, then the algorithm uses the EDD + setup method until all late jobs have been dispatched and continues the method once the first job is non-late.



The new scheduling algorithm

Testing the algorithms

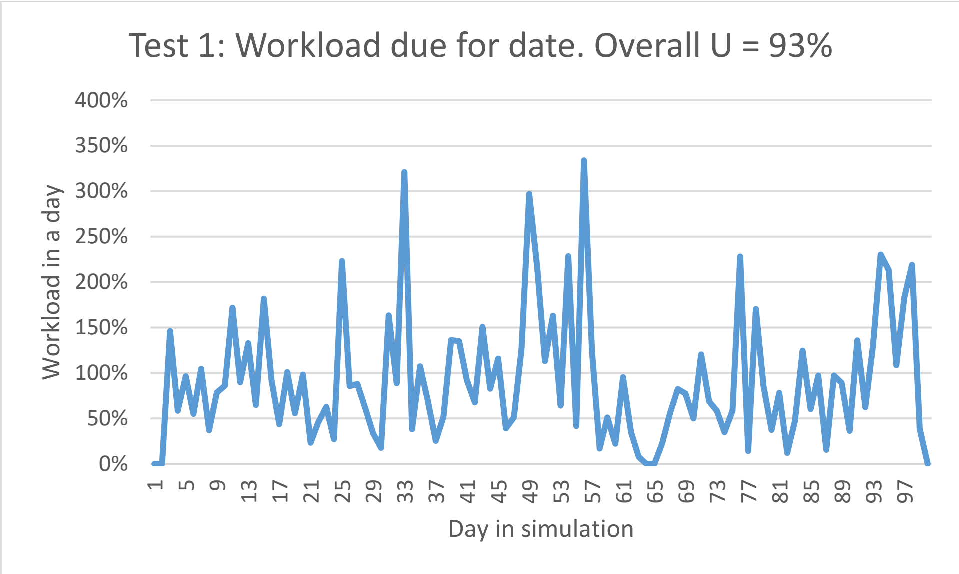
Since the performance of the algorithms is unknown, they were tested with a simple simulation where a simulated work queue with different length jobs dispersed randomly over a schedule was tested. The jobs belong randomly to 4 different setup families with sequence dependent setup times; the same matrix of setup times was used in all tests. All the tests were performed with high utilisation levels. The utilisation level calculation did not consider the setup times, and thus a sequence not considering setup times may end up over capacity. Each test the job set was identical for each of the algorithms within a test. The algorithms were implemented with Python.

There was no test repetition, the tests can only thus be considered indicative. Another key distinction is that since the test assumes a single machine deterministic setup, the utilisation performance cannot be assumed to be indicative of the true utilisation performance. The high utilisation is however necessary to draw differences between the algorithms in such a deterministic model. All of the workload was available to the algorithms at the start, and the algorithms were tasked to forward schedule all of the work.

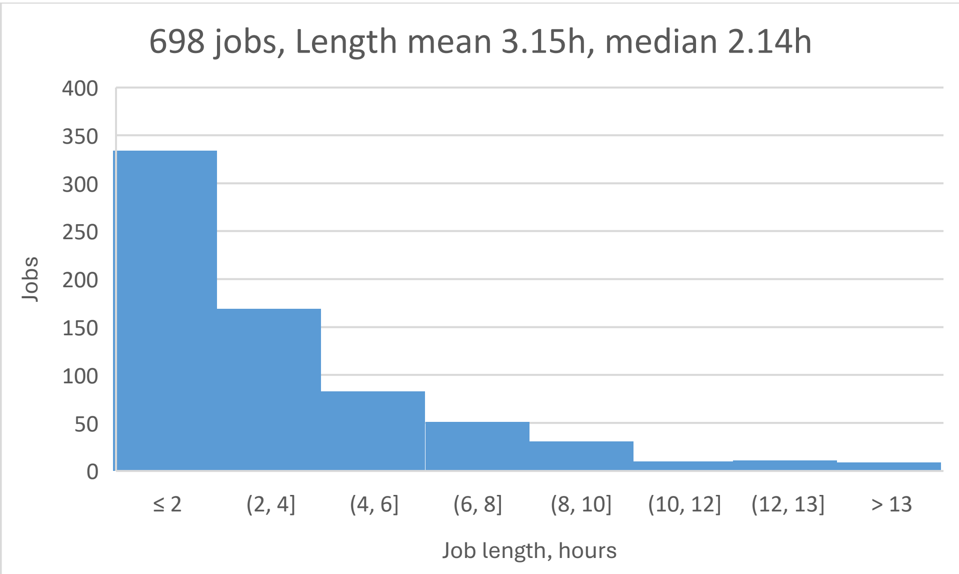
	To 0	To 1	To 2	To 3
From 0	0,00 h	0,60 h	1,50 h	1,20 h
From 1	0,20 h	0,00 h	0,60 h	0,80 h
From 2	0,10 h	0,20 h	0,00 h	0,60 h
From 3	0,20 h	0,40 h	0,50 h	0,00 h

The setup matrix used

Test 1



Workload for each day of the test simulation



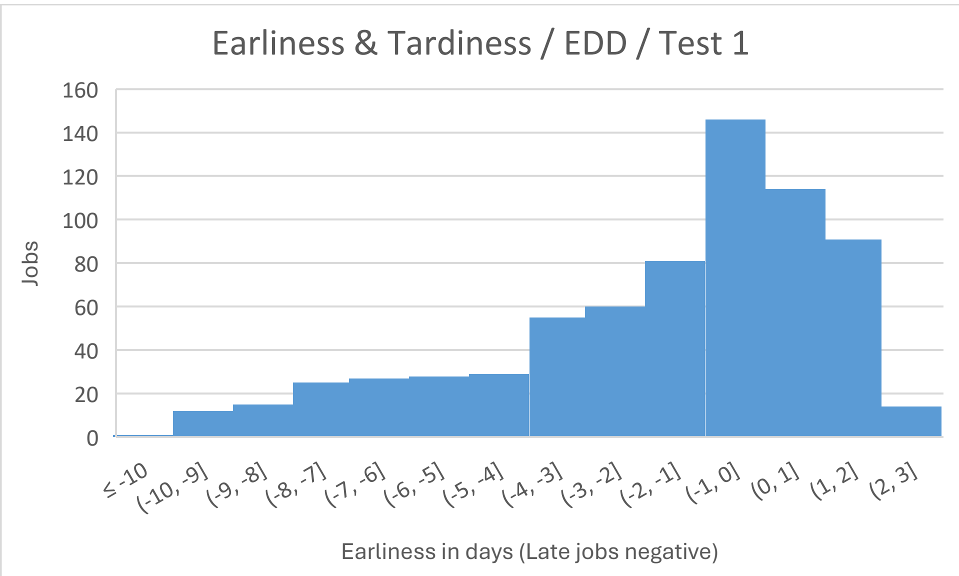
The length of jobs in the test

The first test was run with a set of 698 simulated, relatively short length jobs. The utilisation was moderately high 93% with pronounced spikes in the job load.

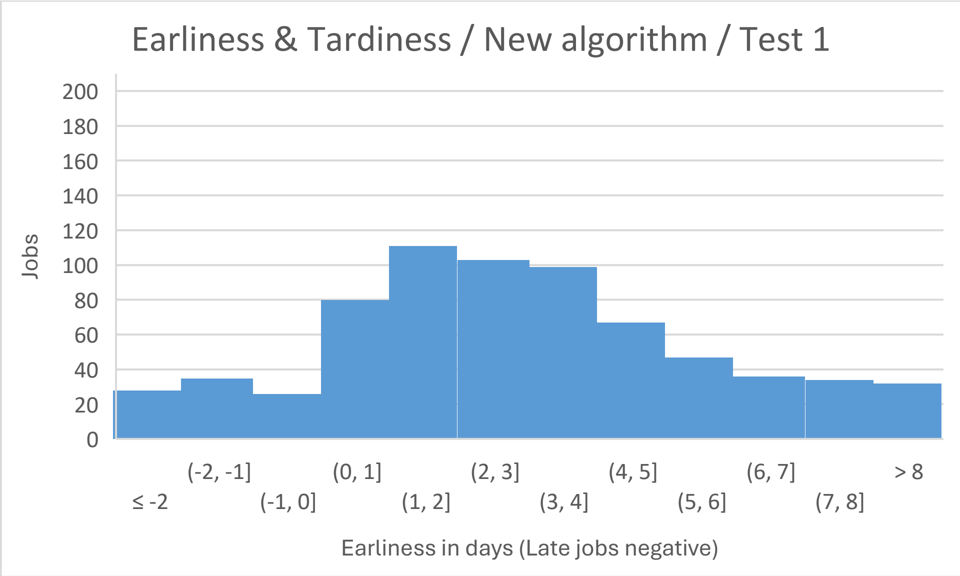
Algorithm	Total setup time	Number of set-ups	Setup time / Processing time	Jobs late
EDD	289.7 Hours	515	13.2 %	479
New algorithm	66.5 Hours	121	3.0 %	89
EDD + Setup	114.3 Hours	216	5.2 %	111

The results of test 1

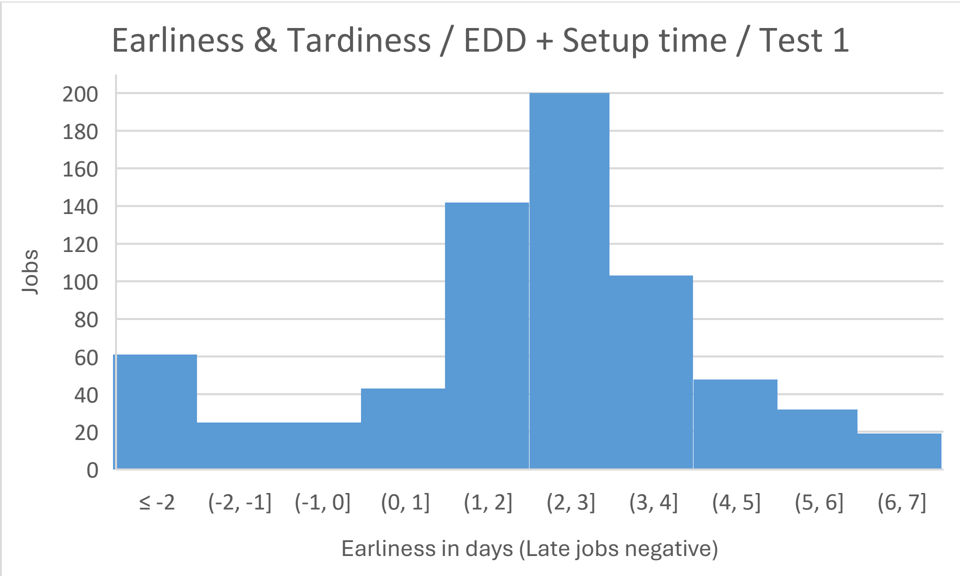
The outcome of the measures in the first test indicates that the new algorithm does perform better than the other algorithms in reducing the number of setups, which directly also reduces the number of late jobs in this case problem, as more time spent on setups in this test case leads to higher capacity overloads and lower effective throughput. There is however a drawback to the new algorithm identifiable from the analysis of the earliness and tardiness of the scheduled jobs. The new algorithm increases the service level through a notable increase of early job finishing. The benefit of 22 less late jobs is likely offset by the extra finished inventory. It is worth noting that in this scenario the algorithms are not allowed to insert pauses, so algorithms with greater utilisation are forced to create finished goods inventory. However the EDD + setup algorithm essentially paces itself, since when the due dates are larger, the setup times are smaller in comparison.



Earliness and tardiness of finished jobs for the EDD rule



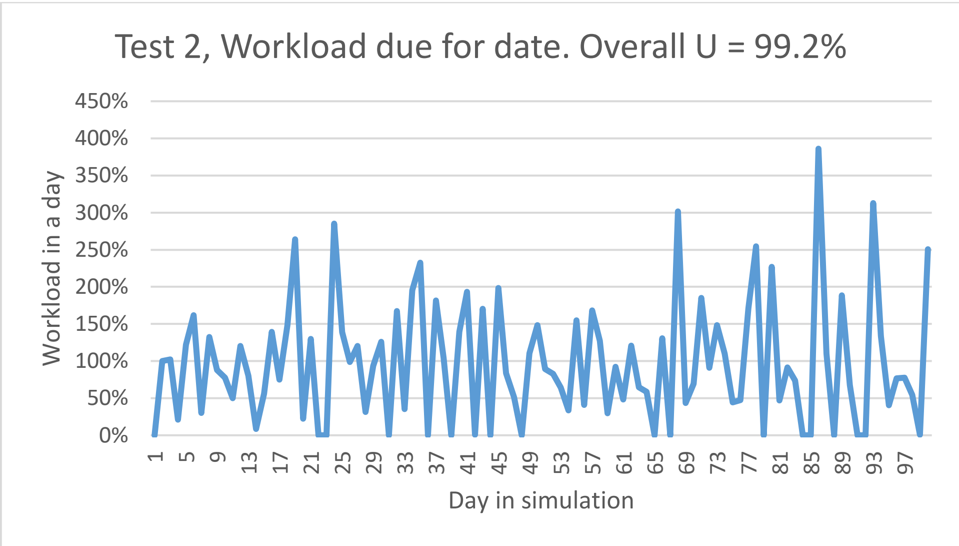
Earliness and tardiness for the new heuristic algorithm



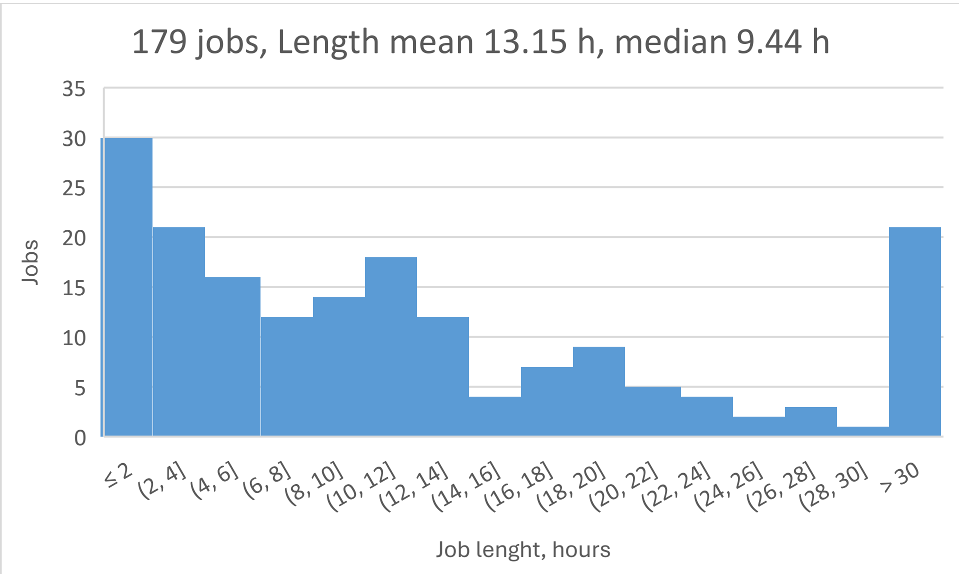
Earliness and tardiness for the EDD + setup time heuristic

Test 2

A second test was run with longer jobs and a very high utilisation level of 99.2%. At this utilisation level the algorithms were overwhelmed and the output for each was roughly similar. The EDD + setup time rule was slightly better than its competitors. In addition, the proportion of the setup times to the processing time made the outcomes of the algorithms not have significant difference, since the jobs were long while the setup times remained the same.



Workload for each day of the test simulation



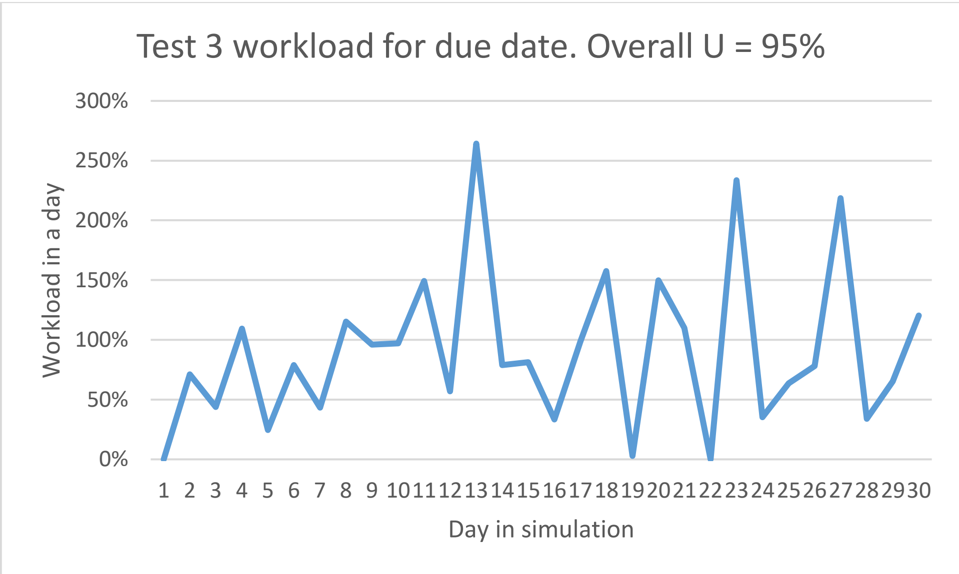
The length of jobs in the test

Algorithm	Total setup time	Number of set-ups	Setup time / Processing time	Jobs late
EDD	78.1 Hours	137	3.3 %	86
New algorithm	74.9 Hours	129	3.2 %	84
EDD + Setup	62.3 Hours	109	2.6 %	69

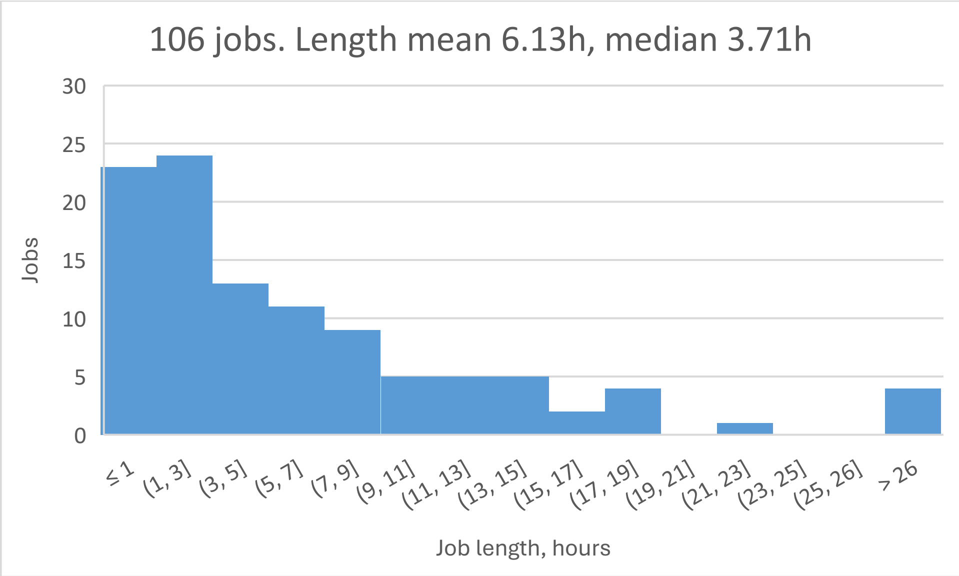
The results of test 2

Test 3

The final test was run with a shorter timeframe of 30 days and a high overall utilisation level of 95%. Pronounced spikes in the daily utilisation levels were present in the job data.



Workload for each day of the test simulation

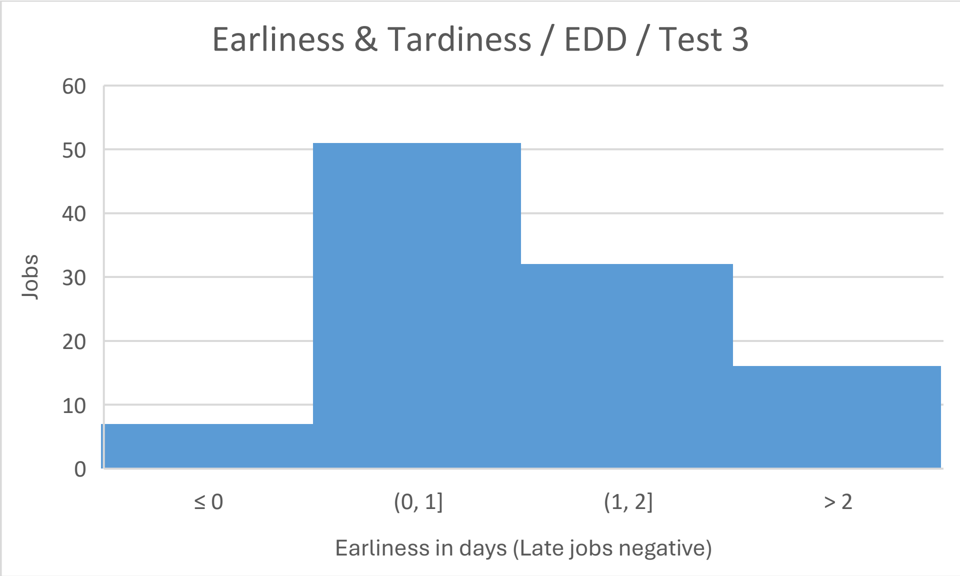


The length of jobs in the test

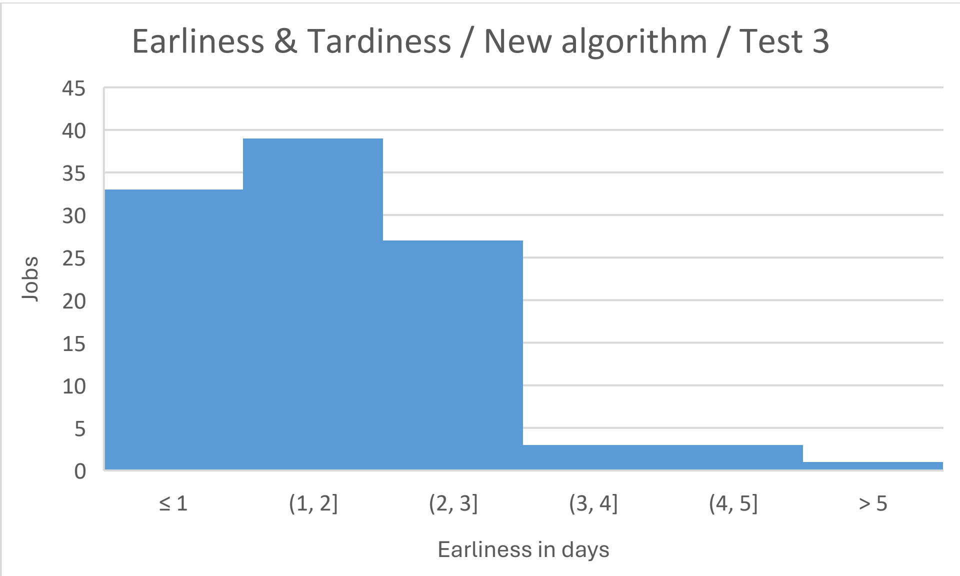
Algorithm	Total setup time	Number of set-ups	Setup time / Processing time	Jobs late
EDD	41.9 Hours	77	6.4 %	7
New algorithm	23.4 Hours	40	3.6 %	0
EDD + Setup	30.7 Hours	53	4.7 %	1

The results of test 3

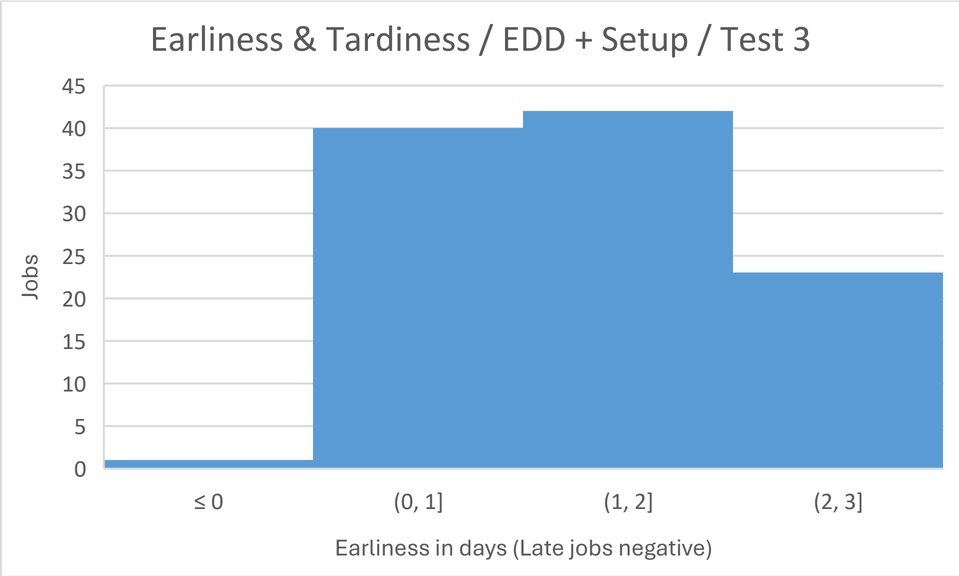
In this test the new algorithm had a slightly better outcome than the EDD + setup algorithm, and the EDD algorithm had unnecessary late jobs. In this case the new algorithm did not create as much of a noticeable increase in finished goods storage when compared to the simple EDD + Setup algorithm.



Earliness and tardiness for the EDD rule



Earliness and tardiness for the new heuristic algorithm



Earliness and tardiness of the EDD + setup heuristic algorithm

Conclusions from the testing of algorithms for CM27

While the new algorithm specifically made for this case did have slightly better results in terms of late jobs and setup time minimisation in this limited testing, it is not recommended. The EDD + Setup time combination dispatching rule appears to be as performant and is magnitudes easier to implement. Considering how the CM27 interface is the low-investment approach to production scheduling implementation in the case, this simplicity is a benefit. Additionally, the combination of EDD and setup times is a common approach that is mentioned often in the literature, so there is further available information on it.

Notable also is that if the setup times are low in relation to the length of the jobs, then the EDD is a recommended choice. It can be implemented directly in the system without modification, and as test number 2 indicates, there is little benefit to considering setup times when setup times do not make much of a difference.

A heuristic algorithm not tested here but potentially interesting when setup times are not relevant is EDD + priority. The priority of orders should not be considered as a first order dispatching factor, as there is no point in producing high priority orders several days early. The priority parameter is only interesting when the production must make compromises. For this use, an EDD + priority algorithm is proposed. It uses the same functional loop as the EDD + setup algorithm, but the setup element is exchanged with a prioritisation element. The value of this dispatch rule, lowest value dispatched first, can be calculated with the formula below:

The EDD + Priority combination rule

$d_j - t_d + \min(\,0, (d_j - t_d - t_b) * \,p\,)$, where

d_j = Due time of the job	t_d = Finishing time of the last dispatched job	t_b = Buffer time	p = Priority number, higher number for higher priority

The buffer time value is used to determine the buffer time, under which the prioritisation element activates. When time until the due date is greater, then the rule simplifies to the EDD rule. When the time until the due date is less than the buffer time, continuously greater prioritisation of the job is performed according to the nearness or lateness of the job. The formula contains a minimum function that selects the output of the priority component or 0, whichever is smaller.