University of Twente

Master Industrial Engineering and Management

An optimisation approach to schedule CNC machines with pallet handling system

Niels van Boxel

March 2024

Master thesis Industrial Engineering and Management

Production and logistics management specialisation

Faculty of Behavioural, Management and Social sciences

University of Twente

An optimisation approach to schedule CNC machines with pallet handling system

Author
Niels van Boxel
MSc Industrial Engineering & Management
March, 2024

HTM Aerotec
Topaasstraat 24
7554 TH Hengelo
The Netherlands

University of Twente
Drienerlolaan 5
7522 NB Enschede
The Netherlands

Supervisor HTM Aerotec
MSc. A.K.H. Kasteel (Arthur)

Supervisors University of Twente
Dr. E. Topan (Engin)
Dr.ir. E.A. Lalla (Eduardo)

Preface

Dear reader,

In front of you lies the Master thesis 'An optimisation approach to schedule CNC machines with pallet handling system'. This research has been performed at HTM Aerotec as the final assignment for the Master Industrial Engineering and Management with a specialization in Production and Logistics management at the University of Twente.

At HTM Aerotec, I learned a lot, got lots of insights and again experienced what it is like to operate within a company. I am grateful for the opportunity that I got and the interesting assignment that came along with it. The scheduling topic was new for me, however I learned a lot about the subject and the practical constraints that make it hard to turn theory into reality. I would like to thank all employees at HTM Aerotec that helped me with my assignment, even though some were not entirely sure about how the thesis would turn out. I was educated on the regional culture, which you do not really experience in the 'University of Twente bubble', and had a lot of fun during the breaks. On top of that, the information about the processes and their thoughts about it really enhanced the understanding of the production process. A special thanks goes out to my company supervisor, Arthur Kasteel, who guided me during the research. Arthur came up with the assignment and gave me the opportunity to work on it at the company. I was given a lot of freedom to put my own spin on the assignment and work on it independently. The meetings were very helpful and because of the similar study background we were able to have detailed and in-depth discussions.

Furthermore, I would like to thank my UT supervisor Engin Topan. He was my first supervisor for my Bachelor assignment, which was a pleasant experience. Because of this, I reached out to Engin for a Master assignment, which is how I ended up at HTM Aerotec. I enjoyed the meetings in which I got good ideas for the assignment and where we also talked about other topics than the research. I would also like to thank my second supervisor Eduardo Lalla for his insights and feedback in the later stages of the research.

This master thesis marks the end of my student life and the end of my time living in Enschede. I was a bit hesitant at first about going to Enschede, but I really enjoyed my five and a half years of studying in Enschede. It has been a very eventful period, with endless fun activities, cool experiences and personal development. At last, I would like to my family, friends, girlfriend and roommates for their support and interest during the course of my study and master thesis.

I hope you enjoy reading this thesis!

Niels van Boxel

Hengelo, February 2024

Management summary

The production facility of HTM Aerotec produces high precision parts for the aerospace- and defence industry. HTM Aerotec has grown over the years, however, the production processes of the 5-axis milling machines with pallet handling system struggle to keep up with the entire supply chain's throughput. Most of the orders at HTM are produced on these machines. Over the last year, the machines operated for an average of 105 hours per week. In theory, these machines should be able to run 24/7 as they can run unmanned due to their pallet handling system. Currently, there is a lack of a sophisticated planning approach for the 5-axis milling machines. This leads to being unable to consistently produce more than 105 hours per week and gives a lack of insights into what factors contribute to good machine utilisation. Therefore, the main research question is stated as follows:

"How can an optimised scheduling strategy be developed to consistently achieve the goal of reaching more than 105 production hours per week for the 5-axis milling machines?"

A context analysis identified the scheduling problem. We found that the performance of the machines is very variable, but averages out to around 105 hours of production per week on average over the last year for both machines. The machines do not have a program roughly 25%. of the time, meaning that there is room for improvement. The machines can switch between machining tasks with ease because they have an automated pallet handling system and a pallet storage area. The loading and unloading of the pallets with products requires operator intervention. Operators are available for 40 hours per week with additional short visits during the weekend. Several constraints further complicate the scheduling process. The two machines studied have a limit on pallet availability, having 40 and 24 pallets available for machines 538 and 539 respectively. Before products of an order can be produced the setup for this order needs to be performed. Besides, certain orders necessitate dedicated fixtures, which are limited and some are not compatible with all pallets. For each order, the products have different characteristics. Products of orders differ in constraints such as the number of products that fit on a pallet, processing time (manned or unmanned), number of production steps on the machine, and release dates. All these factors make it hard for the operator to schedule the products such that production hours are maximised. The scheduling problem consists of deciding when to perform setup for an order, when to load/unload the pallets, sequencing of pallets in the machine, and which pallets to load with what product. The aim is to make a schedule which ensures a high utilisation level of the machines, whilst keeping tardiness low.

The literature review classified the problem as a single-machine scheduling problem with family setups, multi-fixturing pallets, (periodical) resource constraints with product-specific characteristics and constraints. Based on related scheduling research, we identified solution approaches. The problem can be solved in an exact way, which is feasible for small problem instances, or heuristics can be used. With the heuristics, first an initial schedule is generated, which is iteratively improved by the improvement operators in an improvement heuristics with a chosen neighbourhood structure.

The scheduling problem has been modelled in a mathematical model formulation to increase problem understanding. The scheduling problem is not solved exactly due to its NP-hardness and the industry size problems that are used as input. Three different dispatching rules have been deployed for generating the initial solution, being EDD, Multi-Factor, and Random. Two improvement heuristics have been implemented, namely Simulated Annealing and Tabu Search. Besides two neighbourhood structures are used (random and variable neighbourhood). This results in a total of 12 different solution approaches that we test for our scheduling problem.

The best solution approach is to use EDD as the dispatching rule for generating the initial solution and improve the initial solution using TS with random operation selection. The EDD dispatching rule gives the best trade-off between makespan and tardiness. Due to the complexity of the scheduling problem, the algorithm is not capable of overcoming the worse solutions of the other initial solutions generated by the other dispatching rules in a reasonable time span. A sensitivity analysis has been performed on various factors. Operator availability is studied to assist in determining the feasibility and impact of making changes to operator shift times. The currently installed number of pallets is studied to determine whether the number of pallets for the machines is a bottleneck for achieving more production hours. Similarly, it is determined whether the number of dedicated fixtures is a bottleneck, or perhaps it would be beneficial to make some extra. From the sensitivity analysis, the following can be concluded:

- Visiting twice in the weekend is necessary, if this is not possible, visiting on Sunday is slightly preferred.

The system performance already drops by roughly 10% for machine 538 and roughly 30% for machine 539 if an operator only visits one day in the weekend.

- Visiting at a later time in the weekend improves the system performance significantly. Visiting at 16.00-17.30 instead of 10.00-11.30 improved the system performance of the 538 by roughly three percent and the 539 by roughly six percent.
- Reducing or increasing the operator availability hours during the week significantly impacts system performance. This makes it undesirable to work fewer hours given the current situation.
- Reducing the number of pallets gives a bigger decrease in objective value than increasing the number of pallets increases the objective value. However, having four more pallets for both of the machine would have increased the system performance already by five percent.
- The effect of dedicated fixtures availability depends on the mix of products that is produced on the machine. In this case, increasing the availability of dedicated fixtures for the 538 has much less impact than for the 539. This has to do with the fact that the 539 produces more products for which only a limited amount of fixtures is available.
- Increasing robustness of the schedule significantly reduces system performances, however increases the probability of system feasibility. Ten percent schedule robustness increases the objective value by roughly 10 and 15% respectively for the 538 and 539. When increasing the robustness to 20%, the objective value increases by 20% and 40% for the machines.

At last, we compared the algorithmic performance of a schedule of one full month with observed real-world performance in a month. Machine status over a month was tracked, from which we can determine the number of production hours reached. This approach was adopted to ensure a more fair comparison of performance. At the start of the month analysed, it was denoted which products need to be produced in the next months. In this way, the algorithm had access to similar products as the operators. By selecting a set of orders for the algorithm that closely resembles those produced during the observed month, we enhance the value of the performance comparison. The real-world performance was highly above average, compared to performance over the last years. The algorithmic performance still managed to provide better results than the observed real-world performance. The algorithm managed to achieve 7.01% more production hours for machine 538 (138.19 versus 129.14) and 5.71% more production hours for machine 539 (149.68 versus 141.60) in the base case. Table 0.1 shows the utilisation levels that were reached per day of the week for both the observed performance and the performance reached by the algorithm.

Table 0.1: Machine status per day of the week observed in a month in reality and achieved by the scheduling algorithm for machines 538 & 539

Day	Real-world 538	Algorithm 538	Real-world 539	Algorithm 539
Monday	58.24	61.89	66.38	71.61
Tuesday	76.52	87.97	81.68	90.83
Wednesday	83.12	90.43	88.64	92.65
Thursday	81.90	93.33	89.20	94.11
Friday	75.78	96.41	90.83	91.41
Saturday	76.58	93.66	83.98	95.99
Sunday	69.93	48.80	72.25	81.25

The main difference in performance is the number of hours reached on Saturday for both machines. The algorithm for machine 538 managed to reach 17% more hours on Saturday. Besides, the algorithm gets a very high utilisation level during the weekdays. The main limitation for the algorithm for machine 538 is currently the number of hours it can reach on Sunday and therefore also going into Monday. The current weekend visit length is too short for maintaining high utilisation levels.

Running the model has been made accessible for company employees by generating an executable file. An executable file has been made, with which the employees for the company can run the model, based on the input they provide in the accompanying Excel file. A tutorial has been made for the company on how to set the model up and retrieve the output. The output of the model is converted to provide usable information. Insights can be obtained on a higher level, such as when to start producing the first product of an order, the expected lead time of the order, and expected tardiness. More detailed insights can be extracted from the model output, including the sequencing of products for production on the machine, the ideal timing for loading each product, and the pallet allocation for product placement.

Based on the conducted research, several recommendations and directions for further research have been identified. The following are most noteworthy:

- Operator availability heavily constrains the performance of the machines, therefore it is not recommended to shorten operator shifts or reduce the number of weekend visits.
- Have operators visit later on the day when coming in at the weekend as this balances the duration of operator non-availability better.
- Use the model to gain more insights into the feasibility of due dates on the schedule list, lead time of orders, and to identify the bottleneck orders.
- Improve the detail of the schedule list and current machine status.
- Perform research on how this model can be extended to include dynamic rescheduling. Including dynamic rescheduling can make the model operationally deployable. If the operators use the schedule that the algorithm generates, it will be possible to evaluate the model's practical usefulness in achieving more production hours. For this to work, live machine status needs to be linked to the scheduling model.
- Expand the capabilities of the model to identify optimal visit times for operators, both during the week and at the weekend.

Contents

Preface i

Management summary ii

1 Introduction 1

1.1 Company description . . . 1

1.2 Problem identification . . . 2

1.2.1 Action problem . . . 2

1.2.2 Problem cluster and selection of core problem . . . 3

1.3 Research plan . . . 5

1.3.1 Research sub questions . . . 5

1.3.2 Scope . . . 6

2 Context analysis 7

2.1 Internal supply chain . . . 7

2.2 The 5-axis milling machines . . . 8

2.2.1 Machines capabilities and functionality . . . 8

2.2.2 Performance analysis . . . 9

2.3 Production planning . . . 11

2.3.1 Production planner . . . 11

2.3.2 Production department . . . 12

2.4 Resource and production constraints . . . 12

2.4.1 Resource constraints . . . 12

2.4.2 Product-specific factors . . . 13

2.5 Conclusion . . . 14

3 Literature review 15

3.1 Production environments . . . 15

3.2 Taxonomy . . . 16

3.2.1 Scheduling . . . 16

3.2.2 Classification of scheduling problem . . . 17

3.2.3 Related scheduling research . . . 18
3.3 Solution approaches . . . 18
3.3.1 Exact . . . 20
3.3.2 Constructive heuristics . . . 20
3.3.3 Improvement heuristics . . . 21
3.3.4 Neighbourhood operators and variable neighbourhoods . . . 23
3.4 Conclusion . . . 24

4 Solution design **25**
4.1 Input data transformation . . . 25
4.2 Assumptions & simplifications . . . 25
4.3 Problem statement . . . 26
4.3.1 Problem description . . . 26
4.3.2 Scheduling sets, parameters & variables . . . 27
4.3.3 Scheduling mathematical model formulation . . . 28
4.4 Constructive heuristic . . . 30
4.4.1 Initial sequence of jobs . . . 30
4.4.2 Constructing a feasible schedule . . . 31
4.5 Improvement heuristics . . . 32
4.5.1 Choosing an initial solution . . . 32
4.5.2 Neighbourhood operators and neighbourhood structure strategy . . . 32
4.5.3 Simulated annealing . . . 33
4.5.4 Tabu search . . . 33
4.6 Solution approach alternatives . . . 34
4.7 Conclusion . . . 35

5 Experimental analysis **36**
5.1 Problem instances . . . 36
5.2 Experimental design . . . 38
5.3 Initialisation and parameter tuning . . . 39
5.3.1 Initial population size . . . 39
5.3.2 Improvement heuristics parameter tuning . . . 40

5.4 Algorithmic results . . . 41
5.4.1 Initial solution heuristics results . . . 41
5.4.2 Improvement heuristics results . . . 42
5.5 Sensitivity analysis . . . 44
5.5.1 Operator availability . . . 44
5.5.2 Pallet capacity . . . 47
5.5.3 Dedicated fixtures available . . . 47
5.5.4 Machine failure . . . 48
5.6 Algorithmic versus observed real-world performance . . . 49
5.7 Conclusion . . . 50

6 Implementation 52
6.1 Required input data . . . 52
6.2 Running the model . . . 52
6.3 Model output . . . 53

7 Conclusions & recommendations 54
7.1 Conclusion . . . 54
7.2 Recommendations . . . 55
7.3 Limitations and future research . . . 55
7.4 Contribution of the research . . . 56

A Overview solution approach 59

B Parameter tuning 59
B.1 Objective function weight tuning . . . 59
B.2 Improvement heuristic parameter tuning . . . 59

Reader's guide

To enhance the reader's understanding and provide navigation through this thesis, we shortly introduce the chapters.

Chapter 1: Introduction

This research is introduced in this chapter. First, a company description is given, followed by an introduction to the problem that the company is facing. The action- and core problem are identified and the research design is provided.

Chapter 2: Context analysis

A context analysis of the scheduling problem at hand at HTM Aerotec is done. The machine functionality is explained and the performance of the machines is given. After, factors that affect planning are identified.

Chapter 3: Literature review

Provides the literature review. A taxonomy of the scheduling problem is done and the problem is classified. Closely related papers are compared based on problem characteristics and solution approaches. At last, possible solution approaches are discussed.

Chapter 4: Solution design

A more technical in-depth problem description is given. Assumptions and simplifications for modeling the problem are discussed. A mathematical model formulation is provided. Various solution approaches are introduced, showcasing diverse constructive and improvement heuristics.

Chapter 5: Results analysis

The experimental design of the research is given to analyse the performance of the model. The best solution approach per problem instance is selected. Problem instances for the machines are first solved under normal circumstances. After, sensitivity analysis with input parameters is done to assess their impact on the system's performance. At last, the algorithmic performance is compared with the real-world performance over a month.

Chapter 6: Implementation

The implementation steps needed and how this model can be used at HTM Aerotec are discussed. The need for implementation is addressed, followed by how the outcome of the model can be evaluated.

Chapter 7: Conclusion & recommendations

Summarises the findings of the research conducted at HTM Aerotec. The main research question is answered and recommendations to the company are given. Limitations, points for future research, and the contribution of the research are discussed.

List of acronyms

Acronym	Description
FAI	First Article Inspection
ERP	Enterprise Resource Planning
MTO	Make-To-Order
CTP	Capable To Promise
CPO	Computer Planned Order
CNC	Computer Numerical Control
FMS	Flexible Manufacturing System
CODP	Customer Order Decoupling Point
FMC	Flexible Machining Cell
SA	Simulated Annealing
TS	Tabu Search
VNS	Variable Neighbourhood Search
VN	Variable Neighbourhood
EDD	Earliest Due Date
MIP	Mixed Integer Programming
MILP	Mixed Integer Linear Programming
FF	First Fit
BF	Best Fit
RF	Random Fit

1 Introduction

This report describes the Master's graduation assignment performed for the Master Industrial Engineering & Management at the University of Twente. In this chapter, an introduction is provided to the research undertaken at HTM Aerotec. Section 1.1 introduces the reader to the company and provides relevant background information on HTM Aerotec. Next, in Section 1.2 the problem definition is given and the core problem is addressed. Finally, the research design is given in Section 1.3.

1.1 Company description

This thesis is carried out for HTM Aerotec, located in Hengelo (Overijssel). HTM Aerotec is a subsidiary of the HTM Technologies Group (formerly known as PM Group), founded in 1966 (PM-Group, 2022). HTM Technologies specialises in the design, development, and production of high-precision bearings, positioning systems, mechatronic systems, as well as aerospace and military components and modules, serving both civilian and government applications. HTM Technologies consists of five sister companies including HTM Aerotec, HTM Precision Sheet Metal, HTM Industrial Partner, HTM Precision and recently HTM UMI. See Figure 1.1 for the geographical overview of the HTM companies.

Figure 1.1: Geographical overview of the five HTM companies

The HTM Aerotec facility is specialised in advanced aerospace and defence module integration. HTM Aerotec is a first-tier world class supplier in high precision parts and mechatronical integrated modules. HTM Aerotec delivers unique products, technical assets with exacting standards, and detailed administration and documentation (proven with the AS 9100D Standard). The parts that HTM Aerotec produces vary heavily due to specific requirements set by customers. The parts manufactured vary in size, ranging from just a few millimeters to large fuel tanks designed for massive aircrafts. Figure 1.2 shows three examples of parts that are produced within the facility.

Figure 1.2: Examples of parts produced by HTM Aerotec

1.2 Problem identification

HTM Aerotec has experienced significant growth over the past decade. However, within the current operational landscape, specific machines in the production process are operating at maximum capacity, some even planned to full capacity for more than a year. This bottleneck presents a significant challenge to HTM Aerotec's potential for further growth.

1.2.1 Action problem

Anything or any situation that is not how you want it to be is an action problem (Heerkens and van Winden, 2017). Significant investments were made in 5-axis unmanned milling machines a few years ago. This resulted in improved performance for HTM Aerotec. Subsequently, newly emerging bottlenecks within the internal chain were systematically resolved, leading to increased throughput. However, the production processes of the 5-axis milling machines struggle to keep up with the entire internal supply chain's throughput. The daily experience is that the scheduling of the 5-axis machines constantly needs to be adjusted because the requested capacity to meet delivery deadlines cannot be achieved. HTM aims to further grow in throughput and business results in the future. The majority of orders at HTM are produced on the 5-axis machines. In Figure 1.3 the number of production hours per week of two of the 5-axis machines over the years are shown.

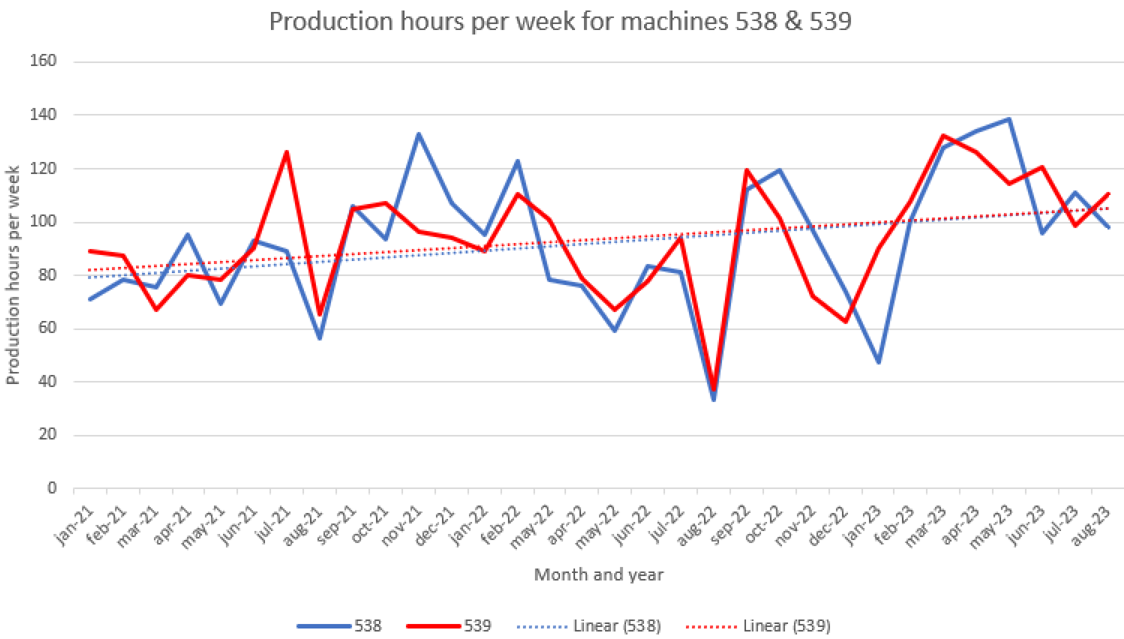

Figure 1.3: Production hours per week over last 2.5 years for machines 538 & 539

Both machines operate for around 95 hours per week on average over the last 2.5 years. However, for both machines an almost identical rising trend line can be seen. What stands out as well, is the large variability of average production hours. In some weeks over 130 hours per week is reached, whilst in others they barely get 70. Over the last year, both machines operated for an average of 105 hours per week, despite there being 168 hours available in a week. In theory, the machines should be able to run 24/7 as they can do "manned" and "unmanned" operations if the ratio between available operations allows for it. "Manned" operations refer to the loading and unloading of pallets that operators have to do at the machines. The machines are capable of producing "unmanned" by using pallet change systems. A pallet change system moves a pallet with a product on it to the machine and can remove it once it is done. This way no human intervention is needed if the pallets are loaded with products. The machines are currently scheduled to full capacity up to one year in advance. This can be done since roughly 80% of the orders at HTM Aerotec are forecast-driven or known well in advance. Time is essence on these 5-axis machines, given that the machines already operate at the current full capacity 24/7. The action problem at hand is therefore defined as:

"HTM Aerotec is currently unable to consistently utilize more than 105 hours per week on the 5-axis milling machines, despite there being 168 hours available in a week"

1.2.2 Problem cluster and selection of core problem

Several factors contribute to the inability to consistently produce an average of more than 105 hours per week. These problems are identified and Figure 1.4 shows the causal relationships towards the observed problem in a problem cluster. A problem cluster is used to map all problems along with their connections. It serves to bring order to the problem context and helps to identify the core problem (Heerkens and van Winden, 2017). A more detailed quantitative analysis of these factors is provided in Chapter 2 within the context analysis.

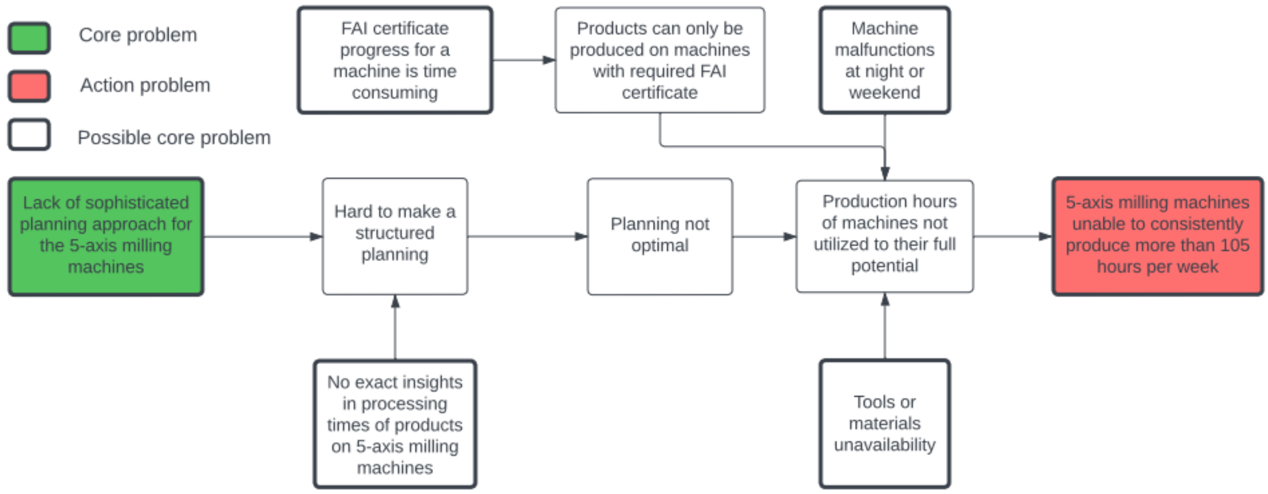

Figure 1.4: Problem cluster

As can be seen in the problem cluster in Figure 1.4, some problems do not have a direct (known) cause themselves and can be considered core problems:

- **FAI certificate progress for a machine is time consuming**: In the industry, HTM Aerotec is in, a First Article Inspection (FAI) is a requirement for most customers. This is needed to verify that a new or modified production step is meeting the specifications detailed in the drawings. Usually, a product has only one FAI and can only be produced on the set of machines that have been indicated in this FAI. However, when the 5-axis milling machine that is in the FAI for a product is fully planned, a partial FAI is needed to be able to produce it on another 5-axis milling machine. This partial FAI certificate progress is quite time consuming and can leave some capacity of the machine unused.
- **Machine malfunctions at night or weekend**: The machines sometimes malfunction at night or during the weekend interrupting unmanned operations and causing significant production time losses. The production has no employees present during these times and the machines give no notifications about the malfunctions. This means that if an employee loads the machine at 10 am on Saturday and the machine malfunctions at 10:10 am, no one will know until an employee comes back on Sunday at 10 am to load all the machines again.
- **Waiting on tools or materials**: The tooling centre indicates that tooling and material unavailability are affecting the number of production hours on the machines. The inventory of tool holders is limited due

to their high cost. The tooling centre indicates that this affects which orders can be put on the machine. From interviews with the machine operators and the production planner, we conclude that this waiting time is not of significant concern to be a problem. This is due to operators ordering the required tools well in advance at the tooling department and work instructions only arrive at the machine when the required tools are available. Furthermore, the tool magazine allows for loading and unloading while the machine is still running.

- **No exact insights in processing times of products on 5-axis milling machines:** For most production steps, the hours per product are booked. For the 5-axis milling machines, this is not done as accurately, since it is hard to manually keep track of which products were worked on at which time. For each product there is a prediction for the processing time, however, it is unsure how accurate this prediction is. If large quantities of a product have to be produced, a deviation in processing time can have a significant impact on the planning. The machine has detailed log data on all operations done, therefore obtaining more insights into the processing times is achievable.
- **Lack of sophisticated planning approach for the 5-axis milling machines:** The production planner periodically creates a list of orders per machine and their respective due dates. The list of orders consists of a prediction of processing time on the machine and setup times. This list is made based on the Computer Planned Orders (CPO) in their Enterprise Resource Planning (ERP) system. For the planner, it is currently not possible to see which mix of orders can lead to being able to produce more hours "unmanned". With the mix of orders, we mean how orders are processed together and/or in what sequence. The machine operators receive a list per machine and they can decide the order they do the list in themselves, as long as they finish an order before the due date in the list. Because of the complexity of this scheduling problem (a large number of variables and constraints involved), it is expected that the scheduling decisions are not perfect, leading to sub-optimal planning.

As can be seen in Figure 1.4, the last core problem, "Lack of sophisticated planning approach for the 5-axis milling machines," was chosen. The current method lacks sophistication in considering maximised unmanned production hours, particularly given the complexity of the scheduling problem. This core problem was prioritised over the others because:

- The "FAI certificate process for a machine is time-consuming" problem primarily affects specific instances rather than the overall scheduling strategy. At the time a FAI certificate process is performed for a product, other products can be produced on the machines.
- "Machine malfunctions at night or weekend" disrupt operations but may be mitigated through maintenance protocols rather than strategic planning adjustments. This is not in the scope of this research, however, could be an interesting topic for future research.
- "Tools or material unavailability" highlights a problem indicated by a single person from the tooling centre. The waiting time caused by this problem is not considered to be of significant concern. Operators order the required tools well in advance, and work instructions only arrive at the machine when the required tools are available.
- The "No exact insights into processing times of products on 5-axis milling machines" challenge, while important, primarily concerns accuracy rather than the strategic approach to scheduling. Additionally, the processing times are generally believed to be accurate on average.

The chosen core problem presents significant potential for enhancing scheduling efficiency and increasing production hours on the 5-axis milling machines, particularly due to its low-cost implementation. The challenge in scheduling the jobs of this order lies in the resource constraints. There is a limited availability of the resources: pallets, fixtures, and availability of the operators. Besides, each job has different characteristics for how long the "manned" operations like product clamping take compared to the "unmanned" operations like the machine run time take. To conclude, the following problem is selected as the core problem of the research:

"Lack of sophisticated planning approach for the 5-axis milling machines"

1.3 Research plan

The goal of the research is to find out what factors contribute to the current 5-axis machine utilisation and how planning can improve this utilisation level. The current way of planning leads to sub-optimal planning due to determining the production order on gut feeling and not using a sophisticated approach. The planner currently does not exactly know what factors contribute to making a good planning that can increase the number of production hours on the 5-axis milling machines. The difficulty in this case lies in the resource constraints and the variety of characteristics of each product that is produced. The main research question is therefore stated as follows:

"How can an optimised scheduling strategy be developed to consistently achieve the goal of more than 105 production hours per week for the 5-axis milling machines?"

1.3.1 Research sub questions

To answer the main research question, research sub-questions are defined:

1. **What is the current situation at HTM Aerotec?**
   (a) *What does the internal supply chain look like and how do the 5-axis milling machines fit in this internal supply chain?*
   (b) *How do the 5-axis milling machines work, what are their limitations and how do they perform?*
   (c) *What does the current production planning procedure for the 5-axis milling machines look like?*
   (d) *What should be taken into account when making the production planning?*

   Chapter 2 answers question 1 and aims to get an insight into the current situation at HTM Aerotec. To answer the main research question first a context analysis needs to be done, where we get an insight into the current production process and how the machines fit in this production process. Then a more in-depth analysis of the limitations and production hours of the machines needs to be done to find room for improvement. Then the current production planning process needs to be analysed to better understand the context. Finally, we need to find out what resource constraints are product-specific constraints are present at HTM Aerotec.

2. **What models are presented in the literature for constructing a production schedule and what optimisation heuristics are available for optimising a production schedule?**
   (a) *What scheduling problems are available in literature and how can we classify the scheduling problem at HTM Aerotec?*
   (b) *What models and methods within the literature are available for constructing a production schedule?*
   (c) *What solution approaches can be used to solve the scheduling problem at HTM Aerotec?*

   A literature study is performed to find relevant models and methods for constructing a production schedule. Models and methods can be combined and modified to fit the situation at HTM Aerotec. Afterwards, optimisation heuristics are studied and a decision needs to be made for a heuristics that fits this situation. An optimisation heuristic is needed as the machine scheduling problem at hand is NP-hard, which Chapter 3 elaborates further upon. The literature study concludes with a summary of the findings and a framework for the rest of the research.

3. **How can the production schedule of the 5-axis milling machines at HTM Aerotec be modelled?**
   (a) *What input data is required?*
   (b) *What are modelling assumptions and simplifications?*
   (c) *What does the mathematical model formulation look like?*
   (d) *How can we construct a feasible schedule and improve it using heuristics?*

   Following the literature study, Chapter 4 delves into the problem formulation, outlining the necessary input data, outlining the model's assumptions and simplifications, and providing insight into the structure and formulation of the mathematical model. Additionally, strategies for constructing an initial feasible schedule and improving it through heuristic methods are explored.

4. **What experiments can be done with the model to investigate the performance?**
   (a) *What problem instances do we use to test the model?*
   (b) *What does the experimental design look like?*
   (c) *How is the model performing and what is the best solution approach?*
   (d) *What is the influence of the model's input parameters on its performance outcomes?*
   (e) *How do the actual production hours achieved by machines in a given month compare to the calculated production hours generated by the scheduling model?*

   In Chapter 5, an in-depth analysis of the model's performance is presented. Problem instances are defined and the experimental design is presented. For each problem instance, the best solution approach is determined. Sensitivity analysis is conducted on the model's parameters to assess their impact on performance. These experiments aim to provide valuable insights to HTM Aerotec regarding the machine environment. At last, a comparison of the model's performance against the actual performance of the machine is done.

5. **How can HTM Aerotec use the model?**
   (a) *What data sources and input is required?*
   (b) *How can the new method be implemented at HTM Aerotec?*

   This chapter examines how HTM Aerotec can use the model in practice. The required data sources and input are addressed. The chapter also explores how the new approach can be implemented at HTM Aerotec, providing how to model can be used easily by the employees.

6. **What conclusions and recommendations can be made to HTM Aerotec?**
   (a) *What are the main conclusions?*
   (b) *What are the main recommendations?*
   (c) *What are the limitations of the research and what needs further research?*
   (d) *What are the practical and theoretical contributions?*

   To conclude the research, conclusions based on the research are drawn and recommendations are made to HTM Aerotec in Chapter 7. The limitations of the research are discussed and directions for future research are given. At last, the theoretical and practical contribution of the research is addressed,

1.3.2 Scope

The research will be limited to the 5-axis milling machines. The company has a lot more machines, but focusing on these machines has the most impact on the throughput of the production.

Furthermore, it should be noted that only the 5-axis milling machines with a pallet handling system are considered for this research. This selective approach is driven by the fact that the machines without a pallet handling system are not capable of running "unmanned". For the 5-axis milling machine without a pallet handling system, the improvement potential is much lower. They can only produce when an employee is present, meaning that a human can intervene when something goes wrong. In this way, not a lot of time is wasted. Besides, more accurate data is available on two of the 5-axis milling machines, namely the 538 and the 539, so we only take these two into account.

The processes that happen before or after producing on the 5-axis milling machines are not in the scope of the research.

2 Context analysis

This chapter provides a context analysis about the current situation at HTM Aerotec. The first research sub question is answered in this chapter:

What is the current situation at HTM Aerotec?

For the rest of the report, machines refer to the 5-axis milling machines unless explicitly mentioned. Section 2.1 explains the internal supply chain and the role of the machines in this internal supply chain. Next, Section 2.2 explains how the machines work, what their limiting factors are and the performance of the machines in the past. Section 2.3 describes the current production planning process in more detail. Lastly, Section 2.4 presents the factors that influence the planning procedure.

2.1 Internal supply chain

As Section 1.1 described, HTM Aerotec produces parts that vary heavily based on specific requirements set by customers. Customers send product design drawings of the product that they wish to have manufactured, after which HTM Aerotec starts producing. A policy where firms start working only after an order has been placed is referred to as make-to-order (MTO) (Chen et al., 2009). Every order follows roughly the same internal supply chain within HTM Aerotec, starting with a customer request and ending with the shipment to the customer. The most important customers place orders for up to three years ahead. Figure 2.1 shows a simplified version of the internal supply chain.

Figure 2.1: Simplified version of internal supply chain

This research focuses on the production planning and production phases, with an emphasis on the 5-axis milling machines. The production planning process is discussed in further detail in Section 2.3. The production phase itself consists of multiple phases, there are over 30 potential production steps for each product. Since there is a large variety of products, the individual production routing for each product is different. Figure 2.2 shows the production routes of two different products. These two are chosen to highlight the variety the production routing can have.

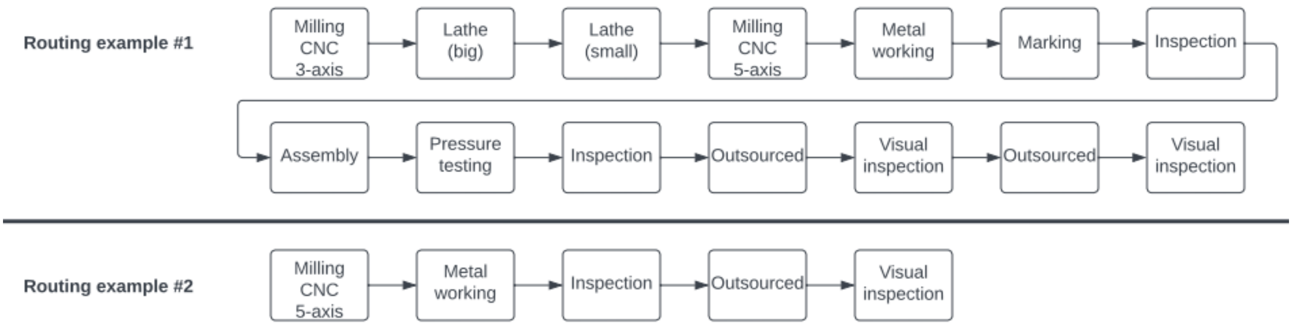

Figure 2.2: Examples of production routing

For new orders, the employees of work preparation (production engineers) determine the routing. This routing is based on the requirements set by the customer. The main driver for deciding the routing is the experience of the production engineer. Based on instances where products were rejected or accepted in the past, the production engineer is capable of determining the required route. There is no systematic approach for determining the routing of the products. For the first product of a new order, an FAI is done to ensure that the first manufactured part meets the design specifications and regulatory requirements. In the FAI process, the product route and the corresponding machines are documented.

A significant part of the orders HTM gets are recurring. The routes for recurring orders are already documented in the FAIs of these products. A partial FAI is required if there is a mutation in the manufacturing process, such as the routing, machines used, tooling, materials, etc. A product can have multiple FAI certificates and a FAI certificate expires if the manufacturing process documented in the FAI for the product is not used for more than two years.

2.2 The 5-axis milling machines

HTM Aerotec has five 5-axis milling machines available. As Table 2.1 shows, four of these machines are capable of running unmanned production, as they have an automated pallet handling system. In this subsection, we first go over the machines and describe how they work, what they can do, and what their practical limitations are. Next, a performance analysis is done on two of the machines (538 & 539). Only machines 538 & 539 have accurate data available, so these machines are the scope of the research.

Table 2.1: 5-axis machines available

Machine ID	Type	Model	Automation possible?
532	CNC Milling 5-axis	Hermle C42	No
533	CNC Milling 5-axis	Hermle C42	Yes, pallet handling system
535	CNC Turn-Mill	GROB G550T	Yes, pallet handling system
538	CNC Milling 5-axis	Hermle C32	Yes, pallet handling system
539	CNC Milling 5-axis	Hermle C32	Yes, pallet handling system

2.2.1 Machines capabilities and functionality

The Hermle machines work similarly. Each machine has a pallet storage area. The handling system can get a pallet with a product on it from the storage and place it in the machine. After the machine is done processing the item, the handling system can get the product from the machine and place it back in the storage room. Machine 533 has a robot arm as a handling system, but works similarly. In this way, the entire setup is capable of producing unmanned as long as there are pallets with products available to work on. Figure 2.3 shows an example layout of the machine setup.

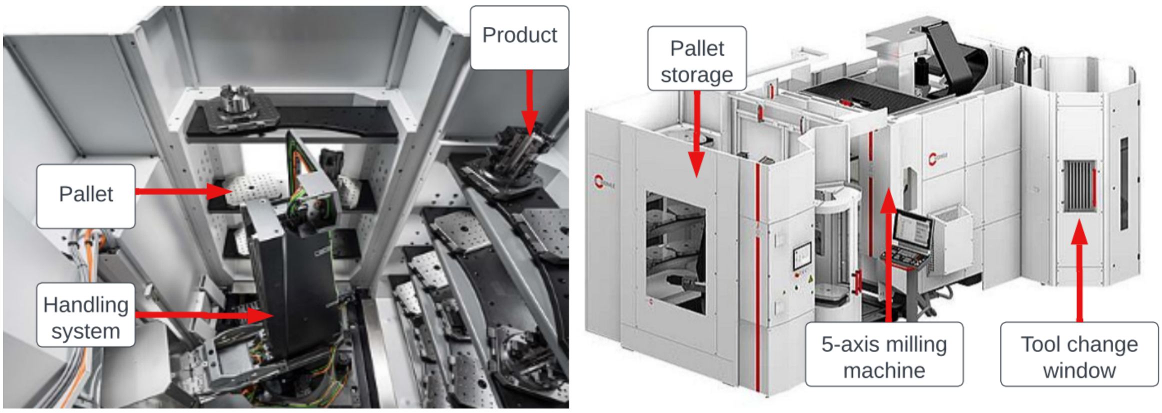

Figure 2.3: Pallet storage (left) and machine layout (right)

Although each Hermle machine works roughly in the same way, the machine specifications differ. Each machine is equipped with different options on for example how many pallets fit in the pallet storage. Table 2.2 shows some of the specifications/limitations for the Hermle machines. The number of pallets that fit in the machine is a big limitation. The 539 has 14 fewer pallets available than the 538. If all pallets in both machines are loaded with 1 product that has a processing time of 15 minutes, the 539 loses 240 minutes of production hours more than the 538. The pallets that are in the 538 are smaller and generally products with a smaller processing time are placed on it.

The GROB machine works a bit differently, compared to the Hermle machines. This is a 5-axis turn-mill machine with a pallet handling system. As the name suggests, mill-turn techniques combine the functionalities of a mill and a lathe. Tool rotation and workpiece rotation are used to machine parts without having to switch

Table 2.2: Machine characteristics

Machine (ID)	Pallets	Tools	Movement X (mm)	Movement Y (mm)	Movement (Z)
Hermle C32 (538)	38	299	650	650	500
Hermle C32 (539)	24	299	650	650	500

machines. This machine has its own set of specifications/limitations, however, this machine will not be part of the research as the data available on this machine is limited.

Both the GROB and Hermle machines are capable of doing operations unmanned and some things need to be done with the help of a machine operator. The unmanned activities refer to the times when a machine is processing a product where no human supervision is required and with the help of the pallet handling system new products can be placed in the machine. Other activities do require human intervention. Clamping the products using fixtures on the pallets and the loading and unloading of the products need to be done by an operator. Tools from the tool magazine need to be switched by the operators. Some products require extra attention when they are on the machine, for example, if it is the first time the product is made or there is an increased probability of failure with a certain product. The information on the predicted amount of manned and unmanned hours that need to be done for each product is available in the ERP system.

If we zoom in on the pallets that are seen in Figure 2.3, we can see what these pallets look like exactly. A pallet is a small flat surface, on which a product and/or fixture can be placed. In Figure 2.4 an empty pallet and a pallet with a product loaded on it can be seen. A product and/or fixture with a product can be assembled onto the pallets due to the holes that are in the pallet. The pallets provide a stable and adaptable surface. They are designed to be easily interchangeable.

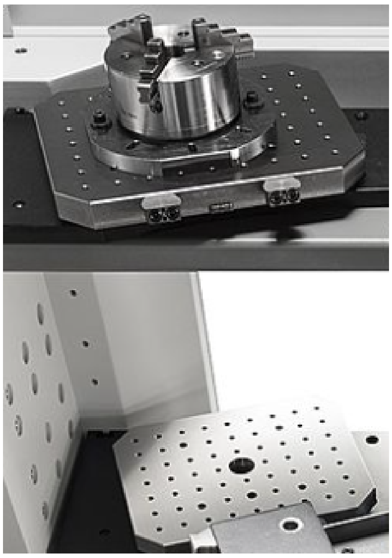

Figure 2.4: Pallet explanation

How the pallets are loaded, differs per product. The operators receive folders with work instructions on how the product(s) need to be placed on the pallets. Each product needs to be clamped on a fixture, which is assembled onto the pallet. Most products require a general fixture, which is readily available. In some cases, products require a dedicated fixture, of which a very limited amount is available. This is due to the fact that these dedicated fixtures can be very expensive, making it generally not worthwhile to make extra dedicated fixtures as only a few dozen of a specific product are made. There is a massive variety of dedicated fixtures, as they are custom-made for a specific product. The amount of products that can be placed on a single pallet also differs per product and fixture type. Besides, a limited number of "towers" are available on which multiple products can be placed. These towers fit on a single pallet.

2.2.2 Performance analysis

In the action problem, we stated that production hours on the machines were on average 105 hours per week, which is lower than desired. Figure 1.3 showed the average production hours per week over the last 2.5 years.

In this section, a more in-depth performance analysis is done to find out more about the lost production hours. In this subsection, we now refer to the percentage of time a machine is doing something. For example, when a machine is producing 105 hours per week, this is 62.5% of the hours in the week. The performance analysis is done on machines 538 and 539 as for these machines accurate data is available.

The status the machines have at each moment is tracked. In this way, the performance of the machines of the last years can be analysed. Table 2.3 shows the percentage of the time a machine is in a certain status. The data on the average status of the machines per month for the last 2.5 years is collected. The table shows the average status over the last 2.5 years and the last year. The column "Active" is the percentage of time the machine is producing a product. The column "No program" is the percentage of time the machine does not have a program of products to produce. This part indicates the improvement potential of the problem. The column "Other" is for things like maintenance, cleaning, etc. The other columns are self-explanatory. In the last year, the performance was better if you compare it to the last 2.5 years. The machines spend more time producing products and less time having no program or "other" activities. There is an increase in emergency stops, which is a direct consequence of more production hours for the machines.

Table 2.3: Machine status (%)

Machine	Active	Pallet change	Interrupted	Emergency stop	No program	Other
Last 2.5 years						
538	54,94 (%)	1,81	4,84	2,08	27,23	9,11
539	55,73	0,71	3,72	1,57	29,85	8,43
Last year						
538	62,28	2,16	4,50	3,40	23,13	4,52
539	61,47	0,64	3,53	2,13	25,91	6,31

As Table 2.3 displays, the machines are currently producing just over 60 percent of the time. Over the last year, both machines did not have a program 25 percent of the time. This means that there is significant room for improvement to be made with better planning. To find out more about this, we dive deeper into the performance of the machines. For Table 2.4, the day-to-day performance of the machines over the last 2 months was analysed. The range of 2 months is chosen because from this period the day-specific data is available. What is interesting is that both machines are significantly less active during the weekend compared to during the week. In the last 2 months, the 538 was producing 73.68% of the time during the week and 35.90% during the weekend. For the 539 this was 72.42% and 60.11% respectively. For the 538 the low performance is due to having 6 Sundays in the last 2 months where zero hours of production were done. For the 539 this was the case on 3 Sundays. Mondays perform worse than other weekdays, due to the operators spending a lot of time unloading the pallets that were loaded for the weekend and loading them again with new products. Besides, the percentage of the time the machines are in the "No program" status is higher than desired. Even during the week, this is around 20% for both machines. This means that almost five hours per day during the working week the machines do not have a program to work on. This can happen when all products on the pallets have been processed and these products do not have long cycle times.

There is even more in-depth data for the machines. The status per hour for the last week (168 hours) is in the machine report. Although the last week is not representative of the average performance of the machines, it gives important insights into how the unmanned production has been throughout the nights and weekend over the last week. It can also help find what the causes of machine downtime are on an operational level. Figure 2.5 shows the average performance of the machines, as well as the difference in performance during the last week and the last weekend. We can see that on average, both machines performed really well in the last week and had one of the best weeks of the year. Machine 538 reached 130 hours of production and 539 got 150(!) hours. For reference, the machines got an average of 93 and 101 hours per week respectively the month before that. This shows that there is a lot of variability in this process. What we can also conclude from the graph is that the moment an employee comes to load the pallets at the weekend is really important. Machine 538 came to a standstill at 11 pm on Friday and someone came at around 3 pm on Saturday. Similarly, machine 538 was out of work before 1 pm on Sunday and was loaded again at 4 pm. At the moment the employees do not have insights into the machine status, which helps choose a moment to load the machines at the weekend. This results in a significant production loss.

Table 2.4: Machine status per day of the week (%)

Day	Active	Pallet change	Interrupted	Emergency stop	No program	Other
			Machine 538			
Monday	65,01 (%)	4,46	0,08	0,00	28,96	1,50
Tuesday	77,45	1,61	2,63	0,00	17,03	1,28
Wednesday	77,41	1,85	5,02	0,08	15,40	0,24
Thursday	69,08	1,44	5,37	0,00	23,54	0,57
Friday	79,47	1,73	5,62	0,00	12,77	0,41
Saturday	50,33	0,99	10,59	0,00	36,36	1,73
Sunday	21,47	1,91	5,21	0,00	71,92	0,01
			Machine 539			
Monday	60,53	0,76	3,34	2,72	32,17	0,48
Tuesday	74,98	0,82	0,14	0,39	23,59	0,08
Wednesday	74,30	0,84	1,48	3,20	20,01	0,17
Thursday	71,98	0,72	1,33	3,37	21,25	1,35
Friday	80,30	0,97	0,60	2,88	14,61	0,64
Saturday	83,88	0,54	0,75	0,00	13,09	1,74
Sunday	36,33	0,21	10,00	4,90	48,48	0,08

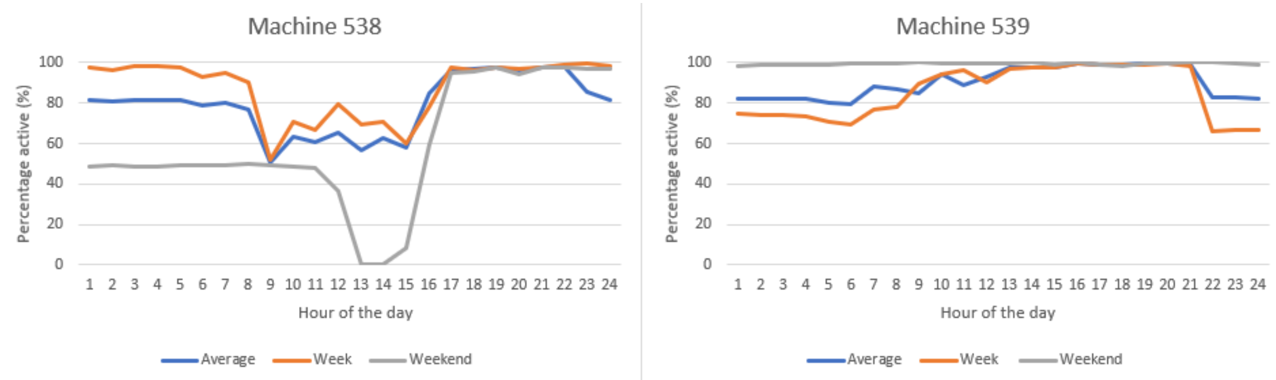

Figure 2.5: Performance of the 538 and 539 over a week

2.3 Production planning

In this subsection, the production planning process is discussed. This is done by first explaining how the production planner makes a more tactical planning and afterwards, the operational planning process of the production department is described.

2.3.1 Production planner

Within HTM-Aerotec the production planner is responsible for scheduling the orders for the production as well as managing and improving the operational planning. The goal is to have a production schedule that guarantees on-time delivery, keeps a high machine utilisation, controls WIP, has acceptable inventory costs and keeps the cash flow into account.

The planner uses data from the ERP system (Glovia G2). Order quantity, routing, required outsourcing, machine labour setup and run times are provided by sales, production engineers and the CAD/CAM department. The ERP system provides a minimum amount of a product that needs to be produced. This is helpful since this reduces total setup time and setup costs. The planner then chooses an amount above that still ensures a good cash flow position. This is not done very precisely, but more roughly based on the current cash flow position. Based on this information and some fixed information like the current expected machine capacities and internal transport times, the production planner can perform a Capable To Promise (CTP) check. This CTP check can be performed using the Glovia G2 add-on Factory Planning. The CTP check is used to calculate an expected delivery date, which sales needs to accept or decline an incoming order.

The planner has access to various planning algorithms within Factory Planning for generating a production schedule; however, these algorithms are not used when creating the schedule lists. Presently, the Factory Planning add-on serves the purpose of providing an estimate of the available machine capacity over time. It functions as an indicator, though it's worth noting that the information derived from the Planning add-on lacks precision. This is attributed to the inability to incorporate certain company-specific details, such as outsourcing, into the system. Therefore, Factory Planning is not used when creating the schedule lists. The planner looks at the Computer Planned Orders (CPOs) and Work Orders (WOs) in the ERP system. The schedule list is arranged chronologically, with orders sorted according to their respective due dates, making sure that the most urgent items are taken care of first. The final schedule is manually fine-tuned using Excel. The finished schedule is then sent to the production department every two to four weeks. Each machine has an individual schedule containing information on the required quantity, due date, etc. Table 2.5 shows a small example of what information is in such a production "schedule" list. The team leaders receive these lists from the production planner. The planner plans with a capacity of 120 hours per week, which includes the setup time. The main responsibility of the planner is to ensure that each machine adheres to the deadlines as indicated in the schedule list.

Table 2.5: Small example of schedule list

Start Date	End Date	WO	Quantity	Item	Open hours	Setup (hrs)	Run time (min)
28-9-2023	29-9-2023	12345	50	123456789	10	2	9,6
5-10-2023	5-10-2023	54321	5	987654321	8	1,5	78

2.3.2 Production department

Once the schedule lists are sent to the production department, it first gets to the production manager. The production manager oversees the entire production process. The production manager can make changes in the order of the schedule list. Rearranging the list and grouping related materials or products helps to reduce setup times and ensure better use of machinery and materials. To maximize unmanned production hours, the production manager occasionally strategically shifts products with long cycle times back a week. In reality, shifting around and changing the orders does not happen continuously. Once every one or two weeks some changes may be made to the order of the schedule list together with operators and planners. The production manager can then choose which orders of the schedule list are given to the machine operators. Work instructions are only given to the machine operators if the required tools and fixtures are available.

The machine operators receive folders with detailed work instructions. The instructions include information on how to position the materials in the machine, which CNC program to use, how it needs to be clamped, etc. Usually, the operators have a few folders with orders to work on at their machines. Assumed is that the tools needed are always available and require minimal changeover time. The operators choose which order they will work on. They try to schedule the orders they have efficiently by having a good mix of products (long cycle times) on the machine at the end of the day and for the weekend. This is hard sometimes when there are priority orders with products that have short cycle times. The operators also base the planning on when they can come back at the weekend to load more products on the pallets.

2.4 Resource and production constraints

In this section, we cover the constraints to be considered. We discuss the resource constraints and product-specific constraints separately to provide a comprehensive understanding of each aspect.

2.4.1 Resource constraints

Resource constraints refer to the limited availability of resources and play a critical role in making a schedule. The resource constraints that are considered in this research are the following:

- **Operator availability** is the first resource constraint that has to be considered in the scheduling of jobs. Operators are available during working hours and usually load the pallets for one hour on both days of the weekend. Because product loading and unloading on the pallets are manned operated, the system must be scheduled such that at the beginning of the night shift all pallets are loaded with parts to be machined. From that moment on production can go on unattended until all these parts have been processed. Therefore, parts are to be processed during a night shift preferably those requiring long processing times. Figure 2.6 shows an example of the hours the operator is available during a normal week. The moment an operator comes in the weekend is not consistent. The aim is somewhere in the middle of the day, however this depends on the operator's plans. They usually stay for 1.5 hours. It is also a possibility that an operator only comes on one of the days or does not visit at the weekend at all.

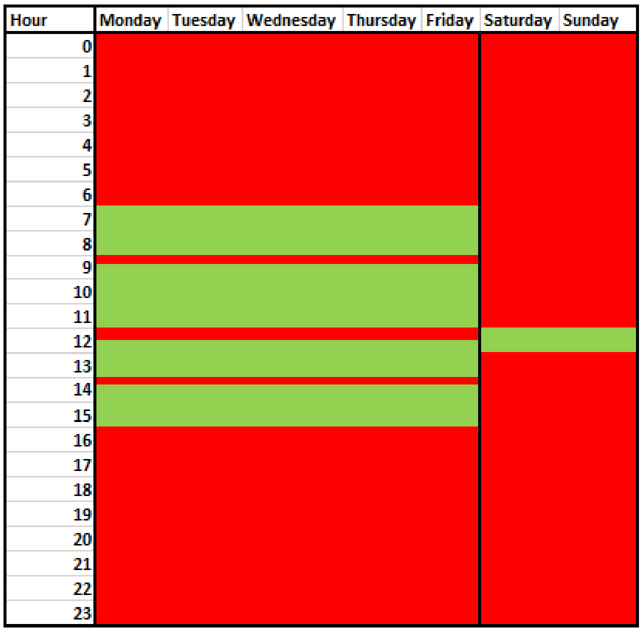

Figure 2.6: Example of operator availability in a normal week

- **Pallets** are available in a limited amount. As Table 2.2 in Section 2.2 shows, machine 538 has 40 pallets available and machine 539 has 24 pallets available.
- **Fixtures** are available in a limited amount. Each product needs to be clamped onto a fixture, which is assembled on the pallets. Some products use general fixtures, however, some products require dedicated fixtures. Since these dedicated fixtures are very expensive and generally are only required for one type of product, there is usually only one fixture of each type available.

2.4.2 Product-specific factors

Other than the resource constraints, the products also have some characteristics and constraints that need to be taken into account when making the schedule:

- **No routing flexibility** is allowed for the products. Each product has a predetermined route and is assigned to be produced on specific machines. This comes from the FAI certificate process. The products can not change to another machine.
- **Dedicated fixtures** are required for some products. The products can only be mounted on a pallet if there is a dedicated fixture available.
- **Multi-fixturing** of products can be done on some pallets. Some towers are available, which allow for the mounting of multiple products on a pallet. And some products are small and multiple of them can be assembled onto fixtures.
- **Processing and setup times** differ per product. Some products require relatively more manned operations than others. These are usually the products with low processing times, but how the products need to be mounted on the pallets can make a difference in setup and labour times as well.

- **Release- and Due dates** are assigned to each work order. All products of the work order need to be finished before the internal due date. Some products can only commence production once the necessary materials have been received or after the completion of the preceding route step. Consequently, the release date for work orders corresponds to the arrival of the required materials.
- **Number of required production steps** on a machine differs per product. For some orders, the products require multiple production steps on the same machine. For example, if a product has 4 production steps on the same machine then for each step, the product needs to be unloaded and positioned differently. It can be positioned on the same pallet, with the same fixture or each operation can require a different dedicated fixture. This is all recorded in the work instructions.

2.5 Conclusion

This chapter offers a comprehensive overview of the current situation at HTM Aerotec and answers the research question: *What is the current situation at HTM Aerotec?* Insights into the internal supply chain, machine performance and capabilities, production planning procedures and resource and production constraints are obtained.

We have identified that each product follows a unique production routing, often documented in the First Article Inspection (FAI) certificate, which remains relatively unchanged for recurring products. The significance of the 5-axis milling machines within the production process was highlighted, with a specific focus on machines 538 and 539.

Moreover, we discussed the limitations and capabilities of these machines, revealing a variable performance that averages to around 105 hours of production per week for both machines over the past year. Notably, 25 percent of the time the machines remain idle, suggesting potential for improvement in utilisation.

Insights into the current production planning process are provided, outlining the procedures followed by the production planner and the subsequent handling by the production department. Finally, we identified key factors that must be considered when developing production plans. This can be split up in resource constraints and product-specific characteristics and constraints.

Moving forward, a literature study needs to be performed to classify and approach the planning problem more effectively, to enhance scheduling efficiency and maximize production output.

3 Literature review

Within this chapter, we provide a literature review. This chapter forms the theoretical framework of the research and answers the second research question:

What models are presented in the literature for constructing a production schedule and what optimisation heuristics are available for optimising a production schedule?

Section 3.1 identifies the production environment of HTM and recognises our problem as a scheduling problem. Section 3.2 provides a taxonomy of the scheduling problem and discusses the solution approaches of closely related papers. Finally, Section 3.3 elaborates on the solution approaches that can be used.

3.1 Production environments

In order to successfully compete, operations in a firm need to be strategically aligned to the market requirements. Companies are incorporating the Customer Order Decoupling Point (CODP) as an important input to the strategic design of manufacturing operations as well as supply chains (Olhager, 2010). The CODP is defined as the point in the value chain of products, where the product is linked to a specific customer order. Different production environments relate to different positions of the CODP. Olhager (2010) make a distinction between four different types of manufacturing environments: make-to-stock, assemble-to-order, make-to-order and engineer-to-order. Make-to-stock is a strategy used to match inventory with anticipated customer demand. Engineer-to-order companies deliver products that are tailored to fit the customer's unique environment. Make-to-order refers to companies that produce bespoke and customized products to particular customer specifications, where production takes place after the customer order has been received (Saniuk and Waszkowski, 2016). Assemble-to-order is seen as a mix of the previous two. Figure 3.1 depicts the CODP for the four different manufacturing environments.

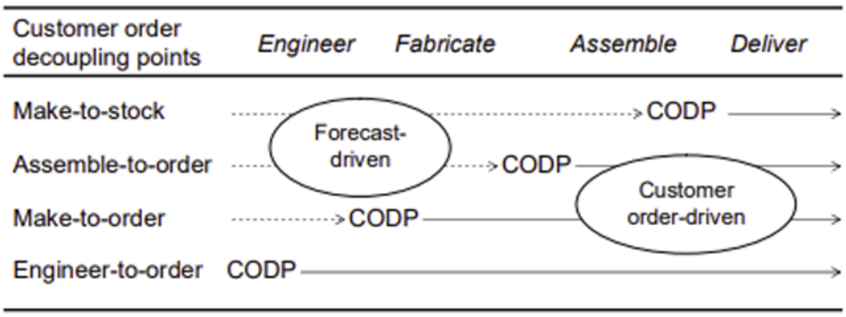

Figure 3.1: Different CODP's (Sharman, 1984)

The production environment at HTM Aerotec can be seen as make-to-order (MTO). The production takes place after a customer order has been placed. Therefore, research in the literature review is mostly on MTO production environments. Figure 3.2 shows a hierarchical framework. The framework distinguishes three processes/levels, namely, the order-selection level, the manufacturing-planning level and the operations-scheduling level.

This research is placed in the offline operational level of resource capacity planning, namely in *scheduling*. The goal of the assignment is to find on an operational level what factors contribute to making a good planning. The planning problem at hand has several resource constraints and product-specific characteristics and constraints. The scheduling problem serves as a feasibility check and can identify in more detail what can lead to more production hours. In the next section, definitions of scheduling are given and a taxonomy of the problem at hand is made.

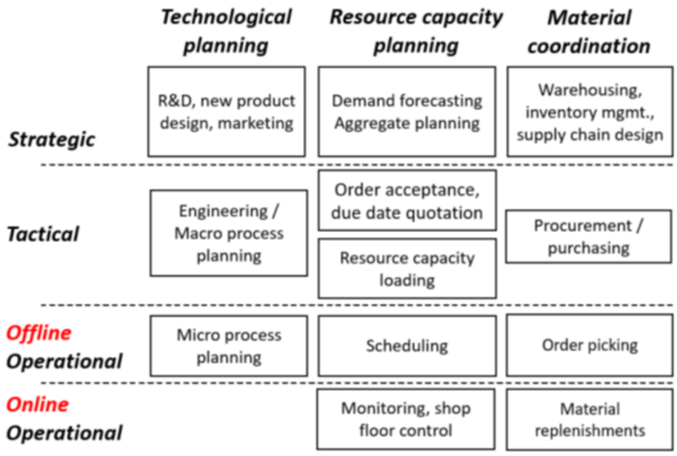

Figure 3.2: Hierarchical framework (Hans et al., 2007)

3.2 Taxonomy

This section provides a taxonomy of the scheduling problem at HTM Aerotec. The taxonomy brings order to the diverse landscape of scheduling research and can help translate our problem into a problem more well-known in the literature. First, some scheduling definitions are provided. Afterwards, the scheduling problem at hand is classified. Furthermore, solution approaches suitable to the scheduling problem are identified from the literature.

3.2.1 Scheduling

Scheduling is a decision-making process that deals with the allocation of resources to tasks over a given time period to optimise on one or more objectives (Pinedo, 2016). In our case, the machines are the resources and the tasks are the products on the pallets that need to be processed. According to Pinedo (2016) and Safarzadeh and Niaki (2019) scheduling problems are classified into some well-known categories regarding machine environment:

Single machine: All jobs need to be produced on a single machine. The single machine environment is simple and provides some basics that apply to more complicated machine environments.

Parallel machines: Involves multiple resources to process a single operation on the jobs. This problem is conventionally organised into three types identical, uniform and unrelated:

- *Identical*: All machines are similar and have the same characteristics.
- Uniform: Each machine has processing speed for all jobs
- Unrelated: No predefined rule for the processing times of the jobs on the machines

Flow shop: Each job has to undergo a series of operations. Often, jobs have the same order. Machines are then assumed to be set up in series and the environment is referred to as a flow shop. *Flexible flow shop* is a variant of the flow shop. This is a combination of flow shop and parallel machines. A job has to be processed at each stage (of the series), any machine will do.

Job shop: In a job shop the routes are fixed, but not necessarily the same for each job. *Flexible job shop* is a generalisation of the job shop and the parallel machine environments. Each stage in the series has a bank of parallel machines and at each stage, a job needs to be produced on one of the machines.

Open shop: Each job needs to be processed on each of the machines. No restrictions regarding the routing through the machine environment.

Other classification categories in which we classify the scheduling problem are the following: objective function, machine constraints and job characteristics. These are discussed in the next subsection.

3.2.2 Classification of scheduling problem

The scheduling problem is classified to obtain a problem more well-known in the literature. Regarding the machine environment, our scheduling problem belongs to the single-machine class. This is because jobs can only be processed on one machine as they are restricted by FAI certificates. We have a similar machine setup as described by Shin et al. (2019), who call it a single-machine Flexible Machining Cell (FMC). The pallet storage and pallet handling system at the machine makes it an FMC and the parts in the FMC can only be produced on a single machine.

Regarding the objective value, the main objective of this research is to increase the production hours per week for the 5-axis milling machines. For a fixed number of jobs (schedule list), the total machining time is fixed and a reduction in the makespan can only increase the utilisation percentage (Baskar and Xavior, 2014). Therefore makespan reduction is the primary objective of this research. The secondary objective of this research is tardiness, which is about the lateness of orders. In principle, the goal is to obtain zero tardiness. However, if this is unavoidable, the tardiness should be minimised.

Regarding the machine characteristics, we have a continuous 24/7 machine availability with fixed capacity and different resource constraints. We have resource constraints on pallets in the pallet storage at the machines. Besides, there are resource constraints on operator availability and moulds/fixtures.

The jobs in our case all have a fixed predetermined processing time. The same holds for the setup time for each production order and the extra labour time required by an operator for each job. A setup for a production order needs to be completed before a product can be processed and this setup can only be done when an operator is available. This is called a family setup. Pallets also need to be loaded and unloaded, which can also only be done when an operator is available. Switching between families (orders) takes more time than switching between jobs of the same family (order), if the setup of the family that is being switched to has not yet been performed (Wemmerlov, 1992). Besides, each order has a due date and some of them have release times. Jobs can only be processed when their required fixture is available. In certain orders, products must undergo multiple production steps using the same machine. If a product entails multiple production steps, each step can only commence once the preceding step has been successfully executed (except for the first step).

Some other elements require some extra explanation. *Production uncertainty* is one of the most important issues regarding scheduling problems (Wojakowski and Warzolek, 2014). In real-world scheduling, it is necessary to find a schedule that is insensitive or robust to production disruptions such as machine failures, absence of workers, etc. Real-time events can be classified into two categories, namely resource-related like machine breakdowns and job-related like job cancellation (Ouelhadj and Petrovic, 2009). The problem of scheduling in the presence of real-time information, named Dynamic Scheduling, is of great importance for the successful implementation of real-world scheduling systems. According to Wojakowski and Warzolek (2014) and Ouelhadj and Petrovic (2009) there are generally three types of approaches used in a dynamic scheduling process under uncertainty:

- In the **reactive scheduling** approach knowledge of possible production disruptions is not taken into account when assigning jobs to resources. No firm schedule is generated in advance and decisions made to rebuild and select the next jobs have a local nature and are based on dispatching rules.
- The **predictive-reactive scheduling** revises schedules in response to real-time events. The new schedule can deviate significantly from the original schedule.
- The **robust pro-active scheduling** approach focuses on building predictive schedules, which satisfy performance requirements predictably in a dynamic environment.

Besides modelling the uncertainty, a decision must be made on the *planning horizon*. A finite horizon means the end of the schedule is set and optimisation is done until the end of that horizon. Infinite horizon means optimising the schedules into infinity, using all known information. A short planning horizon means less computational burden but also less predictive power about the future, while a long horizon may be computationally unfeasible and contain too much inaccurate information (Zhang et al., 2003). A rolling horizon can be used as well. During each planning time frame, optimisation for a predetermined horizon in the immediate future is done and the plan is executed accordingly until the subsequent planning time frame is reached. Then, we devise a new plan based on the next horizon with the new information obtained from the first horizon. In this research, a finite planning horizon is used. In this finite planning horizon, a given set of orders needs to be completed.

Another modelling decision to be made is *representation of time*, which can be either discrete or continuous. Typically, time is presented with real numbers for continuous and discrete event models, and integer numbers for what is defined as discrete-time models. Continuous time is used for representing discrete event systems since signals can be created at non-regular time instants. Discrete-time offers advantages of continuous time simplifying simulator development, since modification happens at every time step, avoiding the use of complex data structures to efficiently manage asynchronous event (Barros, 2016).

In conclusion, our problem is a single-machine scheduling problem with family setups, multi-fixturing pallets, (periodical) resource constraints with product-specific characteristics and constraints.

3.2.3 Related scheduling research

Extensive research on single-machine scheduling has been performed, covering a wide range of situation-specific aspects. Some examples of these are the articles of Zheng and Jin (2019), Rapine et al. (2012), Chen et al. (2013), Figielska (2009), de Athayde Prata et al. (2021). The differences in these researches are mainly on objective, resource constraints, job characteristics and stochasticity. The most common objective is makespan, which is also the objective of our problem.

Every characteristic of our problem appears in the literature. The operator non-availability is discussed in the articles of Rapine et al. (2012) and Chen et al. (2013), where Dang et al. (2023) considers a similar unsupervised shift where manned operations can not take place. Other resource constraints are discussed in the articles of Figielska (2009) and Wu and Cheng (2016). The dissertation from Zeestraten (1989) and the article of Shin et al. (2019) do include resource constraints from pallets, fixtures and operator non-availability. The job characteristics for setup times, due dates and release times are described in the paper by de Weerdt et al. (2021). Furthermore, each article has its solution approach for solving the problem, with de Athayde Prata et al. (2021) comparing the performance of different solution approaches in a single-machine scheduling environment with periodic resources. To the best of our knowledge, the paper of Shin et al. (2019) comes closest in terms of machine characteristics and the paper of Dang et al. (2023) comes closest in terms of operator (non-) availability scheduling constraints.

Although no exactly similar problem is found, the articles reviewed can serve as a source of inspiration for both the formulation of the problem constraints and the development of a solution approach. The solution approach chosen in the literature varies quite heavily. Table 3.1 shows the solution approaches that were used in scheduling problems that are close to our paper. All of the papers make use of heuristics to find solutions to larger problem instances. Half of the papers solve the model exactly to assess the performance of the heuristic. Furthermore, 6 out of the 10 papers provide a Mixed Integer (Linear) Programming (MI(L)P) model of the problem. Providing an MI(L)P model for a scheduling problem serves as a foundation for theoretical understanding, algorithm development, and performance evaluation. It offers an optimal solution benchmark against which heuristic methods can be compared and provides a structured framework for addressing complex scheduling challenges. The table shows a variety of solution approaches. Notably, certain methods, such as simulated annealing and tabu search, have demonstrated repeated success across multiple papers. In the next section, which is on solution approaches, we go over some of the information that is provided by closely related papers on different solution approaches.

3.3 Solution approaches

In this section, we identify suitable solution approaches for the scheduling problem at hand. We identify and assess methodologies that have been used in comparable problems in the literature (see Table 3.1), while also introducing some new literature. For solving these types of machine scheduling problems, two methods are used: exact and heuristic. Given the complexity of finding optimal solutions to large optimisation problems, heuristics are used to find a suitable solution in reasonable computational time (Martinelli et al., 2022). Our problem consists of instances of over 1000 jobs. Therefore, exact approaches are likely not suitable. This is proven by de Athayde Prata et al. (2021) who found that a single-machine scheduling problem with periodical resource constraints is NP-hard, which our problem is an extension of. Even though our problem is NP-hard, we still discuss the exact solution approach. Formulating a model for solving the problem exactly, can increase understanding of the problem. The remainder of the section discusses two categories of heuristics, namely construction heuristics and improvement heuristics. Afterwards, relevant neighbourhood operators are discussed.

Table 3.1: Related research solution approaches

Article	Scheduling problem	Similarity	Exact	Heuristic	Size	Comment
This work	Single machine, (periodical) resource and product constraints, family setups, due- and release dates			Dispatching rules for constructive, SA and TS extended with variable neighbourhood for improvement heuristics	≤ 1800	Problem size larger than similar papers in literature
(Dang et al., 2023)	Parallel machines, unsupervised scheduling	Unsupervised scheduling	MILP	Practitioner heuristic and genetic algorithm	≤ 1201	MILP unable to solve $n \geq 25$, genetic algorithm roughly 20% better than practitioner heuristic
(Martinelli et al., 2022)	Single machine in MTO: a systematic review	Single machine	MILP, Poly, Brand-And-Bound	Genetic algorithm, simulated annealing, tabu search etc.	-	Most popular metaheuristics: Genetic Algorithm (GA), Simulated annealing (SA) and Tabu search (TS)
(de Athayde Prata et al., 2021)	Single machine, periodical resource constraints	Single machine with periodical resource constraints	MIP, MILP	Three-stage constructive, size reduction algorithm and simulated annealing	≤ 300	Three-stage constructive heuristic (aggregation, sequencing and allocation)
(Lee et al., 2020)	Operation scheduling for flexible manufacturing systems with multi-fixturing pallets	Multi-fixturing pallets		Three-stage: input-sequencing, pallet routing and pallet sequencing	-	Dispatching rules used for each phase, no local search applied
(Shin et al., 2019)	Single machine, flexible machining cell with multi-fixturing pallets	Single machine, FMC with multi-fixturing pallets		Two-stage algorithm, each stage constructive and then simple local search	≤ 300	Dispatching rules used for constructing solutions
(Herr and Goel, 2016)	Single machine, family setups and resource constraints	family setups and resource constraints	MIP	Iterated local search	≤ 50	Exact advised for small problem instances
(Chen et al., 2013)	Single machine with operator non-availability (ONA)	Single machine, operator non-availability		Algorithm based on Modified Shortest Processing Time	≤ 50	Operator non-availability more difficult than machine non-availability
(Rapine et al., 2012)	Single machine, small operator non-availability (ONA)	Single machine, operator non-availability		Polynomial algorithm (single ONA), List scheduling (small ONA's)	-	
(Shin et al., 2002)	Single machine, with due dates and release times, sequence dependency	Single machine, due- and release dates and sequence dependency		Tabu search, rolling horizon procedure	≤ 100	Tabu search obtained much better results than rolling horizon procedure
(Tan et al., 2000)	Single processor, minimizing tardiness with sequence-dependent setup times	Single machine, sequence dependency	Brand-And-Bound	Simulated annealing, genetic algorithm and pairwise interchange	≤ 45	Branch-And-Bound for smaller problems, simulated annealing and pairwise provide good results for larger instances

3.3.1 Exact

An exact solution approach refers to a method or algorithm designed to solve a problem with complete accuracy, providing an optimal solution that adheres to the problem's constraints. Exact solutions can be found through exhaustive search techniques, such as linear programming, brand-and-bound algorithms, etc. As Martinelli et al. (2022) posed, exact solutions may not be ideal to solve large problem instances, due to high computational time. However, solving small instances of your problem exactly and comparing them to heuristic approaches can indicate the performance of the heuristic(s). From the papers in Table 3.1, five of them solved the model exactly. The MI(L)P approach was predominantly utilized, accounting for four out of five instances where an exact solution method was employed. In this approach, an MI(L)P model is formulated and then solved using a solver. The second most common exact approach in Table 3.1 is Branch-And-Bound. The Branch-And-Bound method involves partitioning the extensive solution space into disjoint partitions, enabling the search for feasible solutions within each partition. Additionally, efforts can be made to eliminate partial partitions by comparing lower and upper bounds, facilitating a more efficient exploration process. The last exact solution approach that can be seen in the table is Polynomial. A polynomial exact solving approach utilises algorithms with polynomial-time complexity, meaning they solve problems in a number of steps bounded by a polynomial function of the input size. As can be noted from the "Comment" column in Table 3.1, often an exact approach is advised for smaller problem instances. To capture what smaller problem instances are, Table 3.2 is constructed. It is noteworthy that exact approaches are predominantly applicable to smaller problem sizes, with only the study by de Athayde Prata et al. (2021) addressing significantly larger problem instances.

Table 3.2: Problem size limit of exact approaches

Paper	Exact approach used	Problem size limit ($n \leq \ldots$)
(Dang et al., 2023)	MILP	25
(de Athayde Prata et al., 2021)	MIP, MILP	300
(Herr and Goel, 2016)	MIP	15
(Tan et al., 2000)	Branch-And-Bound	45

3.3.2 Constructive heuristics

A constructive heuristic is a systematic approach that commences with an empty solution and incrementally adds one element at a time until a complete solution has been formed. This can be different ways. A dispatching rule can be used to determine the sequence of the job(s) to be scheduled, as detailed by (Sörensen et al., 2018). Notably, these dispatching rules are fast, due to their low computational complexity. Most papers in Table 3.1 use dispatching rules for generating an initial solution, with examples of exceptions being Herr and Goel (2016) and Tan et al. (2000). Moreover, it's worth noting that there are many different types of dispatching rules for constructive heuristics, offering numerous strategies for building solutions efficiently. A few examples of this as given by Ruiz (2015) and Pinedo (2016) are in Table 3.3:

Table 3.3: Dispatching rules

Rule	Definition	Objective
SIRO	Service In Random Order	Ease of implementation
ERD	Earliest Release Date	Variance in throughput times
EDD	Earliest Due Date	Maximum lateness
MS	Minimum Slack	Maximum lateness
SPT	Shortest Processing Time	Sum of completion times, WIP
WSPT	Weighted Shorted Processing Time	Weighted sum completion times, WIP
LPT	Longest Processing Time	Load balancing for parallel machines
SPT-LPT	Combination SPT and LPT	Efficiency and maximum lateness
LNS	Largest Number of Successors	Makespan
SST	Shortest Setup Time	Makespan and throughput
LFJ	Least Flexible Job	Makespan and throughput
LAPT	Longest Average Processing Time	Load balancing for parallel machines
SQ	Shortest Queue	Reduce waiting time
SQNO	Shortest Queue at Next Operation	Machine idleness

Next to the sequencing of the jobs, we address job allocation on the machine, inspired by the paper of de Athayde Prata et al. (2021). This allocation is guided by criteria associated with the bin packing problem. The bin packing policies employed in this context are outlined as follows:

- **First fit (FF)**: Insert a job where possible, considering the resource constraints
- **Best fit (BF)**: Insert the job in the best possible position, considering the resource constraints. The best possible fit results in the best objective value

The paper of de Athayde Prata et al. (2021) illustrates the bin packing policies, as well as what happens if no bin packing policy is used. This can be seen in Figure 3.3. While the terms 'best fit' and 'first fit' are typically linked with bin packing policies, in the context of our scheduling challenge, these terms can take on a comparable role in characterizing allocation strategies for the assignment of jobs to pallets on a machine. In this context, 'best fit' and 'first fit' refer to methods to determine the pallet to allocate the job.

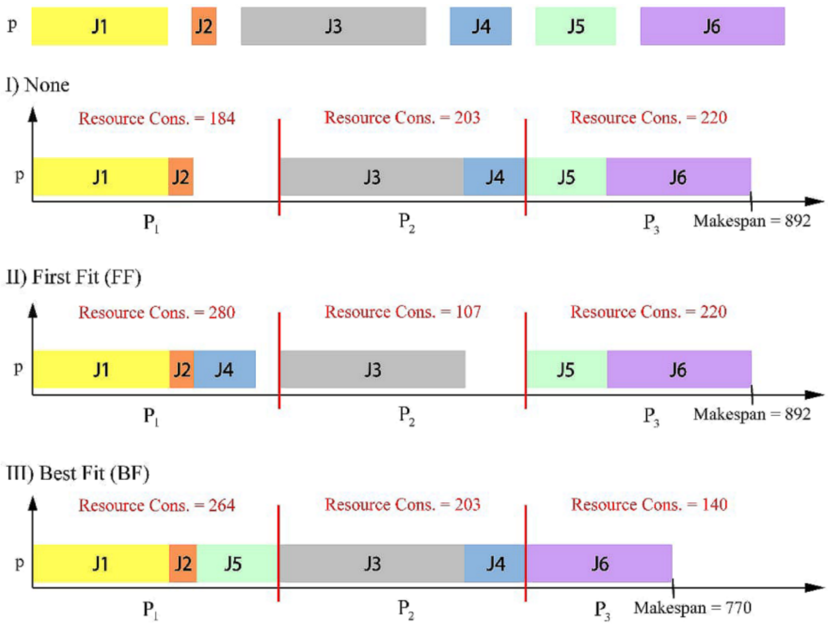

Figure 3.3: Bin packaging policies (de Athayde Prata et al., 2021)

The choice of constructive algorithm depends on the nature of the problems and the objective to be achieved. In most real-world problems, sorting the jobs based on one parameter may not yield acceptable schedules. For these problems a combination of dispatching rules may be used, called composite rules. The constructive heuristic provides a starting point for improvement heuristics, of which two relevant to our problem are discussed in the next section.

3.3.3 Improvement heuristics

Improvement heuristics aim to find a better solution from a starting solution in an iterative manner. Neighborhood operators (see Section 3.3.4) are used to get a neighbour solution. Improvement heuristics handle and accept neighbour solutions differently. A general optimisation algorithm will often either stop at a local optimum or converge to a local optimum. A local optimum is a point in the problem space where no neighbouring options offer an improvement, and a global optimum is a point where no other feasible solution has a better objective value (Rader, 2010). Metaheuristics balance intensification and diversification to overcome being stuck in a local optimum. Diversification is the exploration of much of the feasible region, whereas intensification focuses on a small area for the best solutions. In this section, two of the commonly applied metaheuristics for different types of single-machine scheduling problems are discussed: simulated annealing (SA) and tabu search (Martinelli et al., 2022). These metaheuristics have demonstrated repeated success in papers by Shin et al. (2002), de Athayde Prata et al. (2021), Tan et al. (2000) and Martinelli et al. (2022).

Simulated annealing is one of the older metaheuristic approaches introduced by Kirkpatrick (1983). The idea of SA is that it starts with diversification and ends with intensification. In this way, the algorithm can escape local optima by allowing moves that worsen the objective function. Algorithm 1 shows a general simulated annealing framework. A neighbour solution is always accepted if it is better than the current solution and set as the best solution if it is better than the best solution so far. If the neighbour solution is worse than the current solution, the solution is still accepted with a certain probability. The acceptance probability depends on the difference between the solution and the neighbour solution and the current temperature. As the current temperature becomes lower, the acceptance probability becomes lower as well, resulting in intensification. The cooling scheme of the SA consists of the following elements: Starting Temperature, Length of Markov Chains, Decrease rule for Temperature (alpha), and a stopping criterion. A cooling scheme needs to be determined where first an acceptance ratio close to 1 is obtained and the trade-off between computational time and objective value is reasonable. Besides, the neighbourhood structure needs to be decided for generating neighbour solutions. Simulated annealing is a very good option for solving our problem as manipulating the search space of the problem is straightforward, and it has fewer controlling parameters compared to the majority of existing algorithms in the literature (Amir Mohammad Fathollahi-Fard, 2019).

Algorithm 1: Simulated Annealing (Minimisation problem)

```
Temp ← StartTemp;
Solution ← ConstructInitialSolution;
BestSolution ← Solution;
while not stopping criteria do
    for m ← 1 to MarkovChainLength do
        NeighbourSolutionValue ← FindNeighbourSolution (Solution);
        if NeighbourSolutionValue < SolutionValue then
            if NeighbourSolutionValue < BestValue then
                BestSolution ← NeighbourSolution;
            Solution ← NeighbourSolution;
        else
            if RandomNumber ≤ exp((SolutionValue−NeighbourSolutionValue)/Temp) then
                Solution ← NeighbourSolution;
    Temp ← alpha · Temp;
Result ← BestSolution;
```

Tabu search (TS) moves out of local optima through a deterministic approach, whereas SA uses a probabilistic approach (Rader, 2010). Information from previous iterations guides current and future moves. Based on an initial solution, the neighbour solutions are evaluated. In TS neighbour solutions are kept in a tabu list and ineligible for consideration for a set amount of time (Rader, 2010). This restricts the available neighbourhood and prevents cycling back to recently visited solutions. The tabu tenure dictates the size of the list and the length of memory to use. In each iteration, the best non tabu solution from the neighbourhood is chosen, until some termination condition is met. Algorithm 2 shows the TS algorithm.

Algorithm 2: Tabu Search

```
TabuList ← ();
Solution ← ConstructInitialSolution;
CurrentBest ← Solution;
while not stopping criteria do
    Neighbourhood ← GenerateNeighbourhood(Solution, TabuList);
    BestNeighbour ← ChooseBestNeighbour(Neighbourhood, TabuList);
    Solution ← BestNeighbour;
    if Solution < CurrentBest then
        CurrentBest ← Solution;
    if Length(TabuList) ≥ TabuListLength then
        RemoveOldestElement(TabuList);
    AddElement(Solution, TabuList);
Result ← CurrentBest;
```

3.3.4 Neighbourhood operators and variable neighbourhoods

In the context of single-machine scheduling improvement heuristics, neighbourhood operators are used to explore nearby solutions by making small modifications to the current schedule. The result of this modification is a neighbour solution. There are different types of neighbourhood operators and this section highlights the most important ones for the single-machine scheduling problem. The most common neighbourhood operators are swap, move and inversion. These operators, swap two positions randomly, move a job to another position, and a subset within the sequence is inverted (see Figure 3.4). The operators can be modified slightly by for example swapping pairwise or by swapping jobs based on their characteristics.

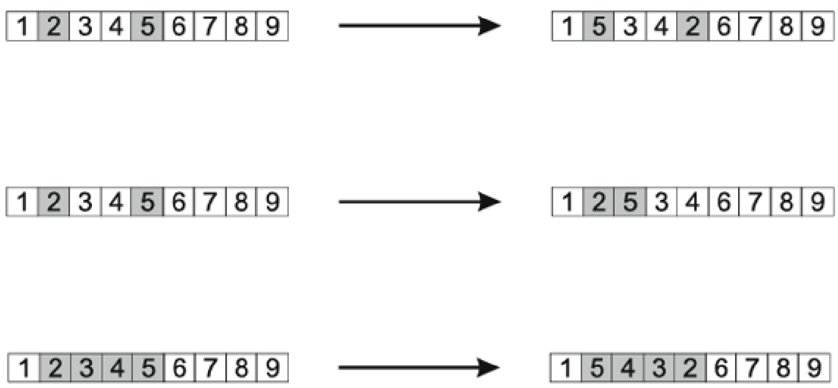

Figure 3.4: Swap (top), move (middle) and inversion (bottom) operators (Eiben and Smith, 2015)

In the case where family setups are present, some additional operators may be useful. The paper of Herr and Goel (2016) discusses four interesting family (batch) neighbourhood operators. The first is a family move operator, which selects a family and inserts it in another position in the sequence. Second, the family swap operator changes the position of two families. Third, the family combine operator combines batches of the same family and chooses a reinsertion point. The final operator breaks families into two parts and inserts both parts in a new position in the sequence. Figure 3.5 illustratively shows these operators.

There are several strategies for selecting neighbourhood operators in improvement heuristics. The most straightforward approach is random selection. However, more sophisticated strategies aim to strike a balance between intensification and diversification, leading to superior results, as demonstrated in studies such as Naderi et al. (2009).

Naderi et al. (2009) write that an effective strategy involves finding a compromise between intensification and diversification. This compromise can be achieved in different ways. For instance, one approach is to conduct an intensive search within the current solution space using small operators, and then transition to a different solution space by employing larger operators, which are also intensively explored. Alternatively, a different strategy involves diversifying the search initially using larger operators to explore various solution spaces. In a later stage of the improvement heuristic, the focus shifts to intensification within a promising solution space.

Variable Neighborhood Search (VNS), as explained by Mohamed Abdel-Bassat (2018), can provide a way to balance intensification and diversification in optimization. The foundations of VNS are built on the fact that a local optimum in one neighbourhood might not hold for all neighbourhoods, whereas the global optimum is the local optimum regarding all possible neighbourhoods, and local optima are frequently relatively close to each other. Based on these principles, VNS offers a dynamic approach to optimization by alternating between randomly and systematically exploring neighbourhoods according to predetermined conditions.

Lalla-Ruiz et al. (2020) assessed the strategy of systematically exploring neighbourhoods, stemming from VNS, for SA. The strategy is referred to as variable neighbourhood (VN). The neighbourhood structure is changed depending on the success of finding better solutions. For the problem in the research conducted by Lalla-Ruiz et al. (2020), extending SA with VN exhibits a better performance compared to the standard SA. Besides, the paper by Behnamian (2013) showed that VN can also be combined with TS and provide a better performance compared to the standard TS.

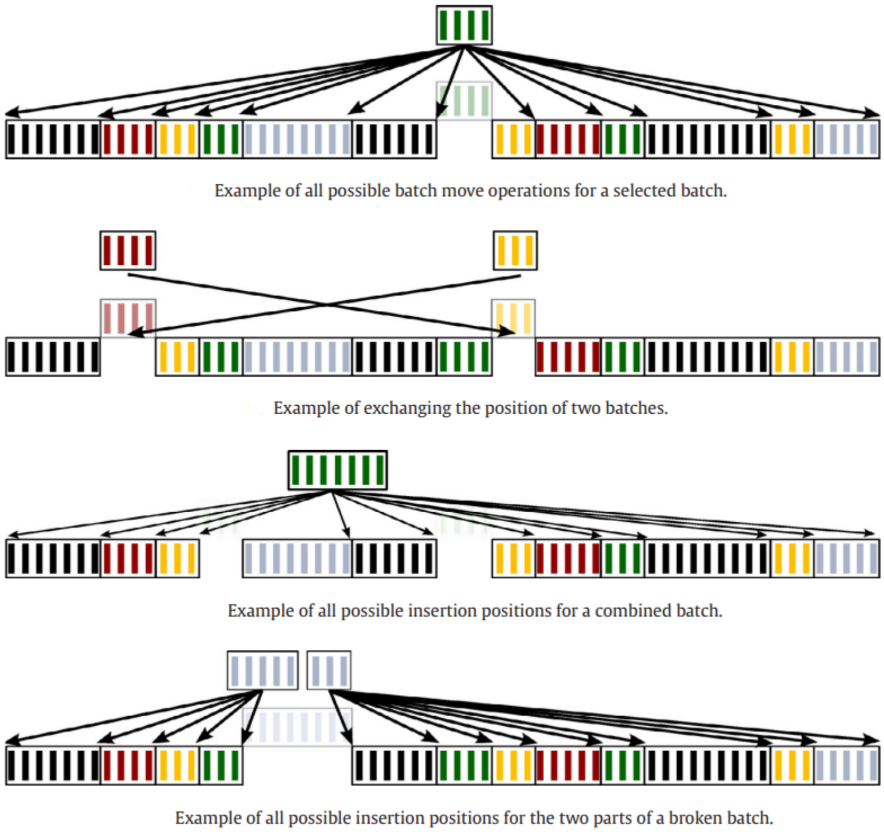

Figure 3.5: Examples of family (batch) operators (Herr and Goel, 2016)

3.4 Conclusion

This chapter provides the literature review on the scheduling problem at hand and answers the research question: *What models are presented in the literature for constructing a production schedule and what optimisation heuristics are available for optimising a production schedule?* The production environment of HTM Aerotec can be identified as MTO. Production takes place after a customer places an order. The research is placed at the operational level of resource capacity planning. Scheduling definitions are presented and a taxonomy of the scheduling problem at hand is done. The scheduling problem is a single-machine scheduling problem with family setup, multi-fixturing pallets, (periodic) resource constraints, and product-specific characteristics and constraints.

Based on the classification we identify that every characteristic of our problem appears in the literature. Single-machine scheduling problems in literature cover only a small variety of (periodical) resource constraints and/or product-specific characteristics and constraints. No exactly similar problem was found. Related research gives a source of information for both the formulation of the problem constraints and the development of a solution approach.

Two different types of solution approaches were identified: exact and heuristic. Exact methods, like MI(L)P and Branch-And-Bound, are used for finding optimal solutions and are computationally expensive for larger problem instances. Constructive heuristics construct an initial solution, which can be constructed using dispatching rules (see Table 3.3) like EDD and with different job allocation strategies (see Figure 3.3). An improvement heuristic can iteratively improve the initial solution by finding and evaluating neighbour solutions that are found using neighbour operators. The selection of neighbour operators can be random or based on a strategy in which diversification and intensification are balanced. Based on related research, we propose to use SA and TS as improvement heuristics. A variety of combinations can be used for the solution approach. The best solution approach can be determined based on the experiments done after modelling the scheduling problem.

4 Solution design

This chapter gives the outline of the model and solution approaches that can be used to solve the scheduling problem. By doing this, the third research question is answered:

How can the production schedule of the 5-axis milling machines at HTM Aerotec be modelled?

In Section 4.1 we describe what data we require and the data transformation that is needed. All assumptions and simplifications that were made are given in Section 4.2. Section 4.3 presents the problem statement. Section 4.4 explains how the constructive heuristic works. Section 4.5 provides the improvement heuristics that are used, whereas Section 4.6 summarises all solution approach alternatives.

4.1 Input data transformation

The main input for the model is the schedule list that is sent to production periodically. There is still some crucial planning information missing from this schedule list. The schedule list lacks details regarding the labour time required per product, the quantity of fixtures available for a production order, the number of products fit on a pallet for each production order and the number of operations required for each product within a production order. This information is (inefficiently) taken from an information system called Teamcenter. The rest of the information can be obtained from the ERP system Glovia G2. The schedule list consists of multiple orders. For each production order an order quantity, setup, run time, release date, due date is given. The setup time refers to the setup that needs to be done for the family (the order) and labour time refers to some manned operations like loading and unloading the pallets. All information related to time in Table 4.1 is in minutes.

Table 4.1: Example order input

Order	Qty.	Steps	Setup	Machine	Labour	Release date	Due date	Fixtures	Multi
1	70	1	120	10,50	4,20	0	7 (days)	100	10
2	50	1	120	13	5	0	9	1	4
3	..	..	..	..	..	..	..	..	..

In order to make solving the problem a bit easier, we transform the production order input data. As we can see in Table 4.1, each production order has a required production quantity and some can be placed in multiples on pallets. For the first order 10 products fit on a pallet and 70 need to be made in total. This means that 7 pallets need to be loaded for this order. For the second order 12.5 pallets can be loaded, meaning that that 12 pallets need to be loaded with 4 products on them and one with 2 products. Run time is multiplied by the amount of products that are placed on a pallet. The setup time and labour time remain the same and setup needs to be once for a product family. This transformation changes the order input to a list of jobs that need to be done. Furthermore, if a production order requires multiple production steps, this is also split up in the job list. This can be seen in Table 4.2. This transformation makes our input data more similar to scheduling problems in literature.

Table 4.2: Example job list

Order	Job	# Products	Operation	Run time	Labour time	Release date	Due date	Fixtures
1	1	10	1	105	4,20	0	7	100
1	2	10	1	105	4,20	0	7	100
1	3	etc.	..	..	..	..	..	..

4.2 Assumptions & simplifications

This section discusses the assumptions and simplifications that we make during the modelling approach. The following assumptions and simplifications are made.

- All input data is deterministic. This means that the setup time, run time and labour time for each product are fixed. For this, the data that is available in ERP is assumed to be correct. In reality, this may slightly deviate.

- Materials and tools are available for production orders without a release date. Materials and tools are assumed to always be available when production orders do not have a release date and for production orders with a release date, these become available from the moment they are released.
- Operators are available during working hours and fixed moments at the weekend. In reality, employees may leave due to sickness or for other reasons. Besides, the weekend availability is inconsistent in reality.
- The machine is empty at the start of the time horizon. No pallets in the machine are loaded with products at the start.
- Machine breakdowns are not considered in the model, which happens occasionally in reality.
- Dedicated fixtures are unique for each item, so they are not shared between items.
- Production orders can be set up in advance as long as the number of dedicated pallets does not surpass the available pallet capacity.
- The planning starts on a Monday at 7.00 am.
- There is no moving time between different operations of the same job. If an operation is completed, labour can immediately start for the next one, provided there is a subsequent operation.

4.3 Problem statement

4.3.1 Problem description

This research considers a single machine scheduling problem with family setups, multi-fixturing pallets, (periodical) resource constraints and product-specific characteristics and constraints. The objectives considered are the minimisation of the makespan, denoted as C_{max}, and the average tardiness. The makespan is equal to the completion time of the last job. Tardiness for a job occurs if it is completed later than the due date. Each machine has a set of production orders PO. These production orders have been split up into a set of jobs J. Each job consists of a sequence of n_j operations. The number of operations required for a job differs per production order. For some production orders only one operation is required, for others, it can be more. For the machine a set of pallets P is available. Each operation O_{ij} for operation i of job j can be processed on the machine for a given uninterrupted machine processing time MP_{ij}. Each operation belongs to a setup family f_{ij}. A setup time of an order, SU_{ij}, needs to be completed before the jobs and operations of this production order can be processed. A setup is needed when the tools and fixtures of a production order are not at the machine yet. The setup time is separate from its corresponding processing time, meaning that the setup of a setup family can occur whenever possible. The set of operations is to be processed by a single machine, which can only process one operation at a time. These orders need to be scheduled over an infinite time horizon. Besides, each operation is characterised by a labour processing time LP_{ij}, associated setup family F_{ij} and amount of dedicated fixtures available DF_{ij}. Jobs have a due date D_j and release date R_j. For jobs requiring multiple operations, an operation can only be performed if the preceding step has been completed. Manned operations like SU_{ij} and LP_{ij} are only allowed when an operator is available. An operation can be placed on a pallet if the dedicated fixture is placed on the pallet at the moment and the preceding operation on this pallet has been processed. The problem consists of deciding:

- When to perform the setup of each setup family
- When to do manned operations for each job
- The sequence in which jobs are processed on the machine
- Starting time of each job
- How to schedule the jobs on the pallets

When making those decisions, there are constraints to consider. For constructing the schedule the following should be taken into account:

- A job can only be processed after its labour time and the setup of the production order it belongs to has been performed
- Labour time on a pallet can be performed after the previous product has been processed on the same pallet
- Labour and setup time can only occur when an operator is available and these cannot overlap
- Labour and setup time can overlap with machine processing time
- Machine processing time can occur when an operator is not available
- Jobs that involve multiple operations are to be executed following the sequence of route steps. For a product to advance to the subsequent step, it must first be loaded onto a pallet and processed by the machine
- Jobs can only be assigned to pallets that have their dedicated fixture assembled on it at that point in time
- A pallet changeover time (labour time) occurs between the processing of jobs
- Jobs can only be scheduled after they are released
- The number of pallets used may not exceed the amount of pallets available for the machine

4.3.2 Scheduling sets, parameters & variables

The following sets are defined.

Set	Description
$\mathcal{PO}$	Set of production orders, indexed by o, $o = \{1, \ldots, \lvert PO\rvert\}$
$\mathcal{I}$	Set of all jobs, indexed by j, $j = \{1, \ldots, \lvert J\rvert\}$
n_j	Set of all operations required for job j, indexed by i, $i = \{1, \ldots, \lvert n_j\rvert\}$
$\mathcal{E}$	Set of all operations
$\mathcal{P}$	Set of all pallets, indexed by p, $p = \{1, \ldots, \lvert P\rvert\}$

The following parameters are defined.

Parameter	Description
$OBWeek$	Operator begin time during the week
$OEWeek$	Operator end time during the week
$OBWeekend$	Operator begin time during the weekend
$OEWeekend$	Operator end time during the weekend
MP_{ij}	Machine processing time for operation O_{ij}
SU_{ij}	Setup time for operation O_{ij}
LP_{ij}	Labour processing time for operation O_{ij}
D_j	Due time of job j
R_j	Release time of job j
$PalletTime$	Time it takes for the machine to change a pallet
M	Big M
F_{ij}	Setup family of operation O_{ij}
DF_{ij}	Dedicated fixtures for operation O_{ij}

The following variables are defined.

Variable	Description
X_{ijp}	Binary variable on whether operation O_{ij} is loaded on pallet p
$Y_{iji'j'}$	Binary variable on whether operation O_{ij} is scheduled before $O_{i'j'}$
$Z_{iji'j'p}$	Binary variable on whether an operation O_{ij} is scheduled before $O_{i'j'}$ on pallet p
SS_{ij}	Start setup time of operation O_{ij}
ES_{ij}	End setup time of operation O_{ij}
SL_{ij}	Start labour time of operation O_{ij}
EL_{ij}	End labour time of operation O_{ij}
SP_{ij}	Start processing time of operation O_{ij}
C_{ij}	Completion time of operation O_{ij}
T_{ij}	Tardiness of operation O_{ij}

4.3.3 Scheduling mathematical model formulation

In this subsection, the model formulation is given. The objective function and constraints of the model are given. The model is a combination of the models by Herr and Goel (2016), Dang et al. (2023) and Low et al. (2006) extended with some elements specific to this research.

$$\text{Min} \quad w * C_{max} + (1-w) * (\frac{1}{|\mathcal{E}|} * \sum_{j \in \mathcal{J}} \sum_{i \in n_j} \max(0, C_{ij} - D_j)) \tag{1}$$

The objective is twofold. The first is about the makespan of the schedule and the second part calculates the average tardiness. The first part is multiplied by a weight w and the second part by $(1-w)$. This weighting allows for adjusting the importance of tardiness relative to the makespan. The specific value for w is currently unknown and will be determined through experimentation.

$$\sum_{p \in P} X_{ijp} = 1 \quad \forall i, j, p \tag{2}$$

$$\sum_{i \in J \setminus \{j\}} Y_{iji'j'} = 1 \quad \forall O_{ij}, Oi'j' \in \mathcal{E} : O_{ij} \neq Oi'j' \tag{3}$$

$$Z_{iji'j'p} + Z_{i'j'ijp} = X_{ijp} * X_{i'j'p} \quad \forall p \in \mathcal{P}, \quad \forall O_{ij}, Oi'j' \in \mathcal{E} : \quad O_{ij} \neq Oi'j' \tag{4}$$

Constraint (2) ensures that each job is scheduled on one of the pallets. Constraint (3) ensures that each job is preceded by another job on the machine. Constraint (4) models the precedence constraints on a pallet. Constraints (3) and (4) are taken from Low et al. (2006). Constraint (4) is non-linear in the form it is stated above. To linearise this constraint, the following needs to be done. First, an auxiliary binary variable $W_{iji'j'p}$ needs to be introduced to represent the product $X_{ijp} * X_{i'j'p}$. The following linear constraints need to be added:

$$W_{iji'j'p} \geq X_{ijp} * X_{i'j'p} - 1 \quad \forall p \in \mathcal{P}, \quad \forall O_{ij}, Oi'j' \in \mathcal{E} : \quad O_{ij} \neq Oi'j' \tag{4.1}$$

$$W_{iji'j'p} \leq X_{ijp} \quad \forall p \in \mathcal{P}, \quad \forall O_{ij}, Oi'j' \in \mathcal{E} : \quad O_{ij} \neq Oi'j' \tag{4.2}$$

$$W_{iji'j'p} \leq X_{i'j'p} \quad \forall p \in \mathcal{P}, \quad \forall O_{ij}, Oi'j' \in \mathcal{E} : \quad O_{ij} \neq Oi'j' \tag{4.3}$$

Constraints (4.1-4.3) ensure that constraint (4) becomes linear if the auxiliary variable $W_{iji'j'p}$ is introduced.

$$ES_{ij} = SS_{ij} + SU_{ij} \quad \forall i, j \tag{5}$$

$$SL_{ij} \geq ES_{ij} \quad \forall i, j \tag{6}$$

$$ES_{ij} = ES_{i'j'} \quad \forall i,j: \quad F_{ij} = F_{i'j'} \tag{7}$$

$$EL_{ij} = SL_{ij} + LP_{ij} \quad \forall i,j \tag{8}$$

$$SP_{ij} \geq EL_{ij} \quad \forall i,j \tag{9}$$

$$C_{ij} = SP_{ij} + MP_{ij} \quad \forall i,j \tag{10}$$

This set of constraints sets the setup times, labour time and completion times. Constraint (5) sets the end of the setup time based on the start of the setup time. Constraint (6) indicates labour for an operation can only be fulfilled when the setup has been completed. Constraint (7) ensures that each operation of a setup family has the same setup times. Constraint (8) sets the end of labour times, where constraint (9) ensures processing happens after labour time for an operation has taken place. Constraint (10) sets the completion time of a job, based on when it has started processing.

$$C_{i'j'} \geq C_{ij} + MP_{i'j'} + PalletTime - (1 - Y_{iji'j'}) * M \quad \forall O_{ij}, Oi'j' \in \mathcal{E}: \quad O_{ij} \neq Oi'j' \tag{11}$$

$$SL_{i'j'} \geq C_{ij} - (1 - Z_{iji'j'p}) * M \quad \forall i, O_{ij}, Oi'j' \in \mathcal{E}: \quad O_{ij} \neq Oi'j' \tag{12}$$

$$C_{ij} \leq SL_{(i+1)j} \quad \forall j, i = 1, \ldots, (n_j - 1), n_j > 1 \tag{13}$$

$$SL_{i'j'} \geq EL_{i,j} \quad \forall O_{ij}, Oi'j' \in \mathcal{E}: \quad O_{ij} \neq Oi'j' \tag{14}$$

$$C_{max} \geq C_{ij} \quad \forall i,j \tag{15}$$

$$C_{ij} \leq D_j + T_{ij} \quad \forall i,j \tag{16}$$

$$\sum_{p \in \mathcal{P}} X_{ijp} \leq DF_{ij} \quad \forall i,j \tag{17}$$

$$SS_{ij} \geq R_j \quad \forall i,j \tag{18}$$

Constraint (11) makes sure machining time does not overlap between different operations (Herr and Goel, 2016). Constraint (12) says that labour on a pallet can only start after the product on that pallet has been processed. Constraint (13) ensures the precedence relations between operations of the same job. Constraint (14) ensures no overlap between labour times. Constraint (15) sets the makespan, whereas Constraint (16) sets the tardiness for each operation. Constraint (17) restricts the amount of pallets on which an operation can be placed to the number of dedicated fixtures available. Constraint (18) ensures that setup can only take place after the release time.

$$OBWeek \leq SL_{ij} \bmod 24 \leq OEWeek - LP_{ij} \quad \forall i,j: \left\lfloor \frac{SL_{ij}}{24} \bmod 7 \right\rfloor < 5 \tag{19}$$

$$OBWeekend \leq SL_{ij} \bmod 24 \leq OEWeekend - LP_{ij} \quad \forall i,j : \left\lfloor \frac{SL_{ij}}{24} \bmod 7 \right\rfloor > 5 \tag{20}$$

$$OBWeek \leq SS_{ij} \bmod 24 \leq OEWeek - SU_{ij} \quad \forall i,j \tag{21}$$

$$\left\lfloor \frac{SS_{ij}}{24} \bmod 7 \right\rfloor < 5 \quad \forall i,j \tag{22}$$

Constraints (19)-(22) ensure that operator availability is respected. Constraint (19) makes sure labour time is between operator start and end times on weekdays and constraint (20) does that for the weekend. Constraint (21) ensures setup times respect operator availability and constraint (22) restricts setups from being done at the weekend.

$$Y_{iji'j'}, Z_{iji'j'p}, X_{ijp} \in 0,1 \quad \forall i,j,p \tag{23-25}$$
$$SS_{ij}, ES_{ij}, SL_{ij}, EL_{ij}, SP_{ij}, C_{ij}, T_{ij} \geq 0 \quad \forall i,j \tag{26-32}$$

Constraints (23)-(32) are the sign constraints of the model.

4.4 Constructive heuristic

Due to the model's complexity and computational expense, we propose heuristics and metaheuristics to efficiently solve larger instances, such as those encountered at HTM Aerotec. Especially for problem instances of our size (see Table 5.2) the literature Table 3.1 shows that MILP is not a feasible method for solving. Subsection 4.4 describes the constructive heuristic and subsection 4.5 describes the improvement heuristic. In this section we first go over how an initial sequence of jobs is made, followed by how we convert an initial sequence to a feasible schedule. At last, we describe how we choose an initial solution and how we deal with the randomness that is introduced in the initial sequence of jobs.

4.4.1 Initial sequence of jobs

The first step of the constructive heuristic is to determine an initial sequence of the jobs to schedule. In Section 3.3 we identified some dispatching rules that can determine the starting sequence for the scheduling problem. Whilst we identified some dispatching rules in the literature review, we are not restricted to only using rules found in the literature review. Dispatching rules, tailored to the unique characteristics of the scheduling problem at hand, have more potential. This allows for leveraging domain knowledge and insights obtained during the context analysis. In our model, the following dispatching rules are implemented:

- **EDD:** In this dispatching rule the due dates are multiplied by a random factor, resulting in a score per job. The job list is then sorted (ascending) based on this score. Combining randomness with EDD can allow a schedule to balance the urgency of meeting due dates with a randomness factor. This prevents strict adherence to due dates that might lead to sub-optimal schedules, whilst still giving some priority to due dates. The random factor it is multiplied by is chosen based on what provides the best results on average. This results in the following priority formula:

$$\texttt{job['Score']} = \texttt{job['Due date']} \times \texttt{random.randint(5,15)} \tag{33}$$

- **Multi-factor priority:** This is a composite rule, meaning that it is a combination of dispatching rules. For this rule, we make a combination between the due date, amount of fixtures available, and the total hours for all items of the production order. For these factors, we first determine a score per job. This is a normalised score best on the minimum and maximum values of all jobs for this factor. The best receives a score of 0 and the worst receives a score of 1. The scores are multiplied by a random factor, resulting in a score per job. The list is then sorted (ascending) based on this score.

$$\texttt{job['Score']} = (\texttt{job['Due date score']} + \texttt{job['Total hours score']} + \texttt{job['Fixtures score']}) \times \texttt{random.randint(5,15)} \tag{34}$$

- **Random:** In this dispatching rule the job list is randomly shuffled. This dispatching rule is unlikely to provide good initial results as due dates are neglected, resulting in high tardiness. A random initial starting solution can help escape local optima, resulting in better solutions after the improvement heuristics.

In the first two dispatching rules we introduce randomness. Introducing randomness plays a crucial role in generating better initial solutions in our case. To showcase this, we provide for a small problem instance what the effect of randomness is. In Figure 4.1, the EDD dispatching rule without randomness is implemented on the left and EDD with some randomness is implemented on the right. The solution that includes randomness is finished much earlier, as setups are done earlier on and pallets are utilised more efficiently. This is the reason for introducing a random element to the dispatching rules. How we deal with this randomness for choosing an initial solution for the improvement heuristics can be seen in Section 4.5.1.

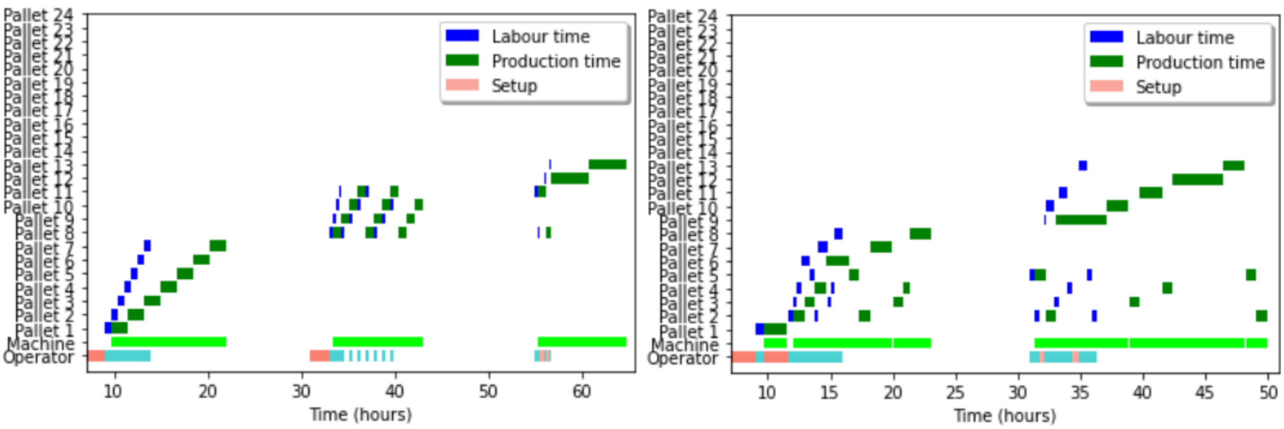

Figure 4.1: Example of EDD (left) and EDD with some randomness (right)

4.4.2 Constructing a feasible schedule

The second phase of the constructive heuristic constructs a feasible initial schedule. The data input for this is the job list, sequenced with a certain dispatching rule. The second phase consists of choosing a job allocation strategy. Based on Figure 3.3 we identified job allocation strategies like first fit (FF) and best fit (BF). Although we are not dealing with a bin packing problem, we can still use these policies in a similar way for allocating jobs to pallets. In our case, FF is not likely a good strategy as some pallets will be loaded much more frequently (the lower ones), which is why we will use the BF strategy. BF chooses the best pallet and best position in the schedule of this pallet. In short, BF results in the lowest makespan value of all available eligible pallets.

The constructive heuristic in Algorithm 3 presents the scheduling heuristic designed to efficiently manage a list of unscheduled jobs on the machine on a high level. The algorithm operates iteratively as long as unscheduled jobs are remaining. Within each iteration, a job is assigned to a pallet and the start and finish times of labour and machine processing times for the job are determined. If a job is the first of an order, a setup is scheduled and pallets are assigned to dedicated fixtures of that order. If a job belongs to an order with multiple operations, all operations of this job are scheduled consecutively.

Algorithm 3: Constructive scheduling Heuristic

```
// Initialise empty scheduling list
Data: List of unscheduled jobs, information about each job
while unscheduled jobs exist do
    for jobs in unscheduled jobs do
        for r ← 1 to RouteSteps(job) do
            Select the first job from unscheduled jobs from the same production order with operation r;
            Determine available pallets for this job and if setup is still required ;
            Determine earliest available feasible pallet p from available pallets ;
            Determine when to load and when to produce the job on p and feasibility;
            Update schedule ;
            Remove the scheduled job from the list of unscheduled jobs;
Result: Feasible schedule for machine
```

4.5 Improvement heuristics

4.5.1 Choosing an initial solution

The first step of the improvement heuristic involves selecting an initial solution. We introduced randomness for determining the initial sequence as this improves the planning quality in our case. However, constructing an initial solution only once may lead to a bad initial schedule due to randomness. In such instances, a substantial number of iterations within the improvement heuristic might be necessary to rectify the schedule stemming from the initial randomness. To prevent this, a similar technique as showcased by Tan et al. (2000) is used. Tan et al. (2000) generate an initial population P_0 of N solutions and choose the solution with the best objective value from the initial population as the initial solution for the metaheuristic. In Chapter 5, an experiment is done to determine the number N of solutions generated for the initial population from which the initial solution is chosen.

4.5.2 Neighbourhood operators and neighbourhood structure strategy

This subsection elaborates upon the neighbourhood operators we consider for our problem. The neighbourhood operators are used to find neighbour schedules, which possibly improve the schedule. We select the following neighborhood operators based on Section 3.3.4:

- **N1:** Swap two jobs in the sequence
- **N2:** Move job to different position in the sequence
- **N3:** Perform multiple moves of jobs in the sequence
- **N4:** Move all jobs in a production order to a different position

Operators N1-N4 are ordered based on the amount of changes they can make to a schedule. N1 and N2 are smaller operators. N1 swaps two different jobs in the sequence, whereas N2 moves a single job to a new position in the sequence. N3 and N4 are larger operators as they change the sequence a bit more. N3 performs multiple (in our case 3) moves of a job in the sequence. N4 moves the position of every job of a production order to another location in the sequence. The N3 and N4 operators result in significantly different neighbour solutions compared to N1 and N2.

We consider different strategies to select neighbour operators. The first strategy randomly selects any of the operators with equal probability. The second strategy is more about balancing intensification and diversification. Results show that improvement heuristics yield superior results if the neighbourhood operators strike a compromise between intensification and diversification (Naderi et al., 2009). To achieve this, we use the neighbourhood structure as explained by Lalla-Ruiz et al. (2020). For using this neighbourhood structure, called variable neighbourhood (VN), we need to have operators in increasing order in terms of how many changes to the current schedule can be made. This is already the case for N1-N4 presented earlier. Algorithm 4 and Algorithm 5 show how the VN switches between operators.

4.5.3 Simulated annealing

The first improvement heuristic is the simulated annealing (SA) heuristic we identified in Section 3.3.3. SA is a metaheuristic, meaning that it can escape local optima (Rader, 2010). SA always accepts a neighbour solution if it is better than the current solution and if it is better than the current best solution, this is stored. If the neighbour solution is worse, the solution is still accepted based on the difference between the objective value and the progression of the heuristic, often denoted by the temperature. The heuristic stops when the stopping temperature is reached. The heuristic returns the best-found solution. Appendix B describes the tuning process of the SA parameters. Algorithm 4 shows the SA algorithm. Lines (1-4) initialize the input for the SA. Line (2) is Algorithm 3. Lines (7-10) update the choice of an operator based on whether operators are selected randomly or VN is used. Line (11) generates a neighbour, based on a chosen operator. For doing this, again part of Algorithm 3 is used, with a certain input sequence. Lines (13-19) update the current solution, operator in case of VN and possibly the best solution. Lines (21-24) update the operator in the case of VN and possibly the solution with the probability as given in line (23).

Algorithm 4: Simulated Annealing (Minimisation problem)

```
   Data: StartTemp, Alpha, MarkovChainLength, joblist, VN
1  Temp ← StartTemp;
2  Solution ← ConstructInitialSolution(joblist);
3  BestSolution ← Solution;
4  VNCount ← 1;
5  while Temp ≤ StopTemp do
6      for m ← 1 to MarkovChainLength do
7          if VN then
8              k ← VNCount;
9          else
10             k ← random.randint(1, 4);
11         NeighbourSolution ← FindNeighbourSolution (Solution, N_k);
12         NeighbourSolutionValue ← CalculateObjective (NeighbourSolution);
13         if NeighbourSolutionValue < SolutionValue then
14             VNCount ← 1;
15             if NeighbourSolutionValue < BestValue then
16                 BestSolution ← NeighbourSolution;
17                 BestValue ← NeighbourSolutionValue;
18             Solution ← NeighbourSolution;
19             SolutionValue ← NeighbourSolutionValue;
20         else
21             if VNCount ≤ 4 then
22                 VNCount += 1;
23             if RandomNumber ≤ exp((SolutionValue−NeighbourSolutionValue)/Temp) then
24                 Solution ← NeighbourSolution;
25     Temp ← Alpha · Temp;
26 Result ← BestSolution;
```

4.5.4 Tabu search

The second and last improvement heuristic that is implemented is a TS. This can be helpful to avoid going back to a similar solution, especially when not randomly selecting jobs or batches for the neighbourhood operators. In the TS algorithm, neighbour solutions are kept in a tabu list. These solutions are ineligible for consideration for a set amount of time, called the tabu tenure (Rader, 2010). In this case, we keep the entire solution in the tabu list. In Appendix B, the parameter tuning process is described. Two parameters are tuned, namely the tabu list length and number of iterations. The TS can be seen in Algorithm 5. Lines (1-4) initialize the input for the TS. Similar to the SA Algorithm, lines (6-9) decide which neighbourhood operator is used to create a neighbour, based on if the strategy is random or VN. Line (10) finds a neighbour solution, but differently than for the SA. We do not generate an entire neighbourhood, as is more standard practice in TS, due to computational time. Instead, we try to identify a promising single neighbourhood, by looking at the current performance of jobs. Jobs that have high tardiness or are currently at the end of the makespan have a higher chance of being moved. Another example is that in a swap, a job with high tardiness is swapped with a job with high earliness. In this way, we do not generate an entire neighbourhood, but still quickly identify a promising neighbour. Line (12) checks whether the neighbour solution is not already in the Tabu list. Line (13) checks whether the neighbour solution is better. If this is true, lines (14-19) update the operator of VN in the case of VN, solution and tabu list. If the neighbour solution is not better, we also update the operator of VN in lines (21-22).

Algorithm 5: Tabu search

```
   // Initialise empty scheduling list, max iterations, tabu list size
   Data: List of unscheduled jobs, Maxiter, TabuListSize, VN
1  Solution ← ConstructInitialSolution;
2  BestSolution ← Solution;
3  VNCount ← 1;
4  iter ← 0;
5  while iter ≤ Maxiter do
6      if VN then
7          k ← VNCount;
8      else
9          k ← random.randint(1, 4);
10     NeighbourSolution ← FindNeighbourSolution (Solution);
11     NeighbourSolutionValue ← CalculateObjective (NeighbourSolution);
12     if NeighbourSolution not in TabuList then
13         if NeighbourSolutionValue < SolutionValue then
14             VNCount ← 1;
15             Solution ← NeighbourSolution;
16             SolutionValue ← NeighbourSolutionValue;
17             TabuList += NeighbourSolution;
18             if len(TabuList) ≥ TabuListSize then
19                 Remove the first solution from tabu list;
20         else
21             if VNCount ≤ 4 then
22                 VNCount += 1;
23     iter += 1;
24 Result ← Solution;
25
```

4.6 Solution approach alternatives

In Section 4.4 and Section 4.5 different ways to do a constructive heuristic and improvement heuristics have been provided. In this section, we highlight all the different solution approach alternatives that will be assessed. We discussed three different initial job sequencing dispatching rules: EDD, Multi-factor and Random. Afterwards, we decided on two different neighbourhood strategies: one is completely random, and one uses the variable neighbourhood principle from VNS. Finally, two different improvement heuristics are implemented: SA and TS. From these options, we can make 12 different solution approaches. These 12 different solution approaches can be seen in Table 4.3. In the next chapter, we decide on the best solution approach per machine, by making a trade-off between solution quality and running time.

Table 4.3: Solution approach alternatives

Approach	Initial sequence	Improvement heuristic	Neighbourhood strategy
1	EDD	SA	Random
2	EDD	TS	Random
3	EDD	SA	Variable neighbourhood
4	EDD	TS	Variable neighbourhood
5	Multi-factor	SA	Random
6	Multi-factor	TS	Random
7	Multi-factor	SA	Variable neighbourhood
8	Multi-factor	TS	Variable neighbourhood
9	Random	SA	Random
10	Random	TS	Random
11	Random	SA	Variable neighbourhood
12	Random	TS	Variable neighbourhood

4.7 Conclusion

This chapter outlines the solution design for solving the scheduling problem at hand and answers the research question: *How can the production schedule of the 5-axis milling machines at HTM Aerotec be modelled?* To ensure comprehensive input data, we identified the necessary information and filled gaps in the current schedule list using data from TeamCenter and the ERP system (Glovia G2). A data transformation was implemented to align our input data more closely with scheduling problems found in the existing literature. All modelling assumptions and simplifications are provided. A concise problem statement of the problem is made and all sets, parameters and variables of the problem have been identified. After, the mathematical model was made using these sets, parameters and variables.

Heuristics are provided to ensure a good solution to the problem in less computational time. Three dispatching rules were proposed for generating an initial sequence. The constructive heuristic transforms a sequence into a feasible production schedule. Two distinct neighbourhood strategies were devised, each employing four operators. The first strategy randomly selects operators, which in turn select random jobs during the entire optimisation. The second strategy uses a variable neighbourhood strategy based on whether the neighbour solution improved the current solution. Two different improvement heuristics are described. The first improvement heuristic is SA, and the second improvement heuristic is TS. This means that we have a total of 12 different heuristic solution approaches, which are evaluated in the next chapter to find out what the best solution approach for this problem is. The list of solution approaches can be seen in Table 4.3. Figure A.1 shows an overview of the general solution approach we take. In the next chapter, we find out which dispatching rule, in combination with a neighbourhood strategy and improvement heuristic works best for our problem.

5 Experimental analysis

This chapter presents the experimental analysis of the research. The goal is to find the best-performing solution approach from Table 4.3 and afterwards conduct sensitivity analyses with the best solution approach. The fourth research question is answered:

What experiments can be done with the model to investigate the performance?

We start the chapter by providing the problem instances that we are going to solve and the running time required for constructing a single schedule in Section 5.1. We define the experimental design in Section 5.2. Initial population size and improvement heuristics parameters are tuned in Section 5.3. We provide the outcome of all solution approaches in the base case and determine the best solution approach in Section 5.4. With the best solution approach, we do a sensitivity analysis on various input factors in Section 5.5. The algorithmic performance is compared with the real-world performance over a month in Section 5.6.

5.1 Problem instances

This section provides a summary of the problem instances, extracted from company data. As explained in Section 4.1, we make use of the schedule list which is made periodically by the production planner. The aggregate information for the problem instances can be seen in Table 5.1. Three problem instances have been created for both machines 538 and 539. Each instance comprises orders that need to be completed within one month, two months, or three months, respectively, for each machine.

Table 5.1: Aggregate information problem instances

Instance	Machine	Pallets	Orders	Quantity per order	Orders with due date	Order release dates	Order setups needed
538-1	538	40	7	48.57	7	0	1
538-2	538	40	16	64.44	16	1	4
538-3	538	40	27	52.63	27	4	8
539-1	539	24	17	54.70	17	0	1
539-2	539	24	32	52.50	32	0	4
539-3	539	24	44	56.67	44	0	10

After converting the schedule list into a job list (see Section 4.1), we obtain a list of jobs and operations to perform for the machines. In Table 5.2, the machine-specific job information is shown. For the columns in which averages are shown, we also denote the standard deviation between brackets. Besides, the times for setup, run time and labour time are indicated in hours. The run time and labour time averages and standard deviation are for operations. Each operation needs to be scheduled on the machine and therefore has an associated run time and labour time.

Table 5.2: Result of converting the schedule list to a list of all operations, including their characteristics

Instance	Machine	Total jobs	Production hours	Operations per job	Total Operations	Setup (order)	Run time	Labour time
538-1	538	300	457.02	2.03	610	1.65 (0.48)	0.75 (0.62)	0.21 (0.11)
538-2	538	887	983.93	1.50	1330	1.80 (0.77)	0.74 (0.57)	0.16 (0.13)
538-3	538	1260	1306.85	1.33	1680	1.78 (0.72)	0.77 (0.59)	0.18 (0.20)
539-1	539	480	670.42	1.24	595	2.15 (0.95)	1.13 (1.26)	0.19 (0.13)
539-2	539	854	1227.09	1.25	1067	1.90 (0.85)	1.15 (1.25)	0.17 (0.13)
539-3	539	1055	2058.80	1.42	1498	1.83 (0.77)	1.37 (1.39)	0.19 (0.15)

In Table 5.2 the information on the operations that need to be completed at HTM Aerotec can be seen. What we can conclude is that 538 typically processes smaller jobs and therefore requires more manned interventions to produce the same amount of production hours. This is why 538 has more pallets available, such that more pallets can be loaded with products to increase the production time. The workload on the 539 is a lot higher. Tardiness is guaranteed for machine 539, as 2058.80 hours need to be finished in 12 weeks. This requires 171.56 hours production hours per week on average for the machines, which is impossible.

Table 5.3 and Table 5.4 provide more insights into the number of operations required and fixtures available for a production order, whereas Table 5.5 depicts the number of products that fit on a pallet for the operations for both problem instances. These three things constrain the model in different ways. The higher the number of operations required, the more constrained the model is. Each operation corresponds to a production step. If a product has multiple production steps then it needs to be unloaded and positioned differently after each step. The number of fixtures gives a restriction on the amount of pallets a product can be placed on. If only one fixture is available for an order, products of this order can only be placed on the one pallet where this fixture is placed. At last, the fewer number of products fit on a pallet, the more constrained the model is. Less manned interventions are needed if more products fit on a pallet.

Table 5.3: Number of operations required (%)

Inst.	**Machine**	**1**	**2**	**3**	**4**
538-1	538	42.9	42.9	0	14.2
538-2	538	63.6	31.3	0	6.1
538-3	538	74.1	22.2	0	3.7
539-1	539	76.5	23.5	0	0
539-2	539	75.0	25.0	0	0
539-3	539	73.8	26.2	0	0

Table 5.4: Number of fixtures available per operation (%)

Inst.	**Machine**	**1**	**2**	**3**	**∞**
538-1	538	38.46	0	0	61.54
538-2	538	33.3	8.3	0	58.4
538-3	538	27.8	5.6	0	66.6
539-1	539	45.0	10.0	0	45.0
539-2	539	37.50	15.0	0	47.5
539-3	539	39.6	15.1	0	45.3

Table 5.5: Number of products fitting on a pallet (%)

Inst.	**Machine**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
538-1	538	100	0	0	0	0	0	0	0	0	0
538-2	538	92.0	4.0	0	4.0	0	0	0	0	0	0
538-3	538	86.1	11.1	0	2.8	0	0	0	0	0	0
539-1	539	60.0	10.0	5.0	15.0	0	0	0	10.0	0	3.6
539-2	539	61.5	7.7	0	23.1	0	0	0	5.2	0	2.5
539-3	539	52.7	10.9	1.8	23.6	0	0	0	3.6	0	3.6

The final piece of information that is needed for the problem instances is the operator availability hours. The operator availability hours are the same for both machines. These can be seen in Table 5.6.

Table 5.6: Operator availability hours

Days	**Start**	**End**
Monday-Friday	7.00	16.00
Saturday-Sunday	10.00	11.30

Heuristics are used to solve the larger problem instances, such as those encountered at HTM Aerotec, in a relatively low running time. Investigating the runtime of these heuristics is therefore interesting. In Figure 5.1, we illustrate the correlation between constructing a single feasible schedule's runtime and the number of operations to schedule. This denotes the time required to formulate a single feasible schedule based on a provided input sequence, which is equal to one iteration in the improvement heuristic. Our observation reveals an almost linear increase in runtime as the number of operations rises. For each iteration in the improvement heuristic, a feasible schedule needs to be made. Despite using heuristics, the runtime for calculating a feasible schedule based on a sequence is quite a lot, especially considering the number of operations outlined in Table 5.2.

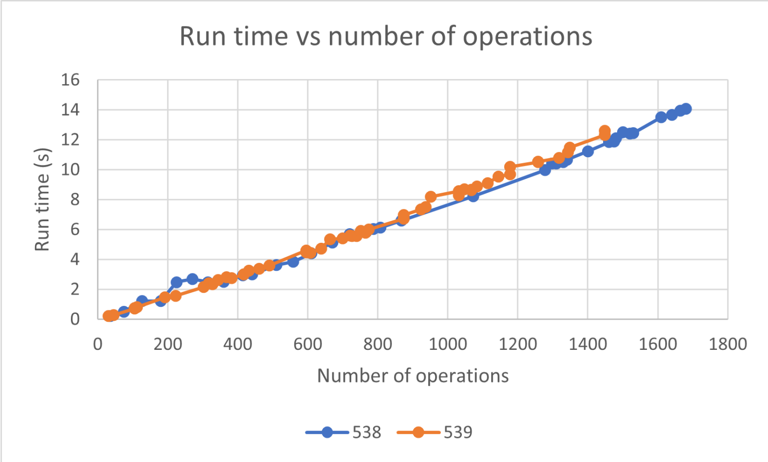

Figure 5.1: Run time versus number of operations for both machines for creating one feasible schedule (one iteration)

5.2 Experimental design

In this section, we define experiments to find algorithmic results of the solution approaches and to find out what the effect of real-world input parameters is on the system performance with the best solution approach.

Table 5.7: Overview of experiments with the goal of each experiment

Number	Experiment	Goal
1	Initialisation and Parameter tuning	Find weights of the objective function (Appendix B). Determine the number N of solutions to add to initial population size P_0, to mitigate risk of randomness and improve objective value. At last, the parameters for the improvement heuristics need to be found. The parameter tuning is needed to perform the rest of the experiments. In-depth explanation can be found in Appendix B
2	Analyse the performance of constructive heuristics	Find out which constructive heuristic performs the best in terms of makespan and tardiness for all problem instances
3	Analyse all solution approaches	Find out which combination of constructive heuristic and improvement heuristic works best and choose the best initial constructive heuristic
4	Compare algorithmic performances over problem instances	Find out the generalizability of solution approaches, and scalability. From this, we choose the best solution approach for all problem instances.
5	Sensitivity analysis on operator availability	To find out what the influence of operator availability is on system performance. Analysis is done on availability during the week, as well as on availability during the weekend. This experiment aims to assist HTM Aerotec in determining the feasibility and impact of changing operator visiting times
6	Sensitivity analysis on pallets available	To find out what the influence of the available number of pallets is on system performance. This experiment identifies whether the initial investment for pallets is good, or extra investment may be necessary to enhance performance
7	Sensitivity analysis on dedicated fixtures available	To find out what the influence of available fixtures is on system performance. With this experiment, it can be determined whether it is worthwhile to produce extra dedicated fixtures
8	Sensitivity analysis on schedule robustness	To find out the effect of making a schedule robust on system performance. This is done to simulate machine failure.
9	Algorithmic versus real-world performance	Find out if scheduling algorithm provides better results than observed in reality

5.3 Initialisation and parameter tuning

5.3.1 Initial population size

The constructive heuristic performs better when an element of randomness is introduced, as explained in Section 4.4. A similar method as showcased by Tan et al. (2000) is used, where we make an initial population P_0 of N solutions. In this section, we determine the value for N that leads to a good trade-off between population size and average and standard deviation of the best objective value of a population. For determining the population size, we experiment with population sizes of 1, 5, 10 until 60 with steps of 5. The population size of 1 is used as a benchmark against which we compare the performance of other population sizes. Five runs are performed with each population size and the average best objective value and standard deviation are noted. The results are depicted in Figure 5.2.

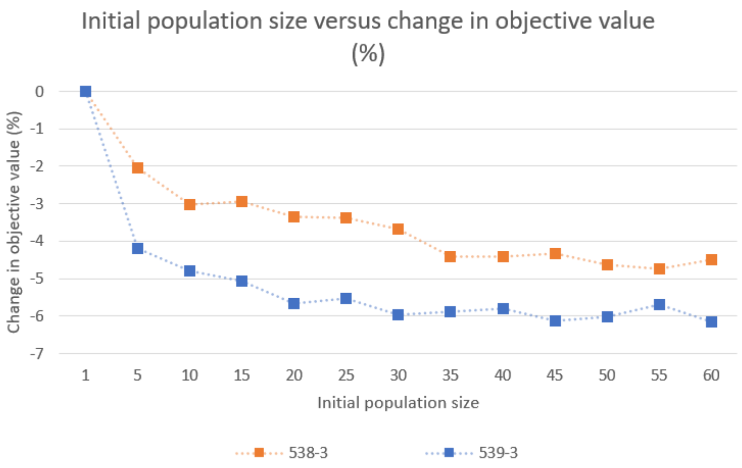

Figure 5.2: Initial population size versus change in average best objective value over 5 runs compared to population size of 1

What we can conclude from Figure 5.2, is that the initial population size makes a significant difference in the objective value. Beyond an initial population size N of 35, no consistent improvements are observed for both problem instances. In addition to having a good average best objective value, it is imperative to maintain a low average standard deviation of the best objective value to mitigate the risk of an unfavourable initial schedule. Figure 5.3 shows the average standard deviation of all problem instances versus the initial population size. From this, we can conclude that the average standard deviation does not decrease significantly after N of 30. Based on the information about the average best objective value and average standard deviation of the best objective value over 5 runs, choosing an N of 35 seems to be sufficient.

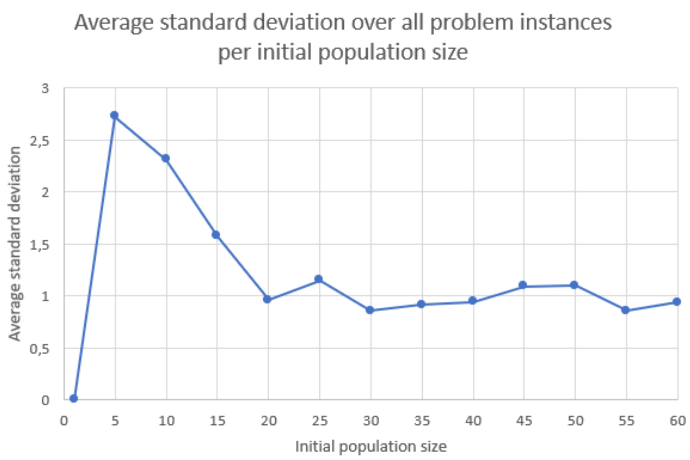

Figure 5.3: Average standard deviation of all best objective values of all problem instances versus initial population size

5.3.2 Improvement heuristics parameter tuning

The improvement heuristics work better if their parameters are tuned to fit the problem. Appendix B includes a more detailed analysis of the parameter tuning, this subsection is a brief explanation. First, the parameters of SA are tuned. The first step in tuning parameters for SA is to determine the starting temperature. We obtain this by solving problem instances 538-3 and 539-3 with the multi-factor dispatching rule and random operator selection. A temperature of 100 was used as a start for this experiment with Markov Chain Length of 100 and alpha of 0.8. For each temperature, the acceptance ratio is stored at the end of the Markov Chain length. This indicates the number of worse neighbours accepted, by the number of worse neighbours proposed. This is shown in Figure 5.4.

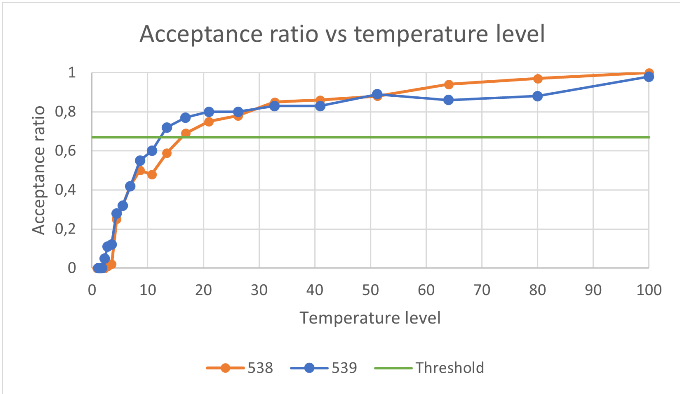

Figure 5.4: Acceptance ratio versus temperature level

Based on Figure 5.4, a starting temperature of 15 is chosen for both machines. A higher initial temperature can result in significantly worse solutions at the start, which can take a lot of iterations to overcome, taking excessively high computational times. Next experiments with stopping temperature, Markov Chain length and decrease factor alpha are performed for problem instance 539-3. For stopping temperature we consider the values 2.5, 5 and 7.5. For Markov chain length we consider 5, 10, 15. For the decrease factor alpha, we consider 0.9, 0.925 and 0.9. Figure 5.5 shows the average objective value versus the average run time over 5 runs for each of the 27 experiments.

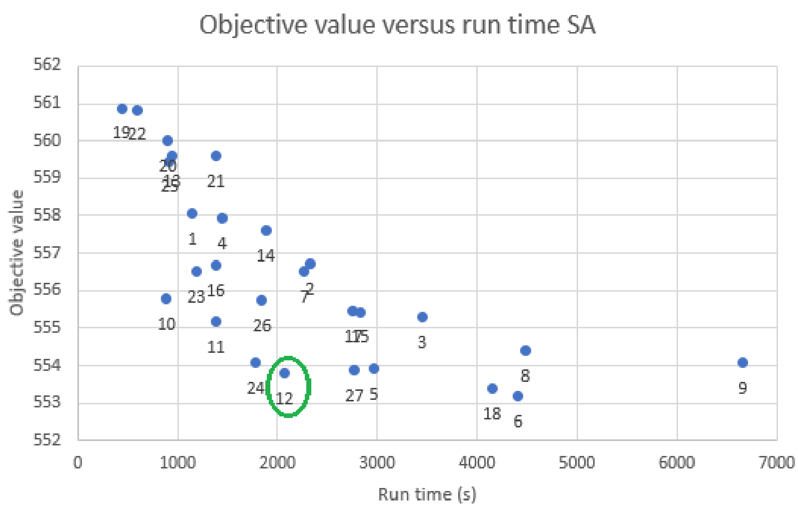

Figure 5.5: Average objective versus average runtime over 5 runs for 27 experiments for SA tuning

Based on Figure 5.5, experiment 12 is chosen as this experiment results in a good trade-off between objective value and run-time. This results in the following SA cooling scheme: start temperature of 15, stop temperature of 5, Markov Chain length of 15 and decrease factor of 0.9.

For the TS, the maximum number of iterations and maximum tabu list size need to be tuned. For the number of iterations, we consider the values 50, 75, 100, 125 and 150. For tabu list size, we consider the values 5, 10 and 15. Similar to the SA tuning, Figure 5.6 shows the average objective value versus the average run time over

5 runs for each of the 15 experiments.

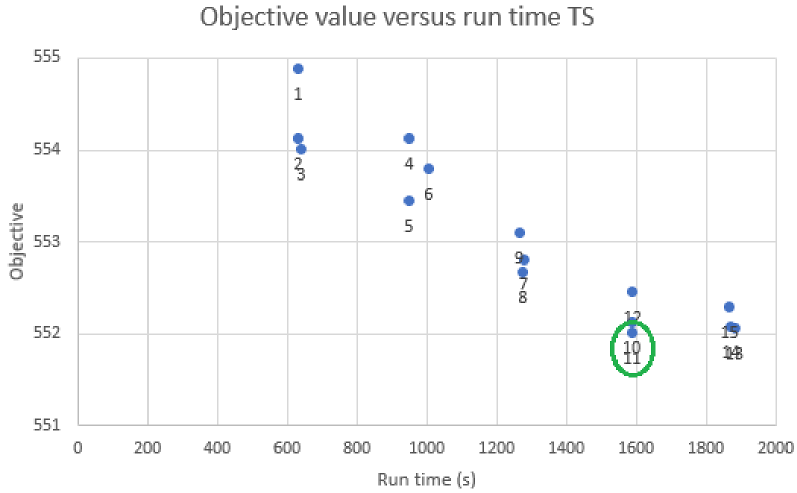

Figure 5.6: Average objective versus average runtime over 5 runs for 15 experiments for TS tuning

Based on Figure 5.6, we choose experiment 11, due to the good trade-off between objective value and run time. This means that for TS we use a maximum of 125 iterations and tabu list size of 10.

5.4 Algorithmic results

This section provides the results of solving the problem instances using the different solution approaches. The performance of the constructive heuristic with different dispatching rules is presented, without applying the improvement heuristics. After, the results of the improvement heuristics are obtained and the best solution approach for each problem instance is selected. For determining the objective value, a weight of 0.1 is used for w in Equation 1. The argumentation for choosing this weight can be found in Appendix B. Besides, the same random number seed is used for all experiments.

5.4.1 Initial solution heuristics results

Table 5.8 provides the average results over 5 runs of creating the initial solution with the three different dispatching rules for all problem instances. The EDD dispatching rule provides the best objective value for all problem instances. This has mainly got to do with the average tardiness achieved, as this is far lower than the other two dispatching rules for both problem instances. The multi-factor dispatching rule is for both problem instances the second best in terms of objective value. The random dispatching rule is good at achieving a low makespan, however does that whilst getting a high average tardiness.

A box plot has been made to provide a comprehensive overview of the quality of the initial solutions generated by the different dispatching rules. This box plot can be seen in Figure 5.7. The box plot shows that the EDD dispatching rule is by far the superior dispatching rule for generating the initial solution. It has a narrower spread of data points and a lower median value, highlighting the effectiveness of using EDD. In the next subsection, it is determined whether EDD still outperforms the other dispatching rules after applying the improvement heuristics.

Table 5.8: Initial solution results

Machine	Dispatching rule	Objective	Makespan	Avg. Tardiness	Run time	Util.
538-1	EDD	66.75	571.28	10.69	138.24	0.80
	Multi-Factor	118.48	573.80	67.89	139.40	0.80
	Random	128.62	544.07	82.46	141.21	0.84
538-2	EDD	123.08	1157.57	8.14	340.16	0.85
	Multi-Factor	218.47	1214.73	107.77	336.21	0.81
	Random	237.16	1157.57	134.89	342.05	0.85
538-3	EDD	159.77	1519.59	8.68	469.20	0.86
	Multi-Factor	283.59	1610.80	136.13	467.89	0.81
	Random	307.85	1547.26	170.14	470.21	0.85
539-1	EDD	92.40	761.84	18.02	135.25	0.87
	Multi-Factor	154.84	758.84	87.73	138.29	0.88
	Random	159.09	753.28	93.07	137.78	0.89
539-2	EDD	193.21	1394.42	59.74	269.50	0.88
	Multi-Factor	323.77	1378.75	206.55	270.09	0.89
	Random	332.66	1363.43	218.13	267.78	0.90
539-3	EDD	343.72	2339.55	121.96	423.50	0.88
	Multi-Factor	575.99	2299.55	384.49	424.01	0.89
	Random	591.80	2250.62	407.49	423.67	0.91

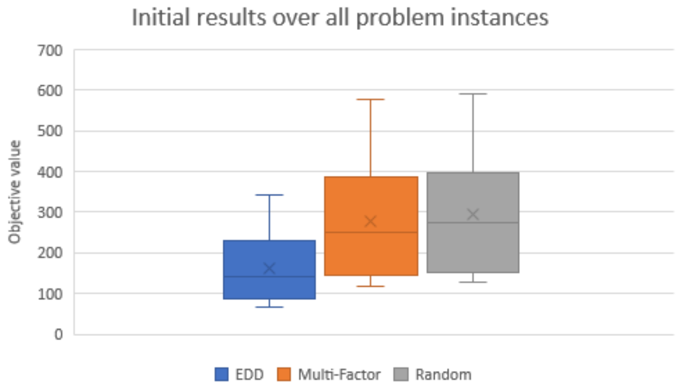

Figure 5.7: Initial solution objective value over 5 runs over all problem instances for the different dispatching rules

5.4.2 Improvement heuristics results

This subsection presents the results of the improvement heuristics. The following parameters were used for SA: starting temperature of 15, stop temperature of 5, alpha of 0.9 and Markov chain length of 15. Besides, we used max iterations of 125 for the TS and a tabu list size of 10. The performance of each of the 12 solution approaches is collected for all problem instances. A box plot has been made to provide a comprehensive overview of the solution approaches in Figure 5.8.

From Figure 5.8, it is hard to conclude what the superior solution approach is. What can be concluded is that all solution approaches where the EDD dispatching rule is used are performing the best. The algorithms are not capable of overcoming (too) bad initial solutions. The run time for creating the initial solution, doing the SA and doing TS can be seen in Table 5.9. There is not a significant difference in run time between using random operator selection or VN. The SA algorithm takes longer, which is logical given that it allows hill-climbing moves at the start, which takes more iterations to overcome.

Table 5.9: Solution approach average run time (s) per problem instance

Instance	Number of operations	Initial solution	SA	TS
538-1	610	139	715	513
538-2	1330	339	1699	1218
538-3	1680	469	2253	1615
539-1	595	137	706	555
539-2	1067	268	1354	1063
539-3	1498	424	2023	1588

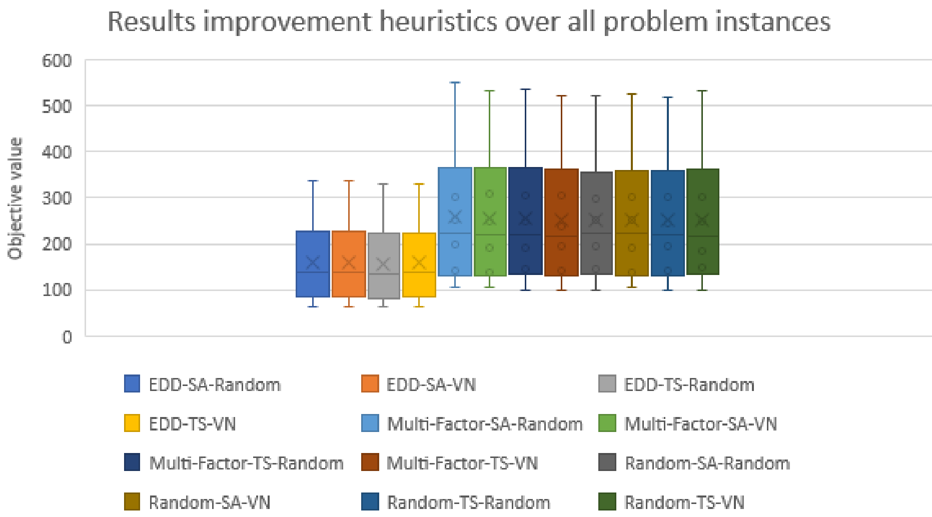

Figure 5.8: Improvement heuristics objective value over 5 runs over all problem instances for the solution approaches

To test which solution approach is the best, we will compare the performance of the four different solution approaches together with the EDD dispatching rule for all problem instances. From this, we can find which algorithm performs the best on average. For each algorithm, we note the minimum, maximum and average over 5 runs. This can be seen in Table 5.10.

Table 5.10: Algorithmic Performances with EDD dispatching rule

Inst.	SA-Random			SA-VN			TS-Random			TS-VN		
	Min	Max	Avg.	Min	Max	Avg.	Min	Max	Avg.	Min	Max	Avg.
538-1	63.12	66.01	64.33	63.34	66.49	65.23	61.12	63.34	**62.77**	61.89	64.89	63.47
538-2	120.21	123.08	122.80	120.89	122.45	121.78	119.10	122.10	**120.60**	120.13	121.98	120.60
538-3	153.11	154.78	153.99	152.89	157.89	154.89	149.12	154.38	**152.22**	149.89	156.10	152.45
539-1	87.12	92.31	89.78	88.61	92.40	90.21	87.30	90.12	**88.92**	90.34	91.89	91.01
539-2	188.07	192.34	190.22	187.92	193.21	191.22	185.45	189.69	**188.43**	185.99	190.46	188.53
539-3	336.56	342.11	338.29	335.88	340.73	337.89	325.67	334.98	**329.97**	328.12	335.52	331.22
Avg	158.03	161.77	159.90	158.26	162.20	160.20	154.63	159.10	**157.15**	156.06	160.14	158.04

As we can see in Table 5.10, TS with random operator selection performs the best for all instances. For some instances other solution approaches come very close, but TS with random operator selection outperforms the others for every instance. In Figure 5.9 the average percentage improvement in percentage change can be seen graphically.

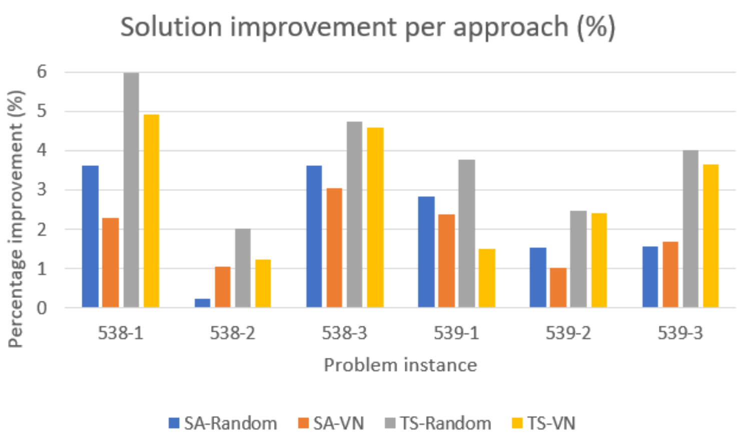

Figure 5.9: Average improvement per approach per problem instance

In conclusion, we can say that the constructive heuristic with EDD is superior for all problem instances. This is mostly due to the fact that this is the only dispatching rule that manages to keep tardiness low. Other dispatching rules are capable of reaching lower makespan, meaning that utilisation is higher. However, since tardiness is an important matter, EDD is by far the best dispatching rule for the constructive heuristic. For all problem instances, we found that TS with random operator selection performed the best after doing improvement heuristics.

5.5 Sensitivity analysis

This section presents the sensitivity analysis. The sensitivity analysis provides more insights and will help understand the outcomes of the base case. The base case represents the scenario as it exists in reality. We go over the effect of operator availability, availability of dedicated fixtures, pallet availability, and robustness of the schedule. For each sensitivity analysis, we provide the difference in objective value compared to the base case, called the gap. In addition to recording the 'gap,' we also indicate the utilisation level in the table, denoted by 'Util'.

5.5.1 Operator availability

The impact of operator availability is analysed in different ways. As we saw in Figure 2.6 in Chapter 2, operators are available for 40 hours per week and for two times 1.5 hours at the weekend. Pallets can only be loaded when an operator is available, so the impact of operator availability is interesting to research. The sensitivity analysis explores various aspects of operator availability, including the frequency of weekend visits, the timing, and duration of weekend visits, as well as the duration of weekday visits.

The first sensitivity analysis is on operator availability at the weekend. We test the frequency of coming at the weekend ranging from 2 (base case, Saturday and Sunday) to 0. This experiment aims to assist HTM Aerotec in determining the feasibility and impact of reducing or eliminating operator visits on weekends. When the visiting frequency is set to 1, we differentiate between visits occurring exclusively on Saturday or Sunday. In Table 5.11 the results of changing the operator visiting frequency at the weekend can be seen.

Table 5.11: Effect of operator visiting frequency in the weekend

	2 (Base)		1 (Saturday)			1 (Sunday)			0		
Instance	Obj	Util	Obj	Util	Gap	Obj	Util	Gap	Obj	Util	Gap
538-1	62.77	0.82	65.23	0.81	3.92%	65.48	0.81	4.32%	74.39	0.76	18.51%
538-2	120.60	0.87	126.86	0.83	5.19%	125.78	0.83	4.30%	133.89	0.81	11.02%
538-3	152.22	0.88	168.18	0.82	10.48%	167.12	0.83	9.79%	189.45	0.78	24.46%
539-1	88.92	0.88	112.42	0.82	26.43%	111.48	0.82	25.37%	129.67	0.73	45.82%
539-2	188.43	0.90	240.34	0.82	27.55%	238.57	0.82	26.61%	283.12	0.74	50.25%
539-3	329.97	0.90	450.12	0.82	36.41%	449.89	0.82	36.24%	487.12	0.75	47.63%

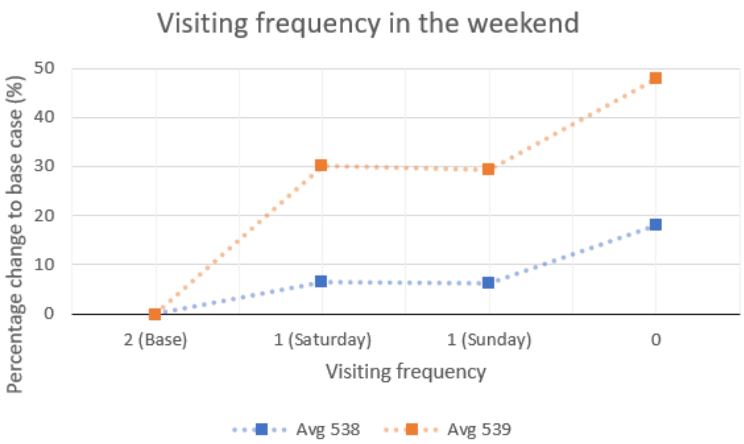

Figure 5.10: Effect visiting frequency at the weekend

From Table 5.11 and Figure 5.10 we can see that the number of visits at the weekend is very important, especially for machine 539. Based on this information, a visiting frequency of two is necessary for both machines. And in the case only one visit is possible, Sunday is on average preferred for a single visit at the weekend. For the bigger problem instances, the effect of visiting less is more apparent than for the smaller problem instances.

Besides the frequency of visiting at the weekend, we also do some sensitivity analysis on the time of visiting at the weekend. In this sensitivity analysis, an operator visits on both days at the weekend. The objective of this experiment is to determine whether the current visiting time is good and to explore the potential of alternative visiting times. For this we try three new settings, one is three hours earlier than the base case and the others are three and six hours later than the base case.

Table 5.12: Effect of operator visiting time at the weekend

	7.00-8.30			10.00-11.30 (Base)		13.00-14.30			16.00-17.30		
Instance	Obj	Util	Gap	Obj	Util	Obj	Util	Gap	Obj	Util	Gap
538-1	63.57	0.81	1.27%	62.77	0.82	60.77	0.83	-3.19%	60.20	0.83	-4.09%
538-2	119.78	0.87	-0.68%	120.60	0.87	119.29	0.87	-1.10%	118.13	0.88	-2.05%
538-3	154.12	0.87	1.25%	152.22	0.88	149.45	0.89	-1.82%	148.89	0.90	-2.19%
539-1	89.02	0.88	0.11%	88.92	0.88	84.58	0.89	-4.88%	82.58	0.90	-7.13%
539-1	191.43	0.89	1.59%	188.43	0.90	180.23	0.92	-4.35%	180.40	0.91	-4.26%
539-1	326.78	0.90	-0.97%	329.97	0.90	323.28	0.91	-2.03%	320.18	0.92	-2.97%

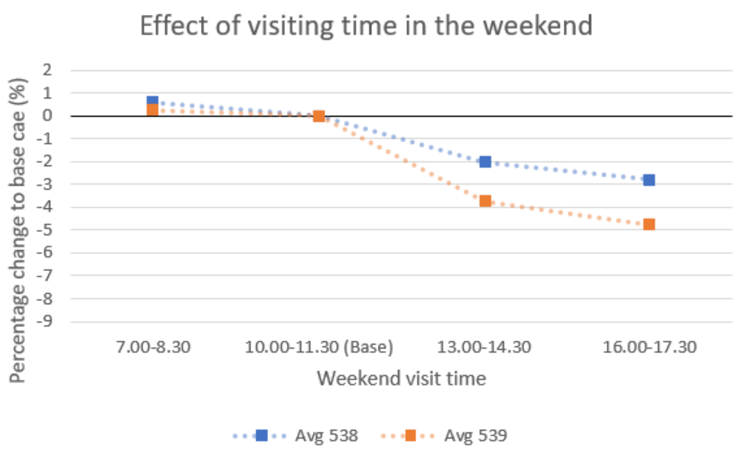

Figure 5.11: Effect of different weekend visiting time versus base case

From Table 5.12 and Figure 5.11 we can conclude that the current time of visiting at the weekend is not a good time. Visiting later on in the day gives much better results. This can be explained by the fact that the duration of operator non-availability is then better balanced. The most favourable time for visiting appears to be at the end of the afternoon.

The last experiment on operator availability at the weekend is on visiting length. The short visiting time at the weekend can be a bottleneck for the algorithm and in the real world. That is why we do a sensitivity analysis with three longer visiting lengths at the weekend, with steps of half an hour.

Table 5.13: Effect visiting length in the weekend

	Base		10.00-12.00			10.00-12.30			10.00-13.00		
Instance	Obj	Util	Obj	Util	Gap	Obj	Util	Gap	Obj	Util	Gap
538-1	62.77	0.82	60.56	0.83	-3.52%	58.48	0.84	-6.83%	56.89	0.85	-9.37%
538-2	120.60	0.87	118.78	0.87	-1.51%	115.89	0.89	-3.91%	112.34	0.91	-6.85%
538-3	152.22	0.88	149.23	0.89	-1.96%	146.79	0.91	-3.57%	144.68	0.92	-4.95%
539-1	88.92	0.88	83.45	0.90	-6.15%	81.89	0.91	-7.91%	80.02	0.92	-10.00%
539-2	188.43	0.90	186.23	0.90	-1.17%	181.49	0.91	-3.68%	178.24	0.92	-5.41%
539-3	329.97	0.90	326.88	0.91	-0.94%	324.01	0.91	-1.81%	316.48	0.92	-4.09%

As can be seen in Table 5.13 and Figure 5.12, the length of visiting at the weekend heavily constrains the model. For both machines, this has a similar effect. Being present at the weekend for 1.5 hours longer than

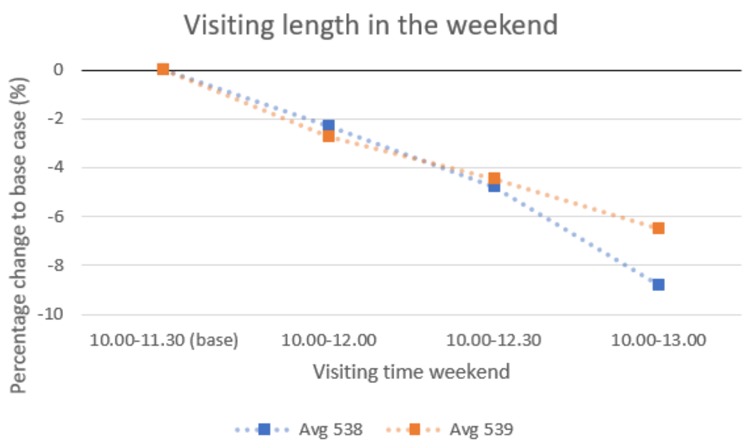

Figure 5.12: Effect of visiting longer at the weekend

usual already results in a decrease of roughly 9% and 6% on the objective value in the model for machines 538 and 539 respectively.

The last sensitivity analysis for operator availability is working hours during the week. For this, we see what happens if we increase or decrease the amount of hours available during the week. This experiment can help HTM Aerotec determine whether the current shift lengths are good, or alternatives are superior. The results can be seen in Table 5.14 and Figure 5.13.

Table 5.14: Effect of operator visiting time during the week

	6.30-16.30			7.00-16.00 (Base)		7.30-15.30			8.00-15.00		
Instance	Obj	Util	Gap	Obj	Util	Obj	Util	Gap	Obj	Util	Gap
538-1	61.01	0.83	-2.80%	62.77	0.82	68.89	0.80	9.75%	75.12	0.78	19.68%
538-2	118.89	0.88	-1.42%	120.60	0.87	124.01	0.85	2.83%	130.78	0.83	8.44%
538-3	149.02	0.90	-2.10%	152.22	0.88	158.89	0.86	4.38%	167.68	0.85	10.16%
539-1	86.56	0.90	-2.65%	88.92	0.88	92.33	0.86	3.83%	96.56	0.83	8.59%
539-2	186.78	0.90	-0.87%	188.43	0.90	200.30	0.87	6.30%	206.34	0.83	9.50%
539-3	324.56	0.91	-1.64%	329.97	0.90	333.40	0.88	1.04%	349.12	0.86	5.80%

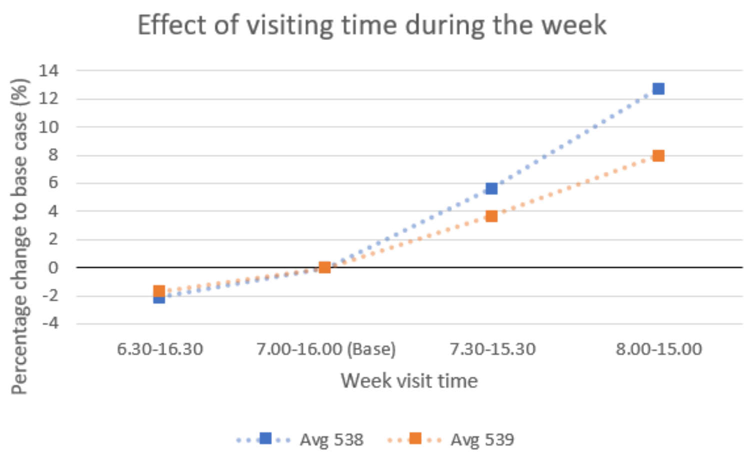

Figure 5.13: Effect of the different week visiting times versus base case

From Table 5.14 and Figure 5.13 we can conclude that it is not desirable to have the operators frequent fewer hours during the week. Therefore, it is not recommended to shorten the work days of the machine operators. Shortening the work days by up to two hours can reduce the performance of the machines by 8% (machine 539) to 14% (machine 538).

The sensitivity analyses on operator availability effectively show the effect of different aspects of operator availability on system performance. Cutting back on weekend visits has a negative effect on performance. Likewise, rescheduling weekend visits emphasizes how crucial it is to go later in the day for better results. Additionally, the analysis of weekday visiting hours highlights the negative consequences of reducing operators' workdays, emphasizing the necessity of preserving sufficient weekday coverage to prevent performance degradation. In summary, operator availability significantly influences system performance. While cutting back on availability hours is not beneficial in the current state at HTM Aerotec, adjusting visiting times, especially during weekends, has the potential for performance improvement.

5.5.2 Pallet capacity

In this subsection, the effect of the pallet capacity is tested. Based on this subsection we can determine whether the current installed pallet storage capacity is good, or an investment may be needed to enhance throughput. Currently, the 538 has 40 pallets and the 539 has 24 pallets. Furthermore, experimenting with the number of pallets available to the machines offers valuable insights into the role of pallet availability on the system's performance. Based on this sensitivity analysis, it can be determined whether the number of pallets the machines have available is a bottleneck in achieving more production hours per week.

Table 5.15: Effect of pallet capacity

	-4			-2			Base		+2			+4		
Inst	Obj	Util	Gap	Obj	Util	Gap	Obj	Util	Obj	Util	Gap	Obj	Util	Gap
538-1	66.56	0.80	6.04%	64.56	0.81	2.85%	62.77	0.82	61.01	0.82	-2.80%	60.58	0.83	-3.49%
538-2	125.68	0.85	4.21%	122.45	0.86	1.53%	120.60	0.87	117.13	0.88	-2.88%	114.39	0.90	-5.15%
538-3	162.45	0.85	6.72%	158.34	0.87	4.02%	152.22	0.88	150.01	0.89	-1.45%	146.78	0.91	-3.57%
539-1	96.40	0.84	8.41%	92.40	0.86	3.91%	88.92	0.88	83.28	0.90	-6.34%	80.03	0.91	-9.99%
539-2	208.34	0.84	10.56%	194.89	0.87	3.43%	188.43	0.90	183.89	0.91	-2.41%	179.78	0.92	-4.59%
539-3	367.89	0.85	11.49%	349.20	0.88	5.83%	329.97	0.90	324.89	0.92	-1.54%	316.23	0.93	-4.16%

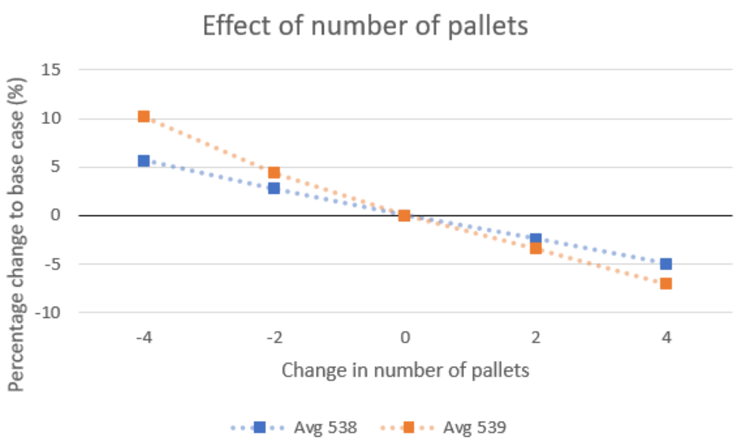

Figure 5.14: Effect of change in pallets versus base case

From Table 5.15 and Figure 5.14 we can see that having a few extra pallets would have been worth it for the initial investment. As expected, the effect is bigger on the 539 as the 538 has 16 more pallets in the base case already. Having four more pallets for machine 539 already increases the objective value by 7.5 percent and the utilisation is also expected to increase by two percent. Had HTM Aerotec initially opted for fewer pallets, the performance decrease would have been higher than the increase in performance gained from adding pallets more pallets initially compared to the base case.

5.5.3 Dedicated fixtures available

In this section, the effect of available dedicated fixtures is investigated to see how they affect system performance. Products can only be mounted on a pallet if the dedicated fixture for it is available. For instance, the restriction of having just one dedicated fixture available at a time limits the number of products that can be mounted on a pallet requiring that specific fixture to only one pallet. Based on this, it can be determined whether it is worth it to make extra dedicated fixtures for orders.

To investigate the influence of dedicated fixtures, dedicated fixtures are added to the orders that have a limited amount of these fixtures available. Experiments are done by adding one fixture, two fixtures, and removing the limit of dedicated fixtures. In Table 5.16 and Figure 5.15 the effect of a change in dedicated fixture availability can be seen.

Table 5.16: Effect of adding dedicated fixtures available

	Base		+1			+2			No limit		
Instance	Obj	Util	Obj	Util	Gap	Obj	Util	Gap	Obj	Util	Gap
538-1	62.77	0.82	60.67	0.83	-3.35%	60.24	0.83	-4.03%	59.97	0.83	-4.46%
538-2	120.60	0.87	118.68	0.87	-1.59%	117.46	0.88	-2.60%	116.89	0.88	-3.08%
538-3	152.22	0.88	151.01	0.88	-0.79%	150.43	0.88	-1.18%	149.45	0.89	-1.82%
539-1	88.92	0.88	83.45	0.89	-6.15%	81.23	0.90	-8.65%	79.78	0.91	-10.28%
539-2	188.43	0.90	182.34	0.91	-3.23%	180.03	0.91	-4.46%	177.34	0.92	-5.89%
539-3	329.97	0.90	314.18	0.92	-4.79%	303.29	0.93	-8.09%	298.34	0.94	-9.59%

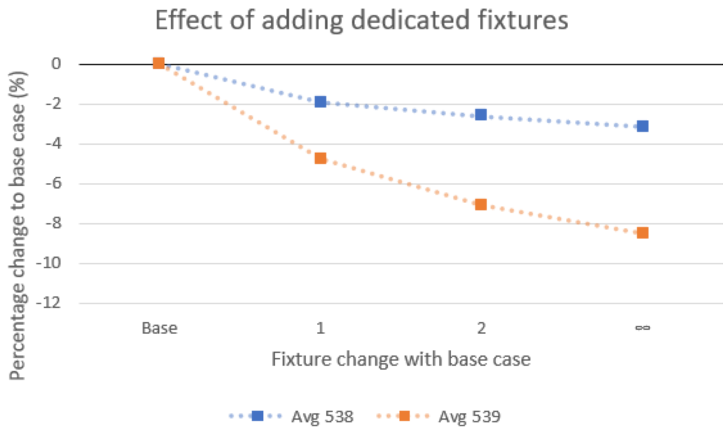

Figure 5.15: Effect of having more dedicated fixtures

For machine 539, dedicated fixture availability has more influence on system performance compared to machine 538. This can be explained by the fact that the 539 has more orders with a limit on dedicated fixtures, as can also be seen in Table 5.4. Therefore, it is beneficial to produce extra dedicated fixtures for orders of machine 539, especially if these orders are recurring and consist of many products.

5.5.4 Machine failure

In this subsection, we test what happens to the system performance if we make the planning more robust. Adding more robustness to the schedule can mitigate the impact of unexpected machine failures on the feasibility of the schedule. Making a more robust schedule with buffers for machine run times can lead to lower performance indicators in terms of efficiency and productivity, however, it significantly enhances the resilience of the planning process. To assess the impact of enhancing schedule robustness, we multiply the machine run time for each job by increments of 5%, ranging from 5% to 20%.

Table 5.17: Effect of making a more robust schedule by adding an extra buffer to machine running time

	Base		+5%			+10%			+15%			+20%		
Inst	Obj	Util	Obj	Util	Gap	Obj	Util	Gap	Obj	Util	Gap	Obj	Util	Gap
538-1	62.77	0.82	67.78	0.79	7.98%	70.23	0.77	11.88%	75.43	0.76	20.17%	79.21	0.74	26.19%
538-2	120.60	0.87	122.01	0.86	1.17%	127.59	0.84	5.80%	135.67	0.81	12.50%	143.12	0.77	18.67%
538-3	152.22	0.88	157.29	0.87	3.33%	166.57	0.84	9.43%	181.59	0.80	19.29%	190.12	0.79	24.89%
539-1	88.92	0.88	92.40	0.87	3.91%	98.79	0.85	11.10%	106.45	0.81	19.71%	111.23	0.78	25.09%
539-2	188.43	0.90	196.40	0.88	4.23%	218.29	0.84	15.84%	240.12	0.81	27.43%	260.12	0.79	38.05%
539-3	329.97	0.90	348.20	0.88	5.52%	371.49	0.86	12.58%	402.28	0.81	21.91%	437.83	0.79	32.69%

As can be seen in Table 5.17 and Figure 5.16, the impact of making the schedule more robust is significant, especially for machine 539. Adding a 20% safety factor to each job already decreases the utilisation by 10%. However, when making the schedule more robust the planning is more likely to be feasible when machine failure

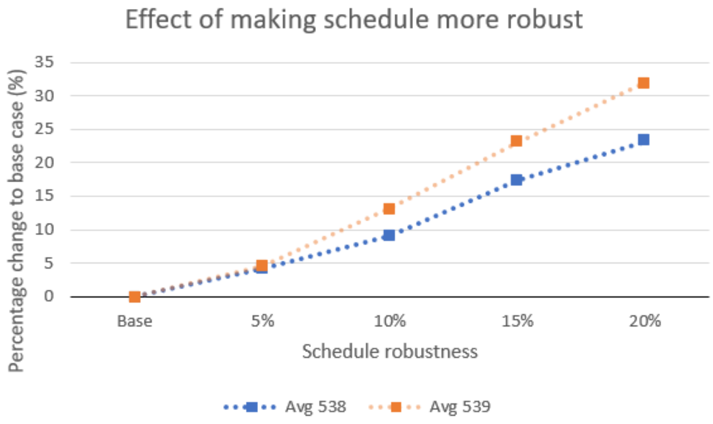

Figure 5.16: Effect of making planning more robust

occurs. Even with the inclusion of a 20% safety factor, the performance surpasses the 105 production hours per week. The number of production hours is still approximately 133 hours for both machines.

5.6 Algorithmic versus observed real-world performance

In the final experiment, we compare the algorithmic performance with the performance that was gotten over the last month for both machines. Since there is no data available on tardiness, we assess the real-world performance by examining the percentage of time the machines were operational each day, which allows us to determine the production hours achieved per week. For the algorithmic analysis, we use the base case with problem instance 538-3 for machine 538 and problem instance 539-3 for machine 539. We look at the first month of the results and retrieve the achieved production hours per week from that period.

Table 5.18: Production hours per week reached with algorithm and real-world

Machine	Real-world	Algorithm	Difference (%)
538	129.14	138.20	7.01%
539	141.60	149.68	5.71%

What we can conclude from Table 5.18, is that both machines had a really good month at the company, especially compared to performances gotten in the last 2.5 years (see Table 2.3). The algorithm was still capable of outperforming the real-world results. The 538 was outperformed by 7.01% and the 539 by 5.71%. If we take a closer look in-depth in the performance we can identify where the differences lie. In Figure 5.18 the utilisation levels over the month can be seen. Besides, in Table 5.19 we can see the percentage of time the machine is running in the algorithm and observed in real world for both machines.

Table 5.19: Percentage of day machining time per day of the week

Day	Real-world 538	Algorithm 538	Real-world 539	Algorithm 539
Monday	58.24	61.89	66.38	71.61
Tuesday	76.52	87.97	81.68	90.83
Wednesday	83.12	90.43	88.64	92.65
Thursday	81.90	93.33	89.20	94.11
Friday	75.78	96.41	90.83	91.41
Saturday	76.58	93.66	83.98	95.99
Sunday	69.93	48.80	72.25	81.25

Based on Table 5.19 we can see that the algorithm is not able to handle the end of the weekend going into Monday at the moment for machine 538. A drop in performance for machine 538 can be seen every Sunday in Figure 5.18. The percentages the machines are running are on the low side there. This can be explained by the fact that in the real-world the operators stay as long as the machines are filled up again, whereas in the model they have the exact window of 1.5 hours. This 1.5 hours is not enough in the model for the 538. To see what happens to the algorithm weekend performance we extend the weekend operator availability by one hour to

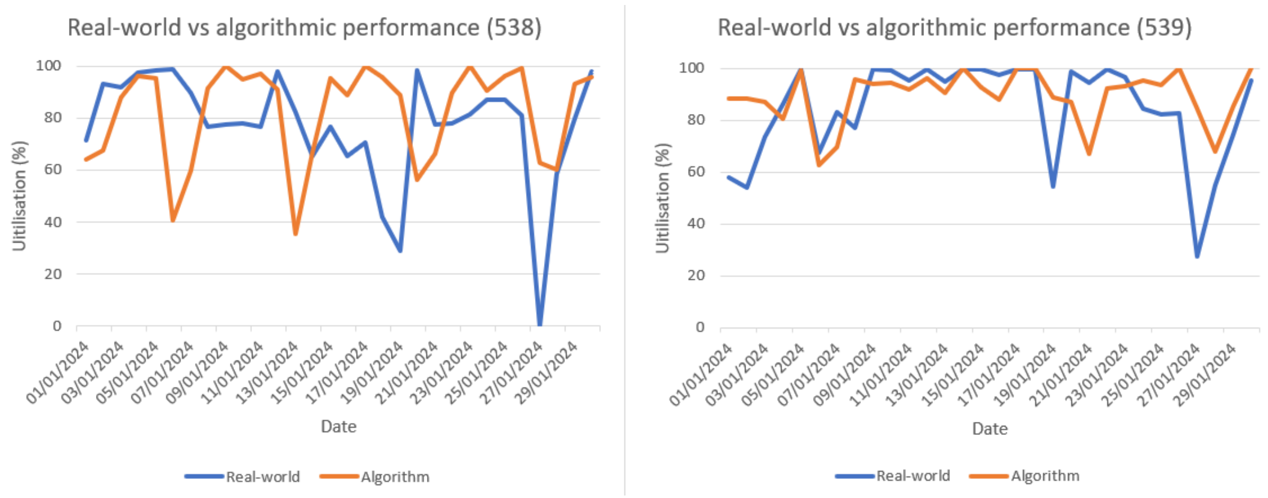

Figure 5.17: Real-world versus algorithmic utilisation in last month for both machines (base case)

10.00-12.30, similar to experiments done in Section 5.5.1. This results in the following. Having one hour extra at the weekend massively improves the performance of machine 538, whereas for machine 539 the difference is considerably smaller.

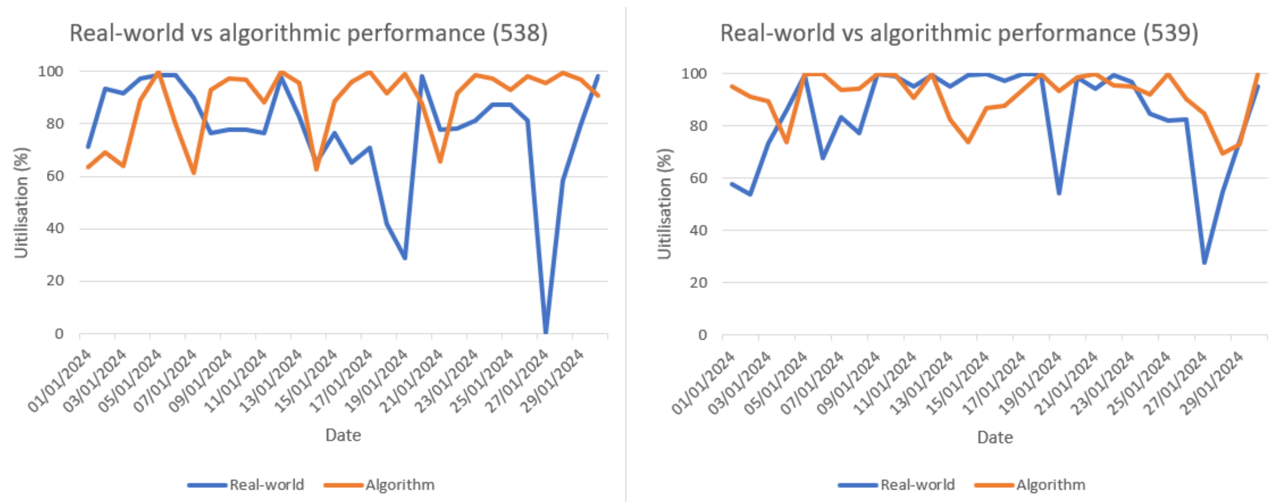

Figure 5.18: Real-world versus algorithmic utilisation in last month for both machines with 1-hour extra weekend visit length

Table 5.20: Production hours per week reached with algorithm and real-world with 1 hour extra visit length in weekend

Machine	Real-world	Algorithm	Difference (%)
538	129.14	148.51	14.99%
539	141.60	153.73	8.57%

5.7 Conclusion

The main goal of this chapter is to answer the research question: *What experiments can be done with the model to investigate the performance?* We have six problem instances in total. We have a smaller size, medium size, and big size problem instance for each machine with which we do all experiments noted in the experimental design. The experimental design first consists of finding input parameters of heuristics, used to find the best solution approach, which is used for the rest of the experiments. Afterwards, sensitivity analysis is described which is performed on operator availability, available pallets, dedicated fixtures, and robustness of schedules. The last experiment compares the performance of the algorithm with the real-time performance achieved in a month.

The algorithmic experiments identified EDD as by far the best dispatching rule for constructing an initial solution. With this constructive heuristic, the best trade-off between makespan and tardiness is gotten. Besides, due to a limit on run time, the algorithm is not able to overcome poor initial solutions generated by the other dispatching rules. Therefore, the EDD dispatching rule was used to compare the improvement heuristics and operator strategy. Consequently, we found that TS with random operator selection is the superior solution approach for all six problem instances. Hence, for the sensitivity analyses TS with random operator selection was used.

The sensitivity analysis provides interesting results. In terms of operator availability, we showed that frequency of visiting is the most important operator availability factor as not coming at the weekend can increase the objective value by up to 50%. Besides, visiting later on the day at the weekend provides better results than the current visiting time at the weekend for both machines. Weekend visiting length is of great importance for machine 538. The operator visiting time during the week significantly impacts system performance. However, visiting one hour less a day still provides decent results. The sensitivity analysis on dedicated fixtures available showed that they do not have a significant impact on machine 538. For machine 539 it may be wise to look into making some more dedicated fixtures, especially for big orders that have a limit on dedicated fixtures available. Making a schedule more robust means achieving a much lower makespan and less utilisation, however, does give more guarantee that the schedule is feasible in reality.

At last, the algorithmic performance of a schedule of one full month was compared with the real-world performance in the same month. The real-world performance was highly above average, compared to performance over the last years. The algorithmic performance still managed to provide better results than the real-world performance. The algorithm managed to achieve 7.01% more production hours for machine 538 and 5.71% more production hours for machine 539 in the base case.

6 Implementation

This chapter describes how the model we proposed can be implemented at HTM Aerotec and answers the fifth research question:

How can HTM Aerotec use the model?

First Section 6.1 describes which input is needed for the model to work. Section 6.2 describes how to easily run the model. At last Section 6.3 describes the output of the model and explains the dashboard. With the information presented in this chapter, HTM Aerotec can use the model to get useful insights into the performance of their machines, allowing for more informed decision-making.

6.1 Required input data

As described in Section 4.1, the main input for the model is the schedule list sent to production periodically. This list includes information on due dates, setup, machine running time, quantity to be produced, and more. However, certain essential planning details are still missing from the schedule list. Table 6.1 provides an overview of the additional input required in the schedule list for the model to work properly.

Table 6.1: Input data model

In schedule list	**Extra data required**
Due date	Setup required
Quantity	Routesteps & operations per item
Item number	Labour time
Total hours per item	Release date
Setup time	Dedicated fixtures available
Run time	Number of products on single pallet

To obtain the additional data necessary for the model, some manual actions were undertaken. This process is time-consuming but can be expedited. Information on the number of route steps and corresponding operations per item is available in the ERP system (Glovia G2). Currently, these are aggregated, and the total runtime is summed, but this data needs to be split up. Labour time is already available in the ERP system within the same line for each order. The determination of setup requirements and release dates, however still requires manual input.

Gathering the data for the number of dedicated fixtures and the number of products that fit on a single pallet for each order is currently the most time-consuming task. Identifying the number of dedicated fixtures involves reviewing the work instructions for a specific item to determine the fixture in use. The work instructions also provide information on the number of products that can fit on a single pallet. Currently, there is no efficient method for looking this up.

Given that the majority of orders are recurring, a more effective strategy would involve expanding the order line data within the ERP system. This expansion would encompass details regarding specific fixtures and the quantity of products that can be accommodated on a single pallet. The number of route steps for each item and operation should be recorded on distinct lines. The only manual input necessary would be to indicate if setup is needed and provide the order's release date.

6.2 Running the model

The model has been made in a Python interpreter. However, (most) people in the company have no to very limited experience using Python. This makes the model by itself unusable for the company. By producing an executable file, the usability, and accessibility of the Python model are improved significantly. All Python scripts are contained in this executable file, which makes it easy for non-technical people to use the model. This method simplifies the execution process by removing the need for the planner to install or configure Python. A tutorial on how to correctly set up and run the executable file, as well as retrieve the output, has been written for the company and provided to them.

With the executable file, the person can handle the input data by making changes to the Excel file that is in the same directory. Running the executable results in some output, which is stored in an Excel file as explained in Section 6.3.

6.3 Model output

From the model, output is generated with various statistics, as Figure 6.1 depicts. The output includes information for each order based on the final schedule resulting from the optimisation model. The statistics contain information on the start and completion times of production for products within a production order, order lead time (measured in days), the percentage of orders that fall behind schedule, and the average delay in days for each order. Besides, on the right side of the figure, a ranking is made based on the worst-performing items in terms of order lead time, percentage later, and average days late. For each order a bottleneck score is calculated based on a normalisation of the three previously mentioned statistics, assigning a score of 1 to orders with the worst performance and 0 to those with the best performance. This approach offers valuable insights into bottleneck items, allowing for a good assessment of schedule feasibility and being able to set good due dates.

Item	Operation	Qty.	Total hours	Due date	Start production	End production	Order lead time	Late (%)	Average days late
101759	1	40	14,80	19/01/2024	08/01/2024	18/01/2024	9,82	0	0
103117	1	40	16,57	19/01/2024	08/01/2024	19/01/2024	11,23	10,53	0,05
139826	1	55	19,67	19/01/2024	08/01/2024	18/01/2024	10,62	0	0
139826	2	55	9,07	19/01/2024	08/01/2024	19/01/2024	11,24	13,21	0,06
103140	1	50	34,75	26/01/2024	09/01/2024	30/01/2024	21,24	20	0,44
103140	2	50	38,50	26/01/2024	10/01/2024	01/02/2024	22,36	24,44	1,04
103140	3	50	38,50	26/01/2024	10/01/2024	02/02/2024	22,98	26,67	1,16
103140	4	50	31,00	26/01/2024	10/01/2024	02/02/2024	22,98	26,67	1,22
139157	1	30	29,00	26/01/2024	09/01/2024	30/01/2024	20,42	14,81	0,42
139157	2	30	6,50	26/01/2024	10/01/2024	30/01/2024	20,1	14,81	0,49
139981	1	48	122,00	26/01/2024	11/01/2024	31/01/2024	19,55	22,92	0,48
139981	2	48	74,00	26/01/2024	12/01/2024	31/01/2024	19,6	22,92	0,89
101606	1	102	43,67	01/02/2024	11/01/2024	06/02/2024	26,78	30	1
100937	1	62	13,50	09/02/2024	17/01/2024	13/02/2024	27,21	21,67	0,66
139824	1	50	61,17	19/02/2024	25/01/2024	23/02/2024	29,64	18,37	0,58
139824	2	50	53,00	19/02/2024	25/01/2024	24/02/2024	29,8	20,41	0,64
101730	1	30	3,67	23/02/2024	25/01/2024	25/02/2024	30,77	20	0,18
100872	1	40	10,00	25/02/2024	31/01/2024	25/02/2024	25,02	5	0,03
102042	1	65	77,00	29/02/2024	01/02/2024	29/02/2024	28,62	5	0,02

Item	Operation	Bottleneck Score	Order lead time	Late (%)	Average days late
101606	1	2,713232358	26,78	30	1
103140	4	2,655766767	22,98	26,67	1,22
103140	3	2,606586439	22,98	26,67	1,16
103140	2	2,413205096	22,36	24,44	1,04
139824	2	2,199251158	29,8	20,41	0,64

Item	Operation	Order lead time	Item	Operation	Average days late
101730	1	30,77	103140	4	1,22
136985	2	29,97	103140	3	1,16
139824	2	29,8	103140	2	1,04
136985	1	29,69	101606	1	1
101687	1	29,68	139981	2	0,89

Item	Operation	Late (%)
101606	1	30
103140	3	26,67
103140	4	26,67
103140	2	24,44
139981	1	22,92

Figure 6.1: Output of model

More detailed output is also presented. In Figure 6.2, a part of the Gantt chart for the first day in the schedule of machine 538 for the first 10 pallets can be seen. The blue bars represent the labour the operators have to perform. In the time that is depicted in the blue bar, the previous product needs to be taken off the pallet and the new product needs to be placed on the pallet. The information on each bar is the Work Order number (which has been changed for this example), the operation (in Dutch "**B**ewerking"), and in which place in the machine sequence this product should be placed. Even more in-depth information about the exact labour times and production times of each product can be seen as well in the output of the Excel sheet.

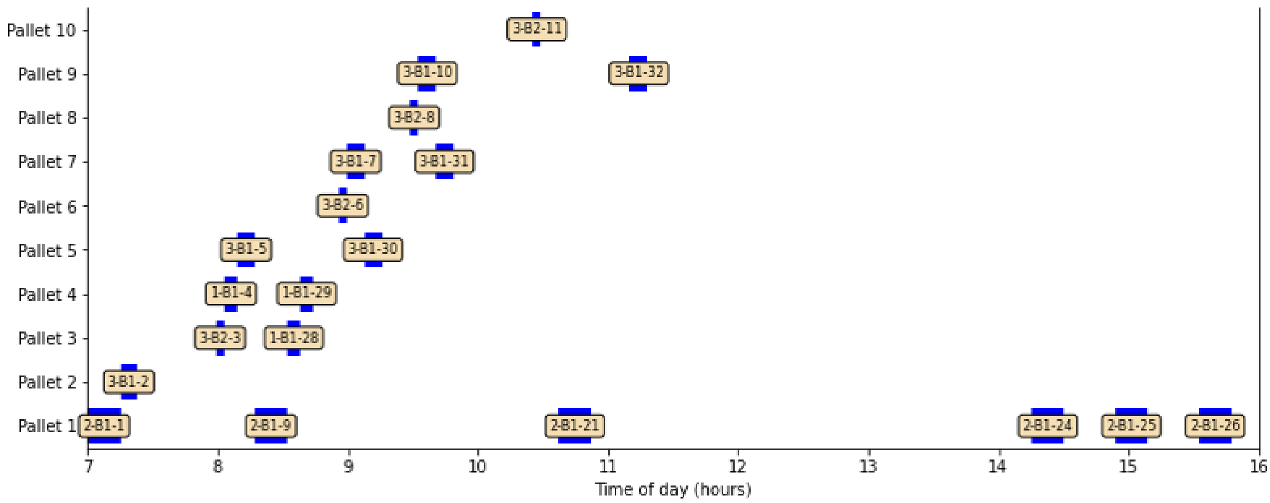

Figure 6.2: Gantt chart for loading of pallets

7 Conclusions & recommendations

In this final chapter, the last research question is answered:

What conclusions and recommendations can be made to HTM Aerotec?

Section 7.1 summarises the main findings and answers the main research question. Section 7.2 provides the recommendations to HTM Aerotec. Section 7.3 discusses the limitations of the research and provides directions for future research. At last, Section 7.4 describes both the academic and practical contribution of the research.

7.1 Conclusion

In this research, solution approaches for scheduling the 5-axis milling machines at HTM Aerotec were explored. Currently, the company does not have a sophisticated planning approach for the machines. Based on the work provided, we can answer the main research question: *How can an optimised scheduling strategy be developed to consistently achieve the goal of reaching more than 105 production hours per week for the 5-axis milling machines?*

The scheduling problem has several practical constraints. Resource constraints are on operator availability, dedicated fixtures, and pallets for the machines. Besides, each product type has its characteristics and constraints. A scheduling algorithm has been programmed in Python to construct a schedule, adhering to all the constraints, based on a given input sequence. For optimising this unique single-machine scheduling problem, 12 solution approaches were proposed, based on information obtained in the literature review. Each solution approach consists of the following steps:

- Generate an initial solution based on a chosen dispatching rule. Three dispatching rules are proposed: EDD, multi-factor, and completely random. In this case, randomness had to be introduced to improve initial solution quality. Adding randomness also meant that the risk of an initial bad solution has to be mitigated by generating an initial population size P_0 of 35 solutions. 35 solutions is a sufficient amount to ensure a good initial start objective value and a low standard deviation of the objective value of the initial solution.
- Improve the initial solution with an improvement heuristic with a neighbourhood strategy. The effectiveness of two improvement heuristics (SA and TS) and two neighbourhood strategies (random and VN) have been evaluated. For both improvement heuristics, input parameters are tuned to ensure a good trade-off between objective value and model runtime.

The best solution approach is to use EDD as the dispatching rule for generating the initial solution and improve the initial solution using TS with random operation selection. The EDD dispatching rule gives the best trade-off between makespan and tardiness. Due to the complexity of the scheduling problem, the algorithm is not capable of overcoming the worse solutions of the other initial solutions generated by the other dispatching rules in a reasonable time span. Using the solution approach, the initial goal of consistently reaching more than 105 production hours per week was easily reached. This can be seen in Table 7.1.

Table 7.1: Improvement of production hours reached, compared with the initial goal

Machine	Production hours reached	Production hours goal	Difference (%)
538	136.98	≥ 105	30.46%
539	149.12	≥ 105	42.02%

The production hours per week increased by 30.46% and 42.02% for machines 538 and 539 respectively. Besides an improvement in production hours, meaningful insights into the performance of the machines and the factors that influence the performance have been obtained. The effect of operator availability during the weekend and the week has been studied. Generally, reducing working hours is not preferable, whether it's during the weekdays or over the weekend. Additionally, scheduling visits later in the weekend tends to yield better outcomes on average. The effects of other resource constraints like pallet availability and available dedicated fixtures have also been studied. For both of these, machine 539 is more sensitive as it has fewer pallets available, and relatively more products are placed on dedicated fixtures.

An executable file has been made, with which the employees for the company can run the model, based on the input they provide in the accompanying Excel file. The output of the model is converted to provide usable information. Insights can be obtained on a higher level, such as when to start producing the first product of an order, the expected lead time of the order, and expected tardiness. More detailed insights can be extracted from the model output, including the sequencing of products for production on the machine, the ideal timing for loading each product, and the pallet allocation for product placement.

7.2 Recommendations

In this section we provide HTM Aerotec with some practical recommendations:

- Visiting twice at the weekend is necessary. If only one weekend visit is possible, visiting on Sunday is slightly preferred. Not visiting at the weekend decreases the performance of machine 538 by just under 20% and for machine 539 this is almost 50(!)%
- Have the operators visit later on the day at the weekend. Visiting at a later time in the weekend improves the system performance significantly. Visiting at 16.00-17.30 instead of 10.00-11.30 improved the system performance of the 538 by three percent and the 539 by six percent.
- For big, recurring orders on machine 539 it is beneficial to add extra dedicated fixtures.
- Use the model to gain more insight into the feasibility of the due dates on the schedule list, lead time of orders, and to identify the bottleneck orders.
- Experiment with operators using the Gantt charts with the loading times to see if this can improve system performance
- Improve the detail of the schedule lists by including route steps & operations on the same machine for each item, labour time, release dates, dedicated fixtures available, and the number of products that fit on a pallet. Based on this information alone it can already be easier to estimate how long an order is going to take.
- Keep machine status more up to date. The insight into the quantity of products that still need to be finished from a production order is subpar. If the insight on quantities still left to produce is not there, it is also not possible to make an exact correct schedule.
- The workload on the machines is higher than they can manage (especially machine 539), so looking for more machines and/or outsourcing is recommended. The scheduled workload on machine 539 for the first 12 weeks of the year is 2058.80. This means that on average 171 production hours per week need to be reached, which is infeasible.

7.3 Limitations and future research

This section provides the limitations of the research and gives directions for future research. The first limitation is that the scope is limited to 5-axis milling machines themselves and surrounding machines are not considered. For this, we assumed that everything of an order is available after the release date, whilst in reality, these machines can be dependent on other machines that come earlier in the production process.

The planning model starts with an empty system on a Monday at 7.00 AM, meaning that none of the pallets are filled at this point. An empty system on a Monday is not very unrealistic, as most items on the pallets have been machined over the weekend and are to be loaded off the pallets on Monday first thing in the morning. However, the planning model currently can not handle starting on a Wednesday for example. Another point to consider is that given that the machines start with an empty system, dynamic rescheduling is not possible and is an interesting topic for future research.

Machine and tool failure are not directly considered when making the planning. One of the main reasons the operator gives for the lower amount of production hours achieved than desired are failures out of their control. We did conduct a sensitivity analysis on schedule robustness, however, future research could be done on how machine and especially tool failure affect the scheduling strategy.

In this research, we did not consider tool changes as part of a planning constraint. This is because of the big tool capacity in the machines and the fact that nobody at the company considered the tool changes to be a constraint. However, as the paper by Dang et al. (2023) showcases, tool changes can have an impact on making a schedule for these types of machines in some cases.

Determining more exactly when and how long to visit for each weekend in the planning could be an interesting direction for future research. The time to visit at the weekend significantly impacts the performance of the system, so research specifically on this subject could be of value to HTM Aerotec.

7.4 Contribution of the research

In this section, the theoretical and practical contribution of the research is described. The scheduling problem researched is a unique research. To the best of our knowledge, the paper of Shin et al. (2019) comes closest in terms of machine characteristics and the paper of Dang et al. (2023) comes closest in terms of operator (non-) availability scheduling constraints. However, both do not include the combination of the operator availability, pallets, dedicated fixtures and product constraints this research has. We made a mathematical model representing the problem, which was not done before for this type of problem. Additionally, we tested the performance of 12 different solution approaches for six different problem instances.

The practical contribution for HTM Aerotec lies in the insights they get into the system. The feasibility of a schedule list can be tested when using it as input to the model. Based on this, information can be obtained about which orders are most likely to be late and what the expected internal lead time of the order is at the company. The current system, Factory Planning, cannot generate plans that are as realistic as those produced by the model. Therefore, this model represents a significant upgrade for HTM Aerotec, enabling more informed, data-driven decision-making. Besides, the output for the model also provides a plan for the operators which shows when and where to start loading a product. Sensitivity analysis on various factors shows the impact of certain aspects on system performance, which they can influence.

References

Amir Mohammad Fathollahi-Fard, Kannan Govindan, M. H.-K. A. A. (2019). A green home health care supply chain: New modified simulated annealing algorithms. *Journal of Cleaner Production 240*, page 11820.

Barros, F. J. (2016). On the representation of time in modeling simulation. *Winter Simulation Conference.*

Baskar, A. and Xavior, M. A. (2014). Optimization of makespan in job and machine priority environment. *Procedia Engineering 97*, pages 22–28.

Behnamian, J. (2013). Decomposition based hybrid vns-ts algorithm for distributed parallel factories scheduling with virtual corporation. *Computers Operations Research*, pages 181–191.

Chen, C.-S., Mestry, S., Damodaran, P., and Wang, C. (2009). The capacity planning problem in make-to-order enterprises. *Mathematical and Computer Modelling 50*, pages 1461–1473.

Chen, Y., Zhang, A., and Tan, Z. (2013). Complexity and approximation of single machine scheduling with an operator non-availability period to minimize total completion time. *Information Sciences 251*, pages 150–163.

Dang, Q.-V., Herps, K., Martagan, T., Adan, I., and Heinrich, J. (2023). Unsupervised parallel machines scheduling with tool switches. *Computers Operations Research 160.*

de Athayde Prata, B., de Abreu, L. R., and Lima, J. Y. F. (2021). Heuristic methods for the single-machine scheduling problem with periodical resource constraints. *TOP*, pages 524–546.

de Weerdt, M., Baart, R., and He, L. (2021). Single-machine scheduling with release times, deadlines, setup times and rejection. *European Journal of Operational Research 291*, pages 629–639.

Eiben, A. and Smith, J. (2015). *Introduction to Evolutionary computing.* Springer.

Figielska, E. (2009). A genetic algorithm and a simulated annealing algorithm combined with column generation technique for solving the problem of scheduling in the hybrid flowshop with additional resources. *Computers Industrial Engineering 56*, pages 142–151.

Hans, E., Herroelen, W., Leus, R., and Wullink, G. (2007). A hierarchical approach to multi-project planning under uncertainty. *omega 35*, pages 563–577.

Heerkens and van Winden (2017). *Solving Managerial Problems Systematically.* Noordhoff 1st edition.

Herr, O. and Goel, A. (2016). Minimising total tardiness for a single machine scheduling problem with family setups and resource constraints. *European Journal of Operational Research 248*, pages 123–135.

Kirkpatrick, Gelatt, V. (1983). Optimization by simulated annealing. *Science*, pages 671–780.

Lalla-Ruiz, E., Heilig, L., and Voß, S. (2020). Assessing simulated annealing with variable neighbourhoods. *Learning and Intelligent Optimization: 13th International Conference*, pages 298–303.

Lee, D.-K., Shin, J.-H., and Lee, D.-H. (2020). Operations scheduling for an advanced flexible manufacturing system with multi-fixturing pallets. *Computers Industrial Engineering 144.*

Low, C., Yip, Y., and Wu, T.-H. (2006). Modelling and heuristics of fms scheduling with multiple objectives. *Computers Operations Research 33*, pages 674–694.

Martinelli, R., Mariano, F., and Martins, C. (2022). Single machine scheduling in make to order environments: A systematic review. *Computers Industrial Engineering 169.*

Mohamed Abdel-Bassat, Laila Abdel-Fatah, A. K. S. (2018). Metaheuristics algorithms: a comprehensive review. *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications*, pages 185–231.

Naderi, B., Zandieh, M., Balagh, A. K. G., and Roshanaei, V. (2009). An improved simulated annealing for hybrid flowshops with sequence-dependent setup and transportation times to minimize total completion time and total tardiness. *Expert Systems with Applications 36*, pages 9625–9633.

Olhager, J. (2010). The role of the customer order decoupling point in production and supply chain management. *Computers in Industry 61*, pages 863–868.

Ouelhadj, D. and Petrovic, S. (2009). A survey of dynamic scheduling in manufacturing systems. *J Sched.*

Pinedo, M. (2016). *Scheduling.* Springer Cham Heidelberg.

PM-Group (2022). Pm history. Accessed September 12, 2023.

Rader, D. (2010). *Deterministic operations research.* John Wiley Sons.

Rapine, C., Finke, G., Brauner, N., and Lehuox, V. (2012). Single machine scheduling with small operator non-availability periods. *Journal of Scheduling.*

Ruiz, R. (2015). *handbook of heuristics.* Springer International publishing.

Safarzadeh, H. and Niaki, S. T. A. (2019). Bi-objective green scheduling in uniform parallel machine environment. *Journal of Cleaner Production 217*, pages 559–772.

Saniuk, A. and Waszkowski, R. (2016). Make-to-order manufacturing – new approach to management of manufacturing processes. *Materials Science and engineering 145.*

Sharman, G. (1984). The rediscovery of logistics. *Harvard Business review 62*, pages 71–80.

Shin, H., Kim, C., and Kim, S. (2002). A tabu search algorithm for single machine scheduling with release times, due dates and sequence-dependent setup times. *Int J Adv Manuf Technol*, pages 859–866.

Shin, J.-H., Yu, J.-M., ho Doh, H., Kim, H.-W., and Lee, D.-H. (2019). Batching and scheduling for a single-machine flexible machining cell with multi-fixturing pallets and controllable processing times. *International Journal of Production Research.*

Sörensen, K., Sevaux, M., and Glover, F. (2018). A history of metaheuristics. *Handbook of Heuristics.*

Tan, K.-C., Narasimhan, R., Rubin, P. A., and Ragatz, G. L. (2000). A comparison of four methods for minimizing total tardiness on a single processor with sequence dependent setup times. *Omega 28*, pages 313–326.

Wemmerlov, U. (1992). Fundamental insights into part family scheduling: The single machine case. *Decision Sciences*, pages 565–595.

Wojakowski, P. and Warzolek, D. (2014). The classification of scheduling problems under production uncertainty. *Research in Logistics Production*, pages 245–255.

Wu, L. and Cheng, C.-D. (2016). On single machine scheduling with resource constraint. *J Comb Optim*, pages 491–505.

Zeestraten, M. (1989). *Scheduling Flexible Manufacturing systems.* PhD thesis, TU Delft.

Zhang, C., jiyin Liu, wah Wan, Y., Murt, K. G., and Linn, R. J. (2003). Storage space allocation in container terminals. *Transportation Research Part B 37*, pages 883–903.

Zheng, F. and Jin, K. (2019). An improved heuristic for single machine lot scheduling problem. *IFAC Papers-Online 52-12*, pages 217–222.

A Overview solution approach

The figure below shows an overview of the general solution approach we take.

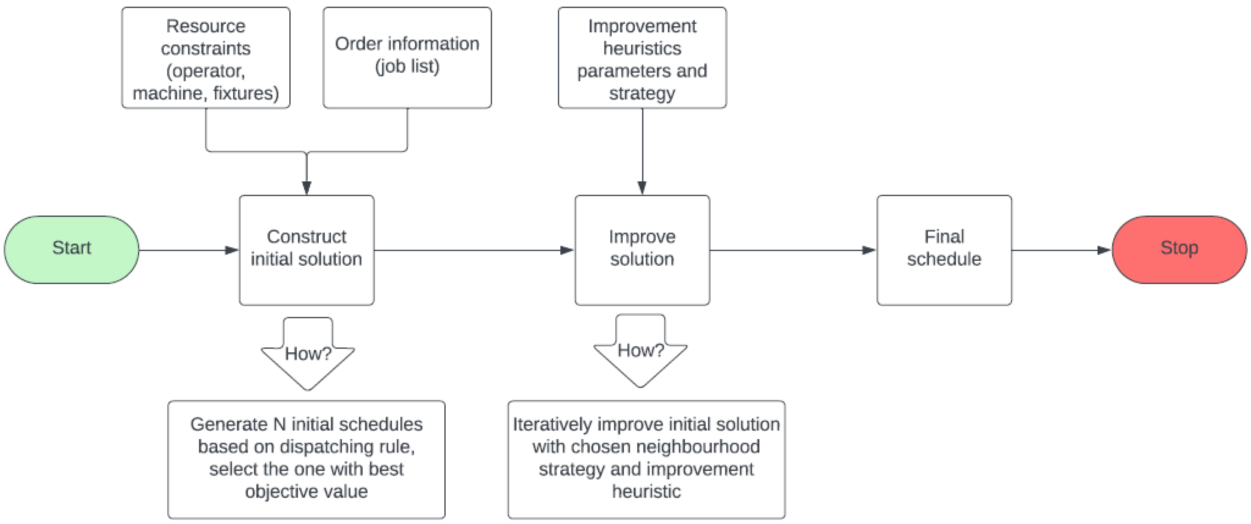

Figure A.1: Overview solution approach

B Parameter tuning

B.1 Objective function weight tuning

In Equation 1 we have the objective function with a weight w. In order to find the value we use for w, 100 schedules for both machine 538 and 539 were generated. For all these schedules we calculated the objective value for w values ranging from 0.1 to 0.9 with a step of 0.1. For each step the best best objective value was selected along with its corresponding makespan and average tardiness values. The makespan and tardiness values can be seen in Table B.1. Based on this we can see that there is no difference between 0.1 and 0.6 for both instances. For machine 538 a different solution was chosen for w of 0.7 to 0.9. This however has a less desired solution with quite a lot more tardiness. Therefore, we choose a weight w of 0.1 as this also results in the lowest objective value, making it easier on the eye.

Table B.1: Objective weight tuning

w	Machine	Makespan	Average Tardiness	Machine	Makespan	Average Tardiness
0,1	538	1610,8	136,13	539	2287,46	387,36
0,2	538	1610,8	136,13	539	2287,46	387,36
0,3	538	1610,8	136,13	539	2287,46	387,36
0,4	538	1610,8	136,13	539	2287,46	387,36
0,5	538	1610,8	136,13	539	2287,46	387,36
0,6	538	1610,8	136,13	539	2287,46	387,36
0,7	538	1605,82	145,66	539	2287,46	387,36
0,8	538	1605,82	145,66	539	2287,46	387,36
0,9	538	1605,82	145,66	539	2287,46	387,36

B.2 Improvement heuristic parameter tuning

The first step for the **SA parameter tuning process** is determining the starting temperature. We obtain the temperature by solving problem instances 538-3 and 539-3 (see 5.1). We do this for the largest problem instances of both machines to see if we need problem-specific parameters or not. We use the multi-factor dispatching rule to construct the initial solution. We start with a temperature of 100 for both. We solve the instances with a Markov chain length of 100 and a cooling factor alpha of 0.8. We stop when the temperature reaches 1. The big decrease factor of 0.8 was chosen to reduce computational time. For each temperature, at the end of each Markov chain we store the acceptance ratio. This indicates the number of worse neighbours accepted divided by the number of worse neighbours proposed. Figure B.1 provides the results of this.

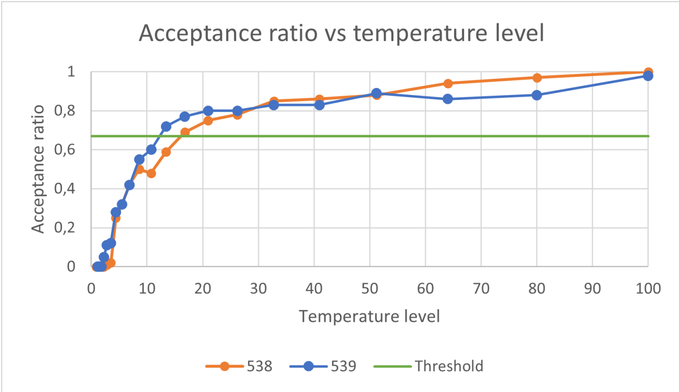

Figure B.1: Acceptance ratio versus temperature level

Based on Figure B.1, we choose a starting temperature of 15 for both the 538 and the 539, which have an acceptance ratio close to 0.67. We choose to not start with a temperature that has an acceptance ratio closer to 1 as the constructive heuristics already provide decent initial results. A higher starting temperature can then result in significantly worse solutions at the start, which may take a lot of iterations to overcome. Besides, due to the relatively high computational time, choosing a lower temperature prevents excessively high computational times.

Next, we have to consider doing experiments with stopping temperature, Markov chain length and decrease factor alpha. For stopping temperature we consider the values 2.5, 5 and 7.5. For Markov chain length 5, 10, 15. For the decrease factor alpha we consider 0.9, 0.925 and 0.95. Table B.2 shows the full factorial experiment including the objective value and run time. From this table we can select the parameter setting that result in a good trade-off between objective value and running time. For getting the results in the table the multi-factor dispatching rule for the problem instance of machine 539 was used. From the table, we conclude that the experiment with stopping temperature 5, decrease factor alpha of 0.9, and Markov Chain length of 15 led to the best trade-off between objective value and run time.

Now the TS parameters have to be found. To find the tabu list size and the number of iterations we do experiments. Since we do 70 iterations for SA, we experiment with TS with iteration numbers close to this. For tabu list length we experiment with 5, 10 and 15. For the Number of iterations we experiment with 50, 75, 100, 125, 150 and to see if a much bigger amount of iterations makes a difference we try 300 and put it in a convergence figure. We solve the problem instance of machine 539 with the multi-factor dispatching rule. Based on Table B.3 we choose max iterations of **125** and tabu list size of **10**.

Table B.2: SA Cooling scheme

StopTemp	Alpha	MarkovChainLength	Objective	Run time
2,5	0,9	5	558.07	1146.02
2,5	0,9	10	556.72	2320.97
2,5	0,9	15	555.28	3448.02
2,5	0,925	5	557.94	1446.60
2,5	0,925	10	553.93	2965.78
2,5	0,925	15	553.18	4411.04
2,5	0,95	5	556.53	2272.99
2,5	0,95	10	554.40	4480.52
2,5	0,95	15	554.08	6656.32
5	0,9	5	555.78	888.44
5	0,9	10	555.16	1383.85
5	**0,9**	**15**	**553.78**	**2065.05**
5	0,925	5	559.60	942.07
5	0,925	10	557.61	1879.82
5	0,925	15	555.44	2826.75
5	0,95	5	556.69	1380.55
5	0,95	10	555.46	2758.63
5	0,95	15	553.40	4145,71
7,5	0,9	5	560.88	436.57
7,5	0,9	10	560.01	901.35
7,5	0,9	15	559.60	1382.53
7,5	0,925	5	560.83	592.22
7,5	0,925	10	556.53	1184.97
7,5	0,925	15	554.08	1782.15
7,5	0,95	5	559.46	920.84
7,5	0,95	10	555.73	1845.35
7,5	0,95	15	553.86	2768.40

Table B.3: Tabu list size experiment

Max iterations	Max Tabu list Size	Objective	Run time
50	5	554.88	631.04
50	10	554.12	630.81
50	15	554.01	640.05
75	5	554.12	948.97
75	10	553.45	947.99
75	15	553.79	1005.28
100	5	552.81	1277.47
100	10	552.67	1275.54
100	15	553.10	1264.58
125	5	552.12	1586.61
125	**10**	**552.01**	**1587.97**
125	15	552.46	1589.32
150	5	552.06	1881.89
150	10	552.07	1871.67
150	15	552.29	1867.92