UNIVERSITY OF TWENTE.

Master Thesis

Improving Product Flow in a Cable Manufacturing Facility through a Buffer Allocation Strategy

by

ir. C. Wilbers

Industrial Engineering and Management
Specialization Production and Logistics Management
Orientation Manufacturing Logistics
Faculty of Behavioural, Management and Social Sciences

Examination committee
Dr. ir. B. Alves Beirigo
Dr. ir. S. M. Meisel
University of Twente

External supervision
ir. J. T. Oude Vrielink
Company X

2024

Preface

Dear reader,

Before you lies the master thesis "Improving Prodcut Flow in a Cable Manufacturing Facility through a Buffer Allocation Strategy". It has been written to meet the graduation requirements of the Industrial Engineering & Management program at the University of Twente, Enschede. I was engaged in researching and writing this thesis from November 2023 to May 2024.

During this research, I learned several new skills, both practical and social. Throughout this period, I spent most of my time at the company that encountered the problem that poses as the subject of this thesis. By fullfilling a role as a production planner on the side I was able to experience the problem firsthand. In addition, I learned how to deal with the IT infrastructure and supply chain software of the company.

I would like to thank my company supervisor ir. Jordy T. Oude Vrielink for his guidance and expertise throughout the process. His insights regularly helped improving the solution quality as well as increasing the usability of the model. Furthermore, I would like to thank my educational supervisors Dr. ir. Breno Alves Beirigo and Dr. ir. Stephan M. Meisel for their supervision during this research.

Finally, I would like to thank you, my reader, I hope you enjoy your reading.

Christian Wilbers
Enschede
June 13, 2024

Abstract

Addressing Company X's complex supply chain challenges involving the Buffer Allocation Problem (BAP), this thesis presents a tailored solution to improve the product flow and corresponding throughput of the manufacturing facility. The BAP is a formidable problem that affects both the total output per unit of time, the number of production orders that are delivered on time, and the total amount of space that is occupied by work-in-progress (WIP). This study proposes a digital model that utilizes optimization algorithms in order to determine the optimal allocation of available buffers while interpreting real-time data. Ultimately, the goal is to increase the throughput while taking space limitations and other constraints into account. The presented model includes a hybrid algorithm that consists of both a generative part and an evaluative part that work together in order to find the best performing allocation. This is all done whilst taking space limitations, flow constraints, and buffer type restrictions into account. Consequently, the tool can be very helpful to the people occupied with production planning at the facility regarding decision making. It provides insight in where currently buffers are allocated through real-time data while also acting as a guideline for where the buffers should be. In addition, the large number of parameters that can be changed enables the user to simulate various scenarios and get an idea of how the buffer allocation would change under different circumstances. As a result, the flexibility of the manufacturing system is increased due to this increase in anticipation level. Besides reviewing these different circumstances, it is also possible to determine what parameters affect the system performance the most. This gives the company additional insights and can therefore contribute to developing a strategy that is concerned with what variables should be improved. In conclusion, this research provides an additional tool to the supply chain operations happening at TKF and improving its market position. A successful implementation of this tool has the potential to significantly improve the throughput of Company X's manufacturing facility while also providing additional insights and the ability to envision other scenarios.

Management Summary

This management summary provides an overview of this thesis, which addresses the Buffer Allocation Problem (BAP) and specifically the variant encountered at the Company X. Below are the key aspects and findings highlighted.

Objective

The primary objective of the research is to develop a tailored solution for optimizing the allocation of available buffers among the machinery, with a focus on maximizing the total throughput. In addition, the company wants to get insight on how to distribute the buffers in a more effective manner.

Key Challenges

1. **Planning Strategy**: The current strategy is mainly based on where the buffers are located at that moment in time without additional reasoning.

2. **Complex Network**: The machinery configuration and different production sequences of the products results in a complex machine network that is difficult to analyze.

3. **Space limitations**: The total available space for the storage of work-in-progress (WIP) limits the amount of buffer that can be allocated per machine.

4. **Material Handling**: There is a limited number of available cable reels. In addition, different machines require different reel types.

Proposed Solution

This thesis proposes a comprehensive solution that utilizes optimization techniques and a real-time data connection to address the BAP for Company X.

Solution Components

1. **Generative Algorithm**: A custom algorithm is created based on existing methods that is able to generate new viable solutions.

2. **Evaluative Method**: A manner through which the user is enabled to determine the throughput given a certain buffer allocation.
3. **Data Connection**: Implementation of a direct data link enables the user to directly retrieve information from the company's supply chain software.
4. **Adjustable Settings**: Integrated adjustable parameters allow the user to simulate various scenarios.
5. **Dashboard**: Visualization of the results provides insight for the production planner on how to steer the production operations.
6. **Performance Metrics**: An overview of all individual performance metrics for each machine, such as utilization.

Expected Outcomes

Implementation of the proposed solution is expected to result in:

- Significant increase in throughput, potentially in the range of 10-20% extra cable reels per unit of time.
- Variables that can lead to an additional increase in throughput will be exposed.
- The time taken up by the planning process is reduced.

Challenges and Risks

- Making decisions purely on the model in order to achieve a certain buffer allocation could go against one's instincts.
- Production orders with high priority could interrupt the product flow and disrupt the buffer allocations.
- Some parameters are based on historical data or predictions and may therefore not be accurate for the future, such as processing times.

Findings

- The model indicated an improvement of at least 17% for all tested algorithms.
- The improved buffer allocation could lead to an increase in generated value of $\sim$€614,286 per week.
- The runtime of the model remains under a minute for all algorithms.
- Improving the overall equipment efficiency and scheduled times for all machines could lead to an even further increase in throughput.

Next Steps

- An implementation plan will need to be set up in order to exactly determine to what level the model is decisive.
- The people who can benefit from this tool will need to be taught on how to use it and how it is built.
- Resource allocation approval needs to be secured.

Executive Summary Highlights

- This thesis proposes a tailored solution to address the BAP at Company X, emphasizing throughput maximization, and increased operational efficiency.
- Potential benefits include a significant increase in throughput, opportunities for further throughput improvements, and reduced time taken up by production planning.

Conclusion

In order to retain or improve the market position of Company X it is important to increase the throughput level with its current resources. This allows the company to sell more products per unit of time without significant additional costs. The potential benefits outweigh the challenges associated with implementation due to its user-friendliness.

Contents

List of Tables

List of Figures

List of Algorithms

Acronyms

AGV automated guided vehicle. xvii, 25

AI artificial intelligence. xvii, 73

BAP buffer allocation problem. ix, x, xiii, xv, xvii, 1–7, 12, 16–22, 24, 26–35, 40, 41, 43–49, 53, 55, 63, 66, 67, 69, 71, 73, 79, 80

CRMH-BAP capacitated and restricted machine hours buffer allocation problem. xvii, 30, 36, 40, 46, 71–73

ERP enterprise resource planning. xvii, 15, 33, 34, 50, 55, 72

Holding Y Holding Y of company X. xvii

KPI key performance indicator. xvii, 15, 19, 44, 49, 62, 72

MTO make to order. xvii, 15

MTS make to stock. xvii, 15

OEE overall equipment efficiency. xvii, 5, 20, 30, 33, 36–38, 46, 51, 56, 57, 61–64, 67, 69, 71, 72

RC reel conversion. xvii, 38, 61–63, 69

ST scheduled time. xvii, 38, 61–63

WIP work-in-progress. xvii, 2–4, 7, 13, 15, 18, 19, 23, 32, 34, 36, 45, 49, 52, 53, 64, 66

X Company X. ix, x, xiii, xvii, 1–7, 9, 11, 12, 14, 16, 17, 19, 20, 24, 27, 29–31, 33, 34, 36, 40, 41, 43–46, 50, 53, 55, 57, 58, 63, 64, 66, 67, 69–73

Chapter 1

Introduction

In this introductory chapter, a comprehensive overview of the context, objectives, and structure of this Master's thesis is given. Section 1.1 discusses the background and context of the research, addressing the significance of optimizing buffer allocations in current-day manufacturing operations. The problem statement, outlining the specific obstacles faced by Company X (X) as well as their intentions are stated in Section 1.2. The research objectives are stated in Section 1.3, providing a clear guideline for the individual goals of this study. The main research question and its derived sub-questions are presented in Section 1.4, aligning them with the content of all further chapters in this report. In addition, the significance of this study is emphasized by highlighting several benefits of having improved buffer allocations in Section 1.5, whereas the scope and limitations are addressed in Section 1.6. Lastly, Section 1.7 summarizes the organization of this report, giving readers an overview of the subsequent chapters and sub-chapters, including the literature review, problem formulation, and research methodology.

1.1 Background and Context

X is a well-established company in the Netherlands and is occupied with the production of connectivity solutions such as electric cabling but also glass fibre cabling. For this research, the main focus is towards the production of electricity cables. Various production steps are involved with the manufacturing of these cables. These will be explained in Section 2.1. However, it is confronted with various challenges that affect the daily operations related to the product flow in the factory. These challenges involve determining the actual buffer sizes, distributing the workforce effectively, optimizing machine schedules, and ensuring on-time delivery of production orders. To tackle these challenges, this Master's thesis is aimed at developing a model that can solve the unique variant of the buffer allocation problem (BAP) that X is faced with while considering corresponding constraints and requirements. The BAP in a manufacturing context is a strategic challenge that requires careful consideration of production processes, fluctuations in demand, and operational constraints. In this case, a buffer is defined as a stock of semi-finished products that are waiting to be processed by a certain machine or production line. Such buffers are used to adjust for variations in production processes. For example, should a previous production step fall behind schedule then the subsequent process could just continue processing products from its buffer. Consequently, the entire manufacturing process runs more smoothly. Allocating the right buffer sizes contributes significantly to achieving lean manufacturing, ensuring the timely delivery of products while minimizing waste and operational costs. Consequently, the BAP is an optimization problem that aims to find the best solution between a trade-off of conflicting objectives.

A buffer size can be seen as the number of semi-finished products that are waiting to be processed by a certain machine. A buffer size can also be measured in time, such as the number of processing hours in which the machine can be active before it runs out of semi-finished products should the previous machine break down for example. When placing the BAP in the context of X, a buffer can be seen as the amount of semi-finished cable length that is waiting to be processed by a specific machine. Cables are stored most of the time on cable reels, these can be seen in Figure 1.1. The buffer size could therefore be measured in the number of cable reels waiting for a machine or the total processing time of waiting semi-finished cables.

1.2 Problem Statement

The primary objective of this research is to find the optimal buffer sizes in between each production stage at the manufacturing facility of X in City A and, specifically, the energy department. The exact problem can be formulated as follows:

The main reason for X to determine optimal buffer sizes is to have a target value for the amount of work-in-progress (WIP). Currently, there are many cable reels occupied with a cable requiring processing however, this leads to a shortage of empty cable reels which are also necessary before a process can start. In addition, the reels take up a lot of space and make it sometimes difficult for forklift operators to manoeuvre around these objects or even find the correct cable reel. An example of a storage area being over-packed with WIP reels can be seen in Figure 1.1. In addition, having large amounts of WIP in between production steps means that the throughput time also increases. Consequently, the company would like to reduce the amount of WIP to create a more efficient product flow.

Figure 1.1: Temporary storage area that is overflowing with cable reels (image is an example and is not the actual facility).

1.3 Research Objectives

The research objectives of this thesis are as follows:

- To analyze and model the BAP that exists at X, taking into account factors, such as available space, manufacturing topology, and related costs.
- To develop a buffer size optimization algorithm that can solve the BAP at X, to reduce WIP while minimizing costs and improving the percentage of on-time deliveries.
- To test the proposed solution using relevant real-time production data provided by X and compare its performance against the current situation.
- To provide practical recommendations and insights to X based on the results obtained, in order to reduce the WIP and improve the percentage of on-time deliveries.

1.4 Research Questions

To guide this research and not lose track of the goal a set of research questions has been formulated. The following research questions are formulated:

1.4.1 Main Research Question

By considering all the characteristics of the manufacturing facility of X in City A, an optimization model to determine buffer sizes for all machines is developed. This results in the following main research question:

- What are the optimal buffer sizes for each production line while maximizing the expected production output at the manufacturing facility of X?

1.4.2 Sub-Research Questions

To answer this question several distinct aspects need to be determined beforehand. Aspects that are of importance for determining the buffer sizes are; current production system performance, optimization objective, optimization algorithms, validation method, and the corresponding constraints and requirements. Therefore, the following five sub-questions:

- How does the current production system perform?
  - What does the current manufacturing layout look like?
  - What KPIs are used to measure the performance?
  - At what rate do new orders arrive?
  - What are the machine characteristics?
- What constraints and requirements should be taken into account?

    - What are the limiting factors?
    - What are the requirements from X?
- What objective should be focused on?
    - What KPIs are available and can be used?
    - What KPI does X value the most?
- What optimization algorithm should be used?
    - How are the different algorithms assessed?
    - What tools or software should be used?
- How are the results going to be validated?
    - What does the test method look like?
    - What variables are going to be assessed?

The goal of this research is to determine the optimal buffer sizes for each production line. These buffer sizes will act as a guideline for the production planning upon which certain decisions can be based. For example, if a certain machine will be scheduled for the upcoming shifts.

1.5 Significance of the Study

As was mentioned in the previous sections, a decrease or even increase in the amount of WIP can have several benefits. Firstly, the average throughput time of a cable will be reduced since there will be less WIP waiting before each machine. Consequently, the total waiting time is reduced and thus the throughput time. Secondly, the holding cost will also decrease because less space is occupied for temporary storage of semi-finished products. Thirdly, the factory floor will be a lot more organized and structured, which means that it will be a lot more simple to keep an overview of where everything is located. Lastly, it could also occur that in some cases the WIP needs to increase to reduce the chance of machines starving out. Meaning that there are no cable reels in the buffer left that require processing by that specific machine. Besides the benefits of a reconfiguration of the WIP level, determining the optimal buffer sizes also has benefits for other obstacles. For example, it helps in adjusting for production variations in previous processes to ensure a smooth throughput. Consequently, it also helps in guaranteeing delivery dates with more certainty.

In Section 1.1 various challenges that are encountered by X were mentioned besides the allocation of buffer sizes, such as distributing the workforce effectively, optimizing machine schedules, and ensuring on-time deliveries. Of these challenges, the allocation of optimal buffer sizes is regarded as the most highly valued by the company. Solving the BAP variant of X automatically also contributes to solving the other challenges. Once the optimal buffer sizes are known it becomes more simple to determine which machines should be running and which machines can afford to miss a shift. Furthermore, the machine schedules can be optimized more easily when the required buffers are known. The production planner has a better overview of what will arrive at the next machine in the upcoming shifts and as a result, it is less complex to determine which buffers should be replenished and with what products. Lastly, the percentage of on-time deliveries can also increase once the optimal buffers are known since this should lead to a smoother flow through the factory. Therefore, X values the challenge of buffer allocations the most.

From the available literature, it became clear that there is not a specific roadmap that can be followed to tackle this problem. In addition, it is possible to approach this problem from various fields of expertise, such as simulation and queueing theory. What is missing in the research is a simplified model that can correlate buffer sizes to the performance of the system. Moreover, there is no existing model for a machine network that takes into account the overall equipment efficiency (OEE) as well as the scheduled hours per machine. This research aims to fill this gap supported by the case study for the manufacturing facility of X. The reviewed literature can be found in Chapter 3. Besides creating a reliable model, it should also be user-friendly and efficient such that it could easily be integrated into the daily operations of a production facility.

1.6 Scope and Limitations

It is important to mention both the scope and the limitations of this research. The goal of this research is to provide the company with a model that can calculate an optimal or near-optimal buffer allocation that can be applied in the company's manufacturing department. The model does not include precise production schedules or layout optimization. In addition, the costs, turnover, or other monetary KPIs will not be included and are regarded as out of the scope of this research. The aim is to provide a simple-to-use tool in which the optimal buffer sizes are determined. However, it could be unavoidable that the model requires simplifications or additional assumptions due to the complex nature of the problem. Furthermore, the lack of available data could also play a role which may require additional presumptions. Another possible limitation is the software available to the company meaning that only a select amount of programs can be used through which the tool can be built and implemented. Furthermore, the model is developed for the manufacturing system during the writing of this report. Consequently, should the production facility change due to new or extra machines for example then this could mean that the model should be modified significantly. These factors affect the generalizability of the model.

1.7 Thesis Structure

The remainder of this thesis is structured as follows:

- **Chapter 2: Problem Context** - This chapter provides additional information about the company, the manufacturing system, and the general production processes involved with producing electric cables.
- **Chapter 3: Literature Review** - In this chapter, an in-depth review of the existing literature is provided that elaborates on the BAP and its variants as well as various solution approaches.

- **Chapter 4: Problem Description** - The fourth chapter formally defines the BAP variant at X, identifying its parameters, constraints, and objectives.
- **Chapter 5: Solution Approach** - This chapter presents the algorithms and methodology developed to address the problem, as well as the required data integration and implementation details.
- **Chapter 6: Experimental Setup** - Here, the experimental setup and procedure are discussed together with the process of data collection and parameter tuning.
- **Chapter 7: Results and Discussion** - In this chapter, the results are analyzed and discussed, while practical recommendations are made based on the experimental findings.
- **Chapter 8: Conclusions and Recommendations** - The final chapter highlights the key findings, contributions, future research directions, and final remarks.

1.8 Summary

This chapter has introduced the background and context of the BAP that is present at X. In addition, the problem statement is elaborated upon in which the complications for X are addressed. This Master's thesis aims to solve these problems with the presented research objectives and answer the corresponding research questions. Furthermore, the significance of the study is indicated by showing the multiple significant benefits that could be acquired when this problem is solved. In the following chapters, more details of the BAP at X are elaborated upon and a solution methodology is proposed.

Chapter 2

Problem Context

In this chapter, the company is introduced and the challenges in their version of the BAP are explored. In the ever-changing landscape of cable manufacturing, companies are continually challenged to optimize their production processes to meet the increasing demands of the market. One aspect that significantly affects the overall efficiency of cable production is the strategic allocation of buffers within the manufacturing facility. Buffers, in this context, refer to a designated amount of WIP to manage fluctuations in throughput rates and ensure a smooth product flow. The BAP arises when a company tries to find a balance between maximizing throughput and minimizing costs. To tackle this problem effectively a thorough understanding of the production processes and manufacturing layout is of importance (Section 2.1). This introduction sets the stage for an elaboration of the BAP, as specific challenges faced by TKF are addressed and that need to be taken into account to increase operational efficiency, reduce costs, and ultimately ensure the company's competitive position in the market (Section 2.2). Combining these challenges creates a specific variant of the BAP that is distinct to X (Section 2.3). In this complex manufacturing environment, it is essential to apply advanced optimization strategies to allocate buffers based on real-time production data and other relevant factors. Solving this version of the BAP for X would be another step in establishing a more efficient product flow in which throughput times are reduced and the available workforce is distributed over the machines in the most effectively.

2.1 Company X: Background and Operations

X is an active company in regions of the Netherlands, Belgium, and Luxembourg regarding the manufacturing of various types of cables. They are known for their flexibility as well as their eye for detail concerning quality. X owns three different facilities from which electric cables are produced. These are located in City A, City B, and City C. The facilities in City B and City C are mostly occupied with producing cables intended for subsea applications such as offshore wind farms. These facilities are out of the scope of this research, which will be focused on the facility located in City A and one production cell specifically. The facility in City A is concerned with the manufacturing of cables intended for various purposes. The product portfolio ranges from medium voltage cables to cables used in the marine sector including vessels.

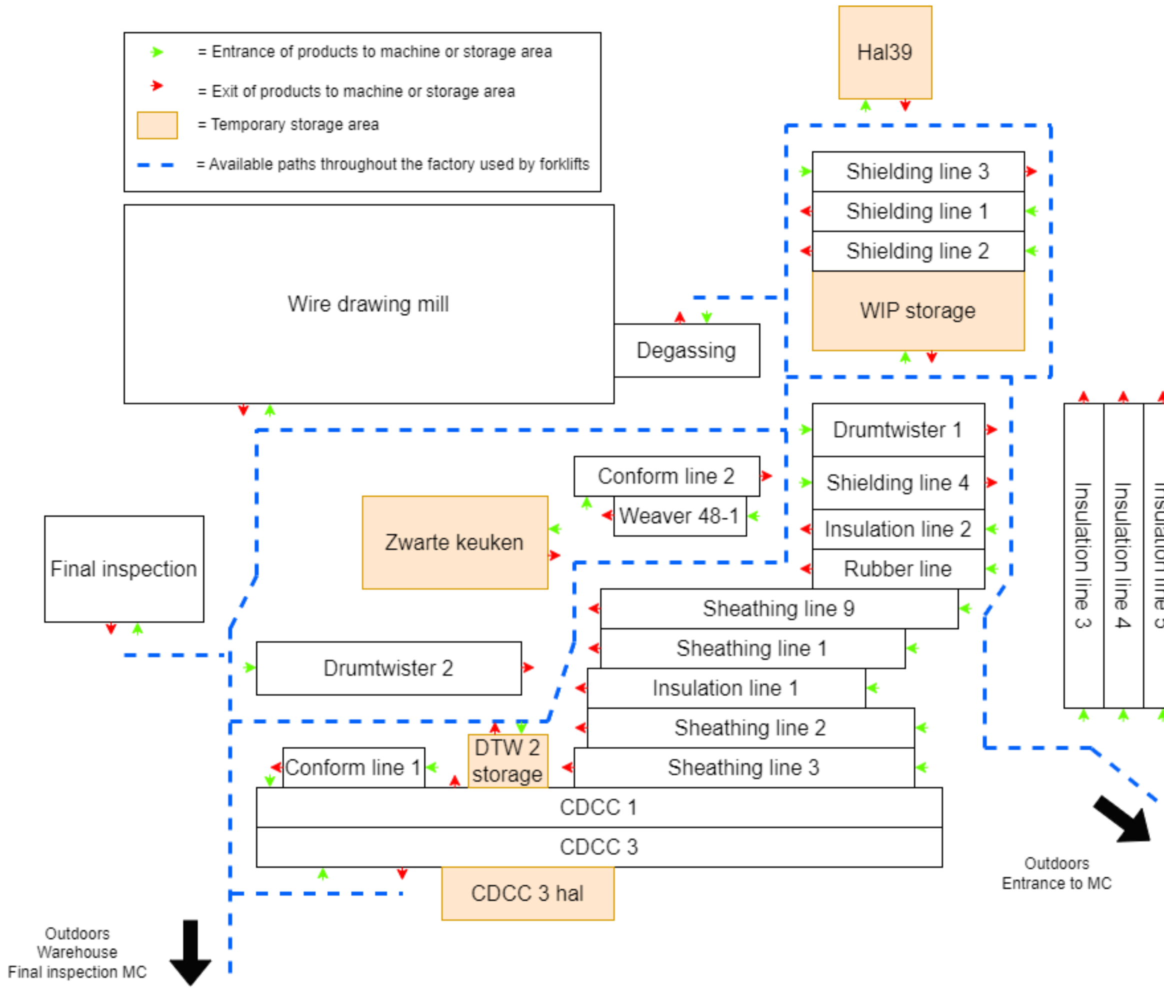

Figure 2.1: Schematic representation of the concerning manufacturing layout.

The company is offering a large range of cable types to the customer. This is also one of the causes of the complexity of the task at hand. In Figure 2.1 a floor plan of the production department can be seen as well as where all machines are located. The department is responsible for producing the larger cable types regarding diameter. It mostly produces medium voltage cables, however large diameter high and low voltage cables are also being manufactured in this department. In general, there are always certain production steps involved with the manufacturing of electric cables that are the same. These steps are visualized in Figure 2.2.

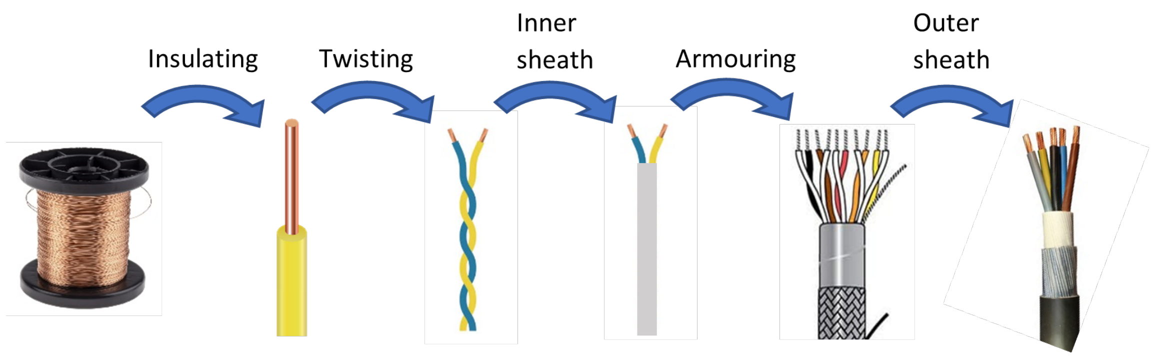

Figure 2.2: Production steps involved with the manufacturing of electric cables.

2.1.1 Cable Production Processes

Electricity cables play a major role in the current living standard of people worldwide. With their ability to transport electricity from one place to another, they increase the quality of living substantially. Without electricity cables one would require batteries or generators everywhere they go and power plants would not be able to exist. Electricity cables come in all shapes and sizes, ranging from subsea cables that need to conduct tens of thousands of volts while at the other hand, your phone requires a significantly smaller charging cable. For this research, the energy department of the manufacturing facility of X in City A is of interest. As a result, the cables that are made there will be elaborated upon.

Every electricity cable always consists of at least two elements. The first one is the conductor which is responsible for conducting the electricity. The other element is the insulation layer that prevents discharges to the surroundings from happening when handling the cable while it is conducting electricity. The cable configuration becomes more complex when more than one conductor is involved, different materials are required, or the cable has to cope with harsh environments. Examples of harsh environments are at the bottom of the sea, in surroundings with a low or high pH, or in places with a high risk of the cable being cut. These examples would require additional insulation with corrosive resistance for example or armouring is required such that a person or animal cannot simply cut or damage the cable. The most general elements of an electricity cable are visualized in Figure 2.3.

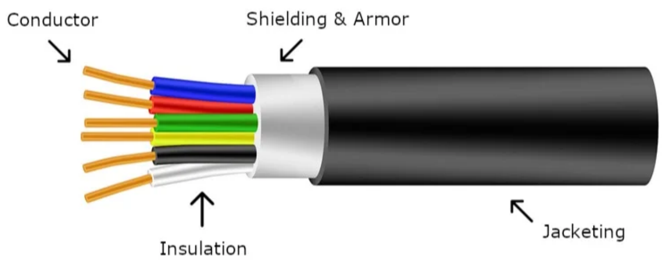

Figure 2.3: Most common elements used in electrical cables Simcona [2023].

Conductor

When manufacturing an electricity cable, the production process starts with the bare conductor. The conductors used at X are either made out of aluminium or copper. While copper has better electric conductivity (factor 1.57), aluminium is a lot cheaper (factor ∼3). Consequently, even if a cable would require more aluminium than copper it could still be cheaper to have aluminium. X buys aluminium and copper in the shape of wire with a standard diameter. To get the desired conductor specifications the material first has to be processed. The copper wire is processed in the wire drawing mill, which is another department and is out of the scope of this research. The aluminium is processed by the conform extrusion lines which soften the material through friction and then force it through a die with the desired diameter. The process is visualized in Figure 2.4.

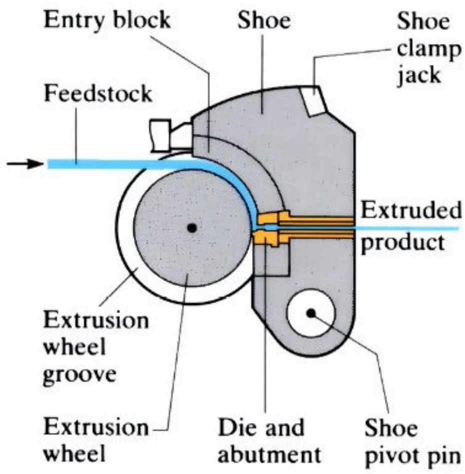

Figure 2.4: Visualization of the conform extrusion process The Open University [2017].

Insulation and Twisting

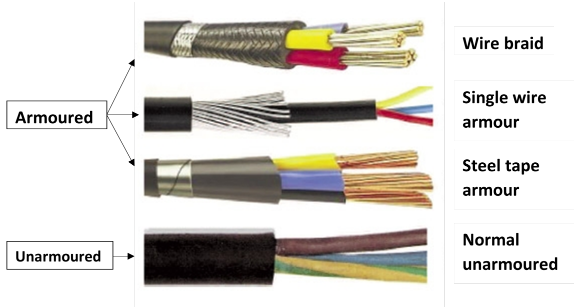

Figure 2.5: Different types of cable armouring Miko Wong [2017].

After the conductor has been manufactured to specification, the next step is to insulate each individual conductor. The purpose of this insulation layer is to prevent electrical charges from jumping out of the conductor to its surroundings, such as people or other cables. In the latter case, there is the possibility of shortening the circuit. Most electricity consists of more than one insulated conductor and the most common configuration includes three insulated conductors namely; earth, neutral, and a live wire. These insulated conductors are not simply bundled after which an outer layer is applied to hold them together. The individual insulated conductors are wounded around each other in a process called twisting. A pair of twisted insulated conductors can be seen in Figure 2.2. The reason for twisting these insulated conductors is to prevent residual stresses inside the conductors. One could imagine if a cable was being placed with turns in only one direction and the insulated conductors were not twisted then the outer conductor would need to cover more distance. As a result, this conductor experiences more tensile stress, increasing the risk of tearing. Once the conductors are twisted the cable is not very suitable for further processing since its outer surface is not smooth but consists of twisted wires. Consequently, the twisted conductors are first fed through an extruder which applies material, often rubber, all around and when the conductors exit the extruder head a smooth singular cable is left over.

Armouring and Outer Sheath

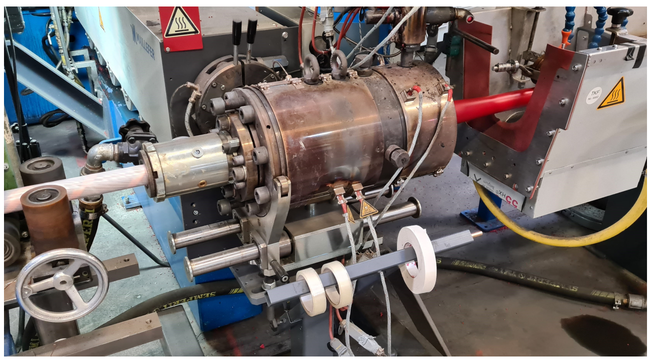

Figure 2.6: Extrusion of the outer sheath for a medium voltage cable.

After the filler material has been applied and the cable has a smooth surface it is fit for further processing. An optional armouring layer could be applied to the cable. This armouring layer has various purposes. Firstly, it acts as a protection layer reducing the ease with which the cable can be cut or damaged. Secondly, it strengthens the cable such that the mechanical stress on the conductor is reduced. However, this is only the case for conductors that are woven, solid conductors can handle more stress. Lastly, the armouring acts as an earthing screen. Should a cable that is placed underground get damaged during excavation for example this armouring ensures that the cable is still grounded. The last step before the cable is finished is always the outer sheath or jacket. This outer sheath does not necessarily have a functional purpose besides indicating the type, length, and other properties of the cable. The outer sheath is applied in a similar way to the insulation layer of the conductors. The cable is fed through an extruder head that applies a layer, often polyethylene, of material around the outside of the cable. This process can be seen in Figure 2.6. The thickness of this layer depends on the size of the cable. Generally, when the diameter of the conductors increases the thickness of the outer sheath also increases. The thickness is often between one and five millimetres.

2.2 Challenges

As a company, X desires to deliver every order on time and according to the highest possible quality standard. However, certain challenges arise when looking at the product flow through the factory. The following challenges are encountered in its daily operations.

2.2.1 Space Limitations

The first challenge encountered is the amount of available space at the manufacturing facility. A large portion of the available factory floor is taken up by all the machines that are required to meet the required throughput. In addition, some of the area is also designated for paths used by forklifts and employees. Consequently, little space is available for temporary storage which needs to be taken into account when determining buffer sizes for each production step.

2.2.2 Material Handling Constraints

In addition to the space limitations, the number of available cable reels is also a constraint. All different processes have a certain set of cable reels on which the processed cable can be winded. If one of these cable reel types should be completely loaded with cables, then it would be possible for production to stagnate due to a lack of empty cable reels. Having enough available cable reels at all times will pose a challenge.

2.2.3 On-time Deliveries

Having production orders finished on time is another challenge that is a consequence of having too much work-in-progress stored between certain production steps. Together with the previously two mentioned challenges there is also the challenge of getting production orders on time and even better would be to get it at the correct time window to prevent any unnecessary holding costs. The challenge is to find a balance in optimal buffer sizes while still ensuring on-time deliveries.

2.2.4 Flow Network Complexity

The range of different cables that are being offered by X in their product portfolio is very wide. Consequently, there are many different production sequences possible through the department. While this large product portfolio can satisfy most customer orders it does result in very complex product flows. Each of these product flows will need to be taken into consideration when determining the buffer sizes.

2.3 The Buffer Allocation Problem at X

The challenges that are mentioned before will all be encountered throughout this research about the BAP at X. This research aims to determine and optimize buffer sizes while taking into account the amount of available space and cable reels, the time windows in which orders need to be delivered and the complex product flows. Solving this problem for X means that a consistent product flow can be achieved throughout the department in which the probability of having a shortage of cable reels is reduced. Consequently, a higher proportion of on-time deliveries can be guaranteed. In addition, the buffer sizes can act as a guideline for the production planners and help them determine which machines should be running and when these machines should be running.

In the list below are the five main product groups shown with their corresponding production sequences. The production sequences also include the various options of machines per step. These machines correspond to the machine network visualized in Figure 2.7. These product groups account for roughly X%, and thus the majority, of the different products that flow through the manufacturing department. Would one take into account the other Y% then the product portfolio would consist out of an enormous variety of different cables. Most of these cables are only made sporadically, meaning at most only once a year which results in a low turnover rate. Consequently, it would be costly to have a cable of such type stored in a buffer for over a year. Therefore, the choice was made to not include the products with low turnover rates.

- **Single core medium voltage**: Conform extrusion line 1 or 2 → CDCC 1, 2 or 3 → Degassing HKS or LCH → Shielding line 1, 2 or 5 → Sheathing line 1, 2, 9 or jacketing line 2→ Final inspection
- **Triple core medium voltage**: Conform extrusion line → CDCC → Degassing → Shielding line → Drumtwister → Sheathing line → Final inspection
- **Low voltage**: Wire drawing mill → Insulation line → Drumtwister → Sheathing line → Shielding line → Sheathing line → Final inspection
- **Low voltage other**: Wire drawing mill or procured conductor → Insulation line → Shielding line → Installation department
- **Alkudia**: Procured conductor → Insulation line → Drumtwister → Sheathing line (filler and inner sheath) → Sheathing line (shielding and outer sheath) → Final inspection

Figure 2.7 shows a network of the various product flows that are possible within the energy department. Due to a multitude of entries as well as exits in combination with a set of possible follow-up machines after each production step a rather complex network is left over. When there is a product flow between two machines this is highlighted through the use of an arrow. Taking into account the main product groups, which are shown in the list above, that flow through the facility a lot of different paths are possible. Only taking these five product groups into consideration there are a total of 54 different paths possible, indicating the complexity of the machine network. The figure also visualizes a buffer that is indicated as a queue to symbolize the amount of

WIP waiting to be processed by that specific machine. It should be mentioned however that the machines indicated as follows; **CDCC 2**, **DEGASSING LCH**, **SL 5**, **DTW 4**, and **JA 2** are located at the facility of City B. Although these machines are not part of the department that will be focused upon they are part of the product flow. Some of the bare aluminium conductors as well as insulated conductors are transported to City B to be processed there due to capacity constraints at the facility in City A. This can also be the other way around in which processed cables are being transported from City B back to City A. This is highlighted by the arrow going from **SL 5** to **M 9**. Consequently, these machines will also be taken into consideration for the remainder of this research. In Figure 2.7, these machines are indicated by a "*" behind their name. The arrows with **START** indicate the machines that are the first step of the various production sequences happening in the energy department. For **CL 1** and **CL 2** the unprocessed aluminium wire is shaped into the right size conductor while **ISO 1** and **ISO 2** provide copper and aluminium conductors with an insulation layer. These conductors are either produced by the wire drawing mill or are procured. The wire drawing mill and procurement of raw resources are out of the scope of this research. Then there are the arrows that indicate after which machines the various products leave the system. These flows are indicated with **EXIT**.

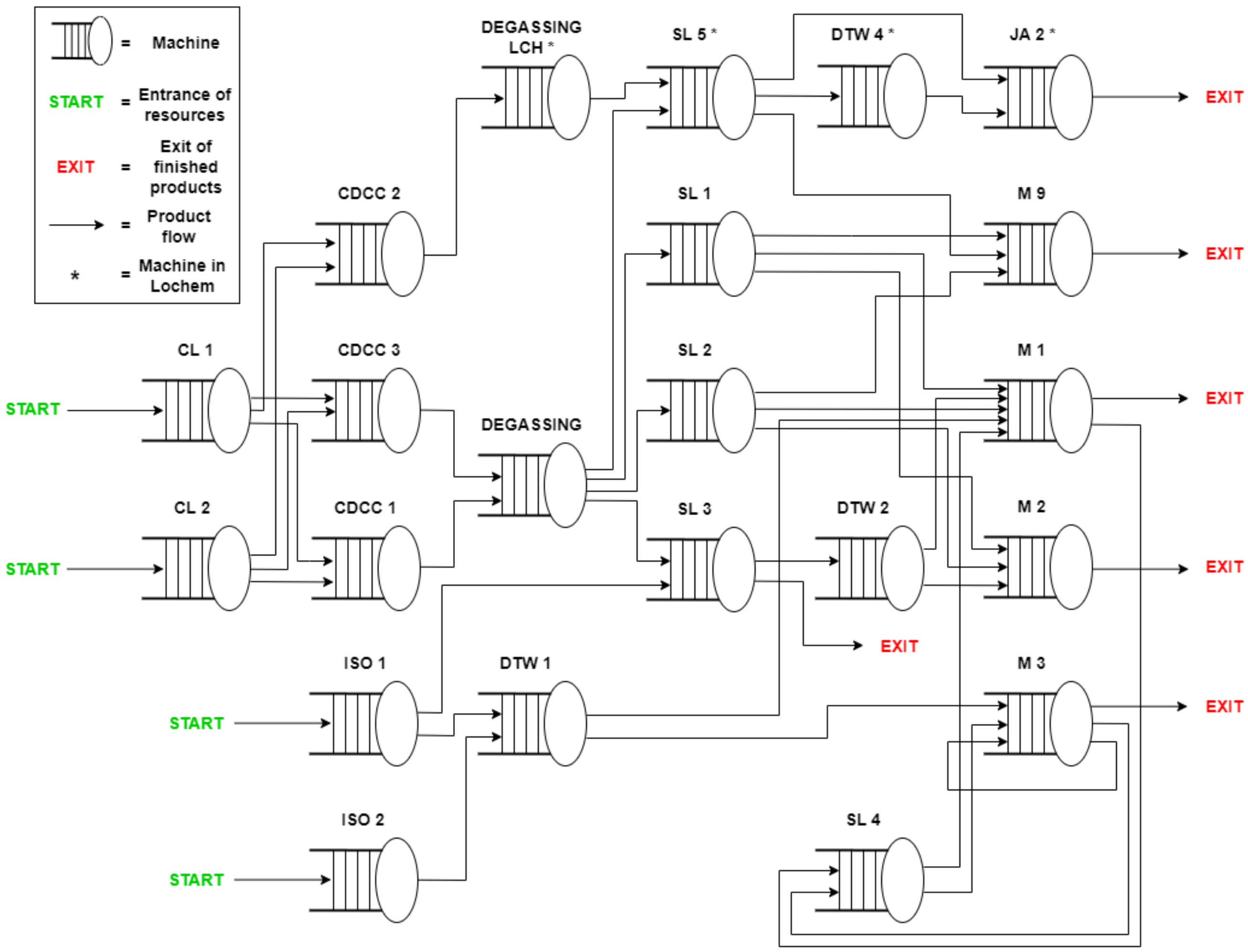

Figure 2.7: Network of all machines and possible product flows.

Figure 2.8 shows a spaghetti diagram for a medium voltage cable with a single aluminium conductor. The pink line visits both the processing steps as well as the in-between temporary storage areas. The cable starts as standard aluminium wire that is procured. It is then extruded in the right shape, in this case by **Conform line 2**, and then it is stored at **CDCC 3 hal** and insulated by **CDCC 3**. After being insulated it needs to go to **Degassing** before it can be shielded at **Shielding line 1**. In between it will be stored at the **WIP storage** and the second to last step is the outer sheath of the cable, which is applied by **Sheathing line 1** in this case. The last step is the final inspection before the cable exits the factory. This figure also shows that the manufacturing layout is not optimal due to large distances between subsequent processes. In addition, the cable goes back and forth the entire department several times also adding a lot of distance. To optimize the product flow even more, the company could look into rearranging the machines. However, this is very costly and requires a different approach. Therefore, the redesign of the manufacturing layout is left out of the scope of this research.

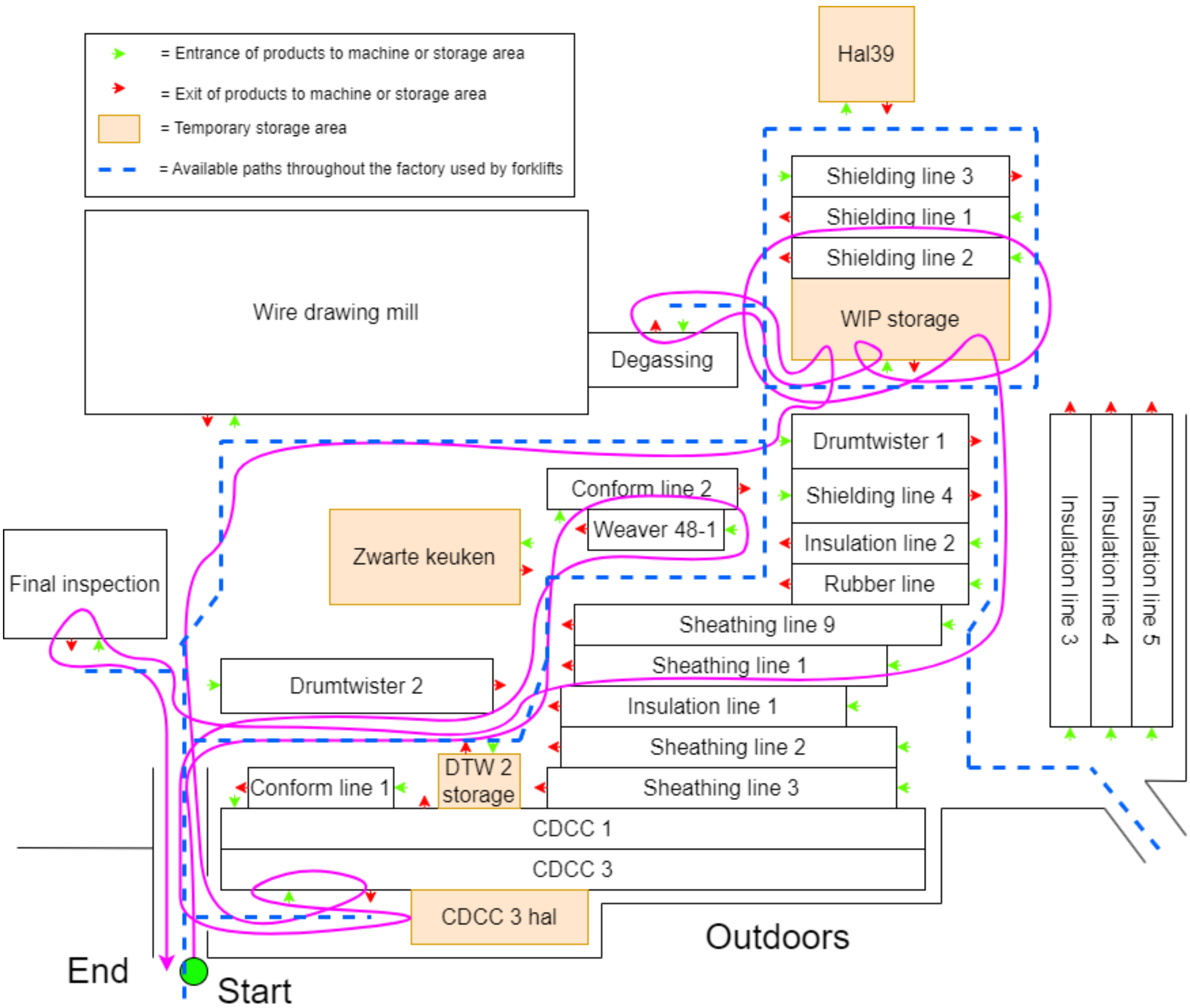

Figure 2.8: Example of how a medium voltage cable would travel through the manufacturing department.

2.4 Current Buffer Strategy

In the current situation, there is no clear strategy being applied by the production planners to optimize the product flow through the factory. Planning decisions are mostly based on the delivery dates of production orders, instinct, and characteristics of the processes. Examples of process characteristics are certain materials being consumed by extruders, one can imagine that an extruder cannot simply switch between one or the other polymer. This would require a set-up of the machine. Planning decisions are therefore also based on reducing the number of set-ups however, this is not always possible due to the limited number of machines performing the same process. Furthermore, machines are scheduled for several shifts depending on the amount of work waiting to be processed by that machine. For example, if the enterprise resource planning (ERP) system shows that the WIP waiting for a specific machine amounts to 16 hours then that machine will only be scheduled for two out of the three upcoming shifts.

There are two main key performance indicators (KPI) that measure the performance of the manufacturing system and the production planning. Firstly, the proportion of on-time deliveries is used to measure supply reliability. This is usually denoted in the form of a percentage of orders that have made it on time out of the factory and also purposes as an indication for customers with what percentage they can expect their order to arrive on-time. Currently, the department aims to have an on-time delivery percentage of 99%, however, this is roughly 95% at the moment. This percentage can partly be explained by the large number of discharges at the final inspection. Another part is due to machine breakdowns, meaning that work will pile up in front of the broken-down machine resulting in long throughput times. Secondly, there is another KPI that measures the output of the manufacturing department. The output of the factory is denoted in the length of cable that has been applied with an outer jacket. When a cable has received its outer jacket it indicates that the cable is finished processing. Consequently, this KPI can also be used to calculate the potential turnover that has been produced over a certain period.

- **KPI**: output [reels per hour], currently: $\sim$0.69 reels per hour

- **KPI**: on-time deliveries [percentage], currently: $\sim$95%

How long production orders spend in WIP highly depends on the production planning. For example, if a machine is set up for a material or configuration that is ordered very sporadically then it could be possible that orders for the upcoming two or three months with that specification are processed in a single cluster. Consequently, these orders could spend significantly longer in WIP waiting for their subsequent process than cable types that are produced on a more regular basis. On average an order will spend about two weeks inside the factory from start to finish. The size of a production order also fluctuates highly. Especially for orders that are make to order (MTO) depend on the customer's wishes. The order sizes for make to stock (MTS) have lower variability since these cables are made more frequently and are used for replenishing the stock.

2.5 Summary

This chapter has provided an in-depth understanding of the problem context by introducing X, its manufacturing processes, and highlighting the challenges it faces in its day-to-day operations. The BAP variant at X significantly affects the performance of the machine network due to various constraints and limitations. Furthermore, there is not a clear buffer strategy applied which can be used by production planning as a guideline. In the following chapters, available literature is reviewed and an exact problem formulation is stated. With these instruments, it is possible to derive an adequate solution methodology. Subsequently, a tailor-made solution is derived that takes into account the constraints as well as objectives.

Chapter 3

Literature Review

In this chapter a comprehensive literature review is presented that discusses the existing literature concerned with the BAP. The purpose of this literature review is to create a foundation upon which an understanding for buffer allocation and its variants can be built. In addition, a set of solution methodologies is explored and put into context of the problem variant encountered at X.

Section 3.1 aims to provide a more in-depth explanation of the BAP in combination with challenges that generally arise for this type of problem. This should also provide an indication of why the BAP can become difficult to solve depending on the complexity of the environment in which it is put into context. Moreover, this section also highlights the significance of this research and why this topic in particular is of scientific importance and could help in solving problems of a similar type.

Subsequently, different variants of the BAP are addressed in which each version has its own unique characteristics. By elaborating on each of these different variants it becomes more clear which specific variant of the BAP is encountered by X.

The next section presents a set of solution methodologies that are commonly used to solve the BAP. These methodologies can be divided into two categories which are both necessary to arrive at an optimal or near-optimal solution. Furthermore, by understanding these methodologies some opportunities might be explored while weaknesses can be identified.

The following section gives an impression of the various tools, programs or other software that can be used to help solve the BAP. These tools are also evaluated based on the limitations at X, regarding licenses or additional expensive equipment for example.

Section 3.5 addresses the specific instance of the BAP that exists at X. Besides the general commonalities with other found instances of the BAP sets of unique objectives, requirements, and constraints are stated. These sets contribute to developing a custom model that is can solve the specific BAP found at X.

The last section is aimed at addressing available literature in which the tackled BAP shows a resemblance to the one at X. These previously conducted case studies can help in determining which combination of solution methodologies is most suitable for this specific instance of the BAP.

The presented literature below provides a basis for determining the most adequate approach of the BAP variant at X. With this established basis the goal is to build a model that can produce optimal or near-optimal buffer sizes that satisfy the needs of X.

3.1 Fundamentals of the BAP

As was discussed before the BAP is focused on finding optimal buffer sizes while taking other factors into account such as the minimization of costs, maintaining or improving throughput, and reducing the space occupied by WIP. According to Smith and Cruz [2005] the BAP is concerned with minimizing the probability with which a blockade in the product flow can occur. Two major challenges are encountered when tackling a BAP. The first one is the lack of an algebraic relation between the buffer size configuration and the resulting performance of the manufacturing system. This means it is complicated to evaluate a possible buffer size configuration. However, there are ways to bypass this obstacle without the use of an algebraic relation. These will be discussed in Section 3.3.

The second major challenge is the increasing complexity of the problem. When the network of machines increases this also increases the number of buffers at which WIP can be stored. Consequently, the number of possible buffer size configurations increases exponentially. Moreover, when the individual buffer sizes can increase this leads to a significantly larger solution space. This means that an algorithm could take a substantial amount of time to come to an optimal solution or will never even find one. An efficient algorithm as well as a fast processor is very beneficial when trying to solve the BAP for larger networks of machines. This is because the BAP is an NP-hard combinatorial optimisation problem [Xi et al., 2020].

3.2 Classification Criteria of the BAP

Various occurrences of the BAP exist and are in the basis similar in that they want to find an optimal or near-optimal buffer allocation. However, each of these instances has its own characteristics. Below are the criteria highlighted through which a variant of the BAP can be classified [Demir et al., 2014, Weiss et al., 2019]. The specific criteria can also be seen in Table 3.1 in which the available literature can be compared to this variant of the BAP.

3.2.1 Machine Network

The characteristics of the machine network can significantly affect how the buffers are allocated. Another meaning of machine network is production line topology. When a machine network consists of a single serial production line it can be more important to place large buffer sizes at the machines subsequent to the bottleneck machine than when one is dealing with a flexible manufacturing system in which the product can follow multiple paths. Another example of a machine network is the assembly line in which multiple production lines eventually converge to a single station at which the final product is assembled. An example of an assembly line topology can be seen in Figure 3.1. Furthermore, there is the general network topology in which a single station is connected to all the other stations. In addition, it is possible to have a serial-parallel production line in which multiple serial lines are placed in parallel and also allow products to switch lines in between processes. Lastly, there is also the cellular manufacturing system in which each product group has its own production line in which the products are not allowed to switch "cells". Comparing these examples to the machine network shown in Figure 2.7 and the product groups shown in the list in Section 2.3 one can conclude that at some stations there are multiple subsequent possibilities indicating a resemblance with the flexible manufacturing system. Moreover, it also shows serial production lines in parallel, thus hinting towards a serial-parallel machine network.

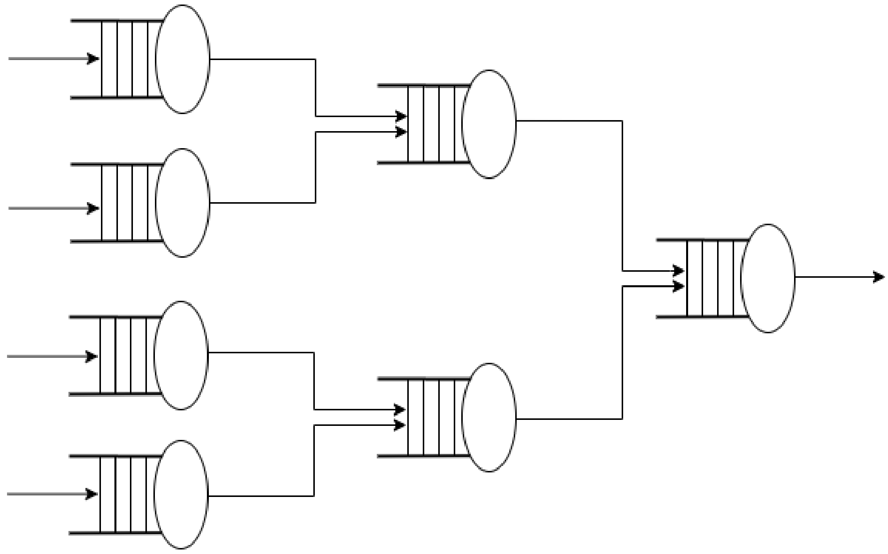

Figure 3.1: Example of an assembly line topology.

3.2.2 Objective Function

Different versions of the BAP also exist depending on the objective function. A company can have various reasons for which they want to solve it. Following, are some of the most used objective functions. Firstly, a company can desire to maximize their output. By maximizing their output it is possible to increase the potential revenue generated. Secondly, a company can aim to reduce the amount of WIP stored in the factory to prevent overcrowding the factory and being at risk of running out of material handling resources. Furthermore, having large buffers at each station also results in large costs, such as holding costs and additional procurement of material handling resources. A company can therefore aim to reduce costs by solving their variant of the BAP. Lastly, there can be various other objective functions such as minimizing the mean waiting time, reducing idle time, minimizing the average throughput time, or minimizing the number of orders that are produced outside their time window. Two of these objective functions are also of importance to X. Firstly, they aim to reduce the amount of WIP that is stored inside the factory in order to lower the risk of running out of empty cable reels. Secondly, the company is revenue-focused and one of their KPIs is the amount of cable that has received its outer jacket. Consequently, they also aim to maximize their throughput.

3.2.3 Solution Method

The solution method that is applied to solve the BAP is another criterion on which classification can be based. Most available literature is based upon iterative optimization methods in which there is a generative and evaluative part. The generative method is responsible for the generation of candidate buffer allocations. These buffer allocations are then evaluated by the evaluative part of the optimization method. Some examples of generative methods are simulated annealing, dynamic programming, and simulation. Examples of evaluative methods are the decomposition method, expansion method, aggregation method, and simulation. These methods are further elaborated in Section 3.3. The choice for a certain solution method often depends on the characteristics of the production system as well as the intent with which the model will be used. In the case of X the company wants the model to act as a guideline for the production planning, such that planning decisions can be made more easily. Consequently, the model should be easy to understand and have short runtime in order to be able to quickly acquire the desired buffer allocation. These requirements should be taken into account when choosing a solution method.

3.2.4 Reliable Production Lines

Lastly, from the available literature about the BAP a division can be made between production lines with reliable machines and production lines with unreliable machines. This significantly affects the buffer allocation. For example, a serial production line with unreliable machines is taken into account. Each of these machines has a different OEE but the same processing time. Consequently, the machine with the lowest OEE could be considered the bottleneck and as a result, it would be more beneficial to have larger buffers at the machines after this bottleneck machine. Due to the larger risk of the bottleneck machine breaking down, this also means that the supply to the subsequent machines halts is more likely. It would therefore make more sense to have larger buffer sizes at these subsequent machines. Comparing this example to a machine network with reliable machines the buffer allocation could be completely different. Placing these criteria in the context of X it would make sense to take unreliable machines into consideration in the form of an OEE to resemble reality as closely as possible. The machines at X are not 100% reliable and therefore it should be taken into account.

3.3 Existing Methods for the BAP

Various methods exist to tackle the BAP. In general, the different solution methodologies can be classified into two segments. The first segment is occupied with the generation of buffer size configurations that could pose as the most optimal solution. The chosen methodology in the first segment works in a feedback loop with another methodology in the other segment. This segment is responsible for evaluating the found buffer size configuration. This set of methods will show if there is an improvement found or not and therefore acts as a kind of feedback loop.

3.3.1 Generative Methods

In this section, the various possible generative methods will be discussed. As was mentioned the generative methods are responsible for coming up with different buffer size configurations. Below are some of the most frequently used generative methods explained.

Simulated Annealing

Simulated annealing is one of the various methods that are used in different variations of the BAP. In the context of the BAP simulated annealing is used as a search algorithm that generates new possible solutions. The method is initialized with a non-optimal configuration which is first evaluated before a new configuration is generated at random. If the objective value is improved with the newly found configuration then this solution is accepted. This process will repeat itself until a certain termination criterion is satisfied. A common disadvantage of other search algorithms is the risk of ending in a local optimum. The simulated annealing algorithm reduces this risk by having the ability to accept solutions that perform worse than the already found best solution. The probability with which a worse-performing solution is accepted is decreased after each iteration following the Boltzmann probability distribution. In addition, after each iteration, the "temperature" is decreased by multiplying it with a certain constant c that has a value between 0 and 1. With the new temperature a new probability is calculated according to the Boltzmann distribution that is stated below. This is only done if the found solution performs worse. In the equation, w represents the probability with which this solution is accepted and E is the difference in performance.

$$w \sim \exp\left(\frac{-E}{cT}\right) \tag{3.1}$$

Spinellis and Papadopoulos [2000] used the simulated annealing algorithm in their methodology for solving the BAP. In their method, they allocate the total available buffer space N over a number of K workstations. The allocated buffers are positive integers otherwise, the number of possible configurations would become excessively large and eventually going to infinity. The objective of their research was to maximize the throughput of the production system, while not exceeding the maximum available buffer space. Once a buffer configuration is found by the algorithm the solution is evaluated with the use of the expansion method, which is explained under Section 3.3.2. An advantage of the simulated annealing algorithm is that a large solution space can be explored. In addition, the algorithm has a built-in regulation with which worse-performing solutions can be accepted thus preventing one ends up in a local optimum. This increases the chance of finding a global optimum. However, the way it finds new solutions is also its disadvantage. It does not incorporate a specific strategy in which potential good configurations are exploited. Consequently, it is possible that many different solutions first need to be evaluated before the algorithm ends up at an optimum or reaches a stopping criterion. As a result, the algorithm could take a long time to run or in some cases not even find a local or global optimum and only present the best solution found until the algorithm was stopped.

Greedy heuristic

Several greedy heuristics exist that are applied for the BAP [Essafi et al., 2010]. One of these heuristics includes an algorithm in which each buffer is added in an iterative manner until all buffers are allocated. The choice where each buffer is placed on the corresponding increase in throughput. The machine with an additional that leads to the biggest increase in throughput gets that buffer allocated. This is repeated until no buffers are left. This algorithm is further referred to as the greatest improvement algorithm.

Dynamic Programming

The dynamic programming algorithm tries to find a solution by dividing the problem into stages. Each stage poses as a sub-problem that requires solving before the next stage can be tried to solve. An example for which the dynamic programming algorithm is commonly applied is the shortest path problem. An example of such a problem can be seen in Figure 3.2. A way to solve this problem with dynamic programming would be to start at the end, so what would be the shortest path from either **E** or **F**? This stage is relatively simple since there is only a single path leading from both points to the end. The next stage would be to determine the shortest path from both **C** and **D**. Since the shortest paths from **E** and **F** are already determined this helps determine the shortest paths in this stage. This process will repeat itself until one ends up at the start.

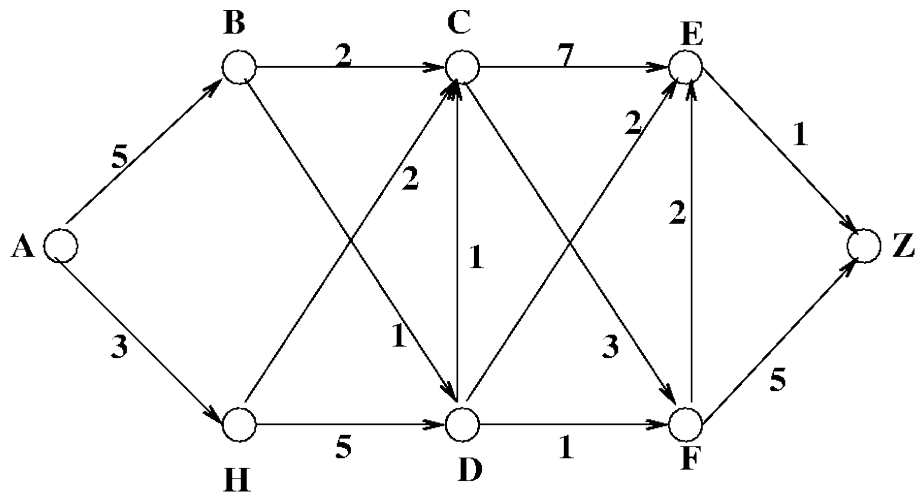

Figure 3.2: Example of a road network for the shortest path problem Bellman [1958].

Huang et al. [2002] have employed a dynamic programming algorithm in order to explore the solution space in their BAP variant. It is proposed to let the workstations represent the stages after which a sub-problem is solved. The state represents the situation in which a certain amount of buffer has already been allocated. The algorithm makes use of a recursion formula that is used to understand the algorithm. Let Z_{i+1} be the best performance for stage $1, 2, ..., i + 1$, given state x_{i+1}, and $Y_{j,i+1}$ the performance of feasible alternative j in stage $i + 1$ with J the maximum number of alternatives. In addition, let P_i be the performance for stages $1, 2, ..., i$, given the state x_i. If the objective is to maximize the throughput, then the following recursion formula is acquired.

$$Z_{i+1} = Max\{Y_{1,i+1} + P_i, Y_{2,i+1} + P_i, ..., Y_{J,i+1} + P_i\} \tag{3.2}$$

This recursion formula comes back in each stage and the result of this formula in the previous stage i given state x_i poses as P_i for the next stage $i + 1$. After the final stage and thus all sub-problems have been solved a new buffer allocation configuration is acquired. An advantage of the dynamic programming algorithm is that it does not randomly search for suitable solutions, but finds the most optimal solution given certain starting conditions. However, dynamic programming does require more complexity when programming this algorithm. In addition, it will take more time to generate a solution compared to other algorithms, such as the simulated annealing algorithm.

Tabu Search

The tabu search algorithm is another method for finding new suitable solutions. Instead of searching randomly for improved solutions it searches more aimed at direct improvement of the objective function. The algorithm is given an initial solution from which its direct neighbouring solutions are derived. From each of these neighbours, the objective value is determined and compared to the current best value. The neighbour with the best found objective value is stored in the tabu list. This neighbour also functions as the base solution from which the neighbours are derived in the next iteration. In order to reduce the risk of the algorithm ending up in a local optimum the algorithm checks if the neighbour is not already stored in the tabu list, if so, then this neighbour is skipped. Consequently, a larger solution space is explored compared to other algorithms such as local search. The tabu list has a certain size indicating the number of solutions that it can store. The algorithm is terminated either when all neighbours derived from a certain solution are already stored in the tabu list, a number of maximum iterations are reached, or through another stopping criteria. An example pseudocode for the tabu search algorithm can be seen below.

Algorithm 1: Pseudocode for a tabu search algorithm

```
Initiate tabu list
Generate initial solution x_0
Initiate best solution x_b
Initiate best objective value y_b = 0
do
    Initiate current best solution y_cb = 0
    {x_1, x_2, ..., x_n} = GenerateNeighbours(x_0)
    for i = 1 to n
        if x_i is not in tabu list and y_i > y_cb
            y_cb = y_i
            x_cb = x_i
            if y_i > y_b
                y_b = y_i
                x_b = x_i
            end
        end
    end
    x_0 = x_cb
    Update tabu list
while Termination condition not satisfied
Return x_b as best solution
```

Demir et al. [2010] developed a tabu search approach for determining the buffer allocation in production lines. The algorithm was developed with the objective of maximizing the throughput of the manufacturing system. In addition, the efficiency of the algorithm was also tested for minimizing the amount of WIP while maintaining the desired throughput. Furthermore, Demir et al. [2010] took also unreliable machines into account. By incorporating failure rates and repair rates the flows between all machines are affected respectively. As an evaluative method they applied the decomposition method, which will be addressed later on. The initial solution that was fed to the algorithm is generated randomly. From the experimental results, they concluded that their tabu search algorithm could be a suitable approach for solving the BAP.

Design of Experiments

As opposed to the previously two mentioned generative methods the design of experiments does not incorporate an underlying search algorithm. A design of experiments is set up beforehand and it is exactly known which buffer allocations will be evaluated. Compared to simulated annealing no random buffer allocations are generated. A set of buffer allocations is made upfront.

Raman and Jamaludin [2008] performed a case study in which the aim was to reduce the WIP for the production system of a small automotive parts manufacturer. The manufacturing system that was focused on is relatively simple compared to the system of X since it only consists of three stages that are in sequential order. Three different buffer strategies were developed up front after which a simulation was executed in order to measure the individual performance of each strategy. In addition, they made each strategy variable. For example, the first strategy is stated as reducing the size of the first buffer compared to the initial buffer. Subsequently, the effect on the performance was measured by reducing this buffer by ten percent after each run.

A benefit of having a fixed set of possible solutions is that there is little to no effort required for finding the optimal solution depending on the size of the solution set. It also provides more room for comparing the performance between them. A disadvantage of a generative method like the design of experiments is that only a very small portion of the solution space is explored. However, this does depend on the complexity of the system. A small production line with only two or three machines is more easily evaluated like it is in the paper of Raman and Jamaludin [2008]. Consequently, a design of experiments is considered not to be optimal as a generative method for the BAP variant of X.

3.3.2 Evaluative Methods

Once a buffer size configuration has been found by the generative method it is then assessed by an evaluative method. This method actually indicates if the found configuration is any good or not. Below are some of the more commonly used evaluative methods listed.

Decomposition Method

The decomposition method evaluates a production system by decomposing every workstation into a single queue. Each of these queues is then analyzed individually. In addition, every queue has its own arrival and service process. By assessing each workstation individually it is less difficult to develop a relation between the allocated buffer size and the throughput of that corresponding machine. Once this relation is established it is possible to determine the throughput of the machines at the beginning of the product flow, which is then the input for the following machines in the production sequence.

Kwon [2006] developed a version of the decomposition method for a productions system that incorporates an automated guided vehicle (AGV) system. Two types of buffers are distinguished. The first type of buffer is the input buffer for a workstation, while the second buffer is the outgoing buffer. The outgoing buffer in this case is the amount of work finished by a machine that requires transporting by an AGV. In the paper Kwon [2006] presents the following relation between an input buffer size and the corresponding throughput TH of the machine.

$$TH = \mu_i \cdot \left(1 - \frac{1-\rho_i}{1-\rho_i^{IB_i+1}}\right) \tag{3.3}$$

In the equation above μ_i represents the service rate of the workstation i. This service rate indicates the number of parts or products that can be processed per unit of time by that machine. Furthermore, ρ_i denotes the utilization of that machine, which is calculated by dividing the arrival rate of products or parts by the service rate $\rho_i = \lambda_i/\mu_i$. The allocated buffer size to machine i is denoted by IB_i.

The main advantage of this evaluative method is the direct relation between individual buffer sizes and the corresponding throughput. Consequently, it is also possible to calculate the total throughput of the system. The throughput of the first production step acts as input for the subsequent machines in the machine network. A disadvantage would be that the service rates are assumed to be constant, which is the case for the paper by Kwon [2006]. As a result, it is not possible to account for outliers or downtime.

Expansion Method

While the decomposition method divides the network into individual queues, the expansion method looks at the entire network in its whole. Moreover, as its name suggests it expands the network. In the expansion network in between each workstation, an additional artificial node is created to be able to register the blocked jobs. This is visualized in Figure 3.3. Once machine i finishes a job this job then tries to enter the finite queue of workstation j. This happens with a probability of $(1 - p_{K_j})$ in which p_{K_j} is the blocking probability of machine j. There is also the probability of the queue still being blocked after retrying. This is denoted by the probability p'_{K_j}.

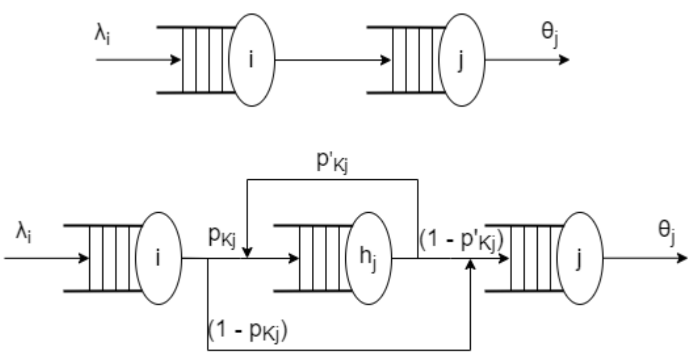

Figure 3.3: Network reconfiguration in the expansion method.

After the network expansion the corresponding parameters such as the blocking probability p_K and the service rate of the holding station μ_h. These parameters can be determined analytically. The equations to solve these parameters are developed by Kerbache and Smith [2000] and Labetoulle and Pujolle [1980]. It should be explicitly mentioned that the blocking probability is a function of the allocated buffer capacity. After the parameter estimation, some compensation is required for the feedback loop that can be seen in Figure 3.3. Should the buffer of workstation j be unsaturated then the service rate of machine i is μ_i and when it is saturated then the service rate of machine i is $\mu_i + \mu'_h$ with $\mu'_h = (1 - p'_K)\mu_h$.

Woensel et al. [2010] developed an expansion method that can be used in production systems with a multitude of machines with general processing times. Once all parameters were estimated they were able to decompose the system and present a formula for the relation between the blocking probability and throughput of a machine i. This relation is presented below in which θ_i represents the throughput of machine i and λ_i the corresponding arrival rate.

$$\theta_i = \lambda_i \cdot (1 - p_K) \tag{3.4}$$

An advantage of the expansion method is that it also introduces an algebraic relation between the allocated buffer size and the throughput although it does not show it directly in Equation 3.4. The calculated blocking probability is a function of the allocated buffer size of the corresponding machine. Similar to the decomposition method a disadvantage is the lack of process time distribution or downtime consideration.

Aggregation Method

The aggregation method operates differently compared to the decomposition method and the expansion method in the fact that it does not expand the network or zoom into each individual workstation. Instead, it uses a recurrent process of combining two machines into one new virtual workstation. This is visualized in Figure 3.4. This resulting virtual workstation has the following parameters; λ^*, μ^*, and u^* which represent the arrival rate, service rate, and the production rate of the upstream machine. These parameters are calculated using differential equations according to the two-machines-one-buffer Markov model [Gershwin and Fallah-Fini, 2007].

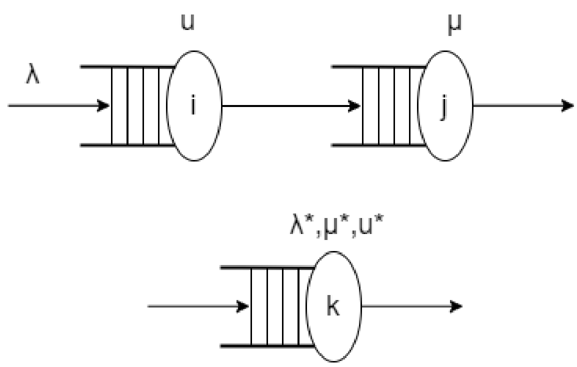

Figure 3.4: The aggregation method converts two machines into a single virtual workstation.

Dolgui et al. [2007] incorporated the aggregation method in their hybrid algorithm to solve the BAP for production lines that are placed in tandem. They used the aggregation method alongside a combination of genetic and branch-and-bound approaches.

Using the aggregation method they were able to convert all machines in the system into a single virtual workstation. For this workstation, they were able to establish a relation between the throughput $V(H)$ and the adjusted parameters for this single station. The algebraic equation is stated below.

$$V(H) = u^* \cdot \frac{\mu^*}{(\lambda^* + \mu^*)} \tag{3.5}$$

From this equation can be concluded that it does not directly relate buffer size to the throughput as was the case in Equation 3.3. However, in determining the adjusted parameters the buffer sizes are taken into account [Terracol and David, 1987]. However, it should be mentioned that the applicability of the presented approach is reasonably limited since the machines are required to be in sequential order with no by-passes or feedback loops. Consequently, the aggregation method is not considered for the BAP at X.

Simulation

Simulation is one of the most frequently used evaluative methods for the BAP. With simulation, there is no need to establish an algebraic relation between the buffer configuration and the throughput of the production system. Amiri and Mohtashami [1987] used simulation in their BAP variant for unreliable production lines. They made use of discrete event simulation, which skips ahead in time until the next event takes place. Continuous simulation also exists and is used for example to predetermine the trajectories of rockets or perform flight simulations. However, these simulations are a lot more workload-intensive per simulated unit of time. The choice for simulation was made due to the complex relationship between workstations that was encountered and the ability to simulate processing times according to a specific distribution. To generate buffer allocation solutions they used a design of experiments in which specific cases were determined beforehand.

Yelkenci and Kilincci [2015] also used discrete event simulation as an evaluative method in their research to tackle the BAP in open serial production lines. To generate possible buffer allocations a hybrid approach was used that combined both a genetic algorithm and simulated annealing. Once a possible solution was generated by the hybrid algorithm, this buffer allocation configuration was simulated for a given period of time. The key performance indicator used was the average production over that period of time. Moreover, the model was built using the simulation software Arena V10.0 while the hybrid algorithm was programmed using Matlab V7.6.

Vergara and Kim [2009] also used simulation to assess buffer configurations in serial production lines. They chose to use a simulation since it allowed for the use of variable processing times. Before a job starts at a workstation the processing time is determined beforehand according to a statistical distribution. This time would then be added to the throughput time of the product. Moreover, it is also possible to simulate variable repair times when a machine breaks down. The key performance indicator was determined by dividing the total number of produced products by the total operational time of the last machine in the production sequence.

The advantage of simulation is that there is no need for an algebraic relation between the buffer allocations and the total system throughput. In addition, it can be a more reliable manner of evaluating a buffer configuration since it can represent reality provided that the simulation is correctly built. For example, a simulation is able to incorporate machine failures and simulate downtime and thus making it more realistic. On the other side, simulations can become very complex and take a long time to run when the number of machines and possible buffer configurations increases.

3.4 Software and Tools

Solving the BAP by hand would be very time consuming. Consequently, everyone who tries to tackle a variant of the BAP uses some sort of tool or a specific software package that is able to either generate buffer configurations quickly or simulate the production system realistically for example. In the previous section, the term discrete event simulation was already mentioned. This type of simulation allows the user to efficiently simulate a production environment or a hospital for example. In some cases, it would be more beneficial to use a continuous simulation like a chemical plant where many pipelines would require simulation.

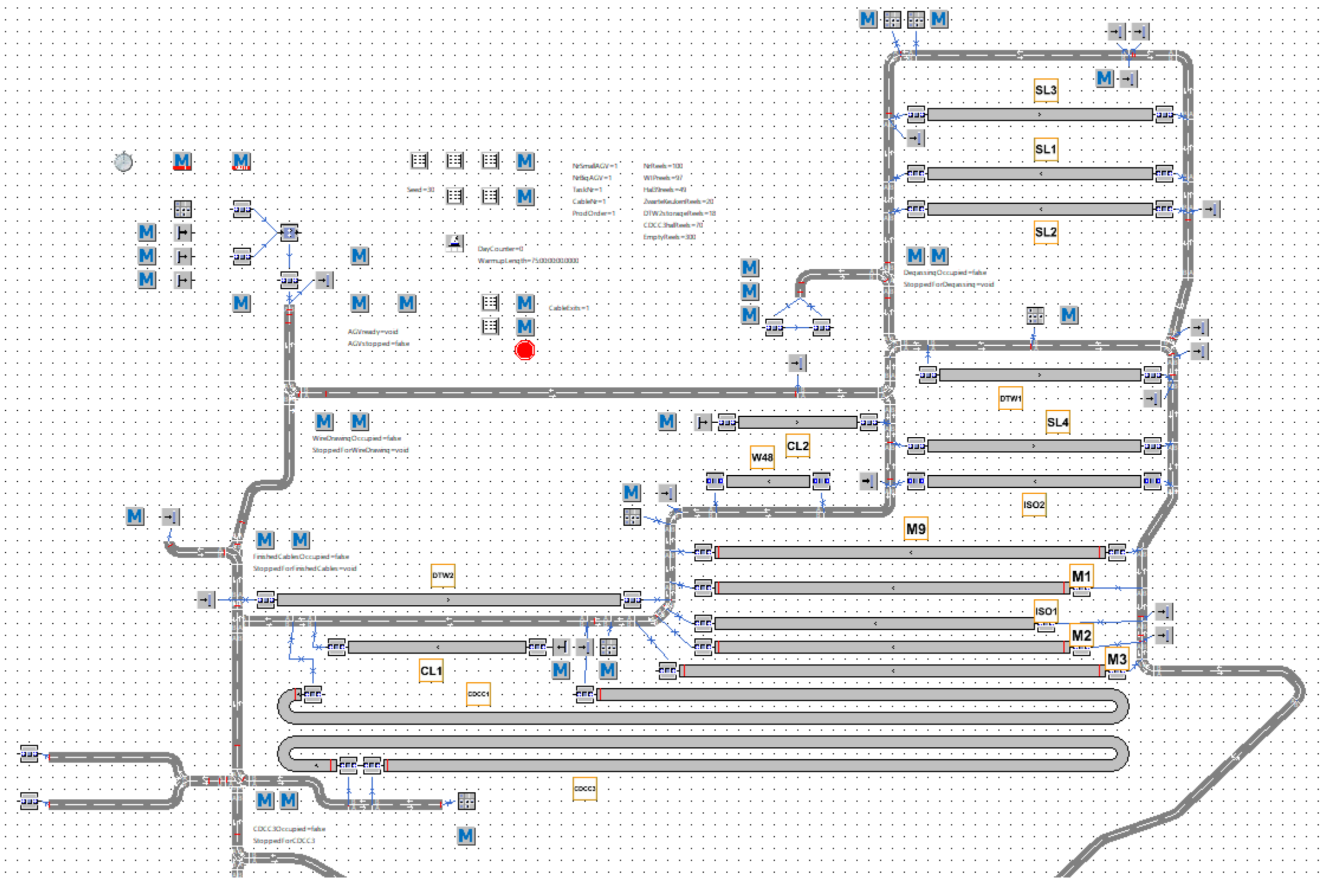

Figure 3.5: Example of a simulation model representing a production system.

For the BAP in most cases, it is possible to use discrete event simulation, since there is only a finite number of parts processed at each moment in time compared to a continuous flow of material. In Figure 3.5 an example of a simulation model can be seen that shows the layout of a manufacturing facility. The program used for this model is Siemens Tecnomatix Plant Simulation. Various other manufacturing environment simulators exist such as Arena, SimPlan, and SysCAD.

In addition to simulation software, there are other tools and software packages used in order to solve the BAP. In order to generate possible solutions with either simulated annealing or dynamic programming for example it is far more efficient to use a computer. A computer is able to generate possible solutions much faster and is, therefore, able to explore the solution space quickly compared to a human. Various tools exist that are able to deploy these algorithms such as programming languages like Python or Java. However, these programs are not optimized to run these algorithms. Programs like Delphi and AIMMS are more focused towards solving combinatorial optimization problems like the BAP.

Another very important aspect when considering a software package or tool for solving this variant of the BAP are the wishes of the company. Firstly, the company way not want to spend significant resources in order to have a tool that is able to solve the BAP for them. Licenses for some of the mentioned programs can become rather expensive, upwards of thousands of euros per year. Furthermore, there is also the usability of the tool. It would be more beneficial to the company if the tool can be used by a multitude of employees. Since the tool will need to be rerun each time a characteristic of the factory changes, such as the processing time of a workstation.

3.5 Buffer Allocation Problem at X

Since no BAP is the same, the variant of X is no different. It has its own characteristics, constraints, and objectives. Firstly, the network of machines as was shown in Figure 2.7 is rather complex and poses a challenge. Most of the conducted research about the BAP focuses on workstations that are in sequential order. The workstations at X are not all ordered in series and there are multiple parallel product flows. Some of these product flows even use the same machine. Moreover, for some product flows there are multiple follow-up machines possible after certain production steps. Furthermore, there are also feedback loops possible in which a product is processed by the same machine more than once. As a result, the complexity of the machine network sets this variant of the BAP apart from most other variants.

Other characteristics of this BAP that make it unique are the production steps itself. Some of the processes produce a multitude of cable lengths, meaning that from one input cable reel, there can be two or more outgoing cable reels or the other way around. These conversion rates are a very important factor that should be taken into account when determining buffer allocations whilst satisfying the corresponding constraints. Furthermore, the level of WIP is not measured in the number of cable reels to be processed by a specific machine but by the amount of processing time that is waiting for a certain workstation. Since cable reels can have different lengths of cable stored the processing time of each step can differ for each independent reel. Consequently, the WIP that is waiting for a specific machine should not be measured in the amount of cable reels.

Besides the characteristics of the complex machine network and conversion rates, there are also certain constraints that make this BAP variant unique. Of course, there is the space limitation which should be taken into account. Inside the manufacturing facility, there is no infinite capacity to store WIP or empty cable reels. As a result, the number of available spaces for cable reels should be taken into consideration. However, this is not a constraint that sets this BAP variant apart from other variants. Besides the available space, a ratio should be set between cable reels that are loaded and can be seen as WIP and empty cable reels. These empty cable reels are of importance since the large majority of production steps require both a cable reel with the cable to be processed and an empty cable reel upon which the processed cable will be winded. To make the matter even more complicated there is also the matter of available cable reels. Some processes require specific types of cable reels regarding their size. The company does not have an infinite number of cable reels for each type.

Moreover, the machines used in the energy department are prone to failure. Unfortunately, there is no data available on when each machines has broken down and what the amount of downtime was. However, the result of these machine failures can be noticed in the OEE that is kept track of. Consequently, in order to have a realistic model the OEE of each machine should be taken into account. On top of the machine failures, there is also the problem of a limited workforce. Each machine is operated by employees and sadly there are not enough people to operate all machines all the time. As a result, machines are not running in some shifts due to either a shortage in the workforce or scheduled maintenance. Since this also affects the product flow it should also be taken into account in the shape of available machine-hours per week. Therefore, the limited workforce and available machine-hours together with the restricted number of cable reels and storage space is what makes this variant of the BAP unique. The BAP variant being tackled in this research can be defined as the capacitated and restricted machine hours buffer allocation problem (CRMH-BAP) for unreliable production lines.

Lastly, there are the objectives for the BAP variant of X. Firstly, the WIP should be reduced where possible. The goal of this objective is to reduce the number of loaded cable reels stored on the factory floor. As a result, the probability of a machine stagnating due to the lack of available empty cable reels is reduced. Furthermore, this also makes it easier to keep track of the number of loaded cable reels. However, it could also be possible that for some workstations the amount of WIP will increase for a certain buffer allocation. This does not pose as a constraint for the BAP of X. The objective would be to reduce the amount of WIP where possible and increase where required. Finally, there is the objective for the throughput rate. For the BAP variant of X the aim is to maximize throughput. The buffer allocation should definitely not lead to a decrease in output since this would also mean a loss in turnover. The company indicated that it is revenue-driven and therefore aims to maximize the output of the facility. This specific variant of the BAP is also put in Table 3.1 to see how it compares to other available literature and what makes it unique.

3.6 Similar Work to this Variant of the BAP

This section focuses on previously conducted research that is relatable to the work presented in this report. Hodgson et al. [2004] present an integer linear programming model in which they allocate glass fibre material. Although optical fibres are not the same as electric cabling they do share a number of similar production steps. The aim of the research was to create a model in which glass fibre material would be allocated such that the least amount of material is wasted. Instead by focusing on the maximization of the throughput the focus was on reducing waste. Furthermore, a random neighbourhood search algorithm was used to come up with different possible solutions and explore the solution space. The configuration would then be put into the objective function of the integer linear programming model to assess the performance. The key performance indicator is the resulting amount of scrap.

Miller [2004] developed a scheduling tool for the manufacturing of optical fibres. The goal of this research was to optimize the production planning such that the time spent on setups is minimized. The aim was to pool orders with similar setup requirements such that little adjustments would be required in between. In addition, the requirements and characteristics of processes further on in the production sequence should also be taken into account. Through the use of a dynamic programming algorithm, it was possible to reach an overall decrease of 25% spent on setups. The solution was presented in the shape of a program that was coded in Excel VBA and Fortran. This program could then be used by the company in order to determine the most optimal production schedule on a short-term basis.

Boomers [2022] also created a scheduling tool in a production environment and in this case, it even involves the same company as this research. The objective of this research was to manage the flow of make-to-stock products from the wire drawing mill department to the other production departments in the factory. In order to manage this flow a tool was developed in which a medium-term production plan could be generated and to which the short-term production schedule could updated. The author used an adjusted version of the capacitated lot sizing problem that can be solved using a mixed integer linear programming model. The algorithm and tool were built using Python. With the developed tool it was possible to achieve a reduction of 7% in costs. Other available literature regarding the BAP can be found in Table 3.1. The columns in this table indicate the characteristics that were discussed in Section 3.2.

3.7 Summary

In this chapter, an in-depth understanding of the BAP was provided. It became clear that the BAP can be classified according to various criteria. These criteria can help in classifying the problem at hand at X and as a result, it is possible to develop an adequate approach. In addition, a hybrid approach is presented in which various generative and evaluative methods are presented that are being used for different variants of the BAP. Moreover, the suitability of each method is also discussed. The chosen method should also be compatible with the chosen tool through which the model will be run as well as presented. Furthermore, the BAP at X is discussed more in-depth regarding requirements and constraints. Lastly, other similar works regarding the industry or other research conducted about the BAP were stated in the section above and in Table 3.1. In the following chapters, the exact variant of the BAP at X is mathematically formulated and the chosen methodology is elaborated.

Author	Machine network	Objective	Solution method		Machine reliability	Capacitated	Limited hours
			Generative	*Evaluative*			
Amiri and Mohtashami [1987]	Assembly line	Maximize throughput	Design of experiments	Simulation	Unreliable	Yes	No
Terracol and David [1987]	Two-machine line	Maximize throughput	Design of experiments	Aggregation method	Unreliable	Yes	No
Lutz et al. [1998]	Serial production line	Reuce WIP	Tabu search	Simulation	Reliable	No	No
Yamashita and Altiok [1998]	Serial production line	Reduce WIP	Dynamic programming	Simulation	Reliable	Yes	No
Spinellis and Papadopoulos [2000]	Serial production line	Maximize throughput	Simulated annealing	Decomposition method	Reliable	No	No
Huang et al. [2002]	Serial production line	Reduce WIP	Dynamic programming	Decomposition method	Reliable	Yes	No
Abdul-Kader and Gharbi [2002]	Serial production line	Minimize cycle time	Design of experiments	Simulation	Unreliable	Yes	No
Smith and Cruz [2005]	General network	Minimize WIP	Powell's method	Expansion method	Reliable	Yes	No
Dolgui et al. [2007]	Serial-parallel	Maximize profit	Branch-and-bound	Aggregation method	Unreliable	Yes	No
Nahas et al. [2009]	Serial-parallel	Maximize throughput	Simulated annealing	Decomposition method	Unreliable	No	No
Vergara and Kim [2009]	Serial production line	Maximize throughput	Design of experiments	Simulation	Unreliable	Yes	No
Demir et al. [2010]	Serial production line	Maximize throughput	Tabu search	Decomposition method	Unreliable	No	No
Woensel et al. [2010]	Flexible manufacturing	Throughput threshold	Powell's method	Expansion method	Reliable	Yes	No
Nahas et al. [2011]	Serial-parallel	Reduce WIP	Design of experiments	Simulation	Unreliable	Yes	No
Chiba [2015]	Serial production line	Reduce WIP	Dynamic programming	Simulation	Reliable	No	No
Shi and Gershwin [2016]	Serial production line	Maximize profit	Design of experiments	Decomposition method	Unreliable	No	No
Nahas [2017]	Serial production line	Minimize costs	Extended local search	Decomposition method	Unreliable	Yes	No
Motlagh et al. [2019]	Flexible manufacturing	Maximize throughput	Genetic algorithm	Simulation	Unreliable	Yes	No
Nejad et al. [2019]	Flexible manufacturing	Minimize cycle time	Branch-and-bound	Analytical model	Reliable	Yes	No
Kose and Kilincci [2020]	Serial production line	Maximize throughput, Reduce WIP	Simulated annealing	Simulation	Unreliable	Yes	No
Xi et al. [2020]	Serial-parallel	Reduce WIP	Local refinement search	Decomposition method	Reliable	No	No
Alaouchiche et al. [2021]	Serial production line	Maximize energy usage	-	Equivalent machine method	Unreliable	Yes	No
Diaz et al. [2021]	Serial production line	Maximize throughput, Reduce WIP	Genetic algorithm	Simulation	Unreliable	Yes	No
Gao [2022]	Serial production line	Maximize throughput	Tabu search	Expansion method	Reliable	Yes	No
Zhang et al. [2022]	Serial production line	Reduce WIP	Design of experiments	Decomposition method	Unreliable	No	No
Amjath et al. [2023]	General network	Maximize throughput	Design of experiments	Expansion method	Reliable	Yes	No
Shi and Gao [2023]	Serial production line	Maximize throughput	Tabu search	Genetic algorithm	Reliable	Yes	No
Kassoul et al. [2024]	Serial production line	Maximize throughput	Local search	Simulation	Unreliable	Yes	No
This research	General network	Maximize throughput	Simulated, annealing Tabu search, Greatest improvement	Decomposition method	Unreliable	Yes	Yes

Table 3.1: Overview of available literature on the BAP.

Chapter 4

Problem Description

This chapter provides a more detailed description of the buffer allocation problem (BAP) variant at X. The first section shows all assumptions that have been made in order to set boundaries and simplifications, but also creating a better understanding of the problem. Section 4.2 presents all the problem characteristics regarding objectives, parameters, and constraints. These characteristics are then materialized in a mathematical formulation shown in Section 4.3. Lastly, an example is presented to provide the reader with a tangible description of the problem, showcasing the challenges involved in optimizing buffer allocation.

4.1 Assumptions

This section provides a list of assumptions that have been taken into account while building the buffer allocation model. These assumptions help in stating the problem's boundaries and simplifications, whilst enhancing the understanding of the problem's context. The key assumptions:

- **Poisson Arrivals**: It is assumed that new production orders arrive according to a Poisson arrival process. This assumption is made such that each machine can be looked at as a simple queueing system.

- **Arrival Rates**: The arrival rates are considered to be static since the model will represent a single moment in time in which the arrival rates cannot change.

- **Active Time**: The machines are not scheduled every shift every day, a percentage of active time is taken into account.

- **Overall Equipment Efficiency**: The machines do not perform 100% reliable, an OEE percentage is taken into consideration.

- **Throughput Rates**: The throughput rates per machine are calculated according to Equation 3.3 in which the buffer allocation is the decision variable. This relation is used to relate the allocated buffer size to the corresponding throughput.

- **Processing Times**: The processing times are assumed to be constant and based on the average of data from the ERP system since the model will represent a single moment in time.

- **Setup Times**: The setup times are assumed to be constant and based on the average that is derived from the ERP system since the model will represent a single moment in time.
- **Flow Dispersion**: In the case of multiple subsequent machines after a process, a pre-specified percentage determines the amount of flow to each possible follow-up machine. This percentage needs to be predetermined to indicate how many processed cables per unit of time go to each possible subsequent machine.
- **Product Flows**: The arrival of production orders at subsequent machines depends on the throughput rate(s) of the previous machine(s). To regard each machine as a queueing system the arrival rates depend on the throughput of previous machines except for the initial production steps.

4.2 Definition

To get an exact definition of the BAP, certain aspects such as the objective, relevant parameters, and the corresponding constraints should be formulated. These aspects act as the foundation of the problem and will expose the characteristics of this variant of the BAP. The problem is defined as follows:

4.2.1 Problem Objectives

As was mentioned in the literature review, there is a multitude of reasons why a company would want to solve the BAP. Together with X the following objective was formulated that they value the most:

- **Maximize Production Output**: The goal is to maximize the total production output per unit of time of the manufacturing system shown in Figure 2.7. The company is revenue-driven, meaning that it aims to produce as much as possible such that there is more product to sell. In this case, the output is determined by the sheathing lines at the end of the manufacturing process. Consequently, the aim is to maximize the throughput rate of these machines which will depend on the supply of the other machines.

In addition to maximizing the output, the company also aims to reduce the WIP level. This reduction in WIP also reduces the throughput time, meaning that customers will receive their orders more quickly. Furthermore, by solving the BAP it should also become more clear where all the WIP is located and make it more plain for the production planner to decide which machines should be running and which not. In addition, available space for WIP is scarce. However, the minimization of WIP is not taken into account for the mathematical formulation since the company values production output more. The company provides a maximum number of reels that can be allocated to machines as a buffer.

4.2.2 Problem Parameters

The BAP at X can be characterized by several parameters. These are listed below:

- **Processing Times**: The time it takes before a specific machine completes a task.
- **Setup Times**: The time it takes to prepare a specific machine before it can start a task.
- **Reel Conversions**: A constant representing the ratio between the number of input and output reels.
- **Flow Dispersion**: A percentage showing what proportion of throughput of a specific machine is allocated to one of the possible follow-up machines.
- **Overall Equipment Efficiencies**: A percentage indicating the effective machine time if it is scheduled.
- **Scheduled Machine Time**: A percentage indicating the amount of time a machine is scheduled.
- **Product-mix Forecast**: A distribution showing what part of the incoming production orders belongs to a certain product group.
- **Degassing Capacity**: The maximum load capacity for the degassing chambers, indicating how many reels can fit in it.
- **Number Of Reels**: The maximum number of cable reels that can be allocated per type.

4.2.3 Problem Constraints

To complete the optimization problem one needs constraints that limit the solution space and make sure that only realistic possibilities are assessed. The following constraints are taken into account:

- **Throughput Dependency**: The throughput rate of preceding machines multiplied with the corresponding flow dispersion acts as the arrival rate for subsequent machines.
- **Reels Capacity**: The number of allocated cable reels per type cannot exceed the maximum available reels.
- **Location Capacity**: The locations City A and City B have a maximum number of cable reels that they are able to store.
- **Utilization Constraint**: The utilization of a machine can never exceed the value of one in order to prevent growing queues.

4.3 Mathematical Formulation

The previously mentioned objective, parameters, and constraints take shape in the mathematical formulation of this variant of the BAP. First, some indices are explained that help understand the formulation. The indices i and j are used to indicate the machines in the manufacturing system. These indices are part of a set N representing all the machines, while the subsets A and B represent the machines located in City A and City B respectively. Furthermore, the index k indicates the type of cable reel which can be part of the set K representing all cable reel types. Lastly, the symbols Q_A and Q_B are used to indicate the reel capacity at the locations City A and City B respectively. A clear mathematical representation of the mixed integer non-linear programming model that can be subjected to algorithmic solutions is shown below.

$$\text{Maximize:} \quad \sum_{i \in N} f_i \cdot TH_i \qquad \text{(Objective)}$$

Subject to:

$$TH_i = ST_i \cdot OEE_i \cdot \mu_i \cdot \left(1 - \frac{1-\rho_i}{1-\rho_i^{\sum_{k \in K} x_{ik}+1}}\right), \quad \forall i \in N \qquad \text{(Throughput)}$$

$$\rho_j = RC_j \cdot \frac{\sum_{i \in N, i \neq j} \left(y_{ij} \cdot TH_i\right)}{\mu_j}, \quad \forall j \in N \qquad \text{(Utilization)}$$

$$\rho_i \leq 1, \quad \forall i \in N \qquad \text{(Max. Utilization)}$$

$$\sum_{i \in A} \sum_{k \in K} x_{ik} \leq Q_A, \qquad \text{(Capacity City A)}$$

$$\sum_{i \in B} \sum_{k \in K} x_{ik} \leq Q_B, \qquad \text{(Capacity City B)}$$

$$\sum_{i \in N} x_{ik} \leq Q_k, \quad \forall k \in K \qquad \text{(Reel capacity)}$$

$$x_{ik} \leq M \cdot z_{ik}, \quad \forall i \in N, \forall k \in K \qquad \text{(Reel types)}$$

$$x_{ik} \in \mathbb{Z}^+, \quad \forall i \in N, \forall k \in K \qquad \text{(Positive integer)}$$

Above the mathematical formulation for the CRMH-BAP at X is stated. Starting with the objective function, TH_i indicates the throughput for machine i. This throughput is measured in the amount of reels that are finished by machine i per hour. This number is multiplied by the parameter f_i which indicates if the machine produces finished products. This parameter states a percentage and is equal to 100% if the machine purely produces final products and 0% if the machine purely produces semi-finished products. The throughput is calculated using the formula presented by Kwon [2006] and expanding it a bit to compensate for machine failures and schedules. The OEE_i and ST_i are both percentages indicating the overall equipment efficiency and the amount of time a machine is scheduled respectively (Throughput). The ρ_i indicates the utilization for machine i and is calculated using the combined throughput of the preceding machines that supply machine i (Utilization). The parameter y_{ij} indicates what percentage of the throughput of machine i is directed towards machine j. The machines **CL 1**, **CL 2**, **ISO 1**, and **ISO 2** in Figure 2.7 are the first machines in the production sequence and therefore have no preceding machines supplying WIP. Consequently, the $\sum_{i \in N} \left(y_{ij} \cdot TH_i\right)$ part is replaced with the arrival rates of orders for these machines in the utilization formula. Furthermore, the fraction is multiplied with the RC_j constant indicating the ratio of the number of input reels compared to the number of output reels. The next constraint is regarding the maximum value of the utilization, which cannot exceed the value of one otherwise the queue before a machine would only get bigger and is not desirable (Max. Utilization). In addition, there are some capacity constraints. The first one indicates the maximum available space for allocating reels at the production facility in City A. This is done by summing all buffers at each machine in City A and for each reel type. Q_A indicates the maximum number of reels that can be stored in City A (Capacity City A). This is also done for the production facility in City B (Capacity City B). There is also a limit indicating the available reels per type, which is denoted by Q_k (Reel capacity).

Moreover, each machine can only accept certain reel types depending on the type of cables that are processed by that machine. This is incorporated by using the binary parameter z_{ik} indicating if machine i can accept reels of type k and is multiplied by a very big number M (Reel types). Lastly, the decision variable x_{ik} can only be a positive integer since it is not possible to have a negative buffer or store only part of a reel (Positive integer).

Symbol	Description
Sets	
N	All machines (A and B are subsets of N for City A and City B)
K	All reel types
Indices	
i	Machine of set N
k	Reel type of set K
Variables	
x_{ik}	Number of reels of type k located at machine i
TH_i	Throughput of machine i in reels per hour
ρ_i	Utilization of machine i
Parameters	
f_i	Fraction of the throughput of machine i destined as final product
ST_i	Fraction of time machine i is scheduled
OEE_i	Overall equipment efficiency of machine i
μ_i	Service rate of machine i in reels per hour
RC_i	Ratio between input and output reels of machine i
y_{ij}	Percentage of the throughput of machine i goes to machine j
Q_A	Maximum number of reels that can be stored in City A
Q_B	Maximum number of reels that can be stored in City B
Q_k	Maximum number of reels that can be stored of type k
z_{ik}	Binary value indicating if machine i can accept reel type k
M	A very big number

Table 4.1: Description of variables and parameters.

4.4 Example

This section will provide an illustrative example of the problem presented in the previous section. The goal is to provide the reader with a tangible representation of the problem. In Figure 4.1 an example of a machine network can be seen of four machines. The objective of this network is to maximize the output of final products, meaning that the throughput of machines 3 and 4 should be maximized ($TH3 + TH4$). To simplify the example a bit further, the different locations have been left out.

Focusing on machine 1 it can be seen that on average 2.5 production orders arrive per hour. Furthermore, the machine has an OEE of 85%, is scheduled 75% of the time, and on average converts one input reel into two output reels. In addition, the machine is able to produce 8.2 output reels per hour and is able to process both $k1$ and $k2$ type reels. It is also important to mention that 70% of the produced reels are directed towards machine 3 and 30% is directed towards machine 4. The relevant parameters and other key elements of this example are further explained below:

- **Arrival rates**: The arrival rates of new production orders are required to calculate the utilization of the first production steps in the production sequence. In this case, it is required to calculate the utilization of machines 1 and 2.

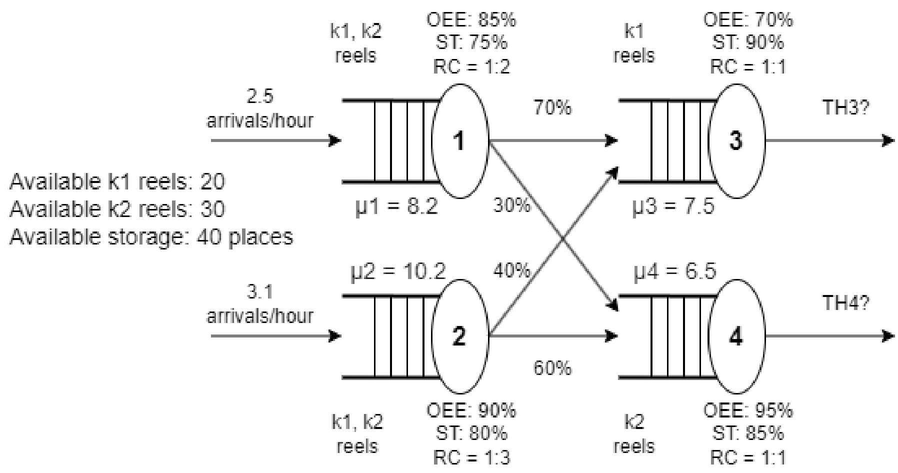

Figure 4.1: Example machine network and corresponding parameters.

- **Processing/Setup Times**: Each machine has an average process time it takes to produce one output reel. This processing time and setup time are included in the processing rates which is denoted by $\mu_1, ..., \mu_4$ corresponding to each machine in the example. The processing rates are measured in a number of output reels per hour.

- **Flow Distribution**: The flow distribution indicates what percentage of the output of a machine goes to another machine. In this example, 70% of the throughput of machine 1 goes to machine 3, 30% goes to machine 4, and 0% goes to machine 2.

- **Reel Types**: Not all machines can process all the different reel types. In the example, two cable reel types are shown, $k1$ and $k2$. While machine 1 can accept both reel types, machine 3 can only accept $k1$.

- **Reel Conversions**: The reel conversion (RC) factors indicate the ratio of output reels compared to a single input reel. The example shows that machine 1 produces 2 output reels out of a single input reel, while machine 3 has a 1:1 ratio.

- **Capacities**: In the example there are 20 reels of type $k1$ available, while there are 30 reels of type $k2$ available. However, these cannot all be allocated since the total capacity is 40 reels. The remaining 10 are used as empty reels or not at all.

- **Overall Equipment Efficiencies**: Each machine has its own OEE depending on how often a machine fails. In the example, machine 1 has an OEE of 75% for example, while machine 4 is more reliable with an OEE of 95%.

- **Scheduled time**: Due to limited skills and the number of employees each machine cannot be operational all the time. In the example a percentage is shown that indicates what proportion of available shifts the machine is scheduled (ST). For example, machine 2 is scheduled 80% of the available time.

With these parameters known, it is possible to try various buffer allocations that can be put in the mathematical formulation presented in Section 4.3. It is the goal of the generative algorithm to find new solutions that satisfy the constraints of the model and find improvements in an iterative manner. This will be further elaborated in the next chapter. For this simplified example the branch and bound algorithm was applied to determine the optimal buffer allocation. The results from the branch and bound algorithm can be seen in Appendix A. From the results, it became clear that there is a set of solutions that all have a similar output. The various solutions can be seen in Table 4.2. As a clarification, the allocated reels at machines 1 and 2 do not indicate the number of reels that subsequent machines will process. The calculated throughput for these allocations is ∼6.8692 reels per hour.

	Machine 1	**Machine 2**	**Machine 3**	**Machine 4**
Solution 1				
k1 reels	1	10	7	0
k2 reels	6	5	0	11
Solution 2				
k1 reels	2	10	7	0
k2 reels	5	5	0	11
Solution 3				
k1 reels	3	10	7	0
k2 reels	4	5	0	11

Table 4.2: Possible buffer allocations to maximize throughput for example.

Reviewing the results from the table above, there is a logical explanation for having three different buffer allocations all corresponding to the same throughput. It can be seen that only the number of reels stored at machine 1 differs between each solution. However, in each solution, the number of $k1$ reels and $k2$ reels sum up to 7 resulting in the same size buffer in each solution. Consequently, the corresponding outputs are all the same.

4.5 Solution Space

The example shown in Section 4.4 was solved by using branch-and-bound to come to an optimal solution. This was possible due to the limited solution space for this simplified example. It is possible to calculate the number of combinations in which a certain amount of reels are distributed over a number of machines. The formula used can be seen below [Albert, 2002]. In the equation m denotes the number of reels to distribute and n the number of machines.

$$\text{Number of possibilities } = \frac{(m+n-1)!}{n! \cdot (m-1)!} = \binom{m+n-1}{n} \tag{4.1}$$

In Table 4.3 the total number of possible solutions for the example can be seen. The two left columns indicate all the possible reel-type configurations while not exceeding the total capacity. The third and fourth column indicate in how many ways that reel type can be distributed over the available machines.

The subtotal column indicates how many possibilities there are for that specific configuration of reels. The total number of possibilities amounts to 504,273 for the ex-

Allocated k1 reels	Allocated k2 reels	Possibilities k1 reels	Possibilities k2 reels	Subtotal
10	30	66	496	32736
11	29	78	465	36270
12	28	91	435	39585
13	27	105	406	42630
14	26	120	378	45360
15	25	136	351	47736
16	24	153	325	49725
17	23	171	300	51300
18	22	190	276	52440
19	21	210	253	53130
20	20	231	231	53361
			Total:	504273

Table 4.3: Number of possible solutions for example.

ample. The same formula has also been used to calculate the approximate number of possible solutions for the BAP at X, taking into account the total capacity available for reels. It can be seen that the total number of possibilities is significantly larger than for the example (24 orders of magnitude) in Table 4.4. This means that the branch-and-bound algorithm would need to calculate a huge amount of additional linear programming relaxations taking up a large amount of time. The model was also built in AIMMS to determine the global optimum, however after 12 hours of running the program did not return a solution. In addition, this run time exceeds the company's requirements for usability.

Reel type	Available reels	No. machines	Possibilities
2800	155	8	$5.09458 \cdot 10^{11}$
2800 wide	45	5	211876
2500	29	3	465
2240	60	5	635376
1800	47	4	19600
		Total:	$\sim 6.25073 \cdot 10^{29}$

Table 4.4: Possible solutions for the BAP at X.

Due to branch-and-bound being computationally intensive as shown by Morrison et al. [2016] and the exponential increase of the solution space making it impossible to completely enumerate, the choice was made to apply other optimization algorithms instead. These methods would not necessarily find a global optimum like branch-and-bound but can provide a near-optimal solution that can act as a guideline for the production planners. Consequently, the mathematical formulation will not be solved through iterations due to computational limits.

4.6 Summary

To conclude, this chapter presented a comprehensive description of the CRMH-BAP encountered at X. The chapter began by outlining the key assumptions that were made while formulating the problem description. These assumptions contribute to defining the scope and the problem context. With the assumptions stated it was possible to exactly define the problem. First, the objectives of the model were explained after which the required parameters were presented that are needed for solving the problem. Furthermore, the formulated constraints help in setting the boundaries in the solution space. These objectives, parameters, and constraints were then translated into a mathematical formulation that can be solved by a computer model. Moreover, an example was presented to provide the reader with a more tangible feel of the problem. Lastly, preliminary tests indicated that the situation at X cannot be solved exactly due to computational limits. This chapter acts as the foundation upon which various algorithms are explored, experiments are conducted, and the BAP variant of X can be precisely described.

Chapter 5

Solution Approach

In this chapter, various solution methodologies and strategies are explored to tackle the buffer allocation problem (BAP) variant of X. A rationale is provided in which the choice for a specific set of algorithms is made (Section 5.1). The remainder of the sections each address a specific aspect of the solution development and implementation (Section 5.2). In addition, the matters of data integration (Section 5.3), solution space, scalability and performance optimization (Sections 4.5 and 5.4) are discussed.

5.1 Method Selection

This Section explains why the choice for specific generative and evaluative method(s) was made. As can be concluded from Section 3.3, various approaches and combinations of methods exist regarding the BAP. Below the various options are evaluated and substantiated about why these options are or are not taken into account.

5.1.1 Generative Method

Various generative methods have been explained in the literature review. The first method is simulated annealing and putting this algorithm into the context of the BAP an initial buffer configuration would be given to the algorithm after which randomly reels are taken from one buffer and inserted at another buffer to generate new solutions. An example of an insertion operation can be seen in Figure 5.1. As a result, this algorithm can explore multiple directions in the solution space depending on the starting temperature, cooling coefficient c, and threshold temperature. However, this is also a downside. Due to these random swaps it could take a long time before the algorithm finally finds the optimal solution and it is not guaranteed if the algorithm finds the optimal solution. Finally, the simulated annealing algorithm is considered for the BAP at X due to its ability to explore the solution space. This ability to explore larger areas of the solution space compared to local search based methods could pose a viable asset in the search for the optimal buffer allocation. Consequently, the papers utilizing extended local search or local refinement search in Table 3.1 are not considered. Furthermore, the simulated annealing algorithm is not an overly complex algorithm to program and is therefore considered in the remainder of this report.

The second generative method that is highlighted in the literature review is dynamic programming. This algorithm generates a solution by solving several sub-problems in several stages with different states. Placing the algorithm in the BAP context, the machine network is divided into several stages in which the machines that produce the final product are the starting stage. This also makes it more complex to program this algorithm. The major benefit of this algorithm is that it does not randomly search for a new solution and already looks for an optimal configuration given a certain state. However, generating a complete buffer allocation for a given initial state takes significantly longer when compared to the simulated annealing algorithm. Due to the large solution space for the given machine network of X the dynamic programming algorithm is not considered further.

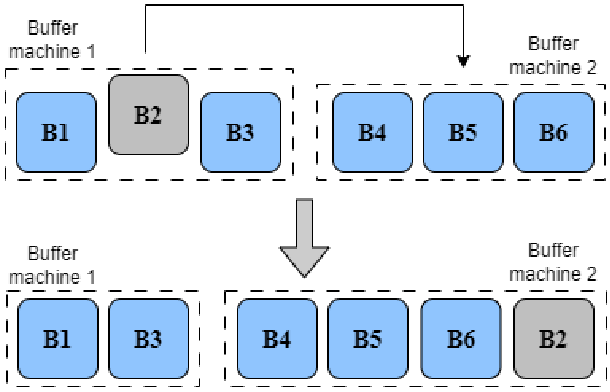

Figure 5.1: Visualization of an insertion operation.

The third method is tabu search and this method also has an in-built prevention method for not ending up in a local optimum similar to simulated annealing. Comparing tabu search against simulated annealing, the main difference is that tabu search looks more actively for improvements while simulated annealing generates solutions randomly. In addition, the tabu search algorithm can have a higher programming complexity due to the additional checks and storage of the tabu list elements compared to simulated annealing. However, this increase in programming complexity is still considered achievable. Consequently, tabu search is also taken into account to compare against the performance of the simulated annealing algorithm. Lastly, another generative method would be the design of experiments in which only predetermined buffer allocations are tested. However, this method does not comply with the requirements of the company who desire a model that can find a buffer allocation on its own and is therefore not considered further. Many other generative algorithms exist that are or can be used for the BAP. However, these are either considered too complex, not suitable for this problem variant or require additional literature research.

To conclude, various generative methods will be used to create solutions that can be compared. These methods are simulated annealing and tabu search. Both algorithms will be evaluated upon the final result as well as runtime. In addition, a different type of generative method will be tested that was not explicitly mentioned in the available literature. This method is called the greatest improvement algorithm (Section 3.3.1). This method starts with an empty buffer for every machine and then iterates over every buffer by adding one reel and evaluating the improvement of the corresponding KPI. The iterations that led to the greatest improvement will be the starting solution for the next iteration until all available reels are allocated. Compared to the other algorithms this method only ends up at a single buffer allocation and it it takes significantly longer to generate this solution due to the extra operations but it only needs to generate a single solution. As a result, the greatest improvement algorithm will also be evaluated.

5.1.2 Evaluative Method

In Section 3.3.2 a set of evaluative methods is presented. These methods include expansion, aggregation, and decomposition. The expansion method extends the machine network by adding another queue in between machines. This additional queue represents the amount of WIP that is blocked before it can start being processed on the second machine. In addition, the method provides a formula that relates the allocated buffer size to the corresponding output of that same machine. However, due to the additional queues required, the machine network becomes very complex. Moreover, these additional queues also make it hard to create a flexible model in which it is possible to scale the problem and machine network. Secondly, the aggregation method also provides a relation between the allocated buffer size and the output of a system. The method does this by simulating a sequential production line of two machines as a single process. This process can be repeated for larger numbers of machines. Sadly, it is only applicable for machines in serial production lines, meaning that it is not suitable for the machine network of X.

Thirdly, the decomposition method assesses every machine in the network as a single queue and Kwon [2006] provided a formula for this method in which the buffer size directly contributes to the throughput of machine. Consequently, it is possible to calculate the throughput of the first machines in the production sequences which then dictate the amount of products per unit time received by the follow-up machines. It is then possible to calculate the throughput levels of these machines. As a result, this process can be repeated until the throughput levels of the last machines. The complexity of the machine network is not necessarily a hindrance in this method since multiple flows leading to a single machine can simply be summed. Consequently, the decomposition method is taken into account for the BAP at X. This is visualized in Figure 5.2. In this figure, **TH1** and **TH2** indicate the throughput rates of machines 1 and 2 respectively. Summing these throughput rates results in the arrival rate for machine 3 ($\lambda\mathbf{3}$). The utilization of machine 3 ($\rho\mathbf{3}$) can be calculated by dividing the arrival rate by the service rate ($\mu\mathbf{3}$).

Besides the three previously mentioned evaluative methods there is also the matter of simulation. Where the expansion, aggregation, and decomposition methods take a more analytical approach, simulation tries to evaluate a buffer configuration through an approximation of reality. The drawback of this method is that to end up with a final result often multiple runs are required since various aspects of a simulation depend on uncertainty. Consequently, it could take up significant time for a final conclusion can be derived for the company which is not desirable. Furthermore, the company would proba-

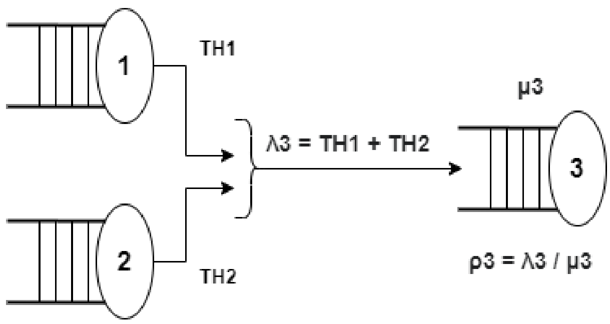

Figure 5.2: Summation of flows that act as arrival rate for follow-up machine.

bly require additional software for which the employees would need training which makes simulation less approachable. Therefore, simulation is not considered for the evaluative method.

To conclude, the decomposition method is considered to be the most adequate and versatile evaluative method for the BAP variant at X. The formula introduced by Kwon [2006] for the decomposition method that relates the buffer allocation to the throughput of a machine is already incorporated in the mathematical formulation in Section 4.3. Other evaluative methods besides the previously mentioned do exist, but these are very complex and highly scenario-specific and are therefore out of the scope.

5.2 Heuristic Design

This section will show how the complete heuristic is used for solving the CRMH-BAP. First, the sequential method is shown together with the decomposition method that is used for evaluating the different solutions generated by the simulated annealing, tabu search, and greatest improvement algorithms. These algorithms are also elaborated with a pseudocode that can be applied for allocating buffers at X.

5.2.1 Sequential Approach

As is explained previously the solution approach that will applied consists of two different methods that are executed sequentially. First, a solution is created by a generative algorithm. In this research, various generative algorithms will be applied to test which is superior. The evaluative method utilizes decomposition in which each machine is seen as a single queue. By then using a formula that is presented below it is possible to calculate the flow happening between each machine depending on the allocated buffer size x to that machine.

$$\text{Throughput} = \text{Scheduled time} \cdot \text{OEE} \cdot \mu \cdot \left(1 - \frac{1-\rho}{1-\rho^{x+1}}\right) \tag{5.1}$$

This equation is very similar to the one presented by Kwon [2006] except for the inclusion of scheduled time and overall equipment efficiency (OEE). Both the scheduled time and the OEE are a percentage. Consequently, the throughput of a machine can be seen as the average throughput per hour it can achieve over a longer period of time. Since the model is intended as a medium-term, one to a few months, guideline for buffer allocations this equation can be regarded as adequate.

5.2.2 Simulated Annealing

One of the generative methods used for the experiments is simulated annealing. A pseudocode for this algorithm used for buffer allocation can be seen in Algorithm 2. The method starts by initializing a starting solution x_0 that satisfies all the constraints. The other variables, x_c, x_b, and x_n indicate the current, best, and next buffer allocation. In the starting solution, two random buffers are taken of which one buffer is incremented by one reel while the other is reduced by one reel. In the pseudocode, these buffers are named $randomBufferUP$ and $randomBufferDOWN$ respectively. Should the resulting buffer allocation still satisfy all constraints then this solution is filled in the objective function. In addition, an acceptance probability is calculated using the current temperature according to the Boltzmann distribution. If the result from the objective function is improved over the previous best found buffer allocation then this allocation is accepted as the current buffer allocation. Should this not be the case then this worse allocation can still be accepted with a chance equal to the calculated probability p. Each time a new best solution is found, the allocation x_b and objective function result E_b are stored. The last step before the process is repeated is calculating the new temperature. This is calculated by multiplying the current temperature with a constant α. The whole process is repeated until the temperature falls below a prespecified threshold.

Algorithm 2: Pseudocode for simulated annealing of the BAP

```
Dictate initial buffer allocation x_0
Initiate current buffer allocation and energy; x_c, E_c
Initiate best buffer allocation and energy; x_b, E_b
Initiate next buffer allocation and energy; x_n, E_n
while T > T_min
    validCandidate = False
    while validCandidate = False
        randombBufferUP = Rnd() * N           'Determine randomly which buffer to change
        randombBufferDOWN = Rnd() * N         'N indicates the number of machines

        x_n = x_c                             'Update the buffer allocation
        x_n(randomBufferUP) = x_n(randomBufferUP) + 1
        x_n(randomBufferDOWN) = x_n(randomBufferDOWN) - 1

        if randomBufferUP = randomBufferDOWN Or ValidSolution(x_n) = False
            validCandidate = False
        Else
            validCandidate = True
        end
    end
    E_n = ObjectiveFunction(x_n)              'Calculate energy of new buffer allocation
    p = Exp((E_c - E_n)/T)

    if E_n > E_c Or Rnd() < p
        x_c = x_n
        E_c = E_n
    end

    if E_c > E_b
        x_b = x_c
        E_b = E_c
    end

    T = T * α                                 'Update the temperature
end
Return x_b as best solution
```

Some functions are used that are also used by the other algorithms. These functions are *ValidSolution* and *ObjectiveFunction*. The first function checks if a new buffer allocation satisfies all the constraints stated in the mathematical formulation and can be achieved by the company. The second function is used to determine the total throughput of the system. The codes for these functions can be found in Appendix B.

5.2.3 Tabu Search

The second generative method is the tabu search algorithm. The pseudocode for this generative method can be seen in Algorithm 3. The tabu search method is initialized similarly to the simulated annealing algorithm with the current, best, and next buffer allocations x_c, x_b, and x_n as well as their corresponding energies E_c, E_b, and E_n. In addition, the object *tabuList* is created to keep track of visited solutions. Instead of a decreasing temperature the tabu search algorithm stops after a certain amount of iterations. The algorithm then checks all neighbours through insertion operations and a double for-loop. In these for-loops, every neighbour is assessed and the neighbour with the best result that is not already in the tabu list will be used as the starting solution during the next iteration.

Algorithm 3: Pseudocode of a tabu search algorithm for the BAP

```
Dictate initial buffer allocation x_0
Initiate current buffer allocation and energy; x_c, E_c
Initiate best buffer allocation and energy; x_b, E_b
Initiate current best buffer allocation and energy; x_cb, E_cb
Initiate next buffer allocation and energy; x_n, E_n
Initialize tabu list
while iter < maxIter
    E_cb = 0
    for i = 1 to length(x_c)                          'Loop through all neighbours
        for j = 1 to length(x_c)
            if i ≠ j
                x_n = x_c
                x_n(i) = x_n(i) + 1
                x_n(j) = x_n(j) − 1
                E_n = ObjectiveFunction(x_n)
                notInList = True

                for k = 1 to length(tabuList)   'Check if solution is already in tabu list
                    if compareArrayes(x_n, tabuList(k)) = True
                        notInList = False
                    end
                end

                if validSolution(x_n) = True and E_n > E_cb and notInList = True
                    x_cb = x_n
                    E_cb = E_n
                end
            end
        end
    end
    x_c = x_cb
    E_c = E_cb
    update(tabuList)                                  'Add new solution to tabu list
    if x_c > x_b
        x_b = x_c
        E_b = E_c
    end
    iter = iter + 1                                   'Update number of iterations
end
Return x_b as best solution
```

After the best-performing neighbour that is not in the tabu list has been found, the buffer allocation (x_{cb}) and result (E_{cb}) are stored. Thereafter, the tabu list is updated. Should the tabu list exceed the limit when the newly found solution is added then the first added solution still in the tabu list is removed. Subsequently, it is checked if the newfound solution has a better performance than the best-found solution so far. If this is the case then the best buffer allocation (x_b) and result (E_b) are updated as well. Lastly, the number of iterations is incremented by one. This process is repeated until a set maximum amount of iterations has been performed or if all neighbours are stored in the tabu list.

5.2.4 Greatest Improvement

In contrast to the other two generative methods, the greatest improvement algorithm is a constructive algorithm instead of testing multiple options. The pseudocode for the greatest improvement method can be seen in Algorithm 4. The method starts with a buffer allocation that has no WIP allocated to any machine. The algorithm adds a single reel per iteration to a buffer that indicates the biggest improvement of the KPI given a specific state. In this case, the state can be identified as having a certain amount of reels already allocated to specific machines. This means that from a given state each possibility is assessed and the option with the biggest improvement is then stored and used as a base for the next iteration. This process is repeated until it is not possible to add a single reel to any machine. What remains is a final buffer allocation in which all available reels are allocated.

Algorithm 4: Pseudocode of greatest improvement algorithm for the BAP

```
Dictate initial empty buffer allocation x_0
Initiate current buffer allocation and energy; x_c, E_c
Initiate best buffer allocation and energy; x_b, E_b
Initiate next buffer allocation and energy; x_n, E_n
while noFurtherIterations = False          'Add reels until constraints are not satisfied
    valid = False
    for i = 1 To N                          'Loop through machines to find the biggest improvement
        x_n = x_b
        x_n(i) = x_n(i) + 1
        if validSolution(x_n) = True
            E_n = ObjectiveFunction(x_n)
            valid = True
            if E_n > E_c
                x_c = x_n
                E_c = E_n
            end
        elseIf valid = False and i = N
            noFurtherIterations = True
        end
    end
    if noFurtherIterations = False                                 'Store the final solution
        x_b = x_c
        E_b = E_c
    end
end
Return x_b as best solution
```

5.3 Integration of Data

Data plays an important role of this research since it determines a lot of the parameters that are required by the solution method. This section explain in what manner the data is gathered and how it is implemented in the model. Furthermore, the impact of this data on the buffer allocation is discussed.

5.3.1 Processing and Setup Times

The processing and setup times play an important role in the mathematical model. Both the processing and setup times determine the service rates with which each machine can process a cable per unit of time. The service rates which will be calculated in the mathematical model are indicated in reels per hour that can be processed. Since each machine is able to process a multitude of cable types the corresponding processing times often differ. In addition, each production order can also differ in cable length which also affects the processing time. Consequently, an average of the processing and setup times needs to be taken into account since it is not possible to simulate a distribution in the analytical approach of this research. For the presented model in Section 4.3 only the product groups presented in Section 2.3 are taken into consideration. Many other different cable types are processed in the factory but they only represent a small portion of the total throughput compared to these five main product groups. In order to determine the average service rate per machine taking into account these product groups a list of historical data is required. This historical data is gathered through the ERP system of X. This is further explained in Section 6.1. Equation 5.2 shows how the average service rate based on historical data is calculated. The number of processed reels N stands for a certain number of processed reels over a certain period of time and each has its own processing and setup time.

$$\text{Service rate } \mu = \frac{\text{Number of processed reels } N}{\sum^{N}(\text{Process time (hours)} + \text{Setup time (hours)})} \tag{5.2}$$

5.3.2 Flow Dispersion

Each machine has a set of possible follow-up machines except for the final processes. The flow dispersion dictates how much of the throughput of a single machine is directed to each of the possible follow-up machines. Consequently, the flow dispersion has a significant effect on the arrival rate of production orders for each machine. The arrival rate is required to calculate the utilization of a machine and the utilization is directly involved in calculating the throughput given a certain buffer size. This can be seen in Equation 5.1 in which ρ is the utilization. It is difficult to determine the flow dispersion based on historical data since one cannot predict the future. Consequently, the flow dispersion will be based on both the forecast for the given product groups in Section 2.3 as well as the experience and knowledge of employees at X. These employees know from experience about the performance of each machine and what cable type is best for which machine. The flow dispersion of one machine to a follow-up machine will be indicated by a percentage and the total flow dispersion from a single machine adds up to 100%. This is also seen in Figure 4.1.

5.3.3 Reel Conversions

Reel conversions are constants that dictate how many output reels can be expected from one input reel for a specific reel. For example, if the reel conversion constant for a machine is two then one can expect on average two output reels for every input reel. This is visualized in Figure 5.3. There are multiple reasons why a cable to be processed could be divided over multiple reels. Firstly, a process often makes a cable bigger in diameter since a layer is added for example, an insulation layer, shielding, or outer jacket. Consequently, less windings over the width of a reel are possible and bigger reels or multiple reels are required to store the processed cable. The reel conversion constants also play a significant role in buffer allocation. For example, if the reel conversion constant is two and a processed cable is distributed over two output reels then the arrival rate for the follow-up machine also increases and as a result, the utilization as well as throughput are affected according to Equation 5.1. These reel conversion constants are determined both by looking at historical data and the number of reels that came out of each machine for a production order as well as experience from employees.

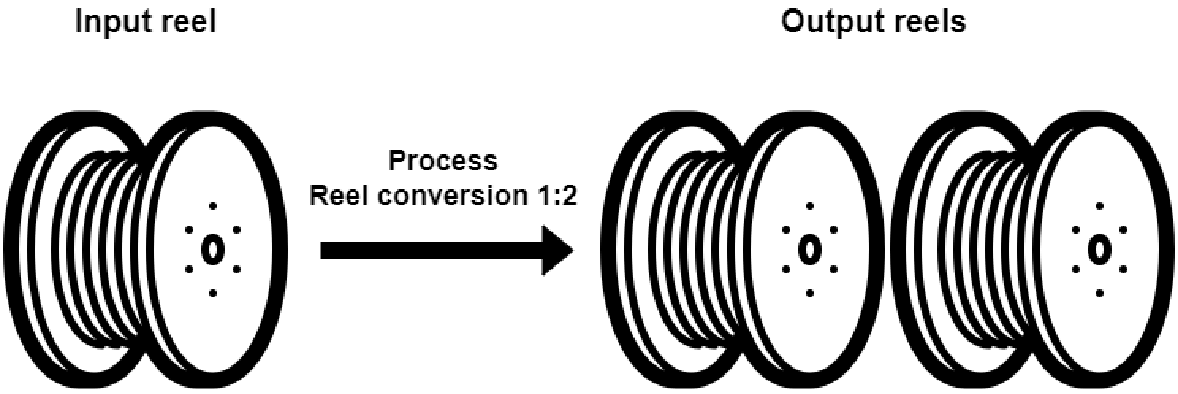

Figure 5.3: Visualization of a cable being processed and spread over two output reels with one input reel.

5.3.4 OEEs and Scheduled Time

The OEEs and scheduled time proportions represent the effectiveness and active periods respectively for each machine. The OEE tells something about how effective the machine can process cables during a shift. For example, an OEE of 50% indicates that the machine only achieves 50% of its potential service rate. The proportion of scheduled time indicates how often the machine is scheduled over a longer period of time. Various reasons exist for why a machine cannot be scheduled for every shift, for example, due to a limited number of employees that are able to operate the machine. Both the OEE and scheduled time affect the buffer allocation as can be seen in Equation 5.1. Before the final throughput is determined it is multiplied by these factors which are both expressed in percentages. As a result, the average throughput over a longer period of time can be approximated. These throughput levels are used to determine the arrival rates for follow-up machines and thus affect the performance of the system and buffer allocation. The values for these factors will have to be based on historical data since it is difficult to determine what the machines will do in the future.

5.3.5 Capacities

There are multiple capacity constraints in the mathematical formulation in Section 4.3. Firstly, there are capacities which are location-dependent, meaning that City A and City B have different capacities for different matters. For example, the number of reels that can be stored in City A inside the factory differs from the number of reels that can be stored in City B due to the size difference of the factory. In addition, City B has a smaller number of machines and as a result the degassing capacity can be smaller. Degassing is required after the CDCC process, which stands for 'completely dry curing and cooling'. Without degassing it takes very long before all the gas that is trapped inside the insulation material to escape. In Table 5.1 the capacities per location can be seen.

Location	**Degassing capacity**	**Reel capacity**	
		Total	*Available for WIP*
City A	32	395	335
City B	25	150	90

Table 5.1: Overview of the capacities per location (values are changed regarding confidentiality).

Besides the locations, there are also different types of reels used at the factory. Each of these types is available in a limited number. One should take into account that not all cable reel types can be accepted by all machines. This is due to the dimensions of the cable reel jacks that lift the cable reel such that it is able to unwind into the machine. In addition, some of the available reels should be reserved for being empty since every machine, except the degassing stage, requires at least two reels. One reel is the input reel with the cable to be processed while the the second reel is used to wind the processed cable. In Table 5.2 the available reels per type and corresponding machines can be seen. The names of the machines correspond to Figure 2.8.

Reel type	**Total**	**Available for allocation**	**Suitable machines**
2800	190	145	EI Degassing; EI Sheathing line 1; EI Sheathing line 2; EI Sheathing LINE 3; EI Sheating line 9; EI Shielding line 1; EI Shielding line 2; EI Shielding line 3
2800 wide	65	35	EI Sheathing line 1; EI Sheathing line 2; LCH Degassing; LCH Jacketing line 2; LCH Shielding line 5
2500	45	39	EI CDCC 1; EI CDCC 3; LCH CDCC 2
2240	92	55	EI Drumtwister 2; EI Sheathing line 1; EI Sheathing line 3; EI Shielding line 4; LCH Drumtwister 4
1800	65	57	EI Drumtwister 1; EI Insulation line 1; EI Insulation line 2; EI Shielding line 3

Table 5.2: Overview of the available reels per type (values are changed regarding confidentiality).

5.4 Scalability and Performance Optimization

Putting scalability in the context of the BAP it is difficult to create a larger or smaller instance of an initial manufacturing environment. Since the model is highly dependent on the characteristics of the machine network. For example, if a machine is either added or removed it requires updating various product flows and the corresponding constraints. In addition, the flow dispersion levels need to be updated and if a machine is added its characteristics also need to be known. However, it should be possible to make this process very user-friendly by creating a form in the digital model that allows the user to add or remove machines. The arrays indicating product flows between machines and throughput constraints should then automatically be updated. Looking at the problem from another perspective it is very easy to adjust the number of available reels that can be used as buffer. Moreover, if a certain parameter is changed over time then this is also very easily updated. For example, if a processing time or reel conversion constant is changed for a machine then this is easily updated in the model making it very flexible.

Regarding performance optimization, it can immediately be stated that the greatest improvement algorithm is a very efficient algorithm regarding computational time. One of the improvements that can be made over the presented tabu search algorithm is to provide the model with an initial solution that is already performing relatively well compared to the majority of other suitable solutions. This can be done by first performing a variable neighbourhood search or other algorithm that looks widely across the solution space. By then giving this initial solution to the tabu search algorithm the algorithm could find the same solution with fewer iterations than it would otherwise have taken without this improved initial solution. This same method could also be applied to the simulated annealing algorithm. An improved initial solution would possibly guide the simulated annealing algorithm in a direction with improved solutions. However, this algorithm does depend on randomness meaning that the computational time varies before it arrives at an optimal solution. For the experiments, multiple initial buffer allocations have been tested in order to provide these algorithms with an already improved buffer allocation compared to the current approach. Regarding the greatest improvement algorithm, there are no algorithmic improvement opportunities since it requires checking every possibility before adding a single item of WIP.

5.5 Summary

To summarize, this chapter provided elaborate insights into the solution approach regarding the different algorithms. It is explained why various generative methods could pose a suitable approach and the choice for a specific evaluative method is substantiated. Pseudocodes are stated that give a clear idea of how the corresponding algorithm can be put to use in the context of a BAP. This approach indicates which of the generative methods is the most suitable for the environment at X. Furthermore, the integration of data is discussed on how the data is gathered and in what way it affects the buffer allocation. In addition, the solution provided insights in why an optimization algorithm would be more suitable for the situation at X. Lastly, the scalability of the BAP in this manufacturing facility is put up for discussion together with potential performance improvements of the algorithms.

Chapter 6

Experimental Setup

In this chapter the experimental setup is outlined for assessing which solution approach is the best suited for the BAP variant at X. Crucial elements such as the data collection and pre-processing (Section 6.1) are discussed and the experimental scenarios (Section 6.2) are stated. Furthermore, the software and computational environment (Section 6.3) used are elaborated upon. Thereafter, the matters of parameter tuning (Section 6.4), performance metrics (Section 6.5), and experimental procedure (Section 6.6) are covered. Lastly, a data sensitivity is presented (Section 6.7). The goal of this chapter is to establish a foundation upon which the effectiveness of the different solution approaches can be assessed.

6.1 Data Collection and Pre-processing

First, the average service rates for all machines had to be determined. This meant that historical data had to be acquired that indicated the processing times per processed cable reel. This historical data is acquired from the ERP system at X. Before the data is retrieved, first a decision needs to be based on the time period over which the service rates will be approximated. This decision is mostly based on consistency and the entire manufacturing system being operational. The consistency has to do with every machine having the same parts and processing parameters over this time period. For example, a sheathing machine could have its extruder replaced enabling it to process more cable per unit of time. These instances affect the average service rate. In addition, some of the machines shown in the network in Figure 2.7 are reasonably new (less than a year). Consequently, it is only possible to take a time period in which these machines are fully operational. Taking these factors into account, a time period concerning six months was left.

Knowing the time period over which data could be gathered, all the information could be retrieved from the ERP system. However, this data is not immediately usable it first needs some filtering before the correct and adequate data is acquired. For each produced reel by a machine in the given time period, it should be known what type of cable it was, how long it took to process the cable, and how long it took to set the process up. This was done by creating a code in Excel VBA that would automatically loop through all the produced reels and determine the required information. This filter would also consider things like maintenance procedures out of the raw data. The number of reels per cable type per machine can be found in Appendix C. In addition, the average processing time was also determined.

Besides the service rates, there is other data to be acquired. This includes the OEEs, scheduled time percentages, reel conversion constants, and the flow dispersion. These parameters could not all be purely based on historical data. For example, for the flow dispersion, the data is partly based on experience and expectation. The historical data does not predict how the product flow is going to look in the future. In addition, the historical data also includes exceptions in which different routings were followed. These exceptions should be neglected when determining the general product flow. In Table 6.1 the dispersion for all the product flows that are visualized in Figure 2.7 are indicated. The percentage represents the proportional output leading from the machine before the arrow to the machine after the arrow. It should be noted that all flows leading from one machine add up to 100%.

Machine 1		**Machine 2**	**Percentage**	**Machine 1**		**Machine 2**	**Percentage**
EI CDCC 1	→	EI Degassing	100%	EI Sheathing line 3	→	EI Sheathing line 3	15%
EI CDCC 3	→	EI Degassing	100%	EI Sheathing line 3	→	EI Shielding line 4	40%
EI Conform line 1	→	EI CDCC 1	55%	EI Sheathing line 3	→	Exit	45%
EI Conform line 1	→	EI CDCC 3	20%	EI Sheathing line 9	→	Exit	100%
EI Conform line 1	→	LCH CDCC 2	25%	EI Shielding line 1	→	EI Sheathing line 1	70%
EI Conform line 2	→	EI CDCC 1	10%	EI Shielding line 1	→	EI Sheathing line 2	25%
EI Conform line 2	→	EI CDCC 3	65%	EI Shielding line 1	→	EI Sheathing line 9	5%
EI Conform line 2	→	LCH CDCC 2	25%	EI Shielding line 2	→	EI Sheathing line 1	40%
EI Degassing	→	EI Shielding line 1	69%	EI Shielding line 2	→	EI Sheathing line 2	30%
EI Degassing	→	EI Shielding line 2	1%	EI Shielding line 2	→	EI Sheathing line 9	30%
EI Degassing	→	EI Shielding line 3	20%	EI Shielding line 3	→	EI Drumtwister 2	50%
EI Degassing	→	LCH Shielding line 5	10%	EI Shielding line 3	→	Exit	50%
EI Drumtwister 1	→	EI Sheathing line 1	55%	EI Shielding line 4	→	EI Sheathing line 1	10%
EI Drumtwister 1	→	EI Sheathing line 3	45%	EI Shielding line 4	→	EI Sheathing line 3	90%
EI Drumtwister 2	→	EI Sheathing line 1	25%	LCH Degassing	→	LCH Shielding line 5	100%
EI Drumtwister 2	→	EI Sheathing line 2	75%	LCH CDCC 2	→	LCH Degassing	100%
EI Insulation line 1	→	EI Drumtwister 1	70%	LCH Shielding line 5	→	EI Sheathing line 9	15%
EI Insulation line 1	→	EI Shielding line 3	30%	LCH Shielding line 5	→	LCH Drumtwister 4	40%
EI Insulation line 2	→	EI Drumtwister 1	100%	LCH Shielding line 5	→	LCH Jacketing line 2	45%
EI Sheathing line 1	→	EI Shielding line 4	10%	LCH Drumtwister 4	→	LCH Jacketing line 2	100%
EI Sheathing line 1	→	Exit	90%	LCH Jacketing line 2	→	Exit	100%
EI Sheathing line 2	→	Exit	100%				

Table 6.1: Overview of the flow dispersion (values are changed regarding confidentiality).

In Table 6.3 the other machine parameters can be seen. Since it is difficult to predict how the OEE of each machine is going to behave in the future it is mainly based on historical data. However, for future calculations, this can easily be adjusted. The proportion of scheduled time is based on historical data in which the number of scheduled shifts was divided by the maximum number of possible shifts in the given time period. The reel conversion constants are purely based on experience for colleagues since these can often differ for a machine, depending on the production order size and customer wishes. Therefore, the presented values are based on the input of employees. In the last column, the service rates that are presented indicate the number of reels that can be processed per hour by the corresponding machine. Furthermore, the arrival rates for the first machines in the production sequences can be seen in Table 6.2. These arrival rates indicate the arrival of new production orders per hour.

Machine	**Arrival rate**
EI Conform line 1	0.053
EI Conform line 2	0.091
EI Insulation line 1	0.374
EI Insulation line 2	0.132

Table 6.2: Arrival rates in system (values are changed regarding confidentiality).

Machine	OEE	Scheduled	Reel conversion	Service rate
EI CDCC 1	95%	97%	2.7	0.597
EI CDCC 3	85%	96%	2.7	0.385
EI Conform line 1	83%	97%	1	0.197
EI Conform line 2	55%	92%	1	0.390
EI Degassing	100%	100%	1	0.014
EI Drumtwister 1	61%	45%	1	0.516
EI Drumtwister 2	59%	91%	2	0.540
EI Insulation line 1	89%	54%	1	0.658
EI Insulation line 2	40%	39%	1	0.473
EI Sheathing line 1	84%	95%	2	0.732
EI Sheathing line 2	66%	48%	2	0.479
EI Sheathing line 3	88%	59%	2	0.756
EI Sheathing line 9	75%	87%	2	0.657
EI Shielding line 1	86%	98%	1	0.450
EI Shielding line 2	90%	64%	1	0.279
EI Shielding line 3	90%	76%	1	0.545
EI Shielding line 4	50%	31%	1	0.344
LCH CDCC 2	90%	90%	2.7	0.132
LCH Degassing	100%	100%	1	0.149
LCH Drumtwister 4	78%	91%	2	0.028
LCH Jacketing line 2	75%	95%	2	0.890
LCH Shielding line 5	86%	98%	1	0.348

Table 6.3: Overview of the machine parameters (values are changed regarding confidentiality).

6.2 Experimental Scenarios

In order to assess the performance of all proposed algorithms an experimental scenario will need to be set up. This scenario will reflect the production environment at X at the moment of writing this report. By putting the algorithms to work on this scenario a better idea of how these algorithms would work in a real-life situation is created. The parameters used for this scenario are presented in the tables in the previous section. In the case of tabu search and the greatest improvement algorithm, a single run would be satisfactory, since these algorithms would end up at the same buffer allocation provided that the parameters have not been changed. However, the simulated annealing algorithm includes randomness. Consequently, multiple runs are required to get a reliable result. This also means that every time the company wants to redetermine the buffer allocation it is required to perform these multiple runs. The minimum required number of runs is determined later on in Section 6.6.

Another scenario of interest would be the situation in which all machines would perform optimally and can be active all the time. Consequently, the parameters OEE and scheduled time would be equal to 100% for all machines. It is interesting for the company to know how the buffer allocation changes when all machinery functions optimally. Besides the fact that this would provide an indication of the maximum achievable output of the manufacturing facility, it also provides insights into how the allocation of reels shifts between machines. The company can then assess if the available space at each production line is sufficient to store the allocated number of reels.

6.3 Software and Computational Environment

Regarding the software for the buffer allocation model various options exist. Even though a high-level programming language like Python would be more efficient in solving this problem the choice was made to use Excel VBA programming. The main reason for this software tool is the familiarity it has among the employees of X. It would take additional time to get to know another programming language or software to be able to use it throughout the facility. The company does not wish to invest this additional time and prefers the use of Excel VBA. Consequently, the buffer allocation model is made using Excel VBA and the dashboard is also created in Excel. The hardware being used for the experiments is a Lenovo Thinkpad P51 from 2017. It is equipped with an Intel Core i7-7700HQ CPU clocked at 2.80 GHz and an NVIDIA Quadro M1200 graphics card. The total RAM memory equals to 16.0 GB.

6.4 Parameter Tuning

Besides the parameters that are presented in Section 6.1, there are also algorithm-specific parameters. These parameters affect the performance of the algorithm and will be further discussed below. Regarding the greatest improvement algorithm, there are no parameters that can be tuned. Consequently, this algorithm will not be discussed.

6.4.1 Initial temperature and cooling rate

Firstly, the simulated algorithm depends on various parameters. Both the initial temperature and cooling rate affect the number of solutions that will be assessed and the probability with which a worse-performing solution. In addition, the initial buffer solution given to the algorithm also affects the algorithm. Since the simulated annealing algorithm only performs a limited number of iterations the initial buffer solution has an influence on the fact that the algorithm will or will not find the optimal buffer allocation. However, the randomness of the simulated annealing algorithm also affects this probability but this randomness is not a parameter that can be tuned. Both the initial temperature and cooling rate are tuned by keeping all other parameters constant and performing experiments with various values for the initial temperature and cooling rate. It should be mentioned that while the initial temperature is tuned the cooling rate is constant and vice versa. The performance indicators that will be tuned upon are the expected output in reels per hour and the running time of the algorithm in seconds. In Table 6.4 the results of the tests can be seen. For the initial temperature variation the cooling rate was kept constant at 0.999 and for the cooling rate variation the initial temperature was kept constant at 10,000.

In order to get a more reliable result each variation has been ran ten times in order to reduce the effect of randomness. Reviewing the results from the initial temperature and cooling rate variations it can be seen that increasing the initial temperature leads to an increase in run time. This is also the case when the cooling rate is increased however, increasing the cooling rate leads to a significantly larger increase in run time since the incremental steps shown in the table lead to more iterations than the incremental steps regarding the initial temperature. Furthermore, it should also

Initial temp.	Throughput	Run time	Cooling rate	Throughput	Run time
100	0.81819	7.128	0.99	0.81819	1.232
1,000	0.81815	9.421	0.999	0.81816	11.745
10,000	0.81821	11.769	0.9999	0.81821	121.135
100,000	0.81815	14.056	0.99999	0.81828	1312.961

Table 6.4: Tests for initial temperature and cooling rate.

be mentioned that the throughput is not significantly changed when either the initial temperature or cooling rate is increased. Consequently, the choice was made to use an initial temperature of 10,000 and a cooling rate of 0.9999.

6.4.2 Tabu list size and number of iterations

Regarding the tabu search algorithm, the tabu list size is one of the tuneable parameters besides the maximum number of iterations. The tabu list size indicates the number of solutions that can be stored in the tabu list. These stored solutions are forbidden for the algorithm to visit again while they remain in the tabu list. Consequently, the larger the tabu list size the lower the chance that the tabu search algorithm will visit previously assessed solutions again thus lowering the probability of ending up in a local optimum. The maximum number of iterations is also a parameter that can be tuned. This parameter indicates the maximum number of iterations that can be performed by the algorithm. Various tabu list sizes and maximum number of iterations have been tried. The results of these tests can be seen below in Table 6.5. For the tabu list size increments the maximum number of iterations was kept constant at 100. In addition, for the maximum number of iterations, the tabu list size was kept constant at 10.

Tabu size	Throughput	Run time	Max. iterations	Throughput	Run time
5	0.81855	47.725	10	0.81850	4.711
10	0.81855	51.323	100	0.81855	50.922
15	0.81855	55.856	250	0.81855	124.781
20	0.81855	56.394	500	0.81855	248.839

Table 6.5: Tests for tabu list size and maximum number of iterations.

From the table above it can be concluded that increasing the tabu size or the maximum number of iterations has little effect on the final throughput of the system. However, the run time does increase each time the tabu size increases. Moreover, the run time increases significantly more if the maximum number of iterations is increased. When the number of iterations increases from 250 to 500 the run time also doubles approximately. Consequently, the tabu list size will be kept at 10 and the maximum number of iterations at 2500 to have a run time similar to the simulated annealing algorithm.

6.4.3 Initial buffer allocation

As was mentioned, the initial buffer allocation affects the performance of the simulated annealing algorithm. Moreover, this initial buffer allocation also affects the performance of the tabu search algorithm. The initial buffer allocation dictates how many operations are required to reach an optimal buffer allocation. Consequently, various buffer allocations are tested to check this effect on both the simulated annealing and tabu search algorithm. Only a few possible initial buffer allocations have been tested due to the large solution space. The results of these tests can be seen in Table 6.6. These instances include scenarios in which most of a single type of reel has been allocated to a single machine (allocations 2 until 6), an instance in which almost all reels are allocated to specific machines while the rest remains with a single reel (allocation 7), and a situation that closely resembles a realistic situation (allocation 1). Allocation 1 has been generated by the greatest improvement algorithm, since this is a constructive heuristic it is interesting to see if the other two algorithms would find an improvement. The tests have been conducted with an initial temperature of 10,000 and a cooling rate of 0.9999 for the simulated annealing algorithm. For the tabu search algorithm, the list size was kept at 10 and the maximum number of iterations was set at 250.

Machine	Reel type	Alloc. 1	Alloc. 2	Alloc. 3	Alloc. 4	Alloc. 5	Alloc. 6	Alloc. 7
EI CDCC 1	2500	11	11	11	27	11	11	27
EI CDCC 3	2500	11	11	11	1	11	11	1
EI Degassing	2800	51	148	51	51	51	51	148
EI Drumtwister 1	1800	10	10	10	10	10	44	44
EI Drumtwister 2	2240	15	15	15	15	56	15	56
EI Insulation line 1	1800	10	10	10	10	10	1	1
EI Insulation line 2	1800	8	8	8	8	8	1	1
EI Sheathing line 1	2800	13	1	13	13	13	13	1
EI Sheathing line 1	2800 wide	1	1	41	1	1	1	41
EI Sheathing line 1	2240	12	12	12	12	1	12	1
EI Sheahting line 2	2800	8	1	8	8	8	8	1
EI Sheathing line 2	2800 wide	1	1	1	1	1	1	1
EI Sheathing line 3	2800	7	1	7	7	7	7	1
EI Sheathing line 3	2240	8	8	8	8	1	8	1
EI Sheathing line 9	2800	29	1	29	29	29	29	1
EI Shielding line 1	2800	26	1	26	26	26	26	1
EI Shielding line 2	2800	4	1	4	4	4	4	1
EI Shielding line 3	2800	17	1	17	17	17	17	1
EI Shielding line 3	1800	17	17	17	17	17	1	1
EI Shielding line 4	2240	8	8	8	8	1	8	1
LCH CDCC 2	2500	7	7	7	1	7	7	1
LCH Degassing	2800 wide	13	13	1	13	13	13	1
LCH Drumtwister 4	2240	17	17	17	17	1	17	1
LCH Jacketing line 2	2800 wide	17	17	1	17	17	17	1
LCH Shielding line 5	2800 wide	13	13	1	13	13	13	1
Initial throughput		0.81853	0.61476	0.67034	0.66676	0.80245	0.79804	0.39107
Simulated annealing	Throughput	0.81853	0.81536	0.80789	0.81077	0.81398	0.81459	0.81368
Simulated annealing	Run time	120.43	121.44	121.81	122.52	122.17	121.33	123.23
Tabu search	Throughput	0.81855	0.81855	0.81855	0.81855	0.81855	0.81855	0.818028
Tabu search	Run time	124.78	122.31	123.99	124.22	125.11	124.78	124.57

Table 6.6: Tests with various initial buffer allocations.

From the table above it can be concluded that the initial buffer allocation does not significantly affect the run time of the algorithms. However, the initial buffer allocation does affect the final throughput. It can be seen that between allocations 1 and 3 with the simulated annealing algorithm, a difference of at least 0.01 reels per hour occurs. For the tabu search algorithm only allocation 7 made an impact on the final throughput of the system. However, the difference is only marginally. Consequently, the choice was made to have allocation 1, which resembles a realistic allocation, as the initial buffer allocation for the experiments.

6.5 Performance Metrics

The performance metrics used in the experiments are similar to the ones used in the parameter tuning. The performance metrics can be divided into two categories. The first category has to do with the solution quality, thus telling something about the expected performance of the found solution. The second category is concerned with the computational efficiency of the model. The quality of the solution will be measured by the corresponding throughput of the resulting buffer allocation. This throughput is measured in produced reels per hour. The company desires to achieve a throughput level as high as possible. The computational efficiency of the model will be measured by the run time of the various algorithms. The shorter the run time, the better an algorithm performs provided that solution quality remains constant or even improves.

6.6 Experimental Procedure

For the experiments the parameter settings that are outlined in Section 6.4 will be used for the simulated annealing and tabu search algorithms. Furthermore, each algorithm is run ten times in order to ensure an average run time is established. The simulated annealing algorithm also requires more than one replication due to the included randomness in the algorithm which affects the final result after each replication. Consequently, a multitude of replications are required to create a reliable output value. The scenario that is given to the algorithms represents the order intake over a period of about six months. The results of each algorithm are compared to each other as well as the actual achieved output over these six months. Based on the determined buffer allocation, corresponding output, and the run time of each algorithm a recommendation is made for the company.

6.7 Data Sensitivity

In order to assess the impact of data uncertainties a sensitivity analysis is conducted. With this analysis it was possible to review the effect of certain data types on the solution performance. The results of this analysis can be seen in Table 6.7. First, a baseline was generated with all the parameters and variables previously described. The following data parameters are changed to assess the impact; OEE, scheduled time, RC, arrival rates, and service rates. Each time one of these parameters was adjusted all the other parameters were kept at their baseline values. For the OEE and scheduled time the values were set at 100%, while the other parameters were simply doubled.

The values in the table indicate the throughput of the corresponding buffer allocation found by the algorithm. This throughput indicates the number of produced reels per hour. From these values, it can be concluded that all data parameters show an improvement in the solution quality if their values are increased. However, some of the parameters show a greater sensitivity than others. The greatest improvement is seen

Algorithm	Baseline	OEE (100%)	ST (100%)	RC (x2)	Arrival rates (x2)	Service rates (x2)
Simulated annealing	0.81823	0.81913	1.48891	1.60481	1.16859	0.81975
Tabu search	0.81855	0.81930	1.50148	1.63846	1.18793	0.81976
Greatest improvement	0.81853	0.81930	1.50094	1.64049	1.18343	0.81976

Table 6.7: Corresponding throughput levels with the different parameter settings.

when the reel conversion constants are doubled. The corresponding throughput also doubles compared to the baseline performance. Unfortunately, doubling these constants is not feasible. This would mean that all machines would suddenly double their added value, which is unlikely. The second and third most influential data parameters are the scheduled time and arrival rates. The arrival rate increases the utilization of all machines, while the increase in active time means additional output over a certain time period. The OEE and arrival rates have the least impact on the solution quality. This is due to the decrease in machine utilization. Consequently, the formula shown in 5.1 will not lead to an increase in throughput.

6.8 Summary

In this chapter, the outlines for the experimental procedure are presented. Beginning with the data collection and pre-processing the parameters are highlighted that are required to be able to determine a buffer allocation. Thereafter, the experimental scenarios are elaborated in which two scenarios were discussed. The first one is the current situation, while the second situation would indicate the performance in a more ideal situation in which all machines never fail and are always active.

In addition, the use of Excel VBA is substantiated and the corresponding hardware used for the experiments is highlighted. Thereafter, the parameter tuning phase was discussed in which the algorithm-specific parameters are tuned for a better performance of the solution. For the experiments themselves, the various KPIs are explained and the experimental procedure is covered. Lastly, the sensitivity of the data parameters is tested and analyzed. This gave insight into the effect of certain parameters on the solution quality.

With the experimental setup and procedures in place, it was possible to conduct the actual experiments and evaluate the performance of each algorithm. The following chapters will show and discuss the acquired results from these experiments. The goal is to understand the effectiveness of the proposed solution.

Chapter 7

Results and Discussions

This chapter presents the results, comparisons, and analyses of the conducted experiments. These results contribute to assessing the proposed solution for addressing the buffer allocation problem (BAP) variant encountered at X. First, a baseline is established for the current situation (Section 7.1), after which a more optimal scenario is reviewed (Section 7.2). Thereafter, a comparison is made between the current approach at X regarding buffer allocation and the proposed buffer allocation (Section 7.3). Lastly, the findings are further discussed and elaborated upon in Section 7.4.

7.1 Baseline Performance

To establish a benchmark for the proposed buffer allocation the algorithms have been applied to a real-life situation at X. This scenario includes the actual system performance over a period of six months, such as the OEEs, scheduled time, arrival rates, service rates, and reel conversion constants. The results for this baseline are plotted in Figure 7.1. A set of ten replications, in which the calculation and algorithm are repeated, was run in order to get a reliable result for the simulated annealing algorithm. The vertical axis indicates the throughput in reels per hour for the corresponding buffer allocation. Appendix D shows how the number of replications is determined.

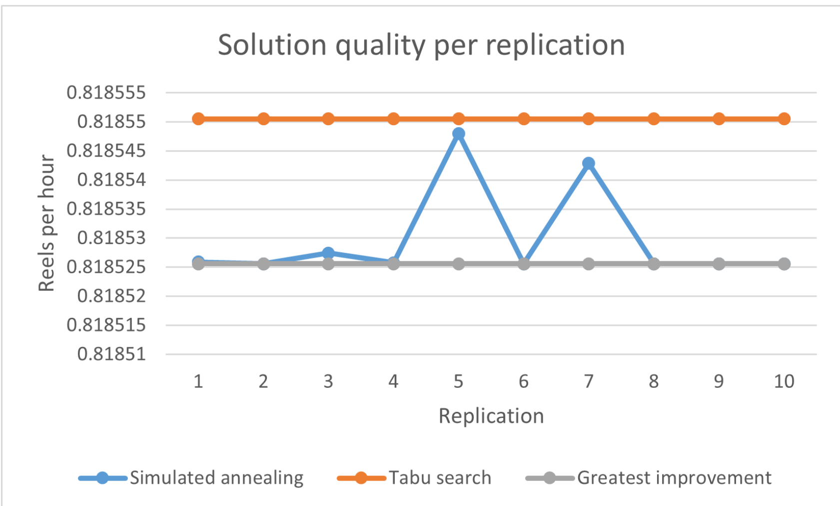

Figure 7.1: Solution qualities of the algorithms per replication.

In addition to the performance of the buffer allocations found by the algorithms, the run time is an algorithm-specific performance indicator. The run time of each algorithm is also taken into account for the final decision regarding the choice of algorithm. The corresponding running times to the replications performed for the baseline can be seen in Figure 7.2.

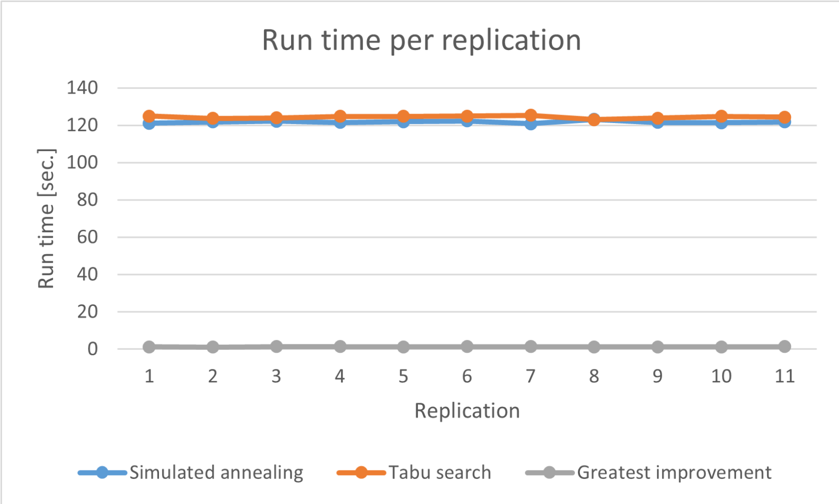

Figure 7.2: Run time of the algorithms per replication.

In the table below the throughput and average running time over the ten replications for each algorithm can be seen. Again the throughput is measured in reels produced per hour for the corresponding buffer allocation. For the simulated annealing algorithm the average throughput over the ten replications was taken.

	Simulated annealing	**Tabu search**	**Greatest improvement**
Throughput [reels/hour]	0.81853	0.81855	0.81853
Run time [sec.]	121.76	124.38	1.33

Table 7.1: Results for the baseline experiments.

7.2 Scenario-Specific Analysis

Besides the real-life situation, another scenario was taken into account for the experiments. In this scenario all machinery performs optimally, meaning an OEE of 100%. In addition, all machines would be available all the time, meaning the proportion of scheduled time is equal to 100%. This could provide meaningful insights into how the buffer allocation would shift in a more optimal scenario. Should the efficiency or active time increase for the machinery then X is able to anticipate how to shift the WIP in order to maximize their throughput. Again for these experiments, ten replications were performed. The results of these replications can be seen in the graph in Figure 7.3. Both the results from the simulated annealing and tabu search algorithms are constant for every replication which is expected. It should be mentioned that the grey line indicating the greatest improvement performance is beneath the orange line of the tabu search algorithm and the blue line indicating the simulated annealing algorithm includes randomness.

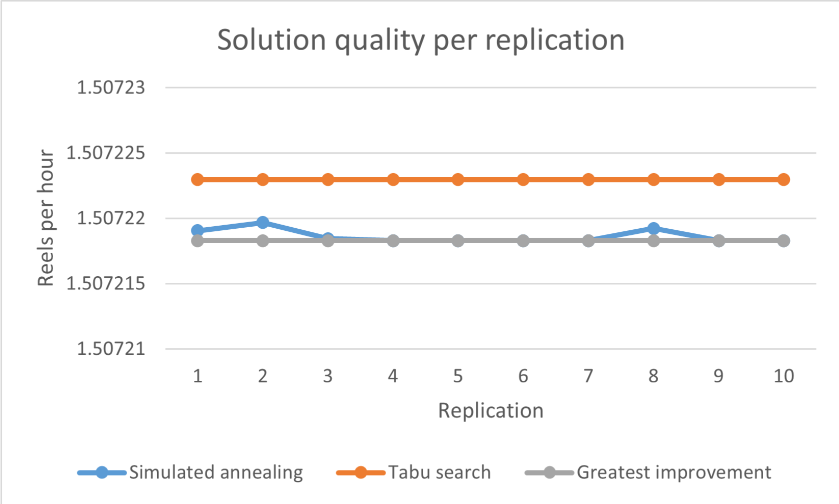

Figure 7.3: Solution qualities for optimal scenario.

The corresponding running times for the replications performed in this scenario can be seen in Figure 7.4. Again the three different algorithms have distinguishable running times. The throughput and average running times for the different algorithms are summarized in Table 7.2. The throughput for the simulated annealing algorithm is again an average over the ten replications. The results itself will be further discussed in Sections 7.3 and 7.4.

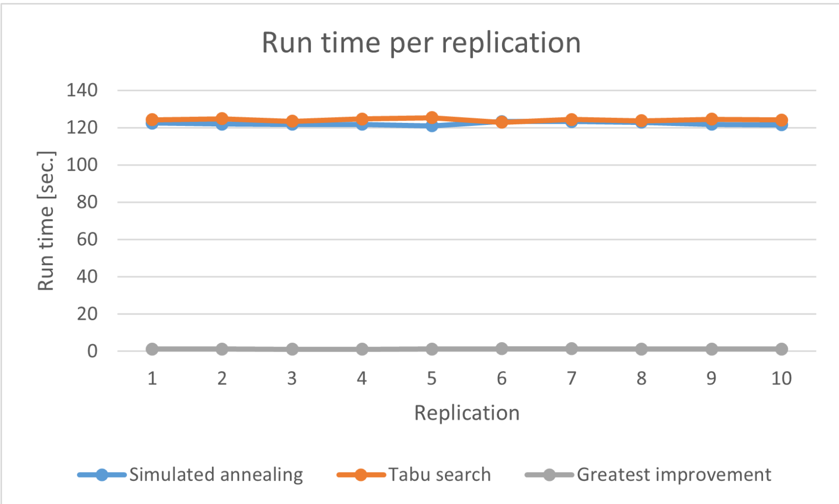

Figure 7.4: Running times for optimal scenario.

	Simulated annealing	**Tabu search**	**Greatest improvement**
Throughput [reels/hour]	1.50722	1.50722	1.50722
Run time [sec.]	122.17	124.23	1.32

Table 7.2: Results for the scenario-specific experiments.

7.3 Comparison with Current Approach

Comparing the results acquired from the baseline experiment to the actual achieved throughput over those same six months it can be concluded that all algorithms achieve an increase in average throughput per hour. This is indicated in Table 7.3. All algorithms indicate that an increase of at least 17% in average throughput should be possible once the buffers are allocated according to their solution. Extrapolating these values for a week means that per week an additional ∼20 reels can be produced.

	Current approach	**Simulated annealing**	**Tabu search**	**Greatest improvement**
Throughput	0.69579	0.81853	0.81855	0.81853
Increase		17.64%	17.64%	17.64%

Table 7.3: Comparison between current approach and proposed solution.

The other major benefit of the proposed approach cannot be quantified through numbers. As was explained in Section 2.4, the current production planning strategy is mainly focused with scheduling the machines that have the largest amount of WIP that is waiting to be processed by those machines. This WIP is measured in machine hours or average machining time per reel. Using this strategy the throughput is not necessarily maximized. Instead, the proposed solution aims to have an efficient allocation of buffers as possible such that each machine that is scheduled has the lowest probability of running out of WIP. Consequently, the scheduled machines can therefore process more cables since there is a lower risk of those machines having to wait for cables that can be processed by that machine. This effect can also be seen due to the increase in throughput shown in Table 7.3. Therefore, having this guideline itself is a significant benefit that cannot be quantified since the model is very versatile and can be used to model various situations.

7.4 Discussion of Findings

In this discussion section, a critical examination of the experimental results is performed through an evaluation of the proposed solution that addresses the BAP variant at X. This in-depth discussion highlights both the strengths and limitations of the proposed solution, thereby establishing an understanding of its applicability and impact. The key components being discussed are; analysis of performance metrics, strengths and limitations, interpretation of scenario-specific results, sensitivity analysis, and the practical implications and applicability.

7.4.1 Performance Metrics Analysis

Looking at the results summarized in Tables 7.1 and 7.2 it can be seen that all algorithms are very close to each other regarding the throughput per hour. However, the randomness included in the simulated annealing algorithm does play a factor in the decision for a preferable algorithm. Due to this randomness, the algorithm is inconsistent in determining a suggested buffer allocation. Therefore, the usability of this algorithm as a guideline for production planning is lower compared to the other two. In addition, the solutions found by the simulated annealing algorithm always performed worse compared to the solution of the tabu search algorithm.

The main difference between the other two algorithms is the running time. Where the greatest improvement algorithm performs steadily below the two-second mark, the tabu search algorithm takes on average between 122 and 125 seconds. This is a rather significant difference in the running time of a factor $\sim$93. It could be argued however that if the total number of to be allocated reels increases the running time for the greatest improvement algorithm also increases, while the tabu search algorithm running remains constant for the same tabu list size and maximum iterations. Still, the running time of 124 seconds can be regarded as reasonable since it is significantly lower than the 12+ hours it would otherwise have taken if it had been solved exactly. However, if the company wants to quickly analyze different scenarios the greatest improvement algorithm would be more suitable.

7.4.2 Strengths of the Proposed Solution

There are multiple strengths associated with this new approach for X to tackle their BAP variant. Firstly, the running time of the model is still within a workable limit such that it can be applied on a daily basis or even more frequently. This allows the company to quickly adapt or predict new scenarios. In addition to the relatively low running time, all the algorithms also indicate an increase in throughput should the buffers be allocated in this manner.

Furthermore, the employee responsible for the production planning does not have to think of a strategy on its own since the presented model can be used as a guideline. Consequently, decisions regarding which machines to schedule in the upcoming shifts or where to create an additional buffer are made more easily. To help in this decision-making process a dashboard with an overview of the entire manufacturing system is made that is able to show the current, target, and minimum buffer for each machine. An example of this dashboard can be seen in Appendix E.

Lastly, the presented model enables the company to predict how the buffer allocation would change if certain characteristics of the manufacturing system change. By changing certain parameters such as the OEE or service rates for example the company can simulate how the new buffer allocation would look like and how the corresponding throughput changes. Moreover, it is also possible to see what effect an increase or decrease of incoming orders has on the buffer allocation and corresponding throughput by changing the arrival rates. This enables the company to better prepare for future situations.

7.4.3 Limitations and Areas for Improvement

There are some limitations and areas for improvement still included in the presented model. Firstly, while most of the parameters can be easily changed in the model it is still a rather complex process if one would want to remove or add new machines in the digital model. If a machine is removed or added then this would require to reroute, delete, or add all the linked product flows to this machine. Consequently, the equations regarding the throughput and utilization for the involved machines will need to be updated. This is significantly more work than just changing a single value and is also more prone to debugging.

A similar case occurs if one would want to add or remove an additional reel type. This means that the method that checks if a proposed allocation is valid will need to be changed such that it includes the constraints regarding the new reel type. For future improvements, the process of changing the number of machines or reel types could be more simplified. This could be done by automatically adding or removing equations and constraints if one would change these characteristics. The user would simply have to tell how a new machine would fit in the manufacturing system and provide the flow dispersion values. This is the same for the reel type in which the user provides the knowledge about which machines are able to use the new reel type and how many are available for allocation.

7.4.4 Interpretation of Scenario-Specific Results

In addition to the scenario representing the current situation, another scenario was inserted in the model to see how the buffer allocation and corresponding throughput would change. This scenario would represent a near-optimal situation in which all machines are 100% operational and active all the time. In Table 7.2 it can be seen that in this scenario the throughput increases significantly compared to the current real-life situation, the number of reels produced per hour more than doubles. Besides this increase in throughput, it is also interesting to see how the buffer allocation changes between these scenarios. Table 7.4 shows the total number of stored reels per machine for both situations. Furthermore, it is also interesting to see the differences between the tabu search and greatest improvement algorithms. The simulated annealing algorithm is not considered in this case since it does not return the same buffer allocation consistently.

Machine	**Tabu search**		**Greatest improvement**	
	Current situation	*Optimal scenario*	*Current situation*	*Optimal scenario*
EI CDCC 1	12	11	11	11
EI CDCC 3	11	12	11	12
EI Degassing	53	64	51	59
EI Drumtwister 1	11	12	10	14
EI Drumtwister 2	15	15	15	15
EI Insulation line 1	13	11	12	12
EI Insulation line 2	8	6	8	6
EI Sheathing line 1	16	30	26	34
EI Sheathing line 2	10	8	9	7
EI Sheathing line 3	12	29	15	36
EI Sheathing line 9	37	24	29	19
EI Shielding line 1	27	22	26	20
EI Shielding line 2	4	3	4	3
EI Shielding line 3	36	19	34	18
EI Shielding line 4	8	8	8	8
LCH CDCC 2	6	6	7	6
LCH Degassing	13	16	13	16
LCH Drumtwister 4	14	13	14	13
LCH Jacketing line 2	18	15	17	15
LCH Shielding line 5	12	12	13	12

Table 7.4: Buffer allocation in each scenario for both the tabu search and greatest improvement algorithms.

Moreover, the simulated annealing algorithm frequently finds a solution that is not an improvement over the solution of the greatest improvement algorithm. The first difference when going from the current situation to the optimal scenario is a shift of reels from the machines "EI Sheathing line 9" and "EI Shielding line 3" to the machines "EI Sheathing line 1" and "EI Sheathing line 3". This is both the case for the tabu search and the greatest improvement algorithm. These are all machines that produce final products that leave the system, so when all machines are fully efficient and always operational the buffers for these machines will be more evenly spread. Another change is the increase in buffer for the "EI Degassing". Again this is the case for both algorithms. The reason for this is that this machine already has a very high OEE and active time compared to the other machines. Consequently, the relative increase in throughput is smaller than that of the other machines and thus requires an increase in buffer to able to reliably supply subsequent machines according to Equation 5.1.

A noticeable difference between the tabu search and the greatest improvement algorithm is the allocated buffers at "EI Sheathing line 1" and "EI Sheathing line 9" in the current situation, 16 and 37 compared to 26 and 29 respectively. A possible reason for this difference could be due to algorithm characteristics. The greatest improvement algorithm does not visit this solution since it first encounters bigger improvements if buffers are added to "EI Sheathing line 1" while the tabu search algorithm does visit these neighbouring solutions.

7.4.5 Robustness and Sensitivity Analysis

From the data sensitivity analysis conducted in Section 6.7 it became clear that some parameters have a bigger influence on the throughput than others. The most influential parameters are the reel conversion constants that determine how many output reels there are compared to input reels. If all these constants double then the final throughput would also roughly double. However, this would not lead to an increase in turnover because the cable would just be divided over multiple reels. Therefore, this parameter is not worth improving. The two parameters following up are worth it. Both the proportion of active time for each machine and the arrival rates of new orders significantly influence the throughput. These parameters could be improved by having additional personnel such that more machines can be operated in each shift and attracting more production orders from clients. Increasing the OEE and service rates for each machine also increases the throughput, but marginally compared to the previously two mentioned parameters. Once the arrival rates exceed the service rate for example, then it would make sense to increase the service rate otherwise the system would overflow with orders. The increasing service rate would have a bigger impact at that moment.

7.4.6 Practical Implications and Real-World Applicability

Regarding the practicality of the model, there are very few adjustments required in order for it to be usable in the daily operations of X. Employees for whom this model is of use will need a short tutorial about how the tool works, what can be changed, and how to operate it. The fact that very little training and adaptions are required is another benefit of this proposed solution. Without any additional costs, the production planning has a guideline and modelling tool in order to tackle the BAP at X. The only practical implication that could be improved is the user interface. This interface should provide a clear and comprehensive overview of all the buffers as well as the parameter tab in which the user is able to make adjustments to the model.

7.5 Summary

In conclusion, this chapter displays the significant findings derived from the results acquired through experiments. It indicates that the proposed solution improves the current situation at X at the moment of writing this report. In addition, the findings are put up for discussion in order to make decisions about which algorithm performs best, the strengths of the proposed solution, limitations, and what is required for a practical application of the model. In the subsequent chapter, a comprehensive overview of the findings is presented together with the final recommendations, possible future research gaps, and the final remarks.

Chapter 8

Conclusions and Recommendations

This concluding chapter covers the key insights, findings, and recommendations from the conducted research of the BAP variant encountered at X. In addition, several conclusions can be drawn from this new approach compared to the current situation at the company. Lastly, various possible future research directions are suggested together with the final remarks which conclude this report.

8.1 Summary of Findings

This section provides a comprehensive yet concise overview of the results and corresponding analysis derived from the experiments. The main findings that were acquired from this CRMH-BAP are as follows:

- From the conducted experiments it became clear that both the tabu search algorithm and greatest improvement algorithm pose as a viable generative algorithm in order to tackle the BAP variant at X. Due to its randomness and lack of performance regarding solution quality the simulated annealing algorithm is not considered suitable.

- Compared to the current approach applied by the company the presented solution reliably indicates that there is still room for improvement regarding throughput. According to the model, an increase in throughput of 17% should be possible.

- From the optimal scenario it became clear that an even further increase in throughput is possible should certain parameters be improved. Should the proportion of active time or OEE increase then the model also shows an increase in output for the manufacturing system.

- In addition to the increase in throughput, the optimal scenario also gave insight into how the buffer allocations shift between the machines when parameters are changed. This knowledge can be very useful for the company in order to increase its adaptability.

8.2 Conclusions

In this section, the derived overarching conclusions from this CRMH-BAP at X are presented. The conducted research highlighted the following critical aspects:

- Besides the throughput KPI that was mainly focused on the model also highlights other factors of the manufacturing. For example, it is able to show the utilizations of each machine which enables the company to quickly determine the bottleneck in the system.
- The presented model is able to determine the buffer allocation for various scenarios, which increases its flexibility. Whether dealing with fluctuating arrival rates of production orders or a reduced number of available reels, the model is able to redetermine the corresponding buffer allocation it considers as most efficient.
- The model consistently provides buffer allocations that surpass the performance of the current approach regarding production planning. Providing this knowledge about room for improvement is a valuable asset for the company.
- Besides indicating the room for improvement the model also poses as a valuable tool for the production planners. The tool can act as a guideline that helps the production planners make decisions regarding which machines to schedule on a short-term basis.

8.3 Recommendations

The recommendations section outlines actionable suggestions based on the research findings and insights. The following recommendations are proposed for X:

- **Implementation**: A complete implementation is recommended in order to improve the throughput as much as possible. Additional insight into the performance of the manufacturing system can also be gathered through modelling various scenarios.
- **Operational Enhancements**: In order to improve the throughput even further it is recommended to improve the active time and OEE for each machine. Other factors such as additional production orders would also improve the performance of the system however, these are not in control of X.
- **Integration with IT Infrastructure**: It is crucial that the model is integrated with the company's ERP system in order to get the most recent and reliable data. This step ensures real-time monitoring and seamless adaptation to changing conditions.
- **Continuous Monitoring and Evaluation**: It is recommended to redetermine the buffer allocation with up-to-date parameters on a monthly basis. In addition, as a guideline, the model could be consulted on a daily/weekly basis in order to determine the production strategy.

- **Training and Knowledge Transfer**: A short tutorial is recommended for production planners who want to use the model. It is essential to know how the parameters can be changed and how the VBA codes work for future improvements.

8.4 Future Research Directions

In this section, promising future research directions regarding the BAP variant at X are presented. The following promising research directions were found:

- **Effect of Parameters**: Further exploring the effect of data parameters on the buffer allocation and corresponding throughput. Developing an accompanying strategy in order to achieve these improved parameters.
- **Multi-Objective Optimization**: Investigating the possibility of achieving the same throughput while minimizing the number of used reels.
- **Expansion Possibilities**: Look into the opportunity of expanding the current machinery with additional production capacity. Determine what type of machine would have the most effect on the throughput.
- **Further Automation**: Explore the integration of upcoming technologies such as artificial intelligence (AI). This could further automate the process of creating a production planning strategy.

These research directions could further improve the ongoing solving of the BAP at X and further improve the daily operations.

8.5 Final Remarks

In this final section, a profound gratitude towards Company X is expressed for providing the opportunity to work on this challenging variant of the BAP. This collaboration allowed for contributions to further explore these types of problems while providing additional insight into the daily operations at the company. In addition to the company, all other stakeholders who have been of help in this research are thanked.

This chapter concludes the exploration of the capacitated and restricted machine hours buffer allocation problem at X, concerning its challenges, proposed approach, actionable recommendations, and opportunities for further research. The aim is to provide an additional guideline for the daily operations at the company while providing new knowledge about this BAP variant. The world of manufacturing logistics is explored further with this additional tool.

Bibliography

Walid Abdul-Kader and Ali Gharbi. Capacity estimation of a multi-product unreliable production line. *International Journal of Production Research*, 40(18):4815–4834, 2002. doi: 10.1080/00207540210000024148.

Yasmine Alaouchiche, Yassine Ouazene, and Farouk Yalaoui. Energy-efficient buffer allocation problem in unreliable production lines. *The International Journal of Advanced Manufacturing Technology*, 114:2871–2885, 2021. doi: 10.1007/s00170-021-06971-1.

Jeanne Albert. Sums of uniformly distributed variables: A combinatorial approach. *The College Mathematics Journal*, 33(3):201–206, 2002. doi: 10.2307/1559029.

Maghsoud Amiri and Ali Mohtashami. Buffer allocation in unreliable production lines based on design of experiments, simulation, and genetic algorithm. *The International Journal of Advanced Manufacturing Technology*, 62:371–383, 1987. doi: 10.1007/s00170-011-3802-8.

Mohamed Amjath, Laoucine Kerbache, James MacGregor Smith, and Adel Elomri. Optimisation of buffer allocations in manufacturing systems: A study on intra and outbound logistics systems using finite queueing networks. *Applied Sciences*, 13(17): 9525, 2023. doi: 10.3390/app13179525.

Richar Bellman. On a routing problem. *Quarterly of Applied Mathematics*, 16(1):87–90, 1958.

J.T.W. Boomers. Improving the medium-term and short-term production planning process of semi-finished products at a cable manufacturing company, 2022.

Eishi Chiba. Heuristics for the buffer allocation problem with collision probability using computer simulation. *Mathematical Problems in Engineering*, 2015:7, 2015. doi: 10.1155/2015/424370.

Leyla Demir, Semra Tunali, and Arne Løkketangen. A tabu search approach for buffer allocation in production lines with unreliable machines. *Engineering Optimization*, 43(2):213–231, 2010. doi: 10.1080/0305215X.2010.481022.

Leyla Demir, Semra Tunali, and Deniz Tursel Eliiyi. The state of the art on buffer allocation problem: a comprehensive survey. *Journal of Intelligent Manufacturing*, 25: 371–392, 2014. doi: 10.1007/s10845-012-0687-9.

Carlos Alberto Berrera Diaz, Tehseen Aslam, and Amos H. C. NG. Optimizing reconfigurable manufacturing systems for fluctuating production volumes: A simulation-based multi-objective approach. *IEEE Access*, 9:144195–144210, 2021. doi: 10.1109/ACCESS.2021.3122239.

A. Dolgui, A.V. Eremeev, and V.S. Sigaev. Hbba: hybrid algorithm for buffer allocation in tandem production lines. *Journal of Intelligent Manufacturing*, 18:411–420, 2007.

Mohamed Essafi, Xavier Delorme, and Alexandre Dolgui. Balancing machining lines: a two-phase heuristic. *Studies in Informatics and Control*, 19(3):243–252, 2010. doi: 10.24846/v19i3y201004.

Sixiao Gao. A bottleneck detection-based tabu search algorithm for the buffer allocation problem in manufacturing systems. *IEEE Access*, 10:60507–60520, 2022. doi: 10.1109/ACCESS.2022.3181134.

Stanley B. Gershwin and Saeideh Fallah-Fini. A general model and analysis of a discrete two-machine production line. *Analysis of Manufacturing Systems*, 2007.

T.J. Hodgson, S.D. Jackson, P. Qu, and R.E. Cannon. A material allocation scheme for optical fibre cable manufacturing: a case study. *International Journal of Production Research*, 42(16):3271–3279, 2004. doi: 10.1080/00207540410001688400.

Ming-Guang Huang, Pao-Long Chang, and Ying-Chyi Chou. Buffer allocation in flow-shop-type production systems with general arrival and service patterns. *Computers Operations Research*, 29(2):103–121, 2002. doi: 10.1016/S0305-0548(00)00060-5.

Khelil Kassoul, Naoufel Cheikhrouhou, and Nicolas Zufferey. Simultaneous allocation of buffer capacities and service times in unreliable production lines. *International Journal of Production Research*, 62(3):644–664, 2024. doi: 10.1080/00207543.2023.2168310.

Laoucine Kerbache and J. MacGregor Smith. Multi-objective routing within large scale facilities using open finite queueing networks. *European Journal of Operational Reserach*, 121(1):105–123, 2000. doi: 10.1016/S0377-2217(99)00018-1.

Simge Yelkenci Kose and Ozcan Kilincci. A multi-objective hybrid evolutionary approach for buffer allocation in open serial production lines. *Journal of Intelligent Manufacturing*, 31:33–51, 2020. doi: 10.1007/s10845-018-1435-6.

Soo-Tae Kwon. On the optimal buffer allocation of an fms with finite in-process buffers. *Computational Science and Its Applications - ICCSA 2006*, 3982:767–776, 2006. doi: 10.1007/11751595_81.

J. Labetoulle and G. Pujolle. Isolation method in a network of queues. *IEEE Transactions on Software Engineering*, 6(4):373–381, 1980. doi: 10.1109/TSE.1980.234493.

Christian M. Lutz, K. Roscoe Davis, and Minghe Sun. Determining buffer location and size in production lines using tabu search. *European Journal of Operational Research*, 106(3):301–316, 1998. doi: 10.1016/S0377-2217(97)00276-2.

Stuart Robert Miller. Dynamic programming based approach to scheduling in the fiber optic cable industry, 2004.

David R. Morrison, Sheldon H. Jacobson, Jason J. Sauppe, and Edward C. Sewell. Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. *Discrete Optimization*, 19:79–102, 2016. doi: 10.1016/j.disopt.2016.01.005.

Maedeh Mosayeb Motlagh, Parham Azimi, Maghsoud Amiri, and Golshan Madraki. An efficient simulation optimization methodology to solve a multi-objective problem in unreliable unbalanced production lines. *Expert Systems with Applications*, 138: 112836, 2019. doi: 10.1016/j.eswa.2019.112836.

N. Nahas, D. Ait-Kadi, and M. Nourelfath. Selecting machines and buffers in unreliable series-parallel production lines. *International Journal of Production Research*, 47 (14):3741–3774, 2009. doi: 10.1080/00207540701806883.

N. Nahas, D. Ait-Kadi, and M. Nourelfath. Selecting machines and buffers in unreliable series-parallel production lines. *International Journal of Advanced Manufacturing Technology*, 62:371–383, 2011. doi: 10.1007/s00170-011-3802-8.

Nabil Nahas. Buffer allocation and preventive maintenance optimization in unreliable production lines. *Journal of Intelligent Manufacturing*, 28(1):85–93, 2017. doi: 10.1007/s10845-014-0963-y.

Mazyar Ghadiri Nejad, Hüseyin Güden, and Béla Vizvári. Time minimization in flexible robotic cells considering intermediate input buffers: a comparative study of three well-known problems. *International Journal of Computer Integrated Manufacturing*, 32(8):809–819, 2019. doi: 10.1080/0951192X.2019.1636411.

Norazmein Abdul Raman and Khairur Rijal Jamaludin. Implementing toyota production system (tps) concept in a small automotive parts manufacturer. *Proceedings of International Conference on Mechanical, Manufacturing Engineering*, 108:276–286, 2008.

Chuan Shi and Stanley B. Gershwin. A segmentation approach for solving buffer allocation problems in large production systems. *International Journal of Production Research*, 54(20):6121–6141, 2016. doi: 10.1080/00207543.2014.991842.

Shuo Shi and Sixiao Gao. Multi-objective optimization of simultaneous buffer and service rate allocation in manufacturing systems based on a data-driven hybrid approach. *International Journal of Industrial Engineering Computations*, 14(4):707–722, 2023. doi: 10.5267/j.ijiec.2023.8.001.

J. MacGregor Smith and F.R.B. Cruz. The buffer allocation problem for general finite buffer queueing networks. *IIE Transactions*, 37(4):343–365, 2005. doi: 10.1080/07408170590916986.

D.D. Spinellis and C.T. Papadopoulos. A simulated annealing approach for buffer allocation in reliable production lines. *Annals of Operations Research*, 93:373–384, 2000. doi: 10.1023/A:1018984125703.

C. Terracol and R. David. An aggregation method for performance evaluation of transfer lines with unreliable machines and finite buffers. *IEEE International Conference on Robotics and Automation*, 4:1333–1338, 1987. doi: 10.1109/ROBOT.1987.1087826.

Miko Wong. Armoured cable vs. unarmoured cable: What's the difference?, December 2017. URL `https://medium.com/@mikowong405/armoured-cable-vs-unarmoured-cable-whats-the-difference-31d4b746999d`.

Simcona. Anatomy of an electrical cable: Conductor, protection, connector, May 2023. URL `https://simcona.com/blog/anatomy-of-a-cable-electrical-industrial`.

The Open University. Conform process, November 2017. URL `https://www.open.edu/openlearn/science-maths-technology/engineering-technology/manupedia/conform-process`.

Hector A. Vergara and David S. Kim. A new method for the placement of buffers in serial production lines. *International Journal of Production Research*, 47(16):4437–4456, 2009. doi: 10.1080/00207540801939022.

Sophie Weiss, Justus Arne Schwarz, and Raik Stolletz. The buffer allocation problem in production lines: Formulations, solution methods, and instances. *IISE Transactions*, 51(5):456–485, 2019. doi: 10.1080/24725854.2018.1442031.

T. Van Woensel, R. Andriansyah, F.R.B. Cruz, J. MacGregor Smith, and L. Kerbache. Buffer and server allocation in general multi-server queueing networks. *International Transactions in Operational Research*, 17(2):257–286, 2010. doi: 10.1111/j.1475-3995.2009.00746.x.

Shaohui Xi, Qingxin Chen, James MacGregor Smith, Ning Mao, Ailin Yu, and Huiyu Zhang. A new method for solving buffer allocation problem in large unbalanced production lines. *International Journal of Production Research*, 58(22):6846–6867, 2020. doi: 10.1080/00207543.2019.1685709.

Hideaki Yamashita and Tayfur Altiok. Buffer capacity allocation for a desired throughput in production lines. *IIE Transactions*, 30:883–892, 1998. doi: 10.1023/A:1007564620313.

Simge Yelkenci and Ozcan Kilincci. Hybrid approach for buffer allocation in open serial production lines. *Computers Operations Research*, 60:67–78, 2015. doi: 10.1016/j.cor.2015.01.009.

Mengyi Zhang, Erica Pastore, Arianna Alfieri, and Andrea Matta. Buffer allocation problem in production flow lines: A new benders-decomposition-based exact solution. *ISSE Transactions*, 54(5):421–434, 2022. doi: 10.1080/24725854.2021.1905195.

Appendix A

Example Branch and Bound

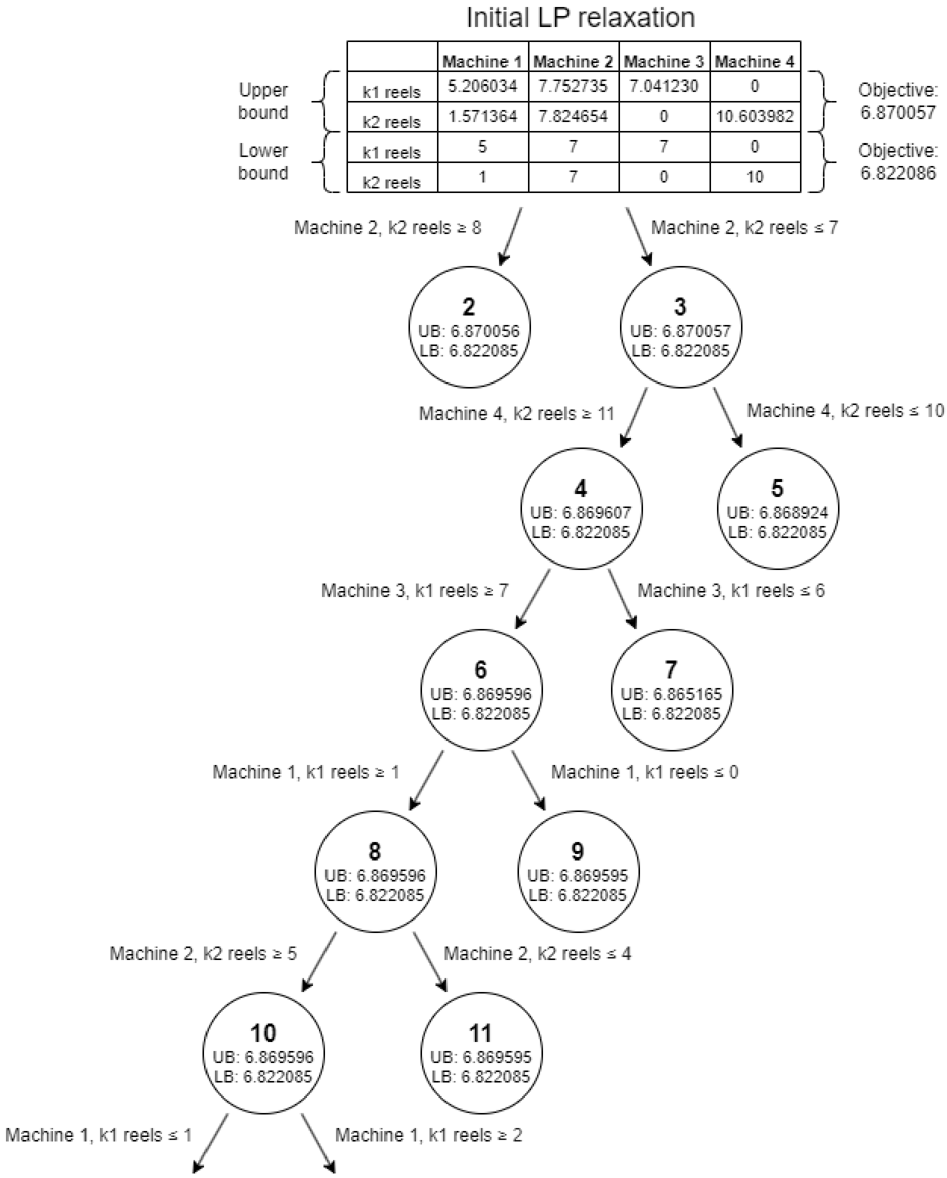

Figure A.1: Branch and bound for BAP example.

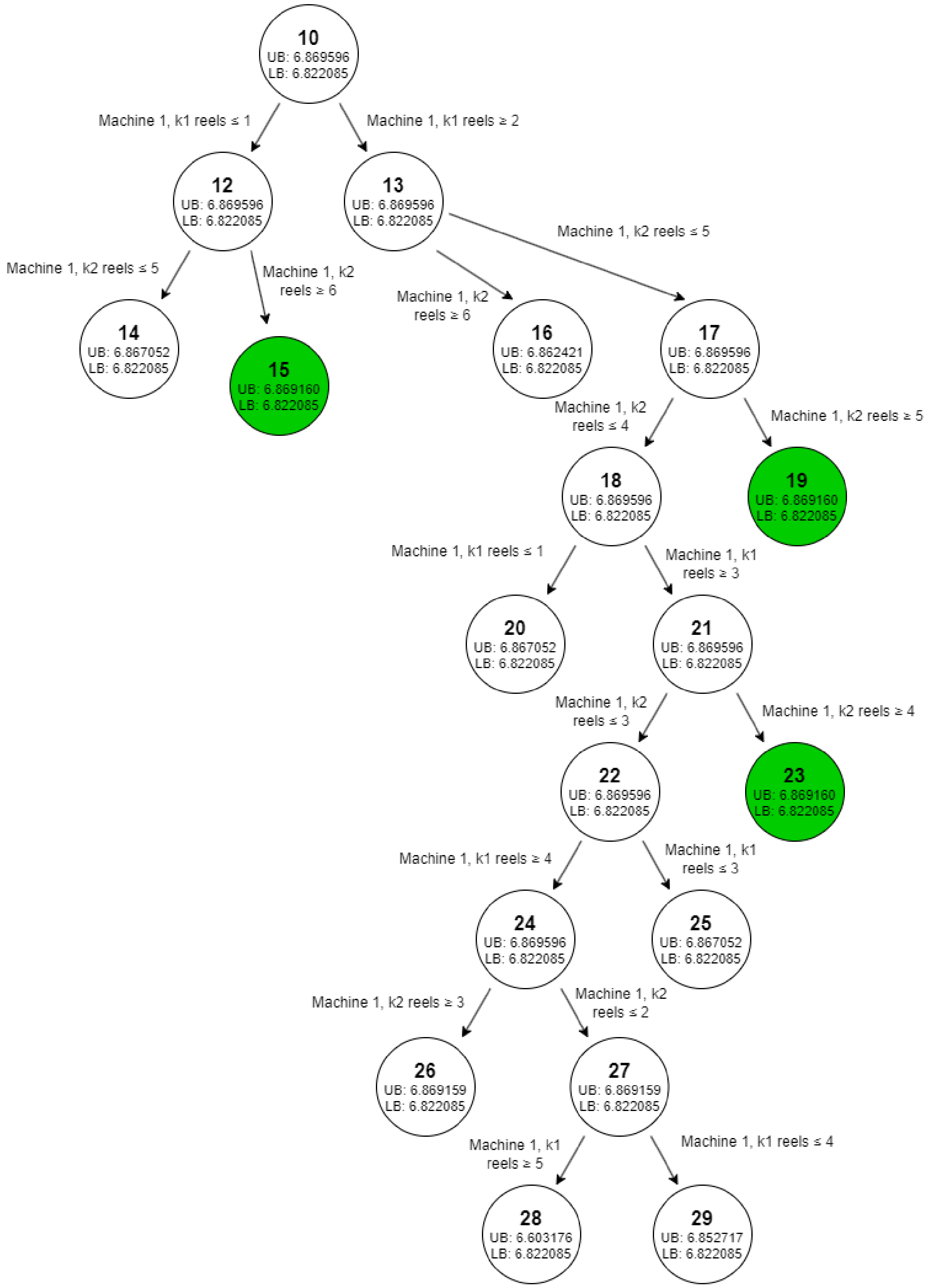

Figure A.2: Branch and bound for BAP example, continued.

Appendix B

VBA Codes

B.1 Objective Function

Listing B.1: Objective function

```
'Function for calculating throughput
Function throughput(activeTime As Double, rho As Double, serviceRate As Double, bufferAllocation As Integer) As Double
    throughput = activeTime * serviceRate * (1 - (1 - rho) / (1 - rho ^ (bufferAllocation + 1)))
End Function

'Objective function
Function ObjectiveFunction(bufferAllocation As Variant) As Double
    'Variables
    Dim parameters As Worksheet
    Dim flowDispersion As Worksheet
    Dim serviceRates() As Variant                'Array with all servicerates
    Dim OEEs() As Variant                        'Array with all OEEs
    Dim activeTime() As Variant                  'Array with active time percentages
    Dim reelConversions() As Variant             'Array with reel conversion consants
    Dim rho As Double                            'Variable used for utilization
    Dim degassingHKS As Integer                  'Degassing capacity in City A
    Dim degassingLCH As Integer                  'Degassing capacity in City B
    Dim maxRho As Double                         'Max allowed utilization

    'Set Parameters
    Set parameters = Worksheets("Parameters")
    Set flowDispersion = Worksheets("Flow␣dispersion")
    serviceRates = Application.Transpose(parameters.Range("Q34:Q55").Value) 'Machines are in alfabetic order
    OEEs = Application.Transpose(parameters.Range("L34:L55").Value)
    activeTime = Application.Transpose(parameters.Range("M34:M55").Value)
    reelConversions = Application.Transpose(parameters.Range("L7:L28").Value)
    degassingHKS = parameters.Range("P20").Value
    degassingLCH = parameters.Range("P21").Value
    maxRho = 0.9999

    'CL 1
    Dim thCL1 As Double
    rho = parameters.Range("Q7").Value / (serviceRates(3) * OEEs(3))
    thCL1 = OEEs(3) * throughput(CDbl(activeTime(3)), minimum(rho, maxRho), CDbl(serviceRates(3)), Int(bufferAllocation(3)))

    'CL 2
    Dim thCL2 As Double
    rho = parameters.Range("Q8").Value / (serviceRates(4) * OEEs(4))
    thCL2 = OEEs(4) * throughput(CDbl(activeTime(4)), minimum(rho, maxRho), CDbl(serviceRates(4)), Int(bufferAllocation(4)))

    'ISO 1
    Dim thISO1 As Double
    rho = parameters.Range("Q9").Value / (serviceRates(8) * OEEs(8))
    thISO1 = OEEs(8) * throughput(CDbl(activeTime(8)), minimum(rho, maxRho), CDbl(serviceRates(8)), Int(bufferAllocation(8)))

    'ISO 2
    Dim thISO2 As Double
    rho = parameters.Range("Q10").Value / (serviceRates(9) * OEEs(9))
    thISO2 = OEEs(9) * throughput(CDbl(activeTime(9)), minimum(rho, maxRho), CDbl(serviceRates(9)), Int(bufferAllocation(9)))

    'CDCC 1
    Dim thCDCC1 As Double
    rho = reelConversions(1) * (thCL1 * flowDispersion.Range("F6").Value + thCL2 * flowDispersion.Range("F9").Value) / _
        (serviceRates(1) * OEEs(1))
    thCDCC1 = OEEs(1) * throughput(CDbl(activeTime(1)), minimum(rho, maxRho), CDbl(serviceRates(1)), Int(bufferAllocation(1)))

    'CDCC 3
    Dim thCDCC3 As Double
    rho = reelConversions(2) * (thCL1 * flowDispersion.Range("F7").Value + thCL2 * flowDispersion.Range("F10").Value) / _
        (serviceRates(2) * OEEs(2))
    thCDCC3 = OEEs(2) * throughput(CDbl(activeTime(2)), minimum(rho, maxRho), CDbl(serviceRates(2)), Int(bufferAllocation(2)))

    'Degassing HKS
    Dim thDegassingHKS As Double
```

```
rho = reelConversions(5) * (thCDCC1 * flowDispersion.Range("F4").Value + thCDCC3 * flowDispersion.Range("F5").Value) / _
    (serviceRates(5) * degassingHKS * OEEs(5))
thDegassingHKS = OEEs(5) * throughput(CDbl(activeTime(5)), minimum(rho, maxRho), CDbl(serviceRates(5)) * degassingHKS, _
    Int(bufferAllocation(5)))

'Degassing LCH
Dim thDegassingLCH As Double
rho = reelConversions(19) * (thCDCC2 * flowDispersion.Range("P4").Value) / (serviceRates(19) * degassingLCH * OEEs(19))
thDegassingLCH = OEEs(19) * throughput(CDbl(activeTime(19)), minimum(rho, maxRho), CDbl(serviceRates(19)) * degassingLCH, _
    Int(bufferAllocation(19)))

'SL 1
Dim thSL1 As Double
rho = reelConversions(14) * (thDegassingHKS * flowDispersion.Range("F12").Value) / (serviceRates(14) * OEEs(14))
thSL1 = OEEs(14) * throughput(CDbl(activeTime(14)), minimum(rho, maxRho), CDbl(serviceRates(14)), Int(bufferAllocation(14)))

'SL 2
Dim thSL2 As Double
rho = reelConversions(15) * (thDegassingHKS * flowDispersion.Range("F13").Value) / (serviceRates(15) * OEEs(15))
thSL2 = OEEs(15) * throughput(CDbl(activeTime(15)), minimum(rho, maxRho), CDbl(serviceRates(15)), Int(bufferAllocation(15)))

'SL 3
Dim thSL3 As Double
rho = reelConversions(16) * (thISO1 * flowDispersion.Range("F21").Value + thDegassingHKS * _
    flowDispersion.Range("F14").Value) / (serviceRates(16) * OEEs(16))
thSL3 = OEEs(16) * throughput(CDbl(activeTime(16)), minimum(rho, maxRho), CDbl(serviceRates(16)), Int(bufferAllocation(16)))

'SL 5
Dim thSL5 As Double
rho = reelConversions(22) * (thDegassingHKS * flowDispersion.Range("F15").Value + thDegassingLCH * _
    flowDispersion.Range("K22").Value) / (serviceRates(22) * OEEs(22))
thSL5 = OEEs(22) * throughput(CDbl(activeTime(22)), minimum(rho, maxRho), CDbl(serviceRates(22)), Int(bufferAllocation(22)))

'DTW 1
Dim thDTW1 As Double
rho = reelConversions(6) * (thISO1 * flowDispersion.Range("F20").Value + thISO2 * flowDispersion.Range("F22").Value) / _
    (serviceRates(6) * OEEs(6))
thDTW1 = OEEs(6) * throughput(CDbl(activeTime(6)), minimum(rho, maxRho), CDbl(serviceRates(6)), Int(bufferAllocation(6)))

'DTW 2
Dim thDTW2 As Double
rho = reelConversions(7) * (thSL3 * flowDispersion.Range("K18").Value) / (serviceRates(7) * OEEs(7))
thDTW2 = OEEs(7) * throughput(CDbl(activeTime(7)), minimum(rho, maxRho), CDbl(serviceRates(7)), Int(bufferAllocation(7)))

'DTW 4
Dim thDTW4 As Double
rho = reelConversions(20) * (thSL5 * flowDispersion.Range("P6").Value) / (serviceRates(20) * OEEs(20))
thDTW4 = OEEs(20) * throughput(CDbl(activeTime(20)), minimum(rho, maxRho), CDbl(serviceRates(20)), Int(bufferAllocation(20)))

'M 1
Dim thM1 As Double
'Different servicerate due to loop
rho = reelConversions(10) * (thSL1 * flowDispersion.Range("K12").Value + thSL2 * flowDispersion.Range("K15").Value + _
    thDTW1 * flowDispersion.Range("F16").Value + thDTW2 * flowDispersion.Range("F18").Value) / _
    ((1 / (1 / serviceRates(10) + flowDispersion.Range("K5").Value * 1 / serviceRates(10))) * OEEs(10))
thM1 = OEEs(10) * throughput(CDbl(activeTime(10)), minimum(rho, maxRho), 1 / (1 / CDbl(serviceRates(10)) + _
    flowDispersion.Range("K5").Value * 1 / CDbl(serviceRates(10))), Int(bufferAllocation(10)))

'M 2
Dim thM2 As Double
rho = reelConversions(11) * (thSL1 * flowDispersion.Range("K13").Value + thSL2 * flowDispersion.Range("K16").Value + _
    thDTW2 * flowDispersion.Range("F19").Value) / (serviceRates(11) * OEEs(11))
thM2 = OEEs(11) * throughput(CDbl(activeTime(11)), minimum(rho, maxRho), CDbl(serviceRates(11)), Int(bufferAllocation(11)))

'M 3
Dim thM3 As Double
'Different servicerate due to loop
rho = reelConversions(12) * (thDTW1 * flowDispersion.Range("F17").Value) / ((1 / (1 / serviceRates(12) + 1 / _
serviceRates(12) * (flowDispersion.Range("K8").Value + flowDispersion.Range("K9").Value))) * OEEs(12))
thM3 = OEEs(12) * throughput(CDbl(activeTime(12)), minimum(rho, maxRho), 1 / (1 / CDbl(serviceRates(12)) + 1 / _
    CDbl(serviceRates(12)) * (flowDispersion.Range("K8").Value + flowDispersion.Range("K9").Value)), Int(bufferAllocation(12)))

'M 9
Dim thM9 As Double
rho = reelConversions(13) * (thSL1 * flowDispersion.Range("K14").Value + thSL2 * flowDispersion.Range("K17").Value + _
        thSL5 * flowDispersion.Range("P5").Value) / (serviceRates(13) * OEEs(13))
thM9 = OEEs(13) * throughput(CDbl(activeTime(13)), minimum(rho, maxRho), CDbl(serviceRates(13)), Int(bufferAllocation(13)))

'JA 2
Dim thJA2 As Double
rho = reelConversions(21) * (thSL5 * flowDispersion.Range("P7").Value + thDTW4 * flowDispersion.Range("P8").Value) / _
    (serviceRates(21) * OEEs(21))
thJA2 = OEEs(21) * throughput(CDbl(activeTime(21)), minimum(rho, maxRho), CDbl(serviceRates(21)), Int(bufferAllocation(21)))

'SL 4
Dim thSL4 As Double
rho = reelConversions(17) * (thM1 * flowDispersion.Range("K5").Value + thM3 * flowDispersion.Range("K9").Value) / _
    (serviceRates(17) * OEEs(17))
thSL4 = OEEs(17) * throughput(CDbl(activeTime(17)), minimum(rho, maxRho), CDbl(serviceRates(17)), Int(bufferAllocation(17)))

'The objective function is defined below
ObjectiveFunction = thSL3 * flowDispersion.Range("K19").Value + thM1 * flowDispersion.Range("K6").Value + _
    thM2 * flowDispersion.Range("K7") + thM3 * flowDispersion.Range("K10").Value + thM9 * flowDispersion.Range("K11") + _
    thJA2 * flowDispersion.Range("P9")
End Function
```

B.2 Valid Solution

Listing B.2: Validate solution

```
'Function for checking if solution is valid
Function validSolution(BA() As Variant) As Boolean
    'Variables
    Dim parameters As Worksheet
    Dim checkValidity As Boolean
    Dim i As Integer
    Dim limHKS As Integer                    'Reel capacity limit in City A
    Dim reelsHKS As Integer                  'Variable for number of reels in City A
    Dim limLCH As Integer                    'Reel capacity limit in City B
    Dim reelsLCH As Integer                  'Variable for number of reels in City B
    Dim lim2800 As Integer                   'Maximum number of 2800 type reels to be allocated
    Dim reels2800 As Integer                 'Variable for number of allocated 2800 type reels
    Dim lim2800wide As Integer               'Maximum number of 2800 wide type reels to be allocated
    Dim reels2800wide As Integer             'Variable for number of allocated 2800 wide type reels
    Dim lim2500 As Integer                   'Maximum number of 2500 type reels to be allocated
    Dim reels2500 As Integer                 'Variable for number of allocated 2500 type reels
    Dim lim2240 As Integer                   'Maximum number of 2240 type reels to be allocated
    Dim reels2240 As Integer                 'Variable for number of allocated 2240 type reels
    Dim lim1800 As Integer                   'Maximum number of 1800 type reels to be allocated
    Dim reels1800 As Integer                 'Variable for number of allocated 1800 type reels

    'Initialize
    Set parameters = Worksheets("Parameters")
    checkValidity = True
    limHKS = parameters.Range("P24").Value
    limLCH = parameters.Range("P25").Value
    lim2800 = parameters.Range("P26").Value
    lim2800wide = parameters.Range("P27").Value
    lim2500 = parameters.Range("P28").Value
    lim2240 = parameters.Range("P29").Value
    lim1800 = parameters.Range("P30").Value

    'Determine cable reels per location
    reelsHKS = 0
    For i = 1 To 22
        reelsHKS = reelsHKS + BA(i)          'All reels in City A
    Next i
    reelsHKS = reelsHKS - BA(3) - BA(4)      'CL1 and CL2 do not consume reels

    reelsLCH = 0
    For i = 23 To 27
        reelsLCH = reelsLCH + BA(i)          'All reels in City B
    Next i

    'Determine the number of cable reels per type
    reels2800 = BA(5) + BA(10) + BA(13) + BA(15) + BA(17) + BA(18) + BA(19) + BA(20)
    reels2800wide = BA(11) + BA(14) + BA(24) + BA(26) + BA(27)
    reels2500 = BA(1) + BA(2) + BA(23)
    reels2240 = BA(7) + BA(12) + BA(16) + BA(22) + BA(25)
    reels1800 = BA(6) + BA(8) + BA(9) + BA(21)

    'Validity check
    If reelsHKS > limHKS Then
        checkValidity = False
    ElseIf reelsLCH > limLCH Then
        checkValidity = False
    ElseIf reels2800 > lim2800 Then
        checkValidity = False
    ElseIf reels2800wide > lim2800wide Then
        checkValidity = False
    ElseIf reels2500 > lim2500 Then
        checkValidity = False
    ElseIf reels2240 > lim2240 Then
        checkValidity = False
    ElseIf reels1800 > lim1800 Then
        checkValidity = False
    End If

    For i = 1 To 27          'Every buffer needs to be above 0
        If BA(i) < 1 Then
            checkValidity = False
        End If
    Next i

    'Store result
    validSolution = checkValidity
End Function
```

B.3 Simulated Annealing Algorithm

Listing B.3: Simulated annealing

```
'Sub for the simulated annealing algorithm
Public Sub SimulatedAnnealing()
    'Variables
    Dim T As Double                                 'Current temperature
    Dim Tmin As Double                              'Minimal temperature before termination of algorithm
    Dim alpha As Double                             'Cooling rate, constant between 0 and 1
    Dim currentBA() As Variant                      'Current buffer allocation from which a new neighbour is derived
    Dim nextBA() As Variant                         'Next buffer allocation to be assessed
    Dim bestBA() As Variant                         'Best buffer allocation found so far
    Dim currentEnergy As Double                     'Performance of current buffer allocation
    Dim nextEnergy As Double                        'Performance of next buffer allocation
    Dim bestEnergy As Double                        'Performance of the best buffer allocation
    Dim acceptanceProbability As Double             'Difference in performance between next and current buffer allocation
    Dim randomBufferUP As Integer                   'Random buffer to be increased by 1
    Dim randomBufferDOWN As Integer                 'Random buffer to be reduced by 1
    Dim validCandidate As Boolean                   'Boolean if generated candidate solution is valid
    Dim solutionsTried As Integer                   'Number of tested solutions
    Dim parameters As Worksheet

    Set parameters = Worksheets("Parameters")

    'Initial temperature and cooling rate
    T = 1000
    alpha = 0.99
    Tmin = 0.1

    'Initial values for best solution
    currentBA = Application.Transpose(parameters.Range("T34:T60").Value)
    currentEnergy = ObjectiveFunction(currentBA)
    bestBA = currentBA
    bestEnergy = currentEnergy
    solutionsTried = 0

    'Main simulated annealing loop
    While T > Tmin
        'Generate a candidate solution
        validCandidate = False
        While validCandidate = False
            randomBufferUP = Rnd() * 26 + 1          'There are 27 machines
            randomBufferDOWN = Rnd() * 26 + 1

            nextBA = currentBA
            nextBA(randomBufferUP) = nextBA(randomBufferUP) + 1
            nextBA(randomBufferDOWN) = nextBA(randomBufferDOWN) - 1

            If randomBufferUP = randomBufferDOWN Or randomBufferUP = 3 Or randomBufferDOWN = 3 Or randomBufferUP = 4 Or _
                randomBufferDOWN = 4 Then  'Buffers 3 and 4 are conform extrusion lines and don't have a buffer
                validCandidate = False
            ElseIf validSolution(nextBA) = False Then
                validCandidate = False
            ElseIf nextBA(randomBufferUP) < 1 Or nextBA(randomBufferDOWN) < 1 Then
                validCandidate = False
            Else
                validCandidate = True
            End If
        Wend

        nextEnergy = ObjectiveFunction(nextBA)

        'Calculate the acceptance probability
        acceptanceProbability = Exp((currentEnergy - nextEnergy) / T)

        'Accept the next solution with acceptanceprobability
        If nextEnergy > currentEnergy Or Rnd() < acceptanceProbability Then
            currentBA = nextBA
            currentEnergy = nextEnergy
        End If

        'Update the best solution if needed
        If currentEnergy > bestEnergy Then
            bestBA = currentBA
            bestEnergy = currentEnergy
        End If

        'Cool down the temperature
        T = T * alpha

        'Update tried solutions
        solutionsTried = solutionsTried + 1
    Wend

    printBuffers bestBA
End Sub
```

B.4 Tabu Search Algorithm

Listing B.4: Tabu search

```
'Function for comparing arrays in the tabu list
Function compareArrays(arr1() As Variant, arr2() As Variant) As Boolean
    Dim i As Integer

    'Check if they are of the same length
    If UBound(arr1) <> UBound(arr2) Then
        compareArrays = False
        Exit Function
    End If

    'Compare each element
    For i = LBound(arr1) To UBound(arr1)
        If arr1(i) <> arr2(i) Then
            compareArrays = False
            Exit Function
        End If
    Next i

    'If every element is the same
    compareArrays = True
End Function

'Sub for the tabu search algorithm
Public Sub TabuSearch()
    'Declare variables
    Dim currentBA() As Variant                        'Current buffer allocation from which a new neighbour is derived
    Dim currentBestBA() As Variant                    'Current best buffer allocation neighbour
    Dim nextBA() As Variant                           'Next buffer allocation to be assessed
    Dim bestBA() As Variant                           'Best buffer allocation found so far
    Dim currentEnergy As Double                       'Throughput of current buffer allocation
    Dim currentBestEnergy As Double                   'Throughput of current best found neighbour
    Dim nextEnergy As Double                          'Throughput of buffer allocation to be assessed
    Dim bestEnergy As Double                          'Throughput of best buffer allocation found so far
    Dim tabuList As Object                            'Tabu list in which previous visited neighbours are stored
    Dim tabuSize As Integer                           'Maximum number of tabu buffer allocations that may be stored in the tabu list
    Dim iteration As Integer                          'Current iteration for the algorithm
    Dim maxIterations As Integer                      'Maximum number of iterations that can be performed after termination
    Dim i As Integer                                  'Index of buffer to be increased by 1
    Dim j As Integer                                  'Index of buffer to be reduced by 1
    Dim k As Integer                                  'Index to loop through tabu list
    Dim notInList As Boolean                          'Boolean for checking if neighbour is not in tabu list
    Dim parameters As Worksheet

    'Initialization
    Set tabuList = CreateObject("System.Collections.ArrayList")
    Set parameters = Worksheets("Parameters")
    tabuSize = 10
    iteration = 1
    maxIterations = 100
    currentBA = Application.Transpose(parameters.Range("T34:T60").Value)
    currentEnergy = ObjectiveFunction(dummyBA)
    bestBA = currentBA
    bestEnergy = currentEnergy
    tabuList.Insert 0, currentBA

    'Tabu algorithm
    While iteration <= maxIterations
        currentBestEnergy = 0

        'Looping through all neighbours
        For i = 1 To UBound(currentBA)
            For j = 1 To UBound(currentBA)
                If i <> j And i <> 3 And j <> 3 And i <> 4 And j <> 4 Then      'CL 1 and CL 2 don't have buffers
                    nextBA = currentBA
                    nextBA(i) = nextBA(i) + 1
                    nextBA(j) = nextBA(j) - 1
                    nextEnergy = ObjectiveFunction(nextBA)
                    notInList = True

                    For k = 0 To tabuList.Count - 1
                        If compareArrays(nextBA, tabuList.Item(k)) = True Then
                            notInList = False
                        End If
                    Next k

                    If validSolution(nextBA) = True And nextEnergy > currentBestEnergy And notInList = True Then
                        currentBestBA = nextBA
                        currentBestEnergy = nextEnergy
                    End If

                End If
            Next j
        Next i

        currentBA = currentBestBA
        currentEnergy = currentBestEnergy

        tabuList.Insert 0, currentBA
        If tabuList.Count > tabuSize Then
            tabuList.RemoveAt tabuSize
        End If
```

```
        If currentEnergy > bestEnergy Then
            bestBA = currentBA
            bestEnergy = currentEnergy
        End If

        iteration = iteration + 1
    Wend

    printBuffers bestBA
Ens Sub
```

B.5 Greatest Improvement Algorithm

Listing B.5: Greatest improvement

```
'Sub for the greatest improvement algorithm
Public Sub GreatestImprovement()
    'Variables
    Dim currentBA() As Variant                     'Current buffer allocation from which a new neighbour is derived
    Dim nextBA() As Variant                        'Next buffer allocation to be assessed
    Dim bestBA() As Variant                        'Best buffer allocation found so far
    Dim currentEnergy As Double                    'Throughput of current buffer allocation
    Dim nextEnergy As Double                       'Throughput of buffer allocation to be assessed
    Dim bestEnergy As Double                       'Throughput of best buffer allocation found so far
    Dim valid As Boolean                           'Boolean to check if there are still reels to allocate
    Dim noFurtherIterations As Boolean             'Boolean to stop algorithm
    Dim i As Integer

    'Initialization
    currentBA = Array(1, 1, 999, 999, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
    currentBA = Application.Transpose(Application.Transpose(currentBA))
    currentEnergy = ObjectiveFunction(currentBA)
    bestBA = currentBA
    bestEnergy = currentEnergy
    valid = False
    noFurtherIterations = False

    'Greatest improvement algorithm
    While noFurtherIterations = False
        valid = False
        For i = 1 To UBound(nextBA)
            If i <> 3 And i <> 4 Then       'The conform extrusion lines do not have a buffer
                nextBA = bestBA
                nextBA(i) = nextBA(i) + 1
                If validSolution(nextBA) = True Then
                    nextEnergy = ObjectiveFunction(nextBA)
                    valid = True

                    If nextEnergy >= currentEnergy Then
                        currentBA = nextBA
                        currentEnergy = nextEnergy
                    End If

                ElseIf i = UBound(nextBA) And valid = False Then
                    noFurtherIterations = True
                End If
            End If
        Next i
        If noFurtherIterations = False Then
            bestBA = currentBA
            bestEnergy = currentEnergy
        End If
    Wend

    printBuffers bestBA
End Sub
```

Appendix C

Machine Productivity

Machine	Single core		Triple core	
	Reels	*Average time [min.]*	*Reels*	*Average time [min.]*
EI CDCC 1	297	190	372	237
EI CDCC 3	336	191	595	271
EI Conform line 1	663	323	510	253
EI Conform line 2	113	344	255	227
EI Degassing	-	4320	-	4320
EI Drumtwister 1				
EI Drumtwister 2			454	111
EI Insulation line 1				
EI Insulation line 2				
EI Sheathing line 1	637	73	305	68
EI Sheathing line 2	193	184	40	107
EI Sheathing line 3				
EI Sheathing line 9	1192	79		
EI Shielding line 1	1004	133		
EI Shielding line 2	39	215		
EI Shielding line 3			208	88
EI Shielding line 4				
LCH CDCC 2	700	980		
LCH Degassing	-	4320	-	4320
LCH Drumtwister 4			135	140
LCH Jacketing line 2	95	49	883	342
LCH Shielding line 5	315	195	76	104

Table C.1: Number of produced reels over a period of six months (values are changed regarding confidentiality).

Machine	Low voltage		Low voltage other		Alkudia	
	Reels	*Average time [min.]*	*Reels*	*Average time [min.]*	*Reels*	*Average time [min.]*
EI CDCC 1						
EI CDCC 3						
EI Conform line 1						
EI Conform line 2						
EI Degassing						
EI Drumtwister 1	283	118			194	114
EI Drumtwister 2						
EI Insulation line 1	566	91	88	91		
EI Insulation line 2					162	127
EI Sheathing line 1	278	98				
EI Sheathing line 2	64	107				
EI Sheathing line 3	440	105			325	79
EI Sheathing line 9						
EI Shielding line 1						
EI Shielding line 2						
EI Shielding line 3			193	503		
EI Shielding line 4	257	174				
LCH CDCC 2						
LCH Degassing						
LCH Drumtwister 4						
LCH Jacketing line 2						
LCH Shielding line 5						

Table C.2: Number of produced reels over a period of six months (values are changed regarding confidentiality).

Appendix D

Minimum Replications

The minimum number of replications is determined by first conducting several replications and storing the required indicator, which is the throughput in this case. Thereafter, the moving average and variance are calculated. The corresponding T-value to the degrees of freedom is also calculated. The Chi-square value is calculated by using the formula $T \cdot \sqrt{\frac{Var}{DOF}}$. The maximum error was set at 0.05, which was immediately achieved. However, as rule of thumb a minimum of 10 replications is used.

Rep.	Throuhgput	Mean	Variance	T-value	Chi-square	Error
1	0.81823	0.81823				
2	0.81796	0.81809	$3.68 \cdot 10^{-8}$	12.7062	0.00172	0.00211
3	0.81805	0.81808	$1.89 \cdot 10^{-8}$	4.3027	0.00034	0.00042
4	0.81828	0.81813	$2.29 \cdot 10^{-8}$	3.1824	0.00024	0.00029
5	0.81815	0.81813	$1.73 \cdot 10^{-8}$	2.7764	0.00016	0.00020
6	0.81802	0.81812	$1.59 \cdot 10^{-8}$	2.5706	0.00013	0.00016
7	0.81797	0.81810	$1.64 \cdot 10^{-8}$	2.4469	0.00012	0.00015
8	0.81817	0.81811	$1.48 \cdot 10^{-8}$	2.3646	0.00010	0.00012
9	0.81832	0.81813	$1.80 \cdot 10^{-8}$	2.3060	0.00010	0.00013
10	0.81841	0.81816	$2.38 \cdot 10^{-8}$	2.2622	0.00011	0.00014

Table D.1: Data to determine minimum number of replications.

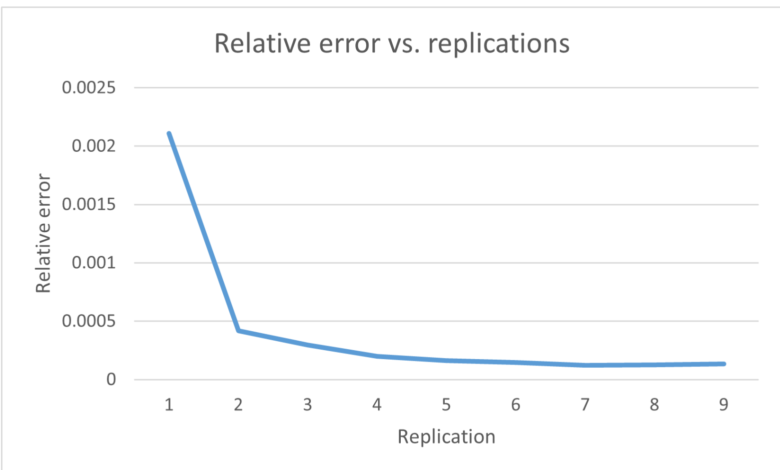

Figure D.1: Relative error after each additional replication.

Appendix E

Guideline Visualization

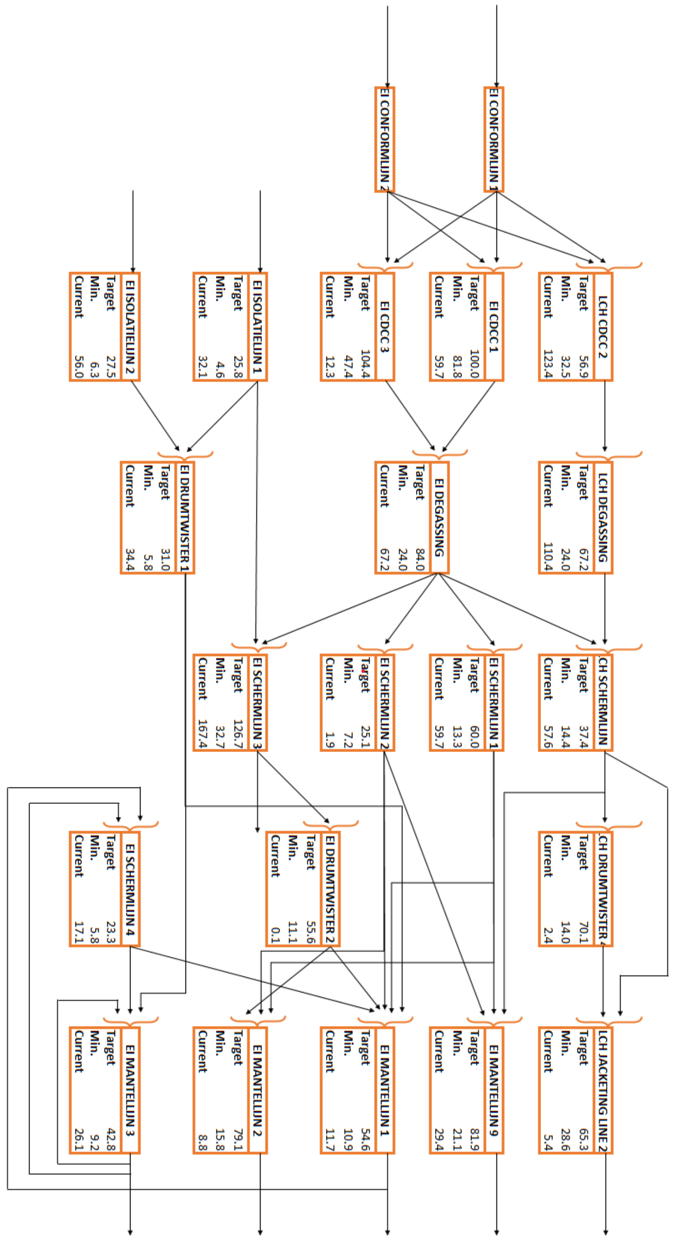

Figure E.1: Dashboard with all buffer allocations.