



LTH

**FACULTY OF
ENGINEERING**

Master of Science Thesis

MIOM05 Degree Project in Production Management
Department of Mechanical Engineering Sciences
Division of Production Management

Data-driven optimisation of production planning: A study on mathematical production planning at Oatly

Authors

Mattias Lewerin & Emil Olefjord Widding

Supervisors

Johan Marklund, LTH

Kristoffer Helgesson, Marcus Johansson, Oatly

Abstract

Title: Data-driven optimisation of production planning: A study on mathematical production planning at Oatly

Authors: Mattias Lewerin & Emil Olefjord Widding

Supervisors: Johan Marklund (LTH), Kristoffer Helgesson (Oatly), Marcus Johansson (Oatly)

Examiner: Danja Sonntag (LTH)

Collaborators: Oatly Sweden Operations & Supply AB

Background: The global plant-based food shift puts high pressure on manufacturing capacity at facilities like Oatly's Landskrona site. This environment faces high combinatorial complexity due to growing product variety and sequence-dependent cleaning requirements. Relying on manual, experience-based planning to navigate these rules limits throughput and introduces organisational risk.

Purpose: To develop a mathematical optimisation model that minimises production makespan for Oatly's high-volume filling lines, providing a decision-support tool to assist planners rather than fully replacing human expertise.

Research Questions: **(1)** How can Oatly's specific production constraints, including sequence-dependent cleaning rules and routing dependencies, be translated into a model to create a technically feasible production schedule? **(2)** To what extent can the total production time for a given order list be minimised by utilising an optimisation model compared to the current experience-based manual planning? **(3)** Is it possible to reduce the number of CIP cycles with a MILP model? **(4)** How may the potential transition to an automated optimisation model affect the stability of the planning process, specifically regarding the reliance on individual human expertise?

Methodology: The study utilises an Operations Research framework to develop a Mixed-Integer Linear Programming (MILP) model in Python. Primary data was gathered through semi-structured interviews, while quantitative validation was performed by retrospectively benchmarking the model's schedules against 39 weeks of historical manual production plans from 2025.

Findings: The MILP model handled sequence-dependent setup constraints and generated faster schedules in 26 of the 39 evaluated weeks under a 2000-second baseline, resulting in a 2.9% reduction in total makespan. The 13 weeks where manual planning resulted in shorter makespans were characterised by high combinatorial complexity. However, extended computational testing (25,000 seconds) on some of the most complex weeks demonstrated that the model has potential to outperform the manual planner when given adequate processing time. The study concludes that a hybrid approach delivers the optimal practical outcome. Using the mathematical model to generate a baseline schedule allows the human planner to focus on managing stochastic disruptions, upstream dependencies, and practical exceptions. This combination of mathematical optimisation and human expertise ensures high capacity output while maintaining system stability.

Keywords: Mixed-Integer Linear Programming, Production Planning, Semicontinuous Manufacturing, Sequence-Dependent Setup Times, Oatly, Combinatorial Optimisation, MILP

Sammanfattning

Titel: Datadriven optimering av produktionsplanering: En studie om matematisk produktionsplanering på Oatly

Författare: Mattias Lewerin & Emil Olefjord Widding

Handledare: Johan Marklund (LTH), Kristoffer Helgesson (Oatly), Marcus Johansson (Oatly)

Examinator: Danja Sonntag (LTH)

I samarbete med: Oatly Sweden Operations & Supply AB

Bakgrund: Den globala övergången till växtbaserade livsmedel sätter enorm press på produktionskapaciteten vid anläggningar som Oatlys fabrik i Landskrona. Denna miljö står inför en hög kombinatorisk komplexitet på grund av ett växande produktutbud och sekvensberoende rengöringskrav. Att förlita sig på manuell, erfarenhetsbaserad planering för att navigera dessa regler begränsar genomströmningen och medför organisatoriska risker.

Syfte: Att utveckla en matematisk optimeringsmodell som minimerar produktionstiden för Oatlys högvolumsfyllningslinjer, för att tillhandahålla ett beslutsstödsverktyg som assisterar planerare snarare än att helt ersätta mänsklig expertis.

Forskningsfrågor: (1) Hur kan Oatlys specifika produktionsrestriktioner, inklusive sekvensberoende rengöringsregler och ruttberoenden, översättas till en modell för att skapa ett tekniskt genomförbart produktionsschema? (2) I vilken utsträckning kan den totala produktionstiden för en given orderlista minimeras genom att använda en optimeringsmodell jämfört med den nuvarande erfarenhetsbaserade manuella planeringen? (3) Är det möjligt att minska antalet CIP-cykler med hjälp av en MILP-modell? (4) Hur kan den potentiella övergången till en automatiserad optimeringsmodell påverka stabiliteten i planeringsprocessen, särskilt när det gäller beroendet av individuell mänsklig expertis?

Metod: Studien använder ett ramverk inom operationsanalys (Operations Research) för att utveckla en blandad linjär heltalsmodell i Python. Primärdata samlades in genom semistrukturerade intervjuer, medan kvantitativ validering utfördes genom att retrospektivt utvärdera modellens scheman mot 39 veckors historiska manuella produktionsplaner från 2025.

Resultat: MILP-modellen hanterade sekvensberoende ställtidsrestriktioner och genererade snabbare scheman under 26 av de 39 utvärderade veckorna med en basberäkningstid på 2000 sekunder, vilket resulterade i en produktionseffektivisering på 2,9 %. De 13 veckor där den manuella planeringen resulterade i kortare produktionstid präglades av hög kombinatorisk komplexitet. Utökade tester (25 000 sekunder) på de mest komplexa veckorna visade dock att modellen har potential att överträffa den manuella planeraren när den ges tillräcklig beräkningstid. Studien drar slutsatsen att en hybridmetod ger det optimala praktiska resultatet. Genom att använda den matematiska modellen för att generera ett grundschema kan den mänskliga planeraren fokusera på att hantera stokastiska störningar, uppströmsberoenden och praktiska undantag. Denna kombination av matematisk optimering och mänsklig expertis säkerställer ett högt kapacitetsutnyttjande samtidigt som systemets stabilitet bibehålls.

Nyckelord: Blandad linjär heltalsmodell, Produktionsplanering, Semikontinuerlig tillverkning, Sekvensberoende ställtider, Oatly, Kombinatorisk optimering, MILP

Preface

This work was carried out in close collaboration with Oatly Group AB and the Division of Production Management at the Faculty of Engineering at Lund University (LTH).

We would like to express our sincere gratitude to Oatly for the opportunity to conduct this thesis. The project allowed us to explore a strategically relevant topic within a leading plant-based food company and apply our academic knowledge in a highly complex, practical setting. We are especially thankful to our supervisors at Oatly, Kristoffer Helgesson and Marcus Johansson, for their support, engagement, and valuable guidance throughout the process. Their feedback and insights helped steer the thesis in the right direction and ensured access to critical resources, data, and personnel.

Finally, we wish to thank our academic supervisor, Johan Marklund, for his guidance and constructive feedback. His experience in operations research and continued support have been greatly appreciated throughout the course of this work. We would also like to acknowledge our examiner, Danja Sonntag, for her time and evaluation of this project.

Lund, May 2026, Mattias Lewerin and Emil Olefjord Widding

Table of Contents

1. INTRODUCTION	1
1.1 BACKGROUND	1
1.1.1 Market Dynamics and the Plant-Based Shift.....	2
1.1.2 Theoretical Characteristics of the Food Industry.....	2
1.2 CASE COMPANY DESCRIPTION	4
1.2.1 The Landskrona Facility and its Production Process.....	4
1.2.2 The Planning Environment	6
1.3 PROBLEM STATEMENT.....	7
1.3.1 NP-Hardness and Mathematical Complexity at Oatly	7
1.3.2 Human Factors and Cognitive Limitations.....	10
1.3.3 Organisational Risk and Knowledge Concentration.....	12
1.4 PURPOSE	12
1.4.1 Theoretical Conceptualisation.....	12
1.4.2 Model Development and Empirical Evaluation.....	13
1.5 RESEARCH QUESTIONS	13
1.6 DELIMITATIONS	13
1.6.1 Focus on High-Volume Edge Lines	14
1.6.2 Geographical and Organisational Scope	14
1.6.3 Process Boundaries	14
1.6.4 Deterministic Modelling Approach.....	14
2. METHODOLOGY	15
2.1 OPERATIONS RESEARCH FRAMEWORK	15
2.2 EVALUATION OF MODELS.....	16
2.2.1 Mixed-Integer Linear Programming.....	16
2.2.2 Rule-Based and Simulation Heuristics	17
2.2.3 Constraint Programming.....	17
2.2.4 Simulation-Based Scheduling	18
2.3 LITERATURE REVIEW	18
2.3.1 Optimisation in Semicontinuous Food Processing.....	18
2.3.2 Comparative Studies of Modelling.....	19
2.4 SELECTION OF MIXED-INTEGER LINEAR PROGRAMMING	19
2.4.1 MILP to Handle Discrete Choices.....	20
2.4.2 Hard Constraints	20
2.4.3 Sustainability and Capacity Goals.....	20
2.4.4 Institutionalising Tacit Planning Knowledge	21
2.5 MOTIVATION OF OTHER CHOICES	21
2.5.1 Software Environment: Python, VSCode, and PuLP.....	21
2.5.2 Continuous Time Variable	22
2.6 DATA COLLECTION.....	22
2.6.1 Primary Data: Semi-structured Interviews.....	23
2.6.2 Secondary Data: Oatly Quantitative Data	23
2.7 RESEARCH QUALITY AND VALIDATION.....	23
2.7.1 Computational Performance.....	23
2.7.2 Verification on Historical Data Sets.....	25
2.7.3 Reliability and Replicability	26
2.7.4 Results Validation and Iterative Feedback.....	28
2.8 ETHICAL CONSIDERATIONS IN INDUSTRIAL RESEARCH.....	30

3. THEORY	32
3.1 FUNDAMENTALS OF PRODUCTION PLANNING AND CONTROL	32
3.1.1 Hierarchical Planning Horizons.....	32
3.1.2 Short-term Scheduling and Sequencing.....	33
3.2 MATHEMATICAL OPTIMISATION AND MODELLING	34
3.2.1 The Scope of Linear Programming Problems	35
3.2.2 Structural Components of the Linear Programming Model	35
3.2.3 Theoretical Assumptions and Mathematical Principles of Linear Programming	38
3.2.4 Mixed-Integer Linear Programming.....	38
3.2.5 The Branch-and-Bound Algorithm	39
4. MODELLING.....	41
4.1 MATHEMATICAL MODEL DEVELOPMENT	41
4.1.1 Model Parameters.....	41
4.1.2 Decision Variables.....	47
4.1.3 Objective Function.....	48
4.1.4 Model Constraints.....	50
4.1.5 Preprocessing	58
4.2 PYTHON MODEL DEVELOPMENT	59
4.2.1 Setting up the Code and Parameters	59
4.2.2 Turning Mathematics into Logic with LLM.....	59
4.2.3 Results and Schedule Generation	60
5. RESULTS	61
5.1 MODEL CONVERGENCE AND COMPUTATIONAL PERFORMANCE.....	61
5.2 EFFICIENCY COMPARISON: MILP VS. MANUAL PLANNING.....	62
5.2.1 2000-Second Simulation for the 39-Week Baseline.....	62
5.2.2 Increased Computational Time for Complex Weeks.....	64
5.3 CLEAN-IN-PLACE CYCLES.....	65
6. DISCUSSION AND CONCLUSIONS.....	66
6.1 PERFORMANCE EVALUATION: MODEL VS. MANUAL PLANNING	66
6.1.1 Week-to-Week Variability of Makespan.....	66
6.1.2 The Case for a Hybrid Planning Approach	66
6.1.3 Impact of Product Complexity and Solver Constraints	67
6.1.4 Validation of results.....	68
6.2 DATA INTEGRITY AND TRANSPARENCY.....	69
6.3 ENVIRONMENTAL IMPACT AND RESOURCE EFFICIENCY.....	69
6.4 ORGANISATIONAL RESILIENCE AND KNOWLEDGE MANAGEMENT.....	70
6.5 PRACTICAL LIMITATIONS AND MODEL BOUNDARY CONDITIONS	71
6.6 COMPUTATIONAL TIME LIMITS AND DIMINISHING RETURNS.....	72
6.7 RECOMMENDATIONS FOR FUTURE RESEARCH.....	73
6.8 CONCLUSION	74
7. REFERENCES	75

List of Figures

Figure 1, Sterilisation/packaging in Oatly's production process.....	5
Figure 2, Order example from the Oatly supply chain department	6
Figure 3, Picture of colour-coded rules for SKU transitions.....	9
Figure 4, Part of the Changeover Matrix.....	10
Figure 5, Manual scheduling example in Excel	11
Figure 6, The five-stage OR process (Hillier & Lieberman, 2010).....	15
Figure 7, Example of diminishing returns	24
Figure 8, Example of a branching tree. Adapted from Hillier & Lieberman (2010).....	39
Figure 9, Relative improvement from the solver's initial incumbent over 2000 seconds of computation time.	61

List of Tables

Table 1, Comparison of several polynomial and exponential time complexity functions. Adapted from Garey & Johnson (1979)	8
Table 2, Variables of LP Problem	36
Table 3, Penalty Coefficients for the Objective Function	46
Table 4, Week by week results.....	63
Table 5, Increased Computational Time for Complex Weeks	65

List of Abbreviations

AMB: Ambient

B1: Brik filling line

CAGR: Compounded Annual Growth Rate

CBC: COIN-OR Branch and Cut

CHILL: Chilled

CIP: Clean-In-Place

CO₂e: Carbon dioxide equivalent

CP: Constraint Programming

CPF: Corner-point feasible

DBR: Drum-Buffer-Rope

E2, E3, E4: Edge filling lines

GT1: Gable Top filling line

IDE: Integrated Development Environment

IP: Integer programming

J1: Jigger filling line

LKR: Landskrona facility

MILP: Mixed-Integer Linear

MO: Manufacturing Order

MPC: Manufacturing Plannir

MPS: Master Production Schedule

OR: Operations Research

PPC: Production Planning and Control

RAM: Random Access Memory

S&OP: Sales and Operations Planning

SDST: Sequence-dependent setup times

SKU: Stock-Keeping Unit

TBA: Tetra Brik Aseptic packaging

TOC: Theory of Constraints

TSP: Travelling Salesman Problem

UHT: Ultra-High Temperature

WIP: Work-in-progress

1. Introduction

Production planning plays a central role in how process industries manage their daily operations. At facilities with high product variety and shared equipment, even small scheduling decisions can influence the overall flow of production. This study examines such a setting and focuses on the practical challenges that arise when coordinating cleaning requirements, machine capacities, and shifting demand.

1.1 Background

The global food industry is currently transforming, driven by an escalating awareness of sustainability, public health, and the environmental impact of traditional livestock-based dairy production (Institute of Sustainability Studies, 2025). At the forefront of this "plant-based revolution" is Oatly, a company dedicated to providing sustainable nutritional solutions that facilitate a shift towards a more resilient food system (Oatly Group AB, n.d.). The market for oat-based dairy alternatives has experienced a large growth; the global market valuation reached approximately USD 3.67 billion in 2024 and is projected to exceed USD 12 billion by 2033, representing a compounded annual growth rate (CAGR) of approximately 14% (Straits Research, 2025). This rapid expansion places high pressure on production facilities to maximise throughput and operational efficiency, particularly at high-volume sites such as Oatly's Landskrona facility (LKR Site).

To navigate this complexity, the application of Operations Research (OR) becomes essential. According to Hillier and Lieberman (2010), OR is a scientific approach to decision-making that involves the use of mathematical models to solve problems related to resource allocation and system design. In the semicontinuous process industry, where food production is often categorised, the planning problem is particularly challenging. It requires the synchronisation of continuous upstream processes (extraction and formulation) with discrete downstream packaging steps while adhering to strict hygiene regulations and sequence-dependent setup constraints (Kopanos et al., 2011).

According to Georgiadis et al. (2019), the increase of unique stock-keeping units (SKUs) and the absence of dedicated lines for every product necessitate the transition from experience-based heuristics to sophisticated mathematical scheduling methods to maintain high capacity utilisation.

1.1.1 Market Dynamics and the Plant-Based Shift

The surge in demand for oat milk is influenced by several factors. Consumers increasingly prioritise products that are lactose-free and have a lower climate footprint compared to regular milk (Straits Research, 2025). Oatly's flagship products report a climate footprint of approximately 0.44 kg CO₂e per litre, representing an 86% reduction in emissions compared to dairy milk (Institute of Sustainability Studies, 2025). According to Agu et al. (2024) this sustainability-first business model drives consumer loyalty, but also introduces a high degree of product variety, ranging from conventional organic drinks to specialised "Barista" editions and flavoured variants.

This growing product variety is particularly evident in the Barista segment, which requires precise formulation, and the shift towards organic products, which necessitates stricter cleaning requirements to prevent cross-contamination. These market trends mean that production lines at the LKR Site must handle frequent transitions between recipes, where every changeover incurs costs in the form of downtime, water consumption, and chemical usage. In the field of OR, these are modelled as sequence-dependent setup times, where the cost of transitioning from product A to product B depends on the specific characteristics of both products (Montoya-Torres et al., 2010).

1.1.2 Theoretical Characteristics of the Food Industry

The food processing industry is theoretically classified as a semicontinuous or hybrid manufacturing environment. This classification stems from the integration of two different production modes: a continuous process upstream, where raw materials are transformed in fluid or bulk form, and a downstream discrete process, where these materials are divided into individual packaging units. According to Vollmann et al. (2005), the primary coordination challenge in systems like these is the synchronisation of the two stages. The continuous flow of the liquid base must be precisely matched with the non-continuous, batch operation of the packaging equipment. Failure to achieve this synchronisation results in either material waste or stagnant equipment, making the interface between fluid processing and packaging a critical optimisation point in a production facility.

A characteristic of the semicontinuous industry is its divergent product structure. As described by Kopanos et al. (2011), food manufacturing typically involves a branching material flow where a relatively small number of raw materials are used to produce a wide variety of end products.

According to Schragenheim & Dettmer (2000), this is known as “V flow”. In a V-type facility, the material flow starts at the bottom with a limited number of inputs and fans out towards the end of the process. At Oatly’s plant in Landskrona and other such plants, a single raw material or product base is diverted into a large number of different SKUs based on varying characteristics, sizes and packaging formats. This diverging structure puts the management focus onto the final stage of the process, as the scheduling decisions made at the packaging lines dictate the requirements and constraints for all previous processing steps.

Managing semicontinuous systems often utilises the principles of the Theory of Constraints (TOC), which focuses on identifying and managing the most restrictive resource in a production process. According to Vollmann et al. (2005), production throughput is typically governed by a bottleneck. In food processing facilities like Oatly, the filling and packaging equipment often serves as this primary constraint. Because the total system throughput is limited by the capacity of these fillers, the production activities upstream should be coordinated using a pull logic to prevent overproduction and excessive work-in-process (WIP) inventory (Vollmann et al., 2005). In a pull-based system, production-based activities are not initiated based on the available raw material; this would signify a push-based system. The activities are instead “pulled” through the factory based on the capacity or requirements of the bottleneck resource.

To ensure this coordination is successful, the Drum-Buffer-Rope (DBR) method is often used. As described by Vollmann et al. (2005), the bottleneck resource acts as the “drum”, setting the “beat” (tempo) for all activities upstream. The “rope” represents the mechanism that handles timing or communication to prevent upstream processing from overproducing, while a “buffer” of material is maintained to protect the bottleneck from running out of work. This method is comparable to Oatly’s production and ensures that the most constrained resource (the bottleneck) in the plant is utilised at its maximal capacity, as stagnation of this resource means downtime for the entire system.

Unlike discrete manufacturing, food production must adhere to strict biological and safety regulations that introduce hard constraints. Pinedo (2016) emphasises that these systems are subject to mandatory hygiene intervals, which are fixed time windows after which production must stop for a full Clean-In-Place (CIP) cycle. These intervals are required to prevent microbial growth and ensure product safety, regardless of the remaining

demand. In the context of Oatly, all machines and lines have a mandatory full CIP after a set time of running.

These cleaning requirements introduce a recurring challenge in scheduling. The planner cannot simply optimise for the shortest setup times between products. Instead, they must also account for the resetting of the hygiene status of the equipment. Georgiadis et al. (2019) argue that these hygiene rules act as “hard constraints” that force a break in the continuous flow, transforming the scheduling into a balance between maximising productive run-time and adhering to hygiene-driven downtime.

1.2 Case Company Description

Oatly, founded in the 1990s based on research in response to lactose intolerance, has evolved into one of the world's largest oat drink companies (Institute of Sustainability Studies, 2025). In 2024, the company produced approximately 576.3 million litres of finished goods, representing a 13.8% increase compared to the previous year (Oatly Group AB, 2024). This volume growth has been accompanied by a strategic restructuring of the supply chain to achieve profitable growth, focussing on core high-efficiency facilities like the Landskrona site (Oatly Group AB, 2024).

1.2.1 The Landskrona Facility and its Production Process

The Landskrona plant serves as Oatly's primary European production hub and a centre for technical innovation. The facility is designed for end-to-end production, from raw oat intake to final packaging. The site is structured around several parallel lines with specific technical characteristics, as shown in Figure 1.

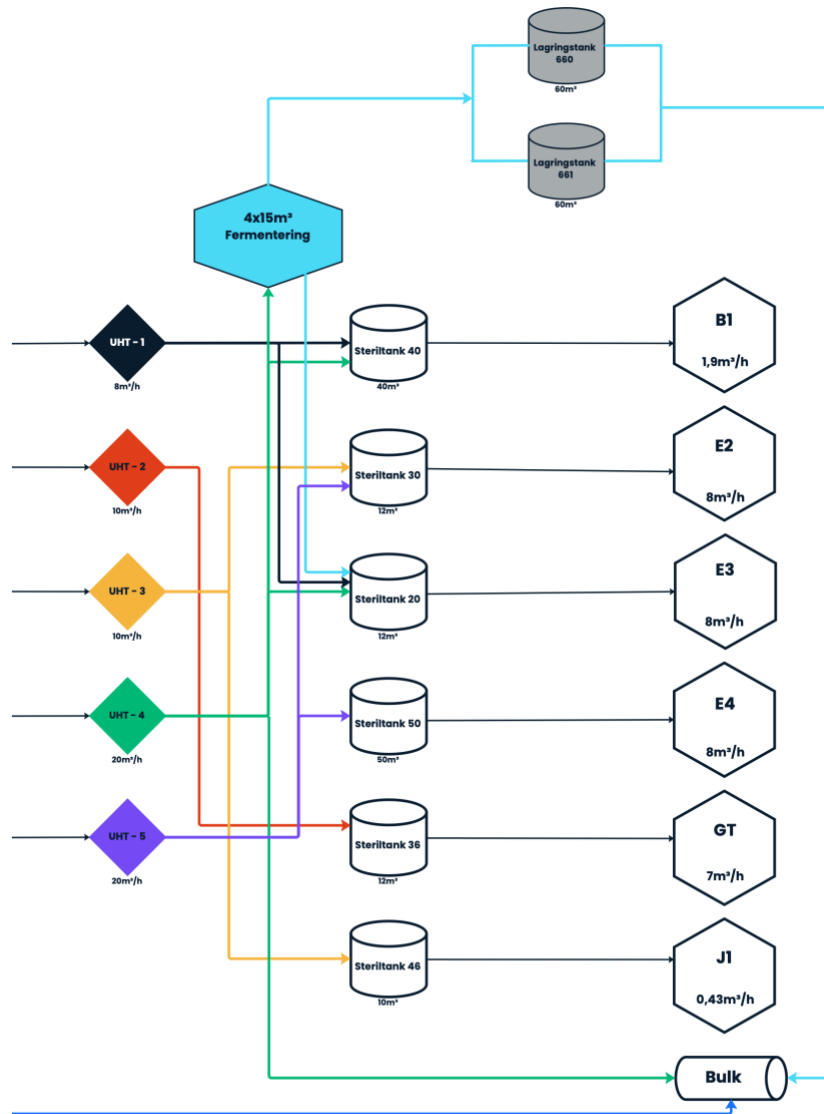


Figure 1, Sterilisation/packaging in Oatly's production process

The manufacturing process at the Landskrona facility is a multi-stage operation starting with raw oat silos and concluding at automated packaging lines. The process is characterised by three primary phases: extraction, formulation (mixing), and the last step which can be shown in Figure 1,

sterilisation/packaging. Each stage is governed by technical parameters such as flow rates (m^3/h), tank volumes, and piping constraints.

In the first phase, a liquid oat base is extracted through milling, enzymatic treatment, and centrifugation. This base is stored in large intermediate tanks before being routed to mixing tanks, where additives such as oils, salts, and flavours are added. Following formulation, the product undergoes Ultra-High Temperature (UHT) treatment to ensure sterility and shelf-life stability. The finalised sterile product is temporarily held in aseptic buffer tanks before being sent to the filling machines.

1.2.2 The Planning Environment

Production planning at Oatly follows a pull-based logic. Each week, the supply chain department provides a finalised order list specifying SKU volumes, delivery dates, and packaging requirements, as shown in Figure 2.

Förpackning	Innehåller	Artikelnr	Artikelnamn	Begärt datum	Leveransdatum	Behovskvantitet
TBA 0,25 Base	500024	61433	Oatly iMat 0,25	07/04/2025	14/04/2025	XX
TBA 0,25 Base	500024	62006	Oatly Hafer Cuisine Zum Kochen 0,25 AT/CHE/DE	07/04/2025	14/04/2025	XX
TBA 0,25 Base	500073	61890	Oatly Hafer Cuisine zum Aufschlagen 0,25 DACH	07/04/2025	14/04/2025	XX
TBA 0,25 Base	500073	62001	Oatly Whippable Creamy Oat 0,25L UK	07/04/2025	14/04/2025	XX
TBA 0,25 Base	500073	62005	Oatly iMat Visp 0,25 DK/FI/NO/SE	07/04/2025	14/04/2025	XX
TBA 1,0 Edge	500080	61574	Oatly Havregurt Naturell 1,0	07/04/2025	14/04/2025	XX
TBA 1,0 Edge	500081	61672	Oatly Havregurt Vanilj/Vanilja 1,0	07/04/2025	14/04/2025	XX
TBA 1,0 Edge	500082	61575	Oatly Havregurt Jordgubb 1,0 DK/FI/NO/SE	07/04/2025	14/04/2025	XX
Jigger	500091	61720	Oatly Oat drink for coffee tea 0,02	07/04/2025	14/04/2025	XX
TBA 1,0 Edge	500310	61578	Oatly Oat Drink Organic 1,0	07/04/2025	14/04/2025	XX
TBA 1,0 Edge	500310	61589	Oatly Hafer/Haver BIO 1,0	07/04/2025	14/04/2025	XX
TBA 1,0 Edge	500349	61771	ICA Havredryck Naturell 1,0	07/04/2025	14/04/2025	XX
Rex	500370	61798	Oatly Oat Drink Organic Semi 1,0 GT	07/04/2025	14/04/2025	XX
Rex	500372	66505	Oatly Oat Drink 1,5% CHILLED	07/04/2025	14/04/2025	XX
Rex	500373	66518	Oatly Oat Drink Chocolate Deluxe 1,0 GT UK	07/04/2025	14/04/2025	XX
TBA 1,0 Edge	500376	61774	ICA Havredryck KRAV Eco 1,0	07/04/2025	14/04/2025	XX
Rex	500377	66507	Oatly Havredryck Mellan KRAV Ekologisk 1,0 GT	07/04/2025	14/04/2025	XX

Figure 2, Order example from the Oatly supply chain department

The planner's objective is to allocate these orders across the available filling lines (E2, E3, E4, B1, GT1, J1) while navigating hardware dependencies. One such constraint is the fixed piping between UHT tanks and filling lines; for example, UHT-5 is dedicated to Line E2 and E4, while UHT-3 feeds Line E2 and J1. These hardwired routings limit the flexibility of the planner and necessitate precise scheduling to prevent tank starvation or overflows.

1.3 Problem Statement

Oatly has a stated objective to achieve profitable growth during 2025 (Oatly Group AB, 2024). Supporting this goal at the Landskrona facility requires maximising the output of the high volume Edge lines (E2, E3, and E4). Currently, these lines output an average of 0.9 million litres per week. The machines cannot operate continuously because they require regular service and mandatory four hour cleaning routines. However, even with these required stops, the current production volume is lower than the theoretical capacity of the lines. This thesis aims to optimise the production schedules to utilise the remaining available machine time, thereby increasing the weekly volume and supporting the company's financial targets.

1.3.1 NP-Hardness and Mathematical Complexity at Oatly

The mathematical difficulty of establishing an optimal production plan at the Landskrona facility stems from the combinatorial nature of sequencing decisions. According to Garey & Johnson (1979), when scheduling a set of n distinct products (SKUs) on a production line where setup times are sequence-dependent, the number of possible permutations is determined by the factorial of the number of jobs. These possible sequences can be expressed as $n!$.

This leads to a phenomenon known as combinatorial explosion (Garey & Johnson, 1979). For a relatively small set of 10 SKUs, there are over 3.6 million possible sequences. However, as product variety increases to the levels observed in modern food processing, such as 50 or more unique SKUs, the search space grows to approximately 3×10^{64} alternatives. As illustrated in Table 1, the growth rate of factorial functions quickly outpaces even high-degree polynomial functions.

Table 1, Comparison of several polynomial and exponential time complexity functions.
Adapted from Garey & Johnson (1979)

Time complexity function	Size					
	10 second	20 second	30 second	40 second	50 second	60 second
n	.00001 second	.00002 second	.00003 second	.00004 second	.00005 second	.00006 second
n^2	.0001 second	.0004 second	.0009 second	.0016 second	.0025 second	.0036 second
n^3	.001 second	.008 second	.027 second	.064 second	.125 second	.216 second
n^5	.1 second	3.2 seconds	24.3 seconds	1.7 minutes	5.2 minutes	13.0 minutes
2^n	.001 second	1.0 second	17.9 minutes	12.7 days	35.7 years	366 centuries
3^n	.059 second	58 minutes	6.5 years	3855 centuries	2×10^8 centuries	1.3×10^{13} centuries

The theoretical framework for understanding this difficulty is provided by the study of computational complexity and the concept of intractability. Garey & Johnson (1979) define a problem as "intractable" if it is so hard that no polynomial-time algorithm can possibly solve it. Computer scientists generally recognise a simple distinction between polynomial time algorithms, which are bounded by $O(n^k)$ for some polynomial function p , and exponential time algorithms. There is wide agreement that a problem has not been "well-solved" until a polynomial time algorithm is known for it, as polynomial time algorithms tend to be "provably efficient" and remain computationally manageable even as the instance size grows (Garey & Johnson, 1979).

Problems are classified as NP-hard when they are at least as complex as the hardest problems in the NP class. In practice, this means that they are non-polynomial in complexity, and the time required to find an optimal solution grows exponentially as the problem size increases. This makes them computationally intractable for standard algorithmic approaches (Garey & Johnson, 1979). The task of sequencing jobs with sequence-dependent setup times is a variant of the Travelling Salesman Problem (TSP), which involves finding the shortest tour through a set of cities (Garey & Johnson, 1979). Because the TSP is a classic example of an inherently intractable NP-hard optimisation problem, there is no known algorithm that can guarantee an optimal solution in polynomial time for large instances (Garey & Johnson, 1979). Problems of this type are consequently characterised by a computational requirement that increases at a rate exceeding the capabilities of deterministic computation as instance sizes grow, making the identification of a global optimum within a reasonable timeframe theoretically impossible for standard algorithmic approaches.

At Oatly, the core of the scheduling difficulty lies in the sequence-dependent setup times governed by Oatly's "Sortbytesmatrix" (Changeover Matrix). This matrix defines the cleaning requirements between any two SKUs. Transitions are categorised by colour-coded rules. For example, "green" transitions require no wash, while "red" transitions require a sterile CIP, as shown in Figure 3.

Ok to switch according to routine MSDOC#1291 (no cleaning after formulation tank)
Cleaning of mixer and transfer line and formulation tank (OK with ACIP if full CIP not applicable)
Cleaning of formulation tank, mixer and transfer line (excluding UHT, sterile tank and filter)
Cleaning from fermentation tank onwards
Cleaning of oat base line

Figure 3, Picture of colour-coded rules for SKU transitions

The complexity in the changeover matrix shown in Figure 4 is further heightened by the large number of possible product combinations; for example, 50 different products create a high variety of potential sequences, although it should be noted that only a subset of these SKUs is processed on the Edge lines.

Sortbytesmatris		Byte till produkt									
Byte från produkt		500310	500434	500024	500340	500080	500343	500344	500349	500358	
Havredryck - EKO KRAV 13%	500310										
Vaniljsås	500434										
iMat raps	500024										
Havrebas till Fermenterad	500340										
Havregurt Naturell	500080										
Havredryck - Berikad 13 % bas-TS	500343										
Havredryck - Choklad Berikad 13% bas-TS	500344										
Havredryck - ICA Naturell 13% bas-TS	500349										
Havredrikk - Coop Norge 13%bas-TS	500358										
Havredryck - ICA Vanilj 13% bas-TS	500350										
Havredryck iKaffe Vanilla-13% bas-TS	500437										
Havredryck iKaffe Popcorn - 13% bas-TS	500442										
Havredryck iKaffe Caramel -13% bas-TS	500438										
Havredryck - ICA Choklad 13% bas-TS	500351										
Havredryck - Choklad Berikad utan jod 13% bas-TS	500441										
Havredryck - Vanilj EKO KRAV	500353										
Havredryck - Choklad EKO 13%	500354										
Havregurt Vanilj	500081										
Havregurt Jordgubb	500082										
Havredryck 1,5% - EKO 13%-TS	500370										

Figure 4, Part of the Changeover Matrix

Determining the optimal sequence of jobs to minimise total setup time is a variation of the TSP (Montoya-Torres et al., 2010). As the number of jobs, represented by various SKUs, grows, the search space expands to a level where it is virtually impossible for a human planner to identify the optimal sequence among the large number of possibilities (Montoya-Torres et al., 2010).

1.3.2 Human Factors and Cognitive Limitations

While intuition is often viewed as playing an important role in decision making, it can be unreliable in complicated business situations (Bonabeau, 2003). As the choices that managers are facing and the data requiring analysis multiply, relying on instinct becomes a “risky delusion” compared to rigorous

analysis (Bonabeau, 2003). This is because human intuition is often a means of ignoring complexity rather than considering it (Bonabeau, 2003). Another clear risk when relying on intuition is that the human brain needs to identify patterns, which often leads planners to see meaningful structure in random data or to wrongly apply past experiences to new phenomena (Bonabeau, 2003).

Usually, production engineers struggle to generate feasible schedules within tight time frames when only relying on experience and empirical rules (Samouilidou et al., 2023). This often results in suboptimal schedules that fail to fully use machine capacity or identify the most efficient product sequences (Azani et al., 2025). Regardless of a planner's long and deep knowledge of the production process, it is theoretically considered impossible for a human to reach near-optimal results for problems with a high combinatorial complexity without the support of mathematical optimisation (Samouilidou et al., 2023).

The performance gap at Oatly is partly attributed to manual, experience-based planning. As shown in Figure 5, schedules are developed using Excel-based models that rely on the expertise and tacit knowledge of individual planners. While these planners possess deep operational knowledge, optimising hundreds of variables (including tank availability, UHT flow rates, and sequence-dependent cleaning) is difficult to achieve manually (Azani et al., 2025). While humans currently manage these variables, they do not reach an optimal solution, highlighting the need for an automated support tool or model.

E3	Edge							
MO	ArtikeLnr	Innehåller	Försäljningsvarunamn	Volym (tråg)	Volym (L)	Start	KL	Amb/Chill
MO10500048180	61578	500310	Oat Drink Organic 1,0	27,500	165,000	Måndag	00:00	AMB
MO10500048217	69000	590004	FF1 Matcha			Tisdag	10:00	INFO
MO10500048178	61589	500310	Hafer/Haver BIO 1.0	27,500	165,000	Tisdag	15:00	AMB
MO10500048182	61574	500080	Havregurt Naturell 1.0	4,300	25,800	Onsdag	19:08	CHILL
MO10500048179	61672	500081	Havregurt Vanilj/Vanilja 1,0	2,150	12,900	Torsdag	00:22	CHILL
MO10500048181	61575	500082	Havregurt Jordgubb 1.0	2,150	12,900	Torsdag	03:30	CHILL
MO10500048198	62168	500418	Oat drink Barista edition BIO	17,741	106,446	Torsdag	10:37	AMB
MO10500048199	62168	500418	Oat drink Barista edition BIO	32,676	196,056	Fredag	05:07	AMB
MO10500048192	61774	500376	ICA Havredryck Eco KRAV 1.0	4,584	27,504	Lördag	18:22	AMB
MO10500048395	62280	500437	Espresso House iKaffe Vanilla 1,0L	4,584	27,504	Lördag	23:54	AMB
MO10500048190	61771	500343	ICA Havredryck Naturell 1,0	18,333	109,998	Söndag	05:25	CHILL

Figure 5, Manual scheduling example in Excel

For Oatly, running the production more efficiently is not merely an economic objective but a part of their sustainability, directly contributing to their goal of reducing water withdrawal to 2.2 litres per litre of product by 2030 (Oatly Group AB, n.d.).

1.3.3 Organisational Risk and Knowledge Concentration

Furthermore, the concentration of planning expertise in a single individual introduces clear organisational risk. Transitioning to an automated model not only enhances capacity and sustainability but also mitigates the risks associated with tacit knowledge being confined to specific personnel (Azani et al., 2025). By shifting towards a mathematical model, the organisation can institutionalise operational knowledge, thereby providing a robust decision-support framework that functions independently of individual intuition (Lillyworks, n.d.).

1.4 Purpose

The primary objective of this master's thesis is to develop a mathematical optimisation model that minimises the total makespan required to complete a given weekly order list at Oatly's Landskrona facility (the order list illustrated in Figure 2). By generating schedules that complete the specified production volume in less time than the current manual, experience-based approach, the model is intended to serve as a decision-support tool that releases additional capacity at the end of the production cycle, which can then be utilised for further manufacturing.

Beyond this primary goal, the thesis pursues two secondary objectives. The first is to reduce the environmental footprint of production: by shortening the total time the facility must remain in active operation, the model lowers the consumption of resources such as electricity, heating, and cooling that scale with runtime, thereby supporting Oatly's broader sustainability targets. The second is to mitigate the organisational risk that arises from concentrating critical planning knowledge in a single individual. By encoding the facility's tacit scheduling logic into a transparent and reproducible mathematical framework, the model reduces dependence on any one planner's expertise and strengthens the long-term stability of the planning process.

1.4.1 Theoretical Conceptualisation

The research is structured into two main phases, beginning with theoretical conceptualisation. This first phase focuses on defining the theoretical requirements for an optimal production schedule within a semicontinuous process industry. It involves identifying mathematical representations for sequence-dependent setup times and complex CIP rules as defined in the

Oatly changeover matrix. Furthermore, it includes the formulation of constraints that accurately reflect the technical hardware dependencies.

1.4.2 Model Development and Empirical Evaluation

The second phase involves the practical development of the model and evaluation of its performance. The primary aim is to create a decision-support tool capable of generating technically feasible and optimised schedules based on real supply chain data. By comparing the model's optimised schedules with historical, manually generated schedules, the study seeks to quantify the potential for efficiency gains. Additionally, this phase analyses how a data-driven model reduces the dependency on individual experience, thereby enhancing the stability and resilience of the planning process. Through these objectives, the study intends to contribute to the field of industrial engineering by demonstrating how OR can be applied in a high-growth production environment to balance profitability with ambitious sustainability targets.

1.5 Research Questions

To fulfil the purpose of the study and address the identified problem, the following research questions have been formulated:

1. How can Oatly's specific production constraints, including sequence-dependent cleaning rules and routing dependencies, be translated into a model to create a technically feasible production schedule?
2. To what extent can the total production time for a given order list be minimised by utilising an optimisation model compared to the current experience-based manual planning?
3. Is it possible to reduce the number of CIP cycles with a MILP model?
4. How may the potential transition to an automated optimisation model affect the stability of the planning process, specifically regarding the reliance on individual human expertise?

1.6 Delimitations

To ensure the depth and accuracy of the research within the 20-week timeframe of a master's thesis, the following delimitations have been established.

1.6.1 Focus on High-Volume Edge Lines

The study is limited to the optimisation of the three Edge filling lines (E2, E3, and E4) at the Landskrona site. These lines represent the majority of the facility's throughput and the highest degree of planning complexity due to SKU variety. The production machines Brik (B1), Jigger (J1) and Gable Top (GT1) are not taken into consideration to ensure the feasibility and right amount of work for this project. Another reason for excluding the GT1 and B1 is also because they are both to be replaced in the near future.

1.6.2 Geographical and Organisational Scope

The research is confined to the LKR Site. While Oatly operates globally, the specific hardware configurations and constraints at LKR are unique and serve as the basis for the model's parameters. Furthermore, the project aims to deliver a validated optimisation model as a decision-support tool; the actual software integration or the design of a front-end user interface is outside the scope of this work.

1.6.3 Process Boundaries

The model focuses on the scheduling of the filling machines and the immediate upstream buffer stages. While it considers the availability of oat base and formulation tank capacities to ensure feasibility, it does not optimise the technical parameters of extraction (e.g., enzyme ratios or milling speed). It also assumes that the normal and fermented bases are always available. Downstream logistics, such as warehouse management and customer distribution, are treated as fixed external requirements and are not part of the optimisation horizon.

1.6.4 Deterministic Modelling Approach

The model is developed as a deterministic optimisation tool. This implies that all inputs (such as processing times, setup durations, and order quantities) are assumed to be known and constant. While real-world production is subject to stochastic disruptions (e.g., equipment failures), the purpose of this study is to establish an optimal “best-case” baseline to demonstrate the potential of mathematical optimisation.

2. Methodology

The approach and methodology of this thesis follow the systematic OR framework presented by Hillier and Lieberman (2010), providing a five-step progression from problem definition to creating a model as shown in Figure 6. This systematic approach is important in the quest for optimised supply chain and production solutions, particularly when addressing the complexities of physical product distribution and manufacturing (Hillier & Lieberman, 2010).

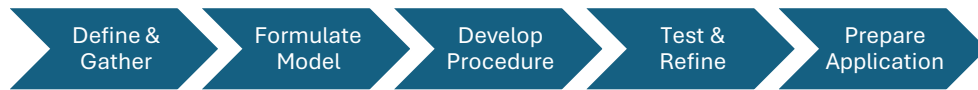


Figure 6, The five-stage OR process (Hillier & Lieberman, 2010).

2.1 Operations Research Framework

Generally, Hillier and Lieberman (2010) emphasise that a successful OR project must begin with a rigorously defined scope and accurate empirical data. The theory stresses that mathematical models are highly sensitive to their input parameters, and solving the wrong problem or using poor-quality data will make even the most complex algorithms useless. At Oatly, this entails a collection of historical records regarding planned versus actual production output, specifically incorporating cleaning matrices and machine status data from previous operational periods.

In the second stage, the framework emphasises the critical task of translating the physical, real-world system into a mathematical representation. Generally, this requires defining a clear objective function to optimise, alongside decision variables and constraints that accurately restrict the model to what is physically possible. Specifically, a scheduling model is constructed with the primary objective of minimising total production time for one order. The model is formulated to respect a set of critical operational constraints, including mandatory hygiene cycles and sequence-dependent setup times. Such a joint consideration of operational constraints is important for achieving sustainable and realistic results (Hillier & Lieberman, 2010).

In the third step, a computer-based procedure is developed to create actionable solutions from the scheduling model. Hillier and Lieberman point out that real-world OR problems are often too complex to solve manually, requiring robust algorithms or dedicated solvers to find optimal or near-optimal solutions. Consequently, this project addresses this stage by

implementing the formulated model into a dedicated optimisation solver. This computational approach allows the system to automatically calculate optimal job sequences, offering a highly efficient alternative to Oatly's traditional manual planning processes.

To ensure the validity and reliability of the proposed solution, the fourth stage focuses on testing and iteratively refining the model. The theory emphasises that because models are inherently simplified approximations of reality, their outputs must be cross-examined against historical data to ensure they produce valid, executable solutions before any real-world deployment. To fulfil this theoretical requirement, the proposed model is evaluated numerically by benchmarking its suggested schedules against actual historical production weeks. This iterative testing process ensures that the model's output not only improves production time but remains feasible within the actual factory environment.

Finally, the framework emphasises that an OR project is not complete until the solution is prepared for practical integration into the organisation's daily operations. This generally involves establishing documentation, user support, and a clear implementation strategy for the decision-makers. However, this study does not involve the actual implementation of the model at Oatly.

2.2 Evaluation of Models

To understand the best way to handle the scheduling problems at Oatly, the various modelling options must be evaluated. Since every production facility operates under a unique set of constraints, it is necessary to examine how different models address these requirements in practice. This section explains four different approaches to industrial scheduling and relates them to the LKR Site to provide a baseline for choosing the most suitable model for the task.

2.2.1 Mixed-Integer Linear Programming

Mixed-Integer Linear Programming serves as a mathematical framework used for problems where constraints and objective functions have a linear structure, but where a subset of decision variables is restricted to integer or binary values while others remain continuous (Hillier & Lieberman, 2010). In the context of the LKR Site, Mixed-Integer Linear Programming (MILP) utilises binary variables to represent discrete “either/or” choices, such as whether a specific manufacturing order (MO) is assigned to a particular filling line or the exact sequence of jobs required to minimise cleaning times. As a scientific approach to decision making, MILP is particularly effective for

solving optimisation problems of high combinatorial complexity, since it can systematically evaluate millions of potential sequences to identify a mathematically optimal plan that minimises downtime and maximises throughput (Samouilidou et al., 2023). It is particularly capable of handling “hard” constraints, such as the mandatory cleaning to meet hygiene rules and specific hardware dependencies, by treating them as hard mathematical constraints. However, the primary challenge of the MILP approach is its computational intensity. As the number of SKUs increases, the search space grows factorially, making it an NP-Hard problem where the time required for a solver to reach an optimal solution increases notably (Garey & Johnson, 1979).

2.2.2 Rule-Based and Simulation Heuristics

Rule-Based models reach scheduling decisions using simulation-based techniques and simple empirical rules, often referred to as heuristics. The logic here is procedural, meaning that the model builds the schedule one step at a time by asking “what should I do next?” based on a list of priorities. In an industrial setting like Oatly, this approach would involve a system following pre-set logic, like prioritising compatible product bases, to build the schedule step by step. These models are highly valued in the process industry for their computational speed, typically generating feasible production plans almost immediately (Kopanos et al., 2011). However, Samouilidou et al. (2023) note that rule-based models often generate suboptimal results because they do not “search” for the best solution sequence across the whole solution space. The quality of the schedule is heavily dependent on the experience and tacit knowledge of the human planner who designed the rules, which makes the process vulnerable to cognitive load and individual error (Azani et al., 2025).

2.2.3 Constraint Programming

Constraint Programming (CP) is a framework where the focus is on feasibility rather than optimisation. While something like MILP uses equations to find a critical value, CP uses domain “pruning” which is a way to reduce the domain to cross out impossible solutions (Apt, 2003). Unlike standard mathematical programming, CP offers high amounts of flexibility in how constraints are stated, including rules ensuring tasks to not overlap and logical rules like IF, AND, OR. A key advantage of CP is its use of “global constraints” which can define complex patterns across multiple variables, which is like commanding multiple variables at the same time instead of doing it one by one (Apt, 2003). Functionally, the distinction between CP and MILP is found in their primary

objective: CP is great at quickly finding a feasible schedule when you have many conflicting rules, but it typically lacks the efficiency of MILP when it comes to finding a high quality or optimal solution.

2.2.4 Simulation-Based Scheduling

Simulation-based scheduling involves building a digital twin or functional representation of the factory to test different “what-if” scenarios based on specific operational rules (Hillier & Lieberman, 2010). Unlike CP or MILP, which are solvers that look for an answer, a simulation is more of a “tester”. This method allows planners to virtually test the production environment to see, for example, what a specific production sequence has on capacity output. These models are very effective for handling uncertainty and seeing how the plant reacts to stochastic variation, like machine breakdown or other equipment failure (Hillier & Lieberman, 2010). However, simulation-based tools are less of optimisers and more of rule-followers. They can only report the outcome of a specific sequence provided to them and cannot search for a better sequence. Simulation therefore becomes more of a way to test human intuition, which Bonabeau (2003) argues can lead to error traps and ignoring complexity rather than considering it. The practical applications and selection between these models are further explored in the coming sections.

2.3 Literature review

Research into product scheduling for the food process industry has focussed on shifting away from manual, experience-based planning towards mathematical frameworks (Kopanos et al., 2011). This section reviews existing studies on production scheduling in semicontinuous environments, focussing on the comparative performance of optimisation models and rule-based tools.

2.3.1 Optimisation in Semicontinuous Food Processing

The food industry is characterised by the complex synchronisation of batch-based upstream extraction and continuous downstream packaging. MILP has emerged as the standard scientific approach for managing this complexity, as it can precisely represent the high combinatorial difficulty commonly occurring in facilities with multiple products (Kopanos et al., 2011). Research by Kopanos et al. (2011) demonstrated that MILP models are effective for real-life dairy plants where sequence-dependent setup times and product family groupings are used to simplify the scheduling of dozens of

SKUs. Georgiadis et al. (2019) further emphasised that as the number of unique SKUs increases, mathematical scheduling becomes essential to maintain high-capacity utilisation and keep track of the factorially growing solution space.

2.3.2 Comparative Studies of Modelling

The effectiveness of a production schedule is determined by the choice between exact mathematical optimisation and more flexible, rule-based, or constraint-driven tools. In practice, many facilities rely on commercial tools that utilise simulation-based techniques to build feasible schedules by following preset rules. While these tools have enough speed for immediate “what-if” analysis and being able to reactively reschedule, there is still a clear performance gap regarding the solution’s quality (Samouilidou et al., 2023).

In a study of large-scale dairy facilities, Samouilidou et al. (2023) found that optimisation-based sequences often outperformed rule-based ones by identifying production paths that reduced total operating costs by up to 61%. The authors emphasise that while rule-based tools generate feasible plans quickly, their quality remains heavily dependent on the tacit knowledge and specific experience of the human planner, which makes the planning process vulnerable to cognitive overload and individual error. Similarly, Cui et al. (2022) examined the optimisation of filling time in the dairy production process, concluding that even simpler linear programming models with one-dimensional rules offer clear benefits over manual machine setting modifications.

The literature provides further nuance by comparing MILP with CP, which offers an alternative way of handling intricate planning environments. While MILP relies on linear equations to find a global optimum, Hillier and Lieberman (2010) describe CP as a framework that focuses on identifying any feasible solution by using domain pruning to systematically remove impossible values. According to Harjunkski et al. (2014), CP can be more efficient than traditional programming in settings characterised by many conflicting constraints, where logical rules like IF, OR, and AND can be expressed more concisely. However, the distinction remains that CP is primarily a tool for finding feasibility.

2.4 Selection of Mixed-Integer Linear Programming

The selection of MILP as the optimisation framework is driven by the specific layout and regulations of the LKR Site. While various modelling techniques

offer individual advantages, MILP is well suited to translate Oatly's physical production environment into a deterministic mathematical structure that can identify a global optimum, not only a feasible solution.

2.4.1 MILP to Handle Discrete Choices

One of the primary motivations for using MILP to solve Oatly's scheduling is the requirement to model discrete decisions. Production at the LKR Site is not a purely continuous flow, instead, it involves specific assignments of production orders to specific filling lines and the need to determine the exact chronological order of each product. In mathematical terms, these represent binary variables that standard linear programming has a hard time accommodating. MILP allows the model to "choose" which line is best suited for a specific SKU while at the same time accounting for the continuous variables of processing time and flow rates.

2.4.2 Hard Constraints

Oatly's production floor is ruled by physical constraints that cannot be simplified without the model losing practical validity. The hardwired piping between specific tanks and machines, along with the strict 36/42 hour mandatory hygiene intervals, acts as hard constraints in the system. Here, MILP provides the necessary tools to treat these rules as hard constraints that the solver must respect no matter what. Unlike rule-based models that may struggle to balance overlapping constraints, MILP can systematically explore the feasible region of solutions to ensure every suggested schedule is technically doable without violating these hard constraints.

2.4.3 Sustainability and Capacity Goals

Since the facility manages a high variety of products with sequence-dependent cleaning requirements, identifying the most efficient path is a combinatorial problem of great scale. This is another reason to choose MILP since it has the capabilities needed to handle $n!$ complexity to find the sequence that minimises non-productive downtime. For Oatly, every transition between two products that can be made without a CIP reduces water and chemical usage, making MILP not only a tool for increased capacity and profit but also for achieving the company's sustainability targets.

2.4.4 Institutionalising Tacit Planning Knowledge

Finally, choosing MILP is motivated by the need to mitigate organisational risk. Currently, the LKR Site relies heavily on the tacit knowledge and intuition of one individual planner. While this still works, this experience-based approach is vulnerable to cognitive load and the loss of expertise. By creating a MILP model with these rules embedded, Oatly can institutionalise its planning knowledge into a data-driven framework. This ensures that the planning process remains stable and consistent, functioning as a decision support tool that is less dependent on human interaction.

2.5 Motivation of Other Choices

Other large decisions include which specific coding language and platform, as well as time representation in the script, should be used. This section describes the rationale behind choosing Python as the primary coding language and the decision behind choosing continuous rather than discrete time variables.

2.5.1 Software Environment: Python, VSCode, and PuLP

The development of the MILP model is conducted in Python, a language selected for its accessibility and user-friendly syntax. While spreadsheet-based tools often struggle with the high-order combinatorial problems inherent in industrial scheduling, Python provides the scalability required to process complex datasets efficiently. The open-source nature of the language and its widespread adoption further motivate the choice of Python (Klotz & Newman, 2013).

Visual Studio Code (VSCode) serves as the Integrated Development Environment (IDE) for the project. It is selected over other tools because it offers better file management and accessibility, which are essential for maintaining structure and replicability. VSCode is a free and open-source IDE which provides a professional workspace that supports extensive debugging and testing without the need of a licence, furthering the choice of VSCode.

To bridge the gap between Python code and mathematical optimisation, the PuLP library is used. This library serves as a high-level modelling interface that translates programmed logic into a structured format compatible with external solvers. PuLP was selected over other modelling languages because its syntax is intuitive and integrates seamlessly with the Python

environment, allowing mathematical rules to be easily verified and replicated without the need for specialised, high-cost software (Mitchell et al., 2011).

While PuLP is used to construct the model's constraints and objectives, the actual optimisation is performed by the COIN-OR Branch and Cut (CBC) solver. In this implementation, CBC serves as the default solver engine and is bundled directly within the PuLP installation, ensuring high portability and ease of setup. When the optimisation is triggered, PuLP communicates the problem to the integrated CBC executable, which then searches the solution space to identify the mathematically optimal production sequence. Utilising CBC, rather than a commercial alternative, provides a robust, open-source solution that maintains the project's full functionality and replicability in both academic and industrial settings without the burden of proprietary licensing fees.

2.5.2 Continuous Time Variable

Another important choice in the model's design is choosing between a continuous and a discrete way to represent time. A discrete way to model time divides the scheduling into fixed buckets of time, whereas a continuous one allows events to occur at any time along a linear timeline (Hillier & Lieberman, 2010). This is important for the LKR Site, where technical parameters such as flow rates require exact timing to prevent starvation of the filling lines. By using continuous variables for the start and finish times, the model adheres to the mechanical actuality of the factory floor, making the production sequence both technically feasible and mathematically precise.

2.6 Data Collection

In contrast to theoretical textbook examples, practical problems encountered by OR teams are often initially described in a vague and imprecise manner (Hillier & Lieberman, 2010). Therefore, the first step in formulating an accurate model is to study the relevant production system and develop a well-defined statement of the problem, including appropriate objectives and physical constraints.

As OR teams typically spend a large amount of time gathering relevant data, this phase is critical for gaining an accurate understanding of the systemic bottlenecks and providing the needed input for the mathematical model (Hillier & Lieberman, 2010). Data required for optimisation models is often not available in the exact format needed for direct use. As a result, data collection is commonly structured into two complementary streams:

qualitative data used to describe human planning logic and quantitative data used to define the system's technical parameters.

2.6.1 Primary Data: Semi-structured Interviews

A large portion of the initial information gathered in practical OR studies consist of "soft" data, which often includes rough estimates based on educated guesses from key personnel (Hillier & Lieberman, 2010). Tacit planning knowledge can be captured through semi-structured interviews with production planners. Such interviews are used to identify undocumented or 'silent' production rules that govern manual planning processes, including experience-based strategies for handling late order changes and for balancing UHT flow rates to avoid aseptic tank starvation.

2.6.2 Secondary Data: Oatly Quantitative Data

While primary data provides the operational context, the core parameters of the model must be derived from secondary data. With the widespread use of enterprise databases, OR teams frequently find that their biggest challenge is not a lack of information, but rather an substantial amount of data (Hillier & Lieberman, 2010). To construct the model's deterministic baseline, technical parameters need to be extracted, filtered, and cleaned from production logs and operational matrices. Additionally, Hillier & Lieberman (2010) describe that historical production records and finalised order lists should be analysed in order to establish typical order quantities, processing rates, and hardwired tank-to-filler routing constraints.

2.7 Research Quality and Validation

To enable a future transition from a theoretical model to a practical planning tool, a thorough assessment of research quality is necessary. This process involves balancing the exponential complexity of the solver with the need for precise results, while simultaneously testing the model's logic against documented historical outcomes. By addressing these factors alongside the criteria for reliability and replicability, the study ensures that the findings are both stable and transparent.

2.7.1 Computational Performance

To ensure the reliability of the results generated by the mathematical model, a structured method is required to determine the duration and scope of the model. According to Hillier and Lieberman (2010), solving MILP problems

is notably more complex than standard linear programming. This complexity arises because the removal of non-integer feasible solutions eliminates the guarantee of finding an optimal corner-point feasible (CPF) solution, which is the foundation of the simplex method's efficiency (Hillier & Lieberman, 2010). In MILP models, the computational difficulty is primarily driven by the number of integer variables rather than the total number of variables or constraints, and the difficulty of finding an optimal solution grows exponentially as the number of integer variables increases (Hillier & Lieberman, 2010).

The method for selecting the computation time balances theoretical solution quality with practical computational limits. By running the model with increasing computation time, the resulting solution efficiency is extracted and plotted against runtime, as shown in Figure 7. This convergence analysis, evaluated alongside the hardware constraints of batch-processing historical data, is then used to determine the final applied time limit. In MILP models, the computational difficulty of finding an optimal solution grows exponentially. This implies that while the solver often identifies good schedules early in the process, yielding a small additional improvement requires a disproportionately large increase in computation time, leading to a point of diminishing returns (Hillier & Lieberman, 2010).

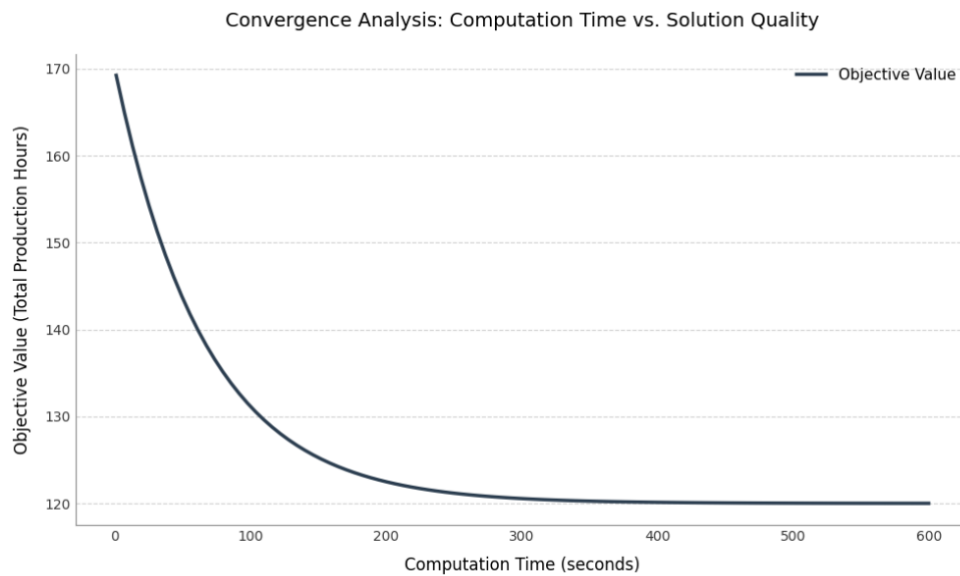


Figure 7, Example of diminishing returns

To identify an appropriate runtime for the MILP model, an experimental procedure is employed where solution quality is evaluated against incremental increases in computation time. This is performed to determine the specific point at which the rate of improvement diminishes rapidly. In MILP solving, early feasible solutions (incumbents) are typically found quickly, but finding marginally better solutions or proving absolute optimality requires exponentially more time (Hillier & Lieberman, 2010). To operationalise this, a convergence analysis is conducted by plotting the objective value (total production hours for a specific order) on the y-axis against the model runtime (computation time) on the x-axis.

2.7.2 Verification on Historical Data Sets

Developing a large mathematical model inevitably involves initial flaws, such as missing parameters or relevant factors, which necessitate a thorough testing process before the model can be used reliably (Hillier & Lieberman, 2010). This process of testing and refining the model to increase its validity is known as model validation (Hillier & Lieberman, 2010). To systematically validate the optimisation model developed in this study, a retrospective test will be conducted. According to Hillier and Lieberman (2010), a retrospective test involves using historical data to reconstruct past scenarios to determine how well the model would have performed had it been deployed at that time.

To operationalise this validation method, the study will utilise historical weekly production data from Oatly during 2025. However, before evaluating the performance, a deliberate data filtering process will be applied. The reliability of the comparative analysis depends heavily on the quality of the underlying data. Therefore, weeks containing missing records, unrecorded timestamps, severe hardware restrictions such as prolonged maintenance, trial runs, or exceptionally low production volumes will be excluded from the dataset. Filtering out these irregularities ensures that the validation reflects normal factory conditions rather than operational anomalies. Including such outliers would provide a misleading baseline, as they do not represent the complex scheduling problems the model is designed to solve. Only the weeks that provide a representative production environment will be used for the final analysis. For each of the weeks, the historical order data will be fed into the mathematical model to generate a hypothetical production schedule. The time required to fulfil the orders according to the model's optimised schedule will then be recorded.

Subsequently, these hypothetical performances will be directly compared to the historical records from those specific weeks. In this context, the model's

output will be benchmarked against the original schedules manually created by the production planner at the start of each week. This approach ensures a fair comparison by evaluating both the manual and optimised plans under the same deterministic conditions, using standard machine speeds, volumes, and fixed cleaning or changeover times.

The primary metric for this comparison is the makespan, the total time required to complete all scheduled batches on machines E2, E3, and E4. By calculating the difference in makespan between the model's optimised sequence and the manual plan, the study will determine whether the mathematical model yields an operational improvement. This method isolates the impact of the model's sequencing logic, showing how much time can be saved simply by organising the production flow more efficiently under standard operating parameters.

Furthermore, this retrospective comparison will serve as a diagnostic tool. If the model generates solutions that are notably different from the manual plans or theoretically impossible to execute, it will indicate areas where the model has shortcomings, such as missing real-world constraints or incorrect parameter estimations, which Hillier & Lieberman (2010) describe as a suitable use of prior data. These insights will be used to make necessary modifications to the model. Finally, while retrospective testing provides a robust framework for validation, Hillier and Lieberman (2010) note that it relies on the assumption that past data is representative of the future, which is not always the case. This inherent limitation will be considered when evaluating the model's long-term applicability.

2.7.3 Reliability and Replicability

According to Bell et al. (2022), reliability refers to the consistency and stability of a study's results, while replicability concerns the degree to which the research procedures are documented, allowing other researchers to reproduce the study. Because this research is centred around mathematical modelling and MILP, these concepts are primarily applied to data integrity, the deterministic nature of the model, and the computational environment rather than traditional measurement scales.

The reliability of the results in this study is considered high due to the nature of the mathematical model. In an optimisation model, providing the exact same parameters, input data, and solver time limits will consistently generate the exact same output (Bell et al., 2022). The historical input data for 2025 is sourced directly from Oatly's internal systems. The order files are provided

directly by the supply department, and the manual reference schedules are retrieved unaltered from the production planners' cloud-based system. Because these documents formed the basis of actual factory operations, the risk of clear, unidentified human data entry errors is minimal; any such errors would have caused immediate production disruptions and subsequently been corrected. Furthermore, to ensure a fair and consistent evaluation, the model's output is compared against the initially planned production schedules rather than the final actual outcomes, which may contain ad-hoc emergency adjustments.

The model's fixed parameters, including equipment speeds and the sequence-dependent cleaning matrix, are based on systematic estimates provided by experienced production planners at Oatly. However, as discussed in Section 2.7.1, the stability of the model's output is influenced by the applied computational time limit. Terminating the optimisation solver before a proven optimal solution is found means that the final solution quality depends on the solver's progress within that specific timeframe.

A primary barrier to complete replicability in this study is data confidentiality, which is a common challenge in business research (Bell et al., 2022). The historical production schedules and order data from Oatly for the year 2025 constitute corporate trade secrets and will not be published in the appendices. Therefore, the specific Python source code developed for data processing will not be shared. Consequently, an independent researcher cannot execute the model on the exact same dataset.

To mitigate this lack of data transparency and ensure a high degree of academic replicability, the data structure and the complete mathematical model are explicitly detailed within the thesis. All decision variables, objective functions, and constraints are presented in standard mathematical notation, enabling another researcher to construct an equivalent model in any programming language. For transparency regarding the software environment, the model in this study is developed in Python using Visual Studio Code, and the optimisation is executed using the open-source PuLP solver.

Finally, it must be noted that the replicability of the exact results under a strict time limit is hardware dependent. Because MILP problems are computationally intensive and scale exponentially in complexity (Hillier & Lieberman, 2010), the number of nodes a solver can explore within a given timeframe relies heavily on the computer's processor and Random Access Memory (RAM). Replicating the study on hardware with lower processing

capabilities using the exact same time limit could result in a noticeably inferior solution compared to a run on a more powerful machine.

2.7.4 Results Validation and Iterative Feedback

When formulating a Mixed-Integer Linear Programming framework to minimise the total production makespan, operations research practitioners often rely heavily on deterministic, quantitative parameters. These parameters include machine throughput rates, tank capacities, and standardised changeover matrices. However, a model built exclusively on these hard mathematical constraints frequently produces solutions that, while mathematically sound, prove practically unfeasible on the factory floor (Merten et al., 2021). The discrepancy between a mathematically calculated minimum makespan and a practically executable production schedule highlights a critical gap in operations research: the absence of qualitative, tacit domain knowledge.

To close this gap and ensure that the mathematical results are genuinely feasible, the research integrates both qualitative and quantitative approaches to provide a comprehensive understanding of the phenomena. In the context of this study, validation is defined as the process of confirming that the mathematical model accurately represents the physical, logistical, and operational reality of the actual production system it intends to simulate. Verification ensures that the mathematical equations function without computational errors, while validation ensures that the output is actually acceptable to the operators and supply chain coordinators who manage the facility.

The validation of the results will be conducted through a continuous, iterative feedback loop. This process is used to uncover the tacit, qualitative rules that govern the planner's decision-making but are entirely absent from the facility's technical documentation. The planned methodology begins with the formulation of a relatively simple baseline objective function. This initial function generates schedules based on the hard constraints provided by Oatly, relying primarily on the quantitative information extracted from the facility's systems and a few preliminary soft constraints identified during initial semi-structured interviews.

Once this baseline model generates initial schedules, these outputs will be continuously presented to the manual planner for review. This review process allows the domain expert to identify gaps, inconsistencies, and opportunities for refinement. By analysing where the mathematical schedule deviates from

what the planner considers logical, safe, or administratively manageable, one can identify missing operational parameters. This creates a systematic back-and-forth mechanism: the mathematical model generates a schedule, the planner critiques its real-world viability and one can then translate this critique into mathematical soft constraints within the objective function, and the model generates a new schedule. This iterative cycle will continue until the developers consistently present schedules that satisfy the planner's criteria for practical usability.

The most critical aspect of this validation methodology is that the validation process directly affects and shapes the objective function itself. The core challenge lies in converting qualitative human preferences into quantitative mathematical variables. During the validation process, it is expected that the manual planner will express operational preferences that do not correspond directly to measurable time savings, but rather to risk mitigation, downstream logistics, or workload balancing.

To illustrate how validation affects the objective function, consider a hypothetical scenario regarding product sequencing. The manual planner might possess qualitative knowledge dictating a strict preference for producing products in a specific order. The mathematical model, evaluating the schedule purely on quantitative processing times and sequence-dependent cleaning durations derived from the changeover matrix, might calculate that transitioning directly from product 1 to product 2 incurs the exact same setup time as transitioning to product 3.

From a purely mathematical perspective, the solver sees no difference between these two sequences; both yield identical time durations. However, for unquantifiable reasons, perhaps related to minimising any trace risk of cross-contamination, aligning with raw material delivery schedules, or matching the physical staging layout of the warehouse, going from product 1 to 2 might be preferable over product 1 to 3. Because the mathematical solver is indifferent to qualitative categories, the validation process must identify these discrepancies and correct the model's indifference.

To account for this, structural penalties must be added into the objective function that takes this specific qualitative preference into consideration. Instead of merely minimising total production time, the objective function is expanded to minimise time plus the sum of all penalty violations. By assigning a mathematical weight to the preferred sequence, the model is forced to recognise the qualitative difference between two sequences that appear temporally identical.

These validation-driven modifications are implemented using soft constraints. Unlike hard constraints, which define physical impossibilities and instantly render a solution infeasible if violated, soft constraints allow the solver to break a rule if doing so provides a substantial time benefit, but penalise the model heavily for doing so. When a qualitative rule is identified during the validation process, it is formalised using binary decision variables multiplied by penalty coefficients. This approach ensures that the model remains flexible, encouraging the mathematical solver to follow the human planner's tacit rules whenever possible, but permitting deviations if strict adherence would cause a substantial delay in production or trigger a mandatory biological cleaning cycle. The iterative validation methodology ensures that these penalty weights are continuously tuned, scaling them up or down until the mathematical model precisely balances theoretical time efficiency with practical factory logic. By relying on this back-and-forth expert feedback, the study helps ensure that the final schedules are not only technically possible on a computer screen but genuinely possible in the real world.

2.8 Ethical Considerations in Industrial Research

Industrial research involving the automation of decision-making processes introduces ethical responsibilities that extend beyond data management to include the impact on the workforce and organisational transparency. In the context of OR, Hillier and Lieberman (2010) emphasise that an OR team has a responsibility to remain objective and act as a servant to the decision-makers, while considering the "human element" of the system being studied.

A clear ethical consideration in this study is the potential shift from experience-based planning to an automated model if the concept of this study were to be implemented. While mathematical optimisation can minimise organisational risk by securing planning knowledge in code, protecting the company against "knowledge concentration" if a key planner were to leave or fall ill, it also creates a delicate dynamic regarding job security. Hillier and Lieberman (2010) note that implementation often fails if the human users of a system feel threatened or excluded. Consequently, this study treats the model not as a complete replacement for the human planner, but as a risk-minimising support tool designed to reduce cognitive load and stress. Ethically, the transition must be managed transparently to avoid alienating the workforce whose expertise formed the basis of the model's parameters.

The use of an automated MILP model raises questions regarding algorithmic responsibility. If a generated schedule leads to operational errors, the ethical

burden lies in the transparency of the model's logic. This study addresses this by ensuring that the model's constraints and objective functions are clearly documented, allowing for human oversight and intervention.

In accordance with ethical standards in business research (Bell et al., 2022), the handling of sensitive industrial data has been conducted with strict confidentiality. To protect the case company's competitive advantage, exact production volumes and specific product identifiers have been anonymised or removed from the published thesis. No personal information, such as employee names or individual performance metrics, has been processed, ensuring that the study focuses on systemic efficiency rather than individual monitoring.

Finally, there is an ethical dimension to the model's impact on sustainability. Although the primary objective of the MILP model is to minimise total production time, the reduction of cleaning cycles as a means to achieve this efficiency has a direct positive impact on environmental performance. By optimising the sequence of production, the model indirectly reduces the consumption of water and chemicals for processes. This aligns the study with broader corporate social responsibility goals, demonstrating that operational efficiency and environmental ethics can be mutually reinforcing.

3. Theory

The effectiveness of a mathematical model depends on how accurately it reflects the real-world system it intends to optimise. This chapter outlines the theoretical concepts used to translate Oatly's production planning into a solvable mathematical format. By reviewing scheduling theory, Mixed-Integer programming, and the specific constraints of the oat milk industry, the following sections provide the building blocks for the model's design and the subsequent evaluation of its performance.

3.1 Fundamentals of Production Planning and Control

Production Planning and Control (PPC) is defined as the administrative and operational framework responsible for managing the flow of materials, the utilisation of labour, and the allocation of equipment to satisfy market demand efficiently. According to Chapman (2006), the PPC system serves as the "information-processing brain" of a manufacturing entity, coordinating physical activities to ensure that products are delivered in the correct quantities at the correct times.

A central theoretical pillar of PPC is the management of built-in trade-offs. Vollmann et al. (2005) describe the Manufacturing Planning and Control (MPC) system as a mechanism to balance competing objectives: customer service (timely delivery), inventory investment (working capital), and operating efficiency (throughput). These objectives are often in conflict, for example, maximising plant efficiency through long production runs may increase inventory levels, while minimising inventory may jeopardise customer service levels. The theoretical role of a planning system is to provide the logical structure required to navigate these trade-offs, ensuring that operational decisions are synced with the broader strategic objectives of the firm (Silver et al., 1998).

3.1.1 Hierarchical Planning Horizons

The planning and control process is theoretically structured as a hierarchical system, transitioning from high-level aggregate planning to granular, short-term execution. This hierarchy, originally formalised by Anthony (1965), segments decision-making into strategic, tactical, and operational horizons. The primary purpose of this segmentation is to manage complexity by filtering information and narrowing the decision scope as the time horizon shortens.

At the highest level, strategic planning spans a long-term horizon typically spanning several years. This phase is characterised by its focus on broad business objectives and high-level capacity requirements, such as major facility investments and market positioning strategies. Because of the long lead time involved in changing physical infrastructure, decisions made at this level define the boundary conditions for all following planning activities.

As the horizon moves into medium length, tactical planning becomes the central focus, primarily through Sales and Operations Planning (S&OP) and the development of a Master Production Schedule (MPS). At this stage, the objective is to balance demand and supply at a “bundled”, aggregate level. As Chapman (2006) emphasises, planning at this level focuses on total volume rather than the specific sequences of individual Stock Keeping Units (SKUs). This ensures that total resources, specifically labour capacity and raw material supply, are sufficient to support the expected production load before detailed scheduling begins.

Finally, operational planning represents the short-term back-end of the MPC system. This level involves the breakdown of the master schedule into specific execution instructions. While the strategic and tactical plans determine what and how much to produce based on volume requirements, the operational level is concerned with the precise timing and sequence of activities. According to Pinedo (2016), this phase is where the theoretical focus shifts to the exact when and in what sequence jobs are processed, ensuring that the physical constraints of the production environment are respected while fulfilling the requirements passed down from the higher levels of the hierarchy.

The integrity of this hierarchy relies on a continuous feedback loop, while instructions flow downwards from strategic levels, real-time data regarding capacity constraints and execution status must flow upwards to ensure that higher-level plans remain feasible (Vollmann et al., 2005).

3.1.2 Short-term Scheduling and Sequencing

Short-term scheduling and sequencing represent the execution phase of the MPC framework, where the focus shifts from volume planning to the precise management of individual machine schedules and the specific timing of each production step (work centre priorities). Chapman (2006) distinguishes between scheduling and sequencing as two distinct but related activities. Scheduling is defined as the process of establishing specific start and finish times for individual jobs on a certain resource, ensuring that the work is

executed within the constraints of available capacity and meeting the requirements of the master plan. Sequencing, however, involves determining the specific chronological order in which a collection of jobs waiting at a resource should be processed.

In complex manufacturing environments, the efficiency of sequencing is heavily influenced by sequence-dependent setup times (SDST). This occurs when the time required to transition a machine from one product to another depends specifically on the characteristics of the former and following items. Pinedo (2016) notes that SDST often transforms a standard scheduling challenge into a combinatorial optimisation challenge equivalent to the TSP, where the objective is to find a sequence that minimises the total “distance” (time in this case) between jobs.

Improper sequencing in SDST environments leads to an excessive load on the resource, where the machine is busy with setup activities instead of production, thereby using capacity that could have been used for throughput. Theoretically, the goal of optimisation at this level is to navigate these dependencies using priority rules or mathematical algorithms to minimise non-productive time, making sure that the plant operates at its maximum capacity while meeting the delivery dates established by the higher-level schedules (Silver et al., 1998).

3.2 Mathematical Optimisation and Modelling

The development of linear programming has been historically categorised as a major scientific advance occurring during the mid-20th century. Since approximately 1950, the impact of this mathematical discipline has been highly influential, establishing itself as a standard tool for decision-making across various industrialised nations. Within modern industrial management, linear programming has facilitated substantial financial savings, and a major proportion of all computer-based calculations is currently devoted to executing these linear models. However, a key limitation of standard linear programming is the assumption of divisibility, which requires that non-integer values be permissible for decision variables.

In many practical problems, such as the assignment of people, machines, and vehicles, decision variables make sense only if they have whole integer values. When the linear programming model is given the one additional restriction that variables must be integers, it becomes an integer programming (IP) problem. If only a portion of the variables requires this indivisibility, the model is referred to as a MILP problem. Together, the academic and

professional prominence of both linear and integer programming is shown by extensive literature, demonstrating how these connected disciplines continually provide the framework to solve complex, real-world problems (Hillier & Lieberman, 2010).

3.2.1 The Scope of Linear Programming Problems

According to Hillier & Lieberman (2010), the most dominant application of linear programming involves the general problem of allocating limited resources among competing activities in the best possible, or optimal, way. More precisely, this allocation problem requires the selection of specific activity levels that must compete for limited resources essential for their performance. The selection of these activity levels subsequently dictates the quantity of each resource that will be consumed by each respective activity.

The diversity of situations where this resource allocation framework applies is vast, ranging from the distribution of production facilities among various products to the allocation of national resources towards domestic needs. Furthermore, linear programming is utilised in portfolio selection, the determination of shipping patterns, agricultural planning, and even the design of radiation therapy (Hillier & Lieberman, 2010). Despite this diversity, the common ingredient in every situation is the requirement to allocate resources to activities by choosing specific activity levels.

Linear programming utilises a mathematical model to describe the problem of concern. The term linear signifies that all mathematical functions in this model must be linear. The term programming in this context does not refer to computer programming but is used as a synonym for planning. Thus, Hillier & Lieberman (2010) describe linear programming as the planning of activities to obtain an optimal result that reaches a specified goal in the best way, in the mathematical model, considering all feasible alternatives.

3.2.2 Structural Components of the Linear Programming Model

In the formulation of a linear programming problem, the decisions to be made are represented by decision variables. Hillier and Lieberman (2010) describe a prototype example shown below involving production planning, where the objective is to determine the number of batches of two products to produce per week to maximise total profit. This is shown in order to better understand the concept. The mathematical notation for these decision variables is defined in Table 2.

Table 2, Variables of LP Problem

VARIABLE	DEFINITION
x_1	Number of batches of product 1 produced per week
x_2	Number of batches of product 2 produced per week
Z	Total profit per week

As established by Hillier & Lieberman (2010), these variables x_1 and x_2 are the decision variables for the model. The relationship between production and profit is expressed through the objective function, which aims to maximise the total profit Z . Based on the production data provided in the prototype example, an objective function can be formulated as (1).

$$Z = 3x_1 + 5x_2 \quad (1)$$

In (1), the coefficients 3 and 5 represent the profit contribution per batch of product 1 and product 2, respectively. The goal is to choose values x_1 and x_2 that maximise this sum while adhering to specific restrictions imposed by limited production capacities. There are also restrictions imposed by the production capacities available in three distinct plants. Each batch of product produced consumes a specific amount of time in these plants, and the total time consumed cannot exceed the total time available.

The first restriction is derived from the capacity of Plant 1. The production data indicate that each batch of product 1 produced per week requires 1 hour of production time in Plant 1. Since only 4 hours per week are available for producing product 1 in this plant, the restriction is expressed mathematically as (2).

$$x_1 \leq 4 \quad (2)$$

This inequality ensures that the level of activity for product 1 does not exceed the scarce resource available in Plant 1. Similarly, Plant 2 imposes a restriction based on its weekly capacity. According to the technical

parameters, product 2 requires a specific amount of time in Plant 2, where the available capacity is 12 hours per week. This restriction is expressed as (3).

$$2x_2 \leq 12 \quad (3)$$

This constraint limits the production rate of product 2 to ensure it remains within the feasible limits of Plant 2's operational capacity. The most complex restriction in the prototype example arises from Plant 3, where both products compete for the same resource. The number of hours of production time used in Plant 3 is determined by the production rates of both products. Specifically, each unit of x_1 uses 3 hours, and each unit of x_2 uses 2 hours. The total available time in Plant 3 is 18 hours per week. The mathematical statement for the Plant 3 restriction is shown in (4), which represents the necessity of allocating the limited time in Plant 3 among the competing activities of producing product 1 and product 2.

$$3x_1 + 2x_2 \leq 18 \quad (4)$$

In addition to the capacity-based restrictions, the mathematical logic of the model requires that the decision variables remain non-negative. Since production rates cannot be negative in a physical sense, it is necessary to restrict the values of x_1 and x_2 to be greater than or equal to zero. These logical constraints are represented as (5) and (6).

$$x_1 \geq 0 \quad (5)$$

$$x_2 \geq 0 \quad (6)$$

To summarise the prototype example in the mathematical language of linear programming, the complete model is structured as: Maximise Z , subject to the capacity restrictions in (2), (3), and (4), and the non-negativity constraints in (5) and (6). The configuration of these coefficients in the linear programming model duplicates the technical information regarding production time per batch and total available hours. This structure allows for the systematic evaluation of all feasible alternatives to identify the optimal result that best reaches the specified goal. The use of such mathematical models has become a standard tool in OR, providing a scientific approach to decision-making that ensures resources are allocated efficiently to achieve the best possible outcome.

3.2.3 Theoretical Assumptions and Mathematical Principles of Linear Programming

The effectiveness of linear programming as a tool for optimisation is based on the adherence of the mathematical model to the requirements of linearity. This implies that the relationship between the activity levels and the consumption of resources must be characterised by linear functions. Hillier & Lieberman (2010) highlight that the development of these methods has advanced scientific computation, enabling the solution of large-scale models that arise in practical industrial and societal applications.

The interchangeability of terms such as linear program and LP model reflects the deep integration of these concepts in the field of OR. Whether the goal is to maximise profit, minimise costs, or allocate national resources, the underlying principles of linear programming remain consistent: the selection of activity levels to optimise a linear objective function while respecting a set of linear constraints. The model in this report serves as a miniature representation of the broader potential of linear programming to solve complex planning problems.

3.2.4 Mixed-Integer Linear Programming

In many practical industrial applications, the assumption of allowing activities to be divided is not applicable because certain decision variables only make sense as whole units, such as the assignment of specific machines or personnel. MILP is the mathematical framework used when a subset of the variables is restricted to integer values, while others remain continuous. Hillier and Lieberman (2010) give the following example in order to better understand the concept of MILP. The general model for a MILP problem involves maximising a linear objective function by summing the contributions of all n decision variables, as shown in (7).

$$\text{Maximize } Z = \sum_{j=1}^n c_j x_j \quad (7)$$

This objective function represents the goal of finding the best possible "plan" (e.g., maximum profit) across all activities. It is subject to m functional constraints that represent resource limitations, formulated as (8).

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \text{ for } i = 1, 2, \dots, m \quad (8)$$

are infeasible, or if they result in an integer solution for all integer-restricted variables.

4. Modelling

The following chapter describes the development of the mathematical model used to optimise weekly production scheduling at Oatly's Landskrona facility. It covers the full model formulation, including parameters, decision variables, objective function and constraints, as well as the preprocessing steps and setup explanation.

4.1 Mathematical Model Development

The mathematical model developed in this thesis is a MILP formulated to optimise the weekly production schedule at Oatly's Landskrona facility. The model determines the assignment of production orders to filling lines, the sequence in which they are processed, and the timing of mandatory cleaning cycles, with the primary objective of minimising the production time across the three Edge filling lines.

The formulation is structured around a set of binary and continuous decision variables governed by a series of constraints that enforce the physical, biological, and operational rules of the LKR Site. These include machine capacity limits, hygiene cycling requirements, sequence-dependent setup times, and product-specific routing restrictions. A weighted objective function balances the primary goal of minimising makespan against several secondary operational preferences, such as avoiding unnecessary cleaning cycles and maintaining logical order sequences.

The following sections define the sets and indices, parameters, decision variables, objective function, and constraints that together constitute the complete mathematical formulation.

4.1.1 Model Parameters

The parameters of the model represent the "fixed truths" of the Landskrona site. These are the physical characteristics of the hardware, such as the speed of a filling machine or the volume of a storage tank, that do not change regardless of the production sequence. By defining these accurately, one ensures that the model respects the laws of physics and the mechanical limits of the equipment.

1. Hygiene limits

Each filling line l at the Landskrona facility is subject to a maximum continuous runtime, denoted H_l , before a mandatory CIP cycle must be performed. This limit exists due to biological and food safety regulations, as microorganisms can accumulate in the filling equipment over time, meaning production cannot run indefinitely without interruption for cleaning. The hygiene limit therefore represents a hard physical boundary on how long a line may operate before it must be taken offline and sanitised. The three Edge lines do not share an identical limit, E3 has a stricter threshold of 36 hours compared to 42 hours for E2 and E4 as shown in (11). This difference reflects the specific equipment characteristics of each line.

$$H_l = \begin{cases} 42 & \text{if } l \in \{E2, E4\} \\ 36 & \text{if } l \in \{E3\} \end{cases}, \quad \forall l \in L \quad (11)$$

Where L is the set of all Edge lines E2, E3 and E4. There are additional rules tied to the cleaning of the equipment, these are introduced later in this chapter.

2. E-Lines Capacity

As mentioned, the Landskrona facility operates three main downstream filling lines, collectively forming the set L . Each line is characterised by a fixed processing rate R_l , representing the maximum volumetric throughput in litres per hour. These rates are determined by the mechanical capacity of each filling machine and remain constant regardless of the product being produced as seen in (12). The capacities are calculated based on the maximum litre throughput during one machine cycle; $250,000/42 = 5952.38$ for E2 and E4 and $210,000/36 = 5833.33$ for E3.

$$R_l = \begin{cases} 5952.38 & \text{if } l \in \{E2, E4\} \\ 5833.33 & \text{if } l \in \{E3\} \end{cases}, \quad \forall l \in L \quad (12)$$

3. Orders

Each production order is represented in the model as a job $i \in J$, where J is the set of all jobs (indexed by i and j). Each production order is characterised by four attributes that together define where it can run, how long it will take, and what cleaning is required before and after it. V_i denotes the total volume to be filled for job i (13), which is the primary driver of processing duration, as a larger volume on a slower line results in a longer runtime, directly affecting the overall schedule length.

$$V_i \in \mathbb{R}_+, \quad \forall i \in J \quad (13)$$

The set of allowed lines $A_i \subseteq L$ defines which filling lines job i is physically permitted to run on per (14). This is determined by the hardwired piping infrastructure of the facility, ensuring a job cannot be assigned to a line outside this set regardless of scheduling considerations.

$$A_i \subseteq L, \quad \forall i \in J \quad (14)$$

The base product ID, denoted b_i , identifies the underlying recipe or product formulation of job i and is defined in (15). This attribute is used to look up the required cleaning time between consecutive jobs on the same line via the changeover matrix. Two jobs with different base product IDs may require anywhere from zero to four hours of cleaning between them depending on the nature of the transition.

$$b_i \in B, \quad \forall i \in J \quad (15)$$

Where B is the set of all base product IDs present in the schedule.

Finally, in (16) the packaging format ID, denoted f_i , identifies the physical packaging type of job i . If two consecutive jobs share the same base product ID but differ in packaging format, a one-hour changeover is required between them.

$$f_i \in F, \quad \forall i \in J \quad (16)$$

Where F is the set of all packaging format IDs present in the schedule.

4. Setup and cleaning times

The setup time between two consecutive jobs captures the time required to prepare a filling line for a new product after the previous job has finished. This duration, denoted σ_{ij} , includes both the physical cleaning of the equipment and any setup required if the packaging type changes. The setup time is not a fixed constant but depends on the specific pair of products being transitioned between and is derived from two primary rules.

The first rule governs format changeovers: if two consecutive jobs share the same base product ID but differ in their packaging format, a one-hour changeover is required to change packaging for the filling machines. If both the base product and packaging format are identical, no changeover time is

needed. The second rule governs product transitions and is based on the changeover matrix. This facility-specific table assigns a requirement to every pair of base product IDs, translating directly into hours of downtime. For example, a green transition requires no cleaning, while a brown transition requires the full four hours. When a product transition occurs, the setup time is defined as the maximum of the format changeover time and the cleaning matrix time, ensuring both operational requirements are satisfied. Both of these rules are defined in (17).

$$\sigma_{ij} = \begin{cases} 0 & \text{if } b_i = b_j \text{ and } f_i = f_j \\ 1 & \text{if } b_i = b_j \text{ and } f_i \neq f_j, \\ \max(\text{matrix}(b_i, b_j), 1) & \text{if } b_i \neq b_j \end{cases} \quad (17)$$

Where $(\text{matrix}(b_i, b_j) \in \{0, 4\})$ denotes the hours in the cleaning matrix required when transitioning from base product b_i to base product b_j . The reason for taking the maximum is that even if the cleaning matrix returns 0 hours for a particular product transition, at least one hour is still needed in practice to reconfigure the line. So, the model ensures the setup time is never less than one hour whenever the product changes, regardless of what the cleaning matrix says.

5. Gurt sequencing rules

The Gurt product family consists of three oat-based yoghurt variants produced exclusively on line E3. Due to the nature of the cleaning requirements between these variants, the order in which they are produced has a clear impact on the schedule. Transitioning between Gurt variants in the correct order requires only a one-hour label changeover, whereas producing them in the wrong order would trigger a four-hour cleaning cycle. The model therefore enforces a strict production sequence for all Gurt jobs. The three variants are identified by their base product IDs in (18) and assigned a rank r per (19), where a lower rank must always be produced before a higher rank. This gives the required sequence: Natural first, followed by vanilla, and finally strawberry.

$$G = \{500080, 500081, 500082\} \subseteq B \quad (18)$$

Where G denotes the set of Gurt base product IDs.

$$r_i = \begin{cases} 0 & \text{if } b_i = 500080 \text{ (Natural)} \\ 1 & \text{if } b_i = 500081 \text{ (Vanilla)} \\ 2 & \text{if } b_i = 500082 \text{ (Strawberry)} \end{cases}, \quad \forall i \in J : b_i \in G \quad (19)$$

6. Product category order

Beyond the Gurt sequencing rules, the model also captures a broader preferred production order across all product categories. This preference is rooted in the asymmetric nature of the cleaning matrix, transitioning from an organic product to a conventional product typically requires little or no cleaning, whereas the reverse direction often triggers a four-hour cleaning cycle. By producing higher-ranked products before lower-ranked ones, the schedule can systematically avoid costly cleaning cycles.

Each base product ID is assigned a category rank $k_i \in \{0, 1, 2, 3\}$ based on its base product ID, where a lower rank indicates a product that should be produced earlier in the sequence. The four categories reflect the product hierarchy at the Landskrona facility, ordered from the most sensitive to the least sensitive in terms of cleaning requirements. Since all product IDs cannot be shown, a general representation can be seen in (20).

$$k_i = \begin{cases} 0 & \text{KRAV/ORGANIC} \\ 1 & \text{Barista KRAV/ORGANIC} \\ 2 & \text{Conventional} \\ 3 & \text{Barista Conventional} \end{cases}, \quad \forall i \in J : b_i \in K \quad (20)$$

Where K denotes the set of base product IDs for which a category rank is defined. Products outside this set, such as specialty items, are not assigned a rank and are therefore not subject to this ordering preference. It should be noted that unlike the Gurt sequencing rules, this ordering is not enforced as a hard constraint. Instead, it is implemented as a soft penalty in the objective function, which is discussed further in Section 4.1.3.

7. Planning horizon

The planning horizon W defines the maximum duration of the schedule, corresponding to one full week of production. Each filling line is allocated a maximum of 168 hours per week as defined in (21), reflecting the total number of hours available from Monday through Sunday. This parameter serves as a soft upper bound on the completion time of each line, the model penalises schedules that extend beyond this limit but does not forbid it,

allowing the solver to exceed 168 hours in situations where the total order volume cannot feasibly be completed within the week.

$$W = 168 \quad (21)$$

8. Penalty Coefficients

Penalty coefficients are used in the objective function (Section 4.1.3) where they are further explained, and are defined as parameters shown in Table 3.

Table 3, Penalty Coefficients for the Objective Function

SYMBOL	VALUE	DESCRIPTION
ALPHA	6	Penalty per forced CIP inserted.
EPSILON	0.01	A small tie-breaker penalty applied to the start times of all jobs, forcing the solver to schedule work as early in the week as possible.
GAMMA	50	The highest penalty in the model. It ensures that the solver treats the 168-hour week as a "hard-as-possible" limit, only spilling into the next week if the volume is physically impossible to fit.
DELTA	10.0	A heavy penalty to ensure that order blocks stay together on the same machine unless moving them saves a large amount of time.
RHO	5	Product category order violations.
BETA	5	Penalty per incompatible transition chosen.
LAMBDA	1.5	Used to keep products consolidated. It is the factor that encourages the solver to finish a product on one line before starting it on another.

9. Big-M constants

The Big-M method is a technique in Mixed-Integer linear programming used to model conditional or disjunctive constraints, that is, constraints that should only be active under certain conditions. By multiplying a large constant M by a binary variable, a constraint can effectively be switched on or off depending on the value of that variable. When the binary variable equals zero the constraint is active, and when it equals one the large constant makes the constraint trivially satisfied and therefore inactive. Two Big-M constants are used in this model. The first, M_T , is defined in (22) and used in the timing constraints and must be large enough to never interfere with feasible start and completion times. Its value of 200 is chosen to exceed the maximum possible schedule length of one week, ensuring it never becomes binding when it should not be. The second, M_A , defined in (23) is used in the hygiene accumulation constraints and must be large enough to dominate any feasible accumulated hygiene time. Its value of 90 is chosen to comfortably exceed the maximum possible accumulated time, which is bounded by the sum of the largest hygiene limit and the longest possible processing duration.

$$M_T = 200 \text{ hours} \quad (22)$$

$$M_A = 90 \text{ hours} \quad (23)$$

It should be noted that while the Big-M values are technically parameters of the model, they are purely mathematical modelling constructs rather than physical characteristics of the facility. Their values must be chosen carefully, too small and the model becomes infeasible, too large and the linear programming relaxation becomes weak, slowing down the solver.

4.1.2 Decision Variables

The model uses a combination of binary variables to represent discrete scheduling decisions and continuous variables to track time and hygiene accumulation.

Binary Variables

These variables represent "yes/no" decisions made by the solver.

$x_{il} \in \{0, 1\}$: Equals 1 if job i is assigned to line l , otherwise it is 0

$y_{ij} \in \{0, 1\}$: Equals 1 if job i is directly sequenced before job j on the same line, otherwise 0

$F_{ij} \in \{0, 1\}$: Equals 1 if a forced CIP is triggered between job i and job j , 0 otherwise

$d_{k,k+1} \in \{0, 1\}$: Equals 1 if consecutive blocks (maximum number of batches within one hygiene cycle) k and $k + 1$ of the same split order are assigned to different lines, otherwise 0

$m_{bl} \in \{0, 1\}$: Equals 1 if at least one job with base product b is assigned to line l , otherwise 0

Continuous Variables

These variables track durations, timestamps, and accumulation levels.

$S_i \geq 0$: Start time of job i

$C_i \geq 0$: Completion time of job i

$P_i \geq 0$: Processing duration of job i

$A_i \geq 0$: Accumulated hygiene time at the end of job i

$C_{MAX} \geq 0$: Global makespan (the finishing time of the last completed Edge line)

$\hat{C}_l \geq 0$: Individual makespan of Edge line l

$O_l \geq 0$: Overload hours (the amount of time Edge line l exceeds the planning horizon $W = 168$)

4.1.3 Objective Function

The objective function in (24) minimises a weighted sum of eight terms, balancing schedule length against several operational penalties. Each term targets a specific aspect of schedule quality that cannot be captured by makespan alone. To ensure industrial relevance, the specific penalties and their operational boundaries were cultivated through multiple feedback loops with the production planners.

$$\min Z = C_{MAX} + \alpha \sum_{i \in J} \sum_{j \in J} F_{ij} + \varepsilon \sum_{j \in J} S_j + \gamma \sum_{l \in L} O_l + \delta \sum_{k, k+1} d_{k, k+1} + \rho \sum_{l \in L} \sum_{(i, j) \in R} y_{jil} + \beta \sum_{l \in L} \sum_{(i, j) \in I} y_{ijl} + \lambda \sum_{b \in B} \sum_{l \in L} m_{bl} \quad (24)$$

Breakdown of Objective Terms:

Makespan (C_{MAX})

The primary objective is to minimise the global makespan C_{MAX} , representing the point in time at which the last job on any Edge line finishes. Through their respective weights, the other terms guide the solver towards operationally preferable solutions within the space of schedules that achieve a short makespan.

Forced CIP penalty

The coefficient $\alpha = 6$ multiplies the total number of forced CIP events across all job pairs, $\sum F_{ij}$. Each time the solver selects to trigger a forced cleaning cycle between two consecutive jobs, 6 is added to the objective. This discourages the solver from triggering unnecessary cleaning cycles and encourages it to align product transitions with natural hygiene resets instead.

Left-pull gravity

The coefficient $\varepsilon = 0.01$ multiplies the sum of all job start times, $\sum S_j$. Since the coefficient is very small, this term does not meaningfully compete with the other objectives. Its sole purpose is to prevent the solver from producing schedules where jobs are unnecessarily delayed, by penalising late start times, the schedule is pulled as early as possible within the planning horizon.

Overload penalty

The coefficient $\gamma = 50$ multiplies the sum of overload hours across all Edge lines, $\sum O_l$, where O_l represents the number of hours by which line l exceeds the 168 hour planning horizon. Each hour of overload on any Edge line adds 50 to the objective, making it one of the most heavily penalised outcomes in the model. This ensures that schedules extending beyond the planning horizon are only accepted when the total order volume leaves no feasible alternative.

Split-order penalty

The coefficient $\delta = 10$ multiplies the total number of cross-machine block splits, $\sum d_{k,k+1}$, where $d_{k,k+1}$ equals one if consecutive blocks of the same split order are assigned to different lines. Each such split adds 10 to the objective, reflecting the operational preference for keeping all blocks of the same order on the same line.

Product rank penalty

The coefficient $\rho = 5$ multiplies the number of product category order violations, $\sum y_{jil}$. The set R contains all job pairs where $k_i < k_j$, meaning job i should run before job j . Each time the solver sequences a lower-ranked product before a higher-ranked one, 5 is added to the objective, discouraging violations of the preferred product category ordering without forbidding them entirely.

Incompatible transition penalty

The coefficient $\beta = 5$, multiplies the number of incompatible transitions chosen by the solver, $\sum y_{ijl}$. The set I contains all job pairs on shared Edge lines where $\sigma_{ij} \geq 4$, meaning the transition requires a full four-hour cleaning cycle. Each time such a transition is selected, 5 is added to the objective, discouraging the solver from freely choosing incompatible product transitions when compatible alternatives exist. While the four-hour cleaning duration is already indirectly accounted for through the minimisation of the total makespan C_{MAX} , having this penalty serves as an additional guiding mechanism for the solver's search logic.

Base spread penalty

The coefficient $\lambda = 1.5$, multiplies the total number of distinct line activations per base product, $\sum \sum m_{bl}$, where m_{bl} equals one if at least one job with base product b is assigned to line l . Each additional line that a base product is spread across adds 1.5 to the objective. This encourages the solver to consolidate all blocks of the same product onto a single line, consistent with how human planners organise production.

4.1.4 Model Constraints

1. Machine assignment

The machine assignment constraints govern how jobs are allocated to filling lines. Together they ensure that every job is assigned to exactly one line, that physical infrastructure limitations are respected, and that product-specific routing requirements are enforced.

Every job must be assigned to exactly one filling line. This is enforced by requiring that the sum of assignment variables across all lines equals one for each job as in (25).

$$\sum_{l \in A_j} x_{jl} = 1, \quad \forall j \in J \quad (25)$$

The physical piping infrastructure of the facility means that not every product can reach every filling line. Each job j has a set of allowed lines $A_j \subseteq L$ determined by the hardwired connections between upstream processing equipment and downstream filling machines. Assigning a job to a line outside this set is physically impossible and is therefore forbidden per (26).

$$x_{jl} = 0, \quad \forall j \in J, l \notin A_j \quad (26)$$

As discussed in Section 4.1.1, all Gurt variants must be produced on line E3 due to their specific equipment requirements. When E3 is among the allowed lines for a Gurt job, all other allowed lines are excluded as in (27), effectively restricting the job to E3 only.

$$x_{jl} = 0, \quad \forall j \in J: b_j \in G, l \in A_j \setminus \{E3\} \quad (27)$$

2. Processing duration and completion time

The processing duration and completion time constraints translate the physical characteristics of each job and line into the model's time variables. These constraints are definitional in nature, they do not restrict the schedule in a business or regulatory sense, but rather establish the mathematical relationship between volumes, speeds, and time that all other constraints build upon.

The completion time C_i of job i is defined as its start time plus its processing duration. The processing duration is determined by the job's volume V_i and the processing rate R_l of the line to which it is assigned. The assignment variable x_{il} ensures that only the rate of the selected line contributes to the

timeline. The completion time for every job in the schedule is determined by (28).

$$C_i = S_i + \sum_{l \in A_i} x_{il} \cdot V_i / R_l, \quad \forall i \in J \quad (28)$$

3. Makespan

The makespan constraints define and bound the length of the production schedule. The model focuses exclusively on the Edge lines when measuring schedule length, as these are the high-volume bottleneck lines whose throughput determines the overall planning outcome for the week.

The global makespan C_{MAX} in (29) represents the point in time at which the last job on the last finishing Edge line finishes. It is defined as an upper bound on the completion time of every job assigned to the Edge lines, ensuring that C_{MAX} always reflects the true end of the schedule.

$$C_{MAX} \geq C_i, \quad \forall i \in J: A_i \cap L \neq \emptyset \quad (29)$$

In addition to the global makespan, the model tracks the individual completion time of each Edge line through a per-machine makespan variable $\widehat{C}_l$ in (30). The Big-M term ensures that only the completion times of jobs actually assigned to line l contribute to $\widehat{C}_l$.

$$\widehat{C}_l \geq C_j - M_T(1 - x_{jl}), \quad \forall j \in J, l \in L \quad (30)$$

The planning horizon $W = 168$ hours acts as a soft upper bound on the per-machine makespan. Any excess beyond this limit is captured by the overload variable O_l in (31) and penalised in the objective function, discouraging schedules that extend into the following week while still allowing it when the order volume leaves no alternative.

$$O_l \geq \widehat{C}_l - W, \quad \forall l \in L \quad (31)$$

4. Sequencing

The sequencing constraints define the relative ordering of jobs on each filling line. They ensure that if two jobs are assigned to the same line, they are processed one after the other without overlapping, and that the required setup time between them is respected.

If job i precedes job j on a shared line l , the start time of j must respect the completion time of i plus the required setup time σ_{ij} . This is enforced in (32) by using a Big-M formulation, where the constant M_T ensures the constraint is only binding when i precedes j and both are assigned to the same line.

$$S_j \geq C_i + \sigma_{ij} - M_T(3 - x_{il} - x_{jl} - y_{ijl}), \quad \forall i, j \in J, i \neq j, l \in L \quad (32)$$

(33) says that for every pair of jobs assigned to the same line, exactly one must precede the other to maintain a continuous production sequence. If both jobs i and j are assigned to line l , the sum of the precedence variables must be at least one.

$$y_{ijl} + y_{jil} \geq x_{il} + x_{jl} - 1, \quad \forall i, j \in J, i < j, l \in L \quad (33)$$

To prevent logical contradictions in the schedule, such as a job preceding itself or a circular sequence, at most one ordering direction is permitted between any pair of jobs as per (34).

$$y_{ijl} + y_{jil} \leq 1, \quad \forall i, j \in J, i < j, l \in L \quad (34)$$

5. Hygiene accumulation and forced CIP

The hygiene accumulation constraints track the continuous runtime on each filling line to ensure that mandatory CIP cycles are performed before the hygiene limit H_l is reached. Each job carries an accumulated hygiene time variable A_i , representing the total continuous runtime on the line at the point when job i finishes, without an intervening CIP cycle.

The accumulated hygiene time of any job must be at least as large as its own processing duration, since a job always contributes its full runtime to the hygiene clock regardless of what preceded it as seen in (35).

$$A_i \geq \sum_{l \in A_i} x_{il} \cdot V_i / R_l, \quad \forall i \in J \quad (35)$$

The accumulated hygiene time, shown in (36), cannot exceed the hygiene limit of the line the job is assigned to. If this limit would be breached, the model must either insert a forced CIP or the sequence is infeasible.

$$A_i \leq \sum_{l \in A_i} H_l \cdot x_{il}, \quad \forall i \in J \quad (36)$$

If job j directly follows job i on the same line and no forced CIP is triggered between them, the hygiene time carries over, meaning j inherits the accumulated runtime of i and adds its own processing duration on top, as seen in (37). The Big-M terms deactivate this constraint when i does not directly precede j or when a forced CIP occurs:

$$A_j \geq A_i + \sum_{l \in A_j} x_{jl} \cdot V_j / R_l - M_A(1 - y_{ijl}) - M_A \cdot F_{ij},$$

$$\forall i, j \in J, l \in L: \sigma_{ij} < 4 \quad (37)$$

When a forced CIP is triggered between job i and job j , the hygiene clock resets and A_j is bounded above by only j 's own processing duration, as seen in (38). The Big-M terms ensure this upper bound is only active when i directly precedes j and a forced CIP is selected.

$$A_j \leq \sum_{l \in A_j} x_{jl} \cdot V_j / R_l + M_A(1 - y_{ijl}) + M_A(1 - F_{ij}),$$

$$\forall i, j \in J, l \in L: \sigma_{ij} < 4 \quad (38)$$

Finally, when the setup time between two jobs is four hours or more, a full cleaning cycle is always performed as part of the transition as in (39). In this case, the hygiene clock always resets to j 's own duration, regardless of whether a forced CIP variable is selected.

$$A_j \leq \sum_{l \in A_j} x_{jl} \cdot V_j / R_l + M_A(1 - y_{ijl}), \quad \forall i, j \in J, l \in L: \sigma_{ij} \geq 4 \quad (39)$$

6. Gurt sequencing

The Gurt sequencing constraints enforce the required production order for the three Gurt variants on line E3, as introduced in the parameters section. Two constraints work together to guarantee that the variants are not only started in the correct order but also run consecutively without any other product being inserted between them.

The first constraint, (40) ensures that all jobs belonging to a lower Gurt rank must start before all jobs belonging to a higher rank. This prevents, for example, vanilla from starting before natural regardless of any other scheduling considerations.

$$S_i \leq S_j, \quad \forall i, j \in J: r_i < r_j, r_i, r_j \in \{0, 1, 2\} \quad (40)$$

The second constraint, (41) prevents any non-Gurt job on E3 from being scheduled between two consecutive Gurt variants. For every pair of consecutive Gurt ranks, any non-Gurt job k that runs on E3 must either be sequenced entirely before the lower-ranked Gurt job or entirely after the higher-ranked one. If job k were inserted between them, both sequencing variables would equal zero while $x_{k,E3}$ equals one, violating the constraint.

$$y_{k,g_A} + y_{g_B,k} \geq x_{k,E3}, \quad \forall k \in J: b_k \notin G, E3 \in A_{k,g_A} \in G_{r,g_B} \in G_{r+1} \quad (41)$$

Where G_r denotes the set of Gurt jobs with rank r and G_{r+1} denotes the set of Gurt jobs with the next consecutive rank.

7. Same-base ordering

When multiple jobs share the same base product ID, the order in which they are sequenced affects the efficiency of hygiene accumulation. Running the larger job first allows the hygiene clock to accumulate as much runtime as possible before a CIP cycle is needed, whereas running a small job first would leave the line with little accumulated time and potentially waste the remaining capacity before the hygiene limit is reached. The model therefore enforces that among jobs sharing the same base product ID, the larger job must always be sequenced before the smaller one.

This is implemented as a hard constraint shown in (42) by fixing the sequencing variable to zero for the reverse ordering. If job i has a smaller volume than job j and both share the same base product ID, job j is forbidden from being sequenced after job i .

$$y_{jil} = 0, \quad \forall i, j \in J: b_i = b_j, V_i < V_j \quad (42)$$

It should be noted that this constraint only applies to standalone jobs. Jobs that are consecutive blocks of the same split order are excluded, as their ordering is governed by the split-order constraints in Section 4.1.4.

8. Small-order prioritisation

In addition to the same-base ordering rule, the model enforces a broader prioritisation rule across jobs with different base product IDs. When the total volume of one order is less than 30% of another order's volume and the two jobs share at least one allowed line, the smaller job is required to run before

the larger one. The reasoning is similar to the same-base ordering rule, scheduling a disproportionately small job after a large one would inefficiently interrupt the hygiene accumulation of the larger run and potentially trigger an unnecessary CIP cycle.

The 30% threshold is defined by the parameter $\tau = 0.3$ and is applied to the total order volume rather than the individual block volume. This distinction is important for split orders, where an individual block may appear small while the full order it belongs to is large. Using the total order volume prevents small tail blocks of large orders from being incorrectly prioritised ahead of genuinely large orders.

As with the same-base ordering constraint, this is enforced as a hard constraint by fixing the sequencing variable to zero for the forbidden direction as seen by (43). Although this prioritisation could be handled through penalty weights in the objective function, a hard constraint was deliberately chosen based on operational requirements from the production planners to guarantee the practical viability of the schedule.

$$y_{jil} = 0, \quad \forall i, j \in J: V_i^{total}/V_j^{total} < \tau, L_{ij} \neq \emptyset \quad (43)$$

Where V_i^{total} is the total volume of the order that job i belongs to, V_j^{total} denotes the total volume of the order that job j belongs to, and $L_{ij} = A_i \cap A_j$ denotes the set of lines shared between jobs i and j .

9. Product Category Ordering

For each ordered pair (i, j) where $k_i < k_j$ (job i has a lower category rank and should therefore precede job j), the binary variable y_{jil} equals 1 if j is scheduled before i , violating the preferred order. The term above adds a cost of ρ per such violation to the objective. The constraint is soft meaning the solver may still choose $y_{jil} = 1$, if doing so reduces the makespan sufficiently, but it is penalised for doing so.

10. Split-order constraints

When a large order cannot fit within the hygiene limit of a single run, it is split into consecutive blocks prior to the model, each represented as a separate job. Here, a block represents the maximum number of batches that a line can produce within its respective hygiene limit; specifically, 9 batches inside the 42 hour limit for the E2 and E4 lines and 7 batches inside the 36 hour limit for the E3 line. While the model has freedom in how it assigns these blocks

to lines, a set of constraints ensures that the blocks of the same order are scheduled in a coherent and practical manner.

Assigning consecutive blocks of the same order to different machines is permitted but discouraged. A binary indicator variable $d_{k,k+1}$ shown in (44) and (45) is introduced for each pair of consecutive blocks, taking the value one if the blocks are assigned to different lines. This is penalised in the objective function, reflecting the operational preference for keeping all blocks of the same order on the same line.

$$d_{k,k+1} \geq x_{kl} - x_{k+1,l}, \quad \forall l \in L \quad (44)$$

$$d_{k,k+1} \geq x_{k+1,l} - x_{kl}, \quad \forall l \in L \quad (45)$$

When two consecutive blocks are assigned to the same line, the later block must start after the earlier one has finished. The Big-M term deactivates this constraint when the blocks are on different lines, as seen in (46).

$$S_{k+1} \geq C_k - M_T(2 - x_{kl} - x_{k+1,l}), \quad \forall l \in L \quad (46)$$

Finally, when two consecutive blocks are on the same line, no other job may be inserted between them. For any other job j on the same line, it must be sequenced either entirely before block k or entirely after block $k + 1$ shown in (47).

$$y_{jk} + y_{k+1,j} \geq x_{jl} + x_{kl} + x_{k+1,l} - 2, \quad \forall j \in J, l \in L \quad (47)$$

11. Symmetry breaking and Variable Reduction

Since E2 and E4 are identical in terms of both processing rate and hygiene limit, the model faces a symmetry problem, any feasible schedule can be mirrored by swapping all jobs between E2 and E4, producing an equally valid solution. Without intervention, the solver would waste time exploring both mirror images of every solution. The symmetry breaking constraints eliminate this redundancy by arbitrarily designating E2 as the higher-volume line, effectively cutting the search space in half without excluding any distinct solutions.

The first constraint requires that the total volume assigned to E2 is at least as large as the total volume assigned to E4, considering only jobs that are eligible for both lines as seen in (48).

$$\sum_{j \in J} x_{j,E2} \cdot V_j \geq \sum_{j \in J} x_{j,E4} \cdot V_j, \quad \forall j: \{E2, E4\} \subseteq A_j \quad (48)$$

The second constraint addresses a computational efficiency concern. For any pair of jobs that share no allowed lines, it is physically impossible for one to directly precede the other on the same line. The sequencing variable for such pairs is therefore fixed to zero as seen in (49) before the solver begins, reducing the number of binary variables the solver needs to explore:

$$y_{ijl} = 0, \quad \forall i, j \in J: L_{ij} = \emptyset \quad (49)$$

Where $L_{ij} = A_i \cap A_j$ denotes the set of lines shared between jobs i and j .

4.1.5 Preprocessing

Before the MILP model is constructed, the raw order data undergoes a series of preprocessing steps to ensure that the jobs passed to the solver are well-formed and easier to handle.

Order consolidation.

Multiple MOs for the same article number are merged into a single consolidated order prior to block splitting. This reduces the number of jobs in the model and allows the solver to determine the optimal block structure for the full consolidated volume rather than receiving it in arbitrarily pre-divided fragments.

Block splitting.

Large orders whose volume exceeds the hygiene limit of the available lines are split into consecutive blocks, each representing one MILP job. For Edge line orders, blocks are sized at a maximum of nine batches (approximately 251,965 L), corresponding to roughly 42 hours on E2 and E4. For orders that would produce more than two nine-batch blocks, the block size is automatically reduced to seven batches (approximately 195,975 L, or 33.6 hours), ensuring compatibility with E3's stricter 36 hour hygiene limit.

Tail merging.

If the final block of a split order is smaller than one full batch, it is merged into the preceding block provided the resulting combined volume does not exceed the hygiene limit of any allowed line. This avoids introducing

disproportionately small jobs into the model that would complicate sequencing unnecessarily.

Warm start.

To provide the solver with a good initial solution, a Longest Processing Time (LPT) heuristic is applied before solving begins. Jobs are sorted by descending processing duration and assigned greedily to the least loaded allowed line. The resulting schedule, including start times, completion times, hygiene accumulation values and sequencing variables, is passed to the solver as an initial incumbent solution, allowing the branch-and-bound search to begin from a feasible baseline rather than from the linear programming relaxation.

4.2 Python Model Development

The transition from the mathematics in Section 4.1 to a working tool was done by writing Python code in Visual Studio Code. To turn the equations into logic that the computer can run, Anthropic's Claude Code was used to help write and check the scripts.

4.2.1 Setting up the Code and Parameters

The code is split into a few different files to keep things organised. A file called `config.py` holds the "fixed truths" of the Landskrona plant, such as how fast the machines can run and the hygiene limits. By putting these physical limits in one place, it is easy to update them if the hardware changes.

The `data_loader.py` file handles the input data. It takes the order lists from the supply chain and the changeover matrix and turns them into numbers the model can use. This ensures that the model gets the right information about volumes, SKU IDs, and cleaning rules.

4.2.2 Turning Mathematics into Logic with LLM

The main part of the model is in `milp_model.py`, which uses the PuLP library to talk to the mathematical solver. Converting the "Big-M" constraints and the cleaning sequences into code was done by iterating with Claude Code. The LLM helped write the loops that define how jobs follow each other and ensured the solver follows the 36 and 42-hour time limits for hygiene.

Claude Code was also used to translate the mathematical symbols from the thesis, like the binary variables for cleaning, into Python code. This helped make sure the actual rules from the factory floor were correctly built into the digital model. No production schedules, order data or other material from Oatly was shared with the LLM.

4.2.3 Results and Schedule Generation

The last part of the system is reporter.py, which takes the numbers from the solver and turns them into a schedule people can read. This script takes the start times and machine assignments and puts them into a timeline. It shows exactly when the machines are filling, changing paper, or stopping for a wash. All the results in this study come from this Python setup, showing how mathematics can be used to replace manual planning.

5. Results

5.1 Model Convergence and Computational Performance

To evaluate the computational performance of the MILP model, a convergence analysis was conducted over a computation time of 2000 seconds. This specific timeframe was chosen to provide the solver with a large window to explore the solution space, allowing for the observation of if and when diminishing returns in schedule optimisation occur.

Figure 9 illustrates the relative improvement of the schedules across this 2000-second period from the solver's own initial incumbent. The data points and the corresponding trendline demonstrate that the solver achieves the most rapid efficiency gains early in the computation process. The trendline exhibits a steep, upward trajectory during the first 500 seconds, capturing approximately 14% relative improvement. The 500-second mark acts as an inflexion point on the curve; at this stage, the steep climb visibly transitions into a more gradual slope, marking the initial onset of diminishing returns.

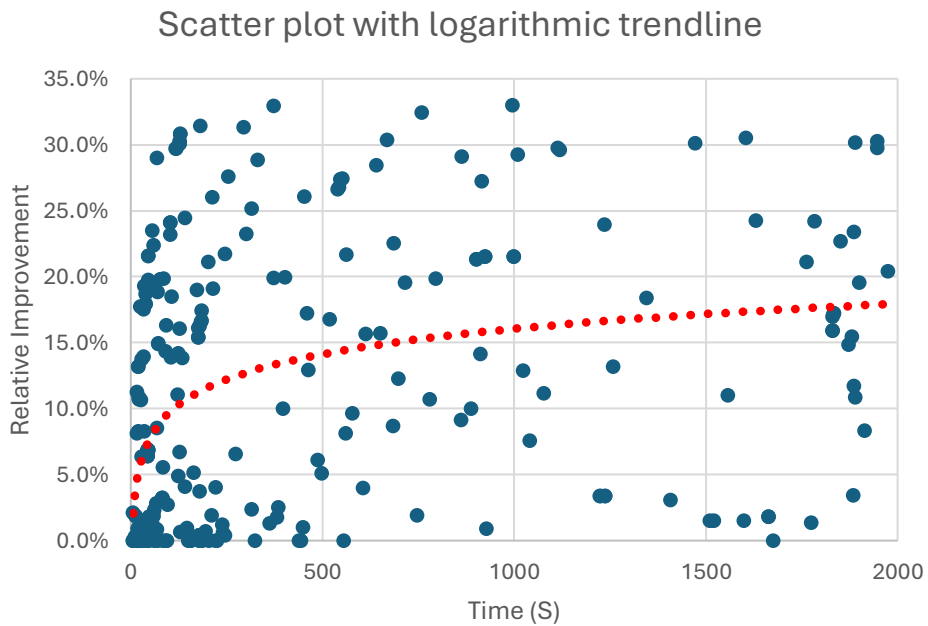


Figure 9, Relative improvement from the solver's initial incumbent over 2000 seconds of computation time.

Despite this slowdown, the results show that the model does not hit a rigid, flat plateau. Between 500 and 2000 seconds, the solver consistently identifies further sequencing optimisations, with the trendline steadily climbing to roughly 18% relative improvement by the end of the run. Because the schedules continued to measurably improve throughout the entire observed period without flattening completely, a practical cutoff was necessary to proceed. The 2000-second limit, dictated by the physical hardware capacity was therefore established as the baseline to generate the final optimised schedules for the subsequent comparative analysis.

5.2 Efficiency Comparison: MILP vs. Manual Planning

5.2.1 2000-Second Simulation for the 39-Week Baseline

The primary focus of the results relies on a direct comparison between the makespan of the manual production plans and the optimised schedules generated by the mathematical model. This comparison spans 39 production weeks with a wide variety of demand levels and product mixes across the three filling lines (E2, E3, and E4).

Over this evaluated period, the optimisation model identified more efficient production sequences compared to the manual approach 26 out of the 39 weeks as seen in Table 4. The manual planning baseline recorded an average makespan of 449.94 hours per week. In contrast, the MILP model achieved a lower average makespan of 436.92 hours per week. This difference translates to a cumulative reduction of 507.52 hours of production time across the 39 weeks, representing an overall efficiency improvement of 2.9%. Furthermore, if the most efficient schedule is selected from either the model or the manual planner for each individual week of 2025, the combined total efficiency improvement reaches 5.1%. However, it should be noted that it should be seen as an upper bound under perfect foresight since manual scheduling would need to be done in parallel to the model in order to achieve 5.1%.

Table 4, Week by week results

Week	Model (h)	Planner (h)	Difference (h)	Difference (%)	Winner
v.9	436.29	437.55	(1.26)	(0.3%)	Modell
v.11	388.48	390.45	(1.97)	(0.5%)	Modell
v.12	452.19	439.44	12.75	2.9%	Planner
v.13	337.29	340.08	(2.79)	(0.8%)	Modell
v.14	492.57	493.54	(0.97)	(0.2%)	Modell
v.15	487.27	482.78	4.49	0.9%	Planner
v.16	496.06	497.46	(1.40)	(0.3%)	Modell
v.17	500.34	482.23	18.11	3.8%	Planner
v.18	490.58	494.07	(3.49)	(0.7%)	Modell
v.19	387.87	391.26	(3.39)	(0.9%)	Modell
v.20	392.19	388.61	3.58	0.9%	Planner
v.21	495.52	486.5	9.02	1.9%	Planner
v.22	488.39	489.83	(1.44)	(0.3%)	Modell
v.23	400.50	402.71	(2.21)	(0.5%)	Modell
v.24	457.46	456.81	0.65	0.1%	Planner
v.25	285.79	291.1	(5.31)	(1.8%)	Modell
v.26	509.54	501.2	8.34	1.7%	Planner
v.27	374.12	372.8	1.32	0.4%	Planner
v.29	493.73	500.31	(6.58)	(1.3%)	Modell
v.30	463.71	501.73	(38.02)	(7.6%)	Modell
v.31	486.28	492.52	(6.24)	(1.3%)	Modell
v.32	473.17	463.7	9.47	2.0%	Planner
v.33	368.51	311.9	56.61	18.2%	Planner
v.36	519.04	484.11	34.93	7.2%	Planner
v.37	489.32	493.29	(3.97)	(0.8%)	Modell
v.39	422.30	426.49	(4.19)	(1.0%)	Modell
v.40	496.36	483.81	12.55	2.6%	Planner
v.41	486.91	485.13	1.78	0.4%	Planner
v.42	333.21	408.34	(75.13)	(18.4%)	Modell
v.43	454.84	497.99	(43.15)	(8.7%)	Modell
v.44	448.24	503.29	(55.05)	(10.9%)	Modell
v.45	421.39	482.73	(61.34)	(12.7%)	Modell
v.46	427.67	479.44	(51.77)	(10.8%)	Modell
v.47	358.03	399.74	(41.71)	(10.4%)	Modell
v.48	445.40	480.51	(35.11)	(7.3%)	Modell
v.49	437.66	488.79	(51.13)	(10.5%)	Modell
v.50	345.19	412.06	(66.87)	(16.2%)	Modell
v.51	366.57	422.58	(56.01)	(13.3%)	Modell
v.52	429.98	490.6	(60.62)	(12.4%)	Modell

The data shows not only the average production times but also how much those times vary between the two methods. The manual production plans exhibited a standard deviation of 55.92 hours over the evaluated period. The schedules generated by the optimisation model showed a similar spread, recording a standard deviation of 58.98 hours. This distribution indicates that while the mathematical model achieves a lower makespan on average, the variation in weekly production time remains comparable to the manual baseline, fluctuating based on the specific product combinations and cleaning constraints of each order list.

To investigate what distinguishes the weeks where the model outperforms the manual planner from those where the planner performs better, a retrospective analysis of scheduling characteristics was carried out across all evaluated weeks. The analysis shows a consistent difference in scheduling complexity between the two groups. Weeks in which the planner outperformed the model had, on average, around 36 production jobs in the MILP solution, compared to approximately 30 in model-winning weeks.

In other words, the weeks in which the planner outperforms the model are systematically more demanding: more products need to be scheduled, and more cleaning cycles are required to handle the transitions between them. This reflects a higher degree of product complexity, weeks with a more varied product mix that generates more setup transitions on the Edge lines. The weeks in which the model performs well tend to be weeks with fewer products and a simpler transition structure.

5.2.2 Increased Computational Time for Complex Weeks

To evaluate the impact of the solver's time limit on highly complex production weeks, an extended simulation was conducted. Weeks 33 and 36 were selected for this test because they represented the instances where the manual planner outperformed the baseline model by a large amount. Both weeks are characterised by high combinatorial complexity, containing a large number of distinct SKUs. For this extended test, the computational time limit was increased from the baseline of 2,000 seconds to 25,000 seconds.

The results, presented in Table 5, demonstrate that with adequate computational time, the MILP model is capable of navigating the complex changeover matrix more efficiently than the manual planner. In both instances, the extended runtime allowed the solver to explore a larger portion of the solution space, consequently identifying sequences that resulted in a shorter total makespan than the manual plans.

Table 5, Increased Computational Time for Complex Weeks

WEEK	OLD MODEL TIME (2,000S)	NEW MODEL TIME (25,000S)	MANUAL PLANNER TIME
V.33	368.51 h	307.12 h	311.90 h
V.36	519.04 h	480.09 h	484.11 h

5.3 Clean-In-Place Cycles

The optimisation model also evaluated the frequency of required CIP cycles by navigating the sequence-dependent setup rules outlined in the changeover matrix. The total number of mandatory and matrix-driven CIPs was tracked for both the manual plans and the generated schedules across the 39 evaluated weeks.

Over this period, the manual production plans required a total of 425 CIP cycles to complete the specified order lists across the three Edge lines. The optimised schedules generated by the MILP model required a total of 446 CIP cycles to process the same MO. On a weekly basis, the manual plans averaged 10.9 CIP cycles per week. The mathematical model averaged 11.4 CIP cycles per week.

6. Discussion and Conclusions

This chapter interprets the results of the MILP model in the context of the Landskrona facility. It evaluates the performance gains relative to manual planning and discusses the broader implications for sustainability and organisational resilience.

6.1 Performance Evaluation: Model vs. Manual Planning

The results from the 39-week comparative study show a performance gap between manual, experience-based planning and the mathematical optimisation model. With an average reduction in total production time of 13.01 hours per week, the model finds a way to complete the same order lists approximately 2.9% faster than current methods. This finding directly answers the study's second research question, suggesting that part of the productivity gap at the Landskrona site comes from the complex scheduling rules that are difficult for human planners to manage manually.

6.1.1 Week-to-Week Variability of Makespan

The week-to-week variability of the makespan is comparable for the two methods, with the manual plans recording a standard deviation of 55.92 hours and the model-generated schedules a marginally higher 58.98 hours. This slightly larger spread reflects the model's willingness to restructure sequences and rebalance load across the Edge lines in response to each week's specific product mix and cleaning requirements, rather than any inherent instability; the difference of roughly three hours is operationally negligible. In other words, the model achieves a lower average makespan without materially widening the range of weekly outcomes. It should be noted that this statistical spread concerns the consistency of the *output* (production time) and should not be confused with the stability of the *planning process* itself which is covered in section 6.4.

6.1.2 The Case for a Hybrid Planning Approach

The extended computational testing demonstrated that the mathematical model is capable of finding schedules with a shorter makespan than manual planning, even in some of the most complex weeks when given adequate processing time. However, this mathematical capability does not eliminate the need for human expertise. Instead, it shifts the justification for a hybrid planning approach.

Rather than relying on the manual planner to compensate for the model's performance under strict time limits, the planner is essential for managing factors that fall outside the model's deterministic boundaries. The MILP framework operates under the assumption of a static environment, meaning it does not account for stochastic disruptions such as sudden equipment failures, raw material delays, or fluctuations in the upstream extraction and formulation processes.

Therefore, the most practical workflow could be to use the model to generate a baseline schedule. This automates the computationally heavy task of navigating the sequence-dependent changeover matrix and optimising the line assignments. The human planner can then review and refine this baseline, applying their tacit knowledge to adjust for upstream dependencies, operator availability, and practical exceptions. This sequential approach leverages the mathematical precision of the digital tool to maximise capacity output, while maintaining the operational flexibility and oversight required to handle the unpredictable reality of the factory floor.

6.1.3 Impact of Product Complexity and Solver Constraints

As seen in Section 5.2.1, product complexity is the primary factor dictating the model's performance under a strict time limit. In scheduling weeks with a relatively simple product mix, fewer jobs, and fewer required CIP cycles, the model finds shorter makespans within the 2,000-second baseline limit. However, in weeks with a more complex product mix, where the number of jobs and CIP cycles is higher, the solver initially fell short. This outcome is consistent with how branch-and-bound algorithms behave under time constraints; as the number of jobs grows, the number of feasible schedules increases factorially, making it progressively harder to find a near-optimal solution within a short, fixed time budget.

The extended computational testing detailed in Section 5.2.2 supports this hypothesis. When the solver was allowed 25,000 seconds to process the highly complex order lists of weeks 33 and 36, it successfully identified schedules that outperformed the manual planner. This supports that the initial performance gap in complex weeks was not caused by a flaw in the model's logic or constraints, but rather by the initial 2,000-second boundary. Given adequate processing time, the mathematical solver should be capable of navigating the combinatorial increase of the changeover matrix more effectively than a human planner, regardless of the week's complexity.

In an operational setting, the model is not bound by the time constraints of a retrospective 39-week academic study. When supporting the planning process for a single upcoming week, the solver can be permitted to run for much longer periods without issue. The extended results prove that utilising this longer runtime directly unlocks further capacity, systematically turning difficult, high-complexity weeks into highly optimised production schedules.

6.1.4 Validation of results

The continuous validation process shaped the behaviour of the mathematical model and explains why the final objective function contains multiple varied penalty variables. By evaluating the real-world practicality of the generated schedules through continuous feedback from the manual planner, one could identify several tacit, qualitative constraints that were necessary for day-to-day operations but absent from the facility's quantitative technical specifications.

Consequently, the objective function was expanded beyond a simple minimisation of time to include soft constraints that address these operational realities. For example, the validation process revealed an operational preference to keep consecutive blocks of the same manufacturing order on a single machine. To account for this, the split-order penalty was introduced into the objective function, assigning a mathematical cost whenever an order is fragmented across different lines. Similarly, feedback indicated a need to consolidate products, which led to the base spread penalty. This penalty discourages the model from spreading the same product base across multiple lines, forcing the solver to prioritise long, continuous batches.

Another adaptation involved the incompatible transition penalty. This soft constraint was added to discourage the solver from freely choosing incompatible product transitions that trigger four-hour washes if compatible alternatives exist. By loading the objective function with penalty variables representing the manual planner's tacit knowledge, the generated schedules transitioned from being merely technically possible to practically feasible in the real world.

Because the large number of the manual planner's soft constraints was taken into consideration, the mathematical solver was restricted to operating within the same logical boundaries that the human planner used. It was actively chosen to make the schedules practically feasible rather than designing an objective function that pursued theoretical time savings at the expense of factory usability. This resulted in schedules that were practically feasible

which could also be an indicator to why the model results were so close to the human planner in many cases.

6.2 Data Integrity and Transparency

The reliability of the comparison between manual planning and the mathematical model depends heavily on the quality of the data. Because of this, the data from the original 52 weeks of 2025 had to be filtered. While 39 weeks had usable data from both methods, 13 specific weeks were excluded from the final study. This was done to ensure the results reflect normal factory conditions rather than missing data or operational anomalies.

The weeks were removed based on specific operational reasons. Weeks 1 through 3 were excluded because no production timestamps were recorded at all. Similarly, weeks 4 through 8 were removed due to missing timestamp data specifically for the E4 line. Week 10 was also excluded from the analysis due to incomplete data records. Week 28 was removed because the manual records showed zero production hours for both the E2 and E3 lines. Week 34 was excluded because the E3 line was out of operation for service from Monday through Thursday, heavily skewing the available capacity. Week 35 was removed because the production schedule included trial runs that disrupted the standard data structure. Finally, week 38 was excluded due to exceptionally low production volumes. Including these unusual weeks would give an unfair baseline, since such severe hardware restrictions, missing data, and low volumes do not represent the complex scheduling problems the model is meant to solve.

By removing these 13 weeks, the study ensures that the reported 2.9% efficiency gain is based on a representative production environment. This data filtering is an important step, as it stops missing data or strange outliers from affecting the final results. The final 39-week dataset gives a stable foundation for comparing how well the model handles the complicated cleaning rules and setup times during a standard week.

6.3 Environmental Impact and Resource Efficiency

The results showed that the model scheduled slightly more CIP cycles than the manual planner, averaging 11.4 cycles per week compared to 10.9. This result highlights a natural balance in the scheduling problem between saving time and saving cleaning resources. Because the model's main objective is to minimise the global makespan (C_{MAX}), the exact point in time when the final machine finishes the weekly schedule, it actively balances the production load

across all three Edge lines. To achieve this, the solver will sometimes accept the penalty of an extra CIP cycle on a line with spare capacity if it allows a bottleneck line to offload volume. While this increases the total number of washes, it parallelises the work more efficiently and reduces the overall completion time for the facility.

The mathematical model treats these cleaning requirements as penalties. If Oatly's priority shifts closer towards lowering the amount of CIP, the penalty weights in the model can be adjusted. By increasing the penalty value for a CIP cycle, the solver could focus more on finding the cleanest path through the production week.

Even though the number of CIP cycles increased slightly, the model could still have environmental benefits. By reducing the total weekly production time by an average of 13.01 hours, the factory saves general resources required to keep the facility running. This reduction in active operating time leads to direct savings in electricity, heating, and cooling. Therefore, moving to a data-driven planning environment offers different ways to support the company's long-term sustainability targets, depending on how time and water are prioritised.

6.4 Organisational Resilience and Knowledge Management

Research Question 4 concerns how the transition to an automated optimisation model may affect the stability of the planning process, specifically regarding its reliance on individual human expertise. A clear finding of this study is the potential for mathematical modelling to strengthen this stability by reducing the plant's dependence on a single planner. Currently, the production planning process relies heavily on the tacit knowledge and intuition of a single individual. While this experience-based approach has historically functioned, it introduces a substantial risk because the critical logic governing the factory floor is not fully documented. By shifting towards a MILP model, Oatly can institutionalise this operational knowledge into a data-driven framework, ensuring that the planning process remains stable regardless of individual personnel changes.

This transition also addresses the cognitive limitations inherent in manual planning. Research suggests that human intuition can be an unreliable tool in highly complex situations, as the brain often attempts to ignore complexity rather than manage it. At Oatly, where planners must balance a large number

of variables including tank availability and sequence-dependent cleaning, the cognitive load is high. The implementation of an automated model reduces this burden, allowing the planner to shift from the tedious task of manual sequencing to a more strategic role of oversight and exception management.

Finally, it is important to clarify that the introduction of such a tool is not intended to replace human expertise. Ethically and practically, the model functions best as a decision-support system that provides a robust, mathematically sound baseline. The human planner remains essential for handling the "soft data" and unexpected operational nuances that a deterministic model might not capture. By combining the speed and precision of MILP with human experience, the organisation can create a more resilient and efficient planning environment that is less vulnerable to individual error or knowledge loss. In direct answer to Research Question 4, the transition is therefore expected to increase the stability of the planning process: by encoding tacit scheduling logic into a transparent, reproducible model, the organisation becomes markedly less dependent on the availability and judgement of any single individual, while that individual's expertise is retained for oversight and exception handling rather than lost.

6.5 Practical Limitations and Model Boundary Conditions

While the results show a theoretical advantage, it is important to recognise the practical limitations of the model. The MILP framework is a deterministic optimisation tool, meaning it assumes all inputs, such as processing times, order quantities, and setup durations, are known and constant. In reality, the factory floor experiences unexpected disruptions like equipment failures or supply chain delays. However, it must be clarified that the historical manual schedules used for comparison are also deterministic. These historical records represent the initial "best-case" plans created at the start of the week and do not account for sudden disruptions. In practice, the planning team uses a separate, live schedule to handle unexpected events as they happen. Therefore, the reported efficiency gains represent a direct comparison between two initial "best-case" plans, rather than a guaranteed daily outcome.

Another critical limitation is the model's sensitivity to hardware performance. Because MILP problems are computationally intensive and their complexity scales exponentially as the number of integer variables increases, the quality of the solution is directly tied to the computer's processing power and memory. Terminating the solver within a fixed timeframe means that a more powerful machine might explore more nodes and identify a better solution

than a standard laptop using the same time limit. This makes the replicability of exact results highly hardware-dependent in professional settings.

Finally, the model relies on several simplification constraints to remain computationally manageable. A primary example is the assumption that raw oat base is always available from the extraction phase. While this allows the optimiser to focus on the complex scheduling of the filling lines, it ignores potential upstream bottlenecks that could impact the feasible start times. Furthermore, the validation process relies on retrospective testing of 2025 data, which carries the inherent assumption that past production patterns are representative of future requirements. These factors must be considered when evaluating the long-term applicability of the model as a standalone tool.

6.6 Computational Time Limits and Diminishing Returns

As explained in the methodology, the goal of the convergence analysis was to identify a specific time limit where the solver reaches a point of diminishing returns. The ideal scenario would be to observe the performance curve hit a completely flat plateau, indicating that additional computing time would yield negligible improvements. This behaviour would justify setting a strict cutoff time for daily use in the factory.

However, the convergence analysis shown in Figure 9 demonstrates different behaviour. There is a steep increase in relative improvement during the first 500 seconds. After this point, the curve shows diminishing returns, but it never completely flattens into a horizontal line. Instead, the trendline continues to climb slowly, approaching an 18 percent relative improvement at the 2,000-second mark.

The decision to establish a baseline solver limit of 2,000 seconds per week was primarily driven by hardware limitations during the batch-processing phase of the study. Running the model consecutively for all 39 evaluated weeks at 2,000 seconds required a continuous computational runtime of 78,000 seconds (over 21 hours). Attempting to push the time limit higher for the entire dataset caused the computers to overheat and struggle to complete the calculations.

The extended simulations conducted on weeks 33 and 36 confirm that the solver does not hit a hard plateau. When the time limit was increased to 25,000 seconds for these highly complex weeks, the model continued to iterate through the large number of possible schedules, consequently yielding time savings that the 2,000-second limit missed.

Because the results were largely bounded by the 2,000-second limit, the reported 2.9% average efficiency gain represents a conservative baseline. In a real-world application at Oatly, the planning team only needs to run the model for one single week at a time. A longer solver run than 2000 seconds can be executed, demanding far less continuous computing strain than the study's 78,000-second batch run. Therefore, the actual efficiency gains in a live production environment could be higher than what was captured in the baseline study, as the facility can actively utilise longer runtimes to unlock further capacity.

6.7 Recommendations for Future Research

The results of this study establish a strong foundation for mathematical optimisation at the Landskrona facility, but several areas remain for further development before the model can be fully integrated into daily operations. A primary recommendation is the development of a user-friendly front-end interface that translates the complex mathematical output into a clear visual timeline. This interface must also allow planners to manually override specific machine assignments when human intuition or sudden operational changes are required. By providing a flexible visual tool, the planning team can leverage the mathematical speed of the solver while maintaining control over the factory floor.

Another important area for future work is scaling the optimisation model to handle upcoming physical changes to the facility. The Landskrona plant is currently replacing the older B1 and GT1 packaging machines and adding an additional high-volume Edge filling line. Future research should expand the model parameters to include this new hardware, which would allow the solver to create a complete, synchronised schedule for the entire packaging department rather than only focussing on a subset of machines. Furthermore, there is a large potential in expanding the optimisation scope backwards to include the upstream extraction and formulation phases. A more holistic model could synchronise the entire facility from raw oat intake to final packaging, potentially uncovering even greater efficiency gains and resource savings.

Finally, future applications of the model should focus on developing procedures for rapid, real-time re-planning. When unexpected events like equipment failures or material delays occur, the planning team needs a fast way to input the current factory status into the model. By building a direct data pipeline for these live updates, the solver could calculate a new, optimised recovery schedule based strictly on those updated conditions.

These advancements would make optimised, resource-efficient scheduling a permanent and flexible part of the site's operational culture.

6.8 Conclusion

This study demonstrated that the complex physical constraints, routing dependencies, and sequence-dependent cleaning rules at Oatly's Landskrona facility can be translated into a mathematical model. By structuring the operational boundaries as hard constraints within a Mixed-Integer Linear Programming framework, the model proved capable of generating technically feasible and optimised production schedules.

When benchmarked against 39 weeks of historical manual planning under a strict 2,000-second baseline, the optimisation model identified shorter production sequences in 26 weeks. This baseline performance achieved a 2.9% reduction in total makespan, saving an average of 13.01 production hours per week.

Furthermore, extended computational testing demonstrated that mathematical optimisation could be more effective for this specific semicontinuous environment, provided the time constraints are managed properly. When the time limit was increased to 25,000 seconds for the most highly complex outlier weeks, the MILP model outperformed the manual plans. This supports the hypothesis that the model is capable of navigating demanding combinatorial complexity better than manual heuristics, given adequate processing time.

Despite this mathematical capability, the transition to an automated model should not aim to entirely replace human expertise. The MILP framework functions best as a robust decision-support tool. By allowing the digital model to handle the heavy computational load of generating an efficient baseline schedule, the human planner is freed to oversee stochastic disruptions, upstream dependencies, and practical factory exceptions. Consequently, combining a data-driven optimisation model with human oversight provides a resilient operational foundation, equipping the Landskrona facility to handle high production complexity and support continued growth.

7. References

- Agu, E. E., Iyelolu, T. V., Idemudia, C., & Ijomah, T. I. (2024). Exploring the relationship between sustainable business practices and increased brand loyalty. *International Journal of Management & Entrepreneurship Research*. <https://doi.org/10.51594/ijmer.v6i8.1365>
- Anthony, R. N. (1965). *Planning and control systems: A framework for analysis*. Graduate School of Business Administration, Harvard University.
- Apt, K. R. (2003). *Principles of constraint programming*. Cambridge University Press.
- Azani, M., Tan, L., Basten, R. J. I., & de Kok, A. G. (2025). Marrying math and mind: Towards production planning systems for industry 5.0. *International Journal of Production Research*. <https://doi.org/10.1080/00207543.2024.2443800>
- Bell, E., Bryman, A., & Harley, B. (2022). *Business research methods* (6th ed.). Oxford University Press.
- Bonabeau, E. (2003). Don't trust your gut. *Harvard Business Review*, 81(5), 116–123.
- Chapman, S. N. (2006). *The fundamentals of production planning and control*. Pearson Prentice Hall.
- Cui, Y., Zhang, X., & Luo, J. (2022). Filling process optimization through modifications in machine settings. *Processes*, 10(11), Article 2273. <https://doi.org/10.3390/pr10112273>
- Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. W. H. Freeman and Company.
- Georgiadis, G. P., Mariño Pampín, B., Adrián Cabo, D., & Georgiadis, M. C. (2019). Optimal production scheduling of food process industries. *Computers & Chemical Engineering*, 131, 106682. <https://doi.org/10.1016/j.compchemeng.2019.106682>
- Harjunkoski, I., Maravelias, C. T., Bongers, P., Castro, P. M., Engell, S., Grossmann, I. E., Hooker, J., Méndez, C. A., Sand, G., & Wassick, J. (2014). Scope for industrial applications of production scheduling models and

solution methods. *Computers & Chemical Engineering*, 62, 161–193.
<https://doi.org/10.1016/j.compchemeng.2013.12.001>

Hillier, F. S., & Lieberman, G. J. (2010). *Introduction to operations research* (9th ed.). McGraw-Hill.

Institute of Sustainability Studies. (2025). *Business spotlights: Oatly*. Retrieved February 19, 2026, from <https://instituteofsustainabilitystudies.com/insights/sustainable-business-spotlights/business-spotlights-oatly/>

Klotz, E., & Newman, A. M. (2013). Practical guidelines for solving difficult mixed integer linear programs. *Surveys in Operations Research and Management Science*, 18(1–2), 18–32.
<https://doi.org/10.1016/j.sorms.2012.12.001>

Kopanos, G. M., Puigjaner, L., & Georgiadis, M. C. (2011). Resource-constrained production planning in semicontinuous food industries. *Computers & Chemical Engineering*, 35(12), 2929–2944.
<https://doi.org/10.1016/j.compchemeng.2011.04.012>

Lillyworks. (n.d.). *How an automated scheduling system revolutionizes production scheduling*. <https://www.lillyworks.com/blog/how-an-automated-scheduling-system-revolutionizes-production-scheduling/>

Merten, D. C., Hütt, M.-T., & Uygun, Y. (2021). A network analysis of decision strategies of human experts in steel manufacturing
<https://doi.org/10.48550/arXiv.2112.01991>

Mitchell, S., Dunning, I., & O'Sullivan, M. (2011). *PuLP: A linear programming toolkit for Python*. Optimization Online. <https://optimization-online.org/wp-content/uploads/2011/09/3178.pdf>

Montoya-Torres, J. R., Soto-Ferrari, M., & González-Solano, F. (2010). Production scheduling with sequence-dependent setups and job release times. *DYNA*, 77(163).
https://www.researchgate.net/publication/49598675_Production_scheduling_with_sequencedependent_setups_and_job_release_times

Oatly Group AB. (2024). *Annual report consolidated financial statements Oatly Group AB (publ)*. <https://investors.oatly.com/static-files/e230b0c1-8573-497a-9b3a-791b30864732>

Oatly Group AB. (n.d.). *Sustainability plan*. <https://www.oatly.com/oatly-who/sustainability-plan>

Pinedo, M. L. (2016). *Scheduling: Theory, algorithms, and systems* (5th ed.). Springer.

Samouilidou, M. E., Georgiadis, G. P., & Georgiadis, M. C. (2023). Food production scheduling: A thorough comparative study between optimization and rule-based approaches. *Processes*, 11(7), Article 1950. <https://doi.org/10.3390/pr11071950>

Schrage, E., & Dettmer, H. W. (2000). *Manufacturing at warp speed: Optimizing supply chain financial performance*. CRC Press

Silver, E. A., Pyke, D. F., & Peterson, R. (1998). *Inventory management and production planning and scheduling* (3rd ed.). Wiley.

Straits Research. (2025). *Oat milk market size, share & trends | Industry report, 2033*. <https://straitsresearch.com/report/oat-milk-market>

Vollmann, T. E., Berry, W. L., Whybark, D. C., & Jacobs, F. R. (2005). *Manufacturing planning and control for supply chain management* (5th ed.). McGraw-Hill.