Ca' Foscari University of Venice

Department of Economics

Operational Research in Production Line Optimization: A Study on Project RefAIne

Final thesis written by

Lorenzo Serafini - 886128

Supervised by Professor Andrea Albarelli

A thesis submitted for the master degree in Data Analytics for Business and Society

Academic Year 2023 - 2024

Contents

Part I
Operational Research in Production Line Optimization

The field of Operational Research, hereinafter sometimes referred to as OR, is a scientific discipline that studies decision analysis and the optimization of complex systems using mathematical and statistical models and algorithms.
Operational research is closely related to automation, in the management of resources, logistics and as the main focus of this thesis, in production to improve efficiency and reduce costs.
From its early mathematical foundations in probability theory and calculus, OR evolved significantly over the centuries, today is involved in many different areas of study such as management, financial engineering, and artificial intelligence.
OR has become an indispensable tool for solving real-world problems across various industries and for companies to be competitive in the Information Age. This thesis explores the historical development of OR, its most used techniques and methodologies, focusing on its application in production line optimization, and presents a case study on Project RefAIne, a practical implementation of OR principles in a manufacturing environment.

The first part of the thesis wants to describe the evolution of OR, from its roots in the 16th century to the boom it had at the beginning of the Information Age, highlights key milestones such as the development of linear programming, the simplex method and the emergence of project management techniques like the Critical Path Method and Program Evaluation and Review Technique. These are just a few advancements that have expanded the applications of the discipline, from military logistics during World War II to modern-day supply chain optimization.

The second part of the thesis shifts focus to Project RefAIne, a case study that demonstrates the practical application of OR in a manufacturing setting. The project aims to optimize production scheduling by integrating data from various sources, automating workflows, and minimizing production costs. By leveraging mathematical programming, the project seeks to address the challenges of workload balancing, machine utilization, and multi-phase workflows in a flexible manufacturing system (FMS).

The purpose of this thesis is to bridge the gap between the theoretical concepts of OR and their practical implementation by showing how OR can be used to improve decision making, increase efficiency and drive innovation in manufacturing systems. This study wants not only contribute to the academic understanding of OR, but will also provide valuable guidance for practitioners seeking to optimize their operations in an increasingly complex and dynamic industrial landscape.

1 Introduction and General Overview

The field of operations research (OR), as it is recognized today, is founded on a rich history of mathematical, scientific, and management-related advances that have taken place over many centuries. Although its formal establishment as a distinct field can be traced back to World War II, the underlying intellectual concepts of OR were developed much earlier, thanks to various contributions in probability theory, mathematical optimization, and scientific management.

The initial foundations of OR are in the studies of probability and statistics that opened the doors to decision making analysis.

During the 19th century, as the Industrial Revolution brought significant changes to manufacturing and commerce, new difficulties in achieving organizational efficiency appeared. In this period a truly great need of efficiency first appeared and the scientific management principles were laid out. Their aim was first to analyze the work processes and then to enhance these processes.

A critical point in the field's development occurred during World War II, when the British military encountered unprecedented challenges while coordinating complex operations and allocating limited resources. The success of early OR teams in military contexts, especially in the deployment of radar and protection of convoys, demonstrated the effectiveness of applying scientific methods to address operational problems. This wartime experience resulted in the swift growth of the field into civilian uses in the post-war period.

The shift from military to civilian applications was marked by significant theoretical progress and the creation of new analytical tools. The appearance of computer technology in the latter portion of the 20th century brought a revolution in the practical capabilities of OR, enabling the solution of increasingly complex problems and expanding the field's accessibility. Currently, OR is continuously evolving, incorporating new technological capabilities and responding to emerging challenges in business, industry, and public services [22].

This section will examine the key developments, innovations, and individuals that have influenced the evolution of operations research. It will explore how diverse concepts of mathematics, science, and management converged to create this problem-solving and decision-making methodology.

2 Early Mathematical Foundations (1564-1847)

2.1 Probability Theory and Games of Chance

1564 Girolamo Cardano is frequently recognized as the first mathematician to explore gambling and calculate probabilities. Cardano's method involved determining probabilities by finding the ratio of successful outcomes to the overall number of possible outcomes, assuming all outcomes had the same chance of occurring. This represents a fundamental advance in understanding probability and decision-making under uncertainty, both of which are essential to OR. Even though Cardano's book *Liber de Ludo Aleae* was not published until 1663, his ideas regarding probability set the stage for later progress. The formal study of probability, which grew from the analysis of games of chance, serves as an essential resource for dealing with uncertainty in OR. The capacity to measure risk, which originated in these early studies of games, is crucial for making well-informed decisions in various OR applications, including finance, logistics, and risk management [20, 24].

1654 Through their correspondence, Blaise Pascal and Pierre de Fermat developed a solution for dividing stakes in an unfinished gambling game, a problem now known as the "problem of points." Their work was fundamental to the development of probability theory and influenced later studies of risk and uncertainty [20, 24].

2.2 Mathematical Optimization and Calculus

1665 Newton created an iterative method for estimating the solution (root) of a general equation, $f(x) = 0$. This approach, known as Newton's method, is essential to both numerical analysis and optimization and can be modified to solve nonlinear constrained optimization problems. It can also be applied to interior-point methods for addressing linear programming problems.
Newton defined the condition necessary to find an extremum (maximum or minimum) of a function, $f(x)$, by setting its derivative equal to zero. This is a fundamental idea in optimization theory and is employed in OR to identify the optimal solution to a problem.
In his work *De Methodis Serierum et Fluxionum* (completed in 1671 but published posthumously in 1736), Newton explained how to determine the maximum or minimum of a quantity by examining when its "fluxion" neither increases nor decreases. This idea is critical for optimization because identifying where a function's rate of change is equal to zero helps identify a possible maximum or minimum [20, 24, 40].

2.3 Statistical Developments

1713 Jacob Bernoulli formulated the law of large numbers, which states that as the number of trials in a random experiment increases, the average outcome will tend towards its expected value. This is a key theorem in probability and statistics [57].

1715 Mathematicians discovered that the expansions of several common transcendental functions were specific instances of the general series now known as the Taylor series. Brook Taylor, a follower of Newton, stated the general result in his *Methodus Incrementorum Directa et Inversa*. Taylor series led to the theory of approximation, which makes use of a polynomial function to approximate a differentiable function within a defined error range [11, 24].

1733 Abraham de Moivre described the normal distribution as an approximation to the binomial distribution, which is crucial to statistical modeling and inference [24].

1736 Leonhard Euler, a Swiss mathematician, famous for establishing graph theory, described the layout of the seven bridges of Königsberg that linked the two banks of the Pregel River and two of its islands. He answered the question: Is it possible to cross each of the seven bridges in a single continuous walk without crossing any bridge more than once? The answer was no. Euler demonstrated that for a graph configuration to have a path with the required characteristics, each node (land area) must be linked to an even number of arcs (bridges) [13, 57].

1788 Joseph-Louis de Lagrange, a French mathematician, introduced his method of Lagrange multipliers in his renowned work *Mécanique Analytique*. This method determines the extrema of functions when they are subject to equality constraints. Notably, Lagrange's technique permits finding the maximum or minimum value of a function while adhering to certain restrictions on the variables [23].

1795 Both German mathematician Carl Friedrich Gauss and French mathematician Adrien-Marie Legendre are credited for independently discovering the method of least squares. This method provides a way to find the best fit for a set of data points by minimizing the sum of the squares of the errors. It is a fundamental method for computing the unknown parameters in the general regression model, which is commonly seen in operations research applications and statistical analyses [10].

1810 Pierre-Simon Laplace formulated the general central limit theorem: the sum of a sufficiently large number of independent random variables will follow a distribution that is approximately normal. His work introduced new and highly advanced analytical techniques to the field of probability theory [26, 56].

1826 Jean-Baptiste-Joseph Fourier's work on inequalities in linear systems had a practical influence as the foundation for linear programming, a core technique in operations research [22].

2.4 Network Theory and Boolean Algebra (Kirchhoff, Boole)

1845 The German physicist Gustav R. Kirchhoff discovered two well-known laws describing the flow of electricity through a network of wires. Kirchhoff's laws, which are the conservation of flow at a node and the law of potential, have direct applications to modern network and graph theory. Kirchhoff also demonstrated how to construct a set of $n+m-1$ circuits in a connected graph with m nodes and n arcs [44].

1847 The English mathematician and logician George Boole introduced Boolean algebra, a system of symbolic logic that represents basic operations in binary mathematics and serves as the foundation of computer science [9].

3 Era of Scientific Management (1881-1913)

3.1 Time Studies and Scientific Management Principles

1881 Frederick W. Taylor, an American engineer and management consultant often called "the father of scientific management", introduced his important method of time study in 1881 while working as general foreman at the Midvale Steel Company. He was interested in finding the answers to two core questions: "What is the most effective method for completing a job?" and "What constitutes a fair amount of work for a day?" Taylor's practical applications were shown at the Bethlehem Steel Company (1898), where, through careful observation and experimentation, he determined that a steelworker's shovel should hold exactly 21.5 pounds to maximize the daily work when shoveling ore. His scientific management principles stressed the structured analysis and optimization of work processes to maximize productivity, and as a consultant, he used these methods in various industries.

This focus on efficiency and optimization aligns with the core principles of operations research, which aims to discover the most effective means of managing complex systems. Taylor's method of using scientific techniques to analyze and improve operations are fundamental to several modern OR methods, especially those focused on process optimization and efficiency. The importance of using a systematic approach to analyze and improve operational processes continues to be a core theme in OR, originating from these early concepts of scientific management [4, 58].

3.2 Gilbreth's Motion Studies

1885 Contemporaneous with Frederick W. Taylor's time studies was the development of motion studies by Frank B. Gilbreth. Their work is often associated, yet there was a substantial philosophical difference: Taylor was concerned primarily with reducing process times; Gilbreth sought to make processes more efficient by reducing the motions involved. He analyzed the labor process in a scientific context and conducted his first motion study on bricklaying. His goal was to understand the working habits of industrial workers to find a way to increase production and eliminate waste caused by poor misdirected and inefficient motions. In his study of the motion of bricklaying, Frank Gilbreth reduced the motions of a construction worker in laying brick from 18 to 5, which resulted in an increase in the rate of bricklaying from 120 to 350 per hour. [4].

3.3 Gantt Charts and Project Scheduling Methods

1900 Henry L. Gantt, an associate of Frederick Taylor, devised a method of project planning in which managers could represent the interconnected steps of a project with a series of bars on a chart. This tool allowed the user to show the precedence relationships among the steps, indicate completion time, and track real performance. This chart today remains a basic management tool, particularly in the construction sector [27].

1909 Agner Krarup Erlang, a Danish mathematician and statistician, was exposed to the challenges of the telephone system. Erlang's first significant publication on modeling telephone traffic showed that incoming calls can be described by the Poisson distribution. In a paper from 1917, he calculated the Erlang loss formulas and set the foundations for modern queuing theory [22].

3.4 Early Inventory Management

1913 The widely known square-root formula, formula 1, for the optimal economic order quantity (EOQ), developed by Ford W. Harris is a cornerstone in inventory management. It is used to determine the optimal order size that minimizes total inventory cost. In the formula, S represents the setup cost for placing an order, D is the demand rate for the product, and h is the holding cost per unit [61].

$$\text{EOQ} = \sqrt{\frac{2SD}{h}} \tag{1}$$

4 Pre-World War II Developments (1915-1936)

4.1 Linear Equations and Combat Modeling

1914 Frederick Lanchester, a British engineer, created mathematical models that describe the relative strength of military forces, models that became fundamental to military operations research [22].

1915 E. Stiemke, an German mathematician, established the conditions for the existence of a positive solution to a set of linear equations, a significant contribution to linear algebra and its uses [22].

4.2 Advancement of Statistical Methods

1920 Ronald Aylmer Fisher, a British polymath today considered the father of modern statistics, proposed the maximum likelihood method as a general approach to point estimation, creating a main course of statistical reasoning. Fisher also introduced the term "likelihood" to distinguish it from probability [57].

4.3 Initial British Military Applications

1936 The British Air Ministry established the Bawdsey Manor Research Station to investigate how radar could be utilized to intercept enemy aircraft. The work performed by a team of RAF officers and civilian scientists at Biggin Hill airport is widely viewed as the primary applied research activity that initiated what would soon be called operational research. The term "operational research" is attributed to A. P. Rowe, superintendent of Bawdsey Research who, together with his team, examined the efficiency of the tracking techniques and operations rooms originated from the radar interception experiments at Biggin Hill [35].

5 World War II and Military Applications (1942-1947)

5.1 British Navy

1942 Patrick M. S. Blackett's memorandum titled *Scientists at the Operational Level* influenced the establishment of the US Navy's Anti-Submarine Warfare Operations Research Group (ASWORG). Blackett established an OR group at the British Admiralty. This group determined that larger merchant convoys were more effective in reducing losses due to U-boat attacks [35].

5.2 US Army

The first group of US operations research analysts went to England to work with the Eighth Bomber Command of the Army Air Forces. They improved bombing accuracy by suggesting that the most skilled bombardier be in the lead plane, that all bombs be released at the same time, and that the planes fly in a tight precision formation. This resulted in a substantial increase in the number of bombs hitting the target [18, 22].

The US Navy's Antisubmarine Warfare Operations Research Group (ASWORG) examined the threat of German U-boats in the Atlantic. Bernard O. Koopman's report *Search and Screening* described a probabilistic approach to the optimal distribution of search efforts [18, 22].

5.3 Establishment of Project RAND & Project SCOOP

1945 The US government established Project RAND (Research and Development) to develop mainly economic analysis and research for its armed forces in the areas of military strategy, the health industry, aerospace, universities and others. Today, more than 1,700 researchers are currently involved in the project for military planning and other government issues [25].

1947 A US Air Force research group, based at the Pentagon and led by Marshall K. Wood and George B. Dantzig, focused on creating solutions for Air Force requirements scheduling. This included establishing time-based material requirements for war plans. It was at Project SCOOP (Scientific Computation of Optimal Programs) that Dantzig first stated the mathematical form of the general linear program. Along with Wood, he established the related mathematical and economic theories of program scheduling. Dantzig also invented the simplex method for solving such problems here [25].

5.4 Solving LP Problems

George B. Dantzig introduced the simplex method as a procedure for solving linear programming (LP) problems. The simplex method solves linear programming problems through a step-by-step algebraic approach. It progresses by identifying increasingly better solutions at the corners of the feasible region. The method starts by finding a basic feasible solution and adding slack variables to transform inequality constraints into equations. Next, it checks

if the current solution can be improved. If improvement is possible, the method identifies which non-basic variable should enter the solution to maximize improvement and which existing basic variable should leave. The equations are then recalculated using Gaussian elimination to reflect these changes. This cycle is repeated until the optimal solution is reached.
Although being an algebraic process, it can be visualized as geometric movements within the solution space. The method is computationally efficient and widely implemented in software to handle large-scale problems. While other approaches, such as interior-point method, have since emerged, the simplex method remains a cornerstone in LP problem-solving [15].

5.5 The First Definition of OR

Charles Kittel, an American physicist, defined OR as "a scientific method of giving executive departments a quantitative basis for decisions." This definition was later altered by Charles Goodeve, a Canadian chemist, as: "Operations research is a scientific method of giving executive departments a quantitative basis for decisions regarding the operations under their control" [22].

6 Postwar Evolution (1950-1970)

6.1 Early developments of OR

1950 W. Edwards Deming, an American economist, consulted on sampling techniques in Tokyo and to the Japanese Union of Scientists and Engineers (JUSE). He was recognized for his contributions to improving manufacturing in the United States and Japan during World War II; in particular, he advised Tokyo and the Japan Union of Scientists and Engineers (JUSE) on sampling techniques. Deming's method of statistical quality control was widely adopted by Japanese companies and played an important role in the recovery of the Japanese economy [22].

The first academic OR journal, the *Operational Research Quarterly*, was published in March 1950 by the British OR Club. The journal name was changed to *Journal of the Operational Research Society* in 1978 [22].

John F. Nash extended von Neumann's minimax theorem to show that every finite, n-person, general sum game has at least one equilibrium outcome in mixed strategies. This concept is known as the Nash equilibrium. Nash was awarded the Nobel Prize in economics in 1994 for this work [45].

Dynamic programming, developed by Richard Bellman, is an optimization technique for multi-stage decision problems based on the principle of optimality [7].

Charles W. Thornwaite was the first to apply OR to agriculture. He designed a planting schedule based on a crop's climatic calendar to manage the harvesting of peas [59].

1951 Economists and mathematicians used linear programming and related mathematical and computational methods to optimize the blending of aviation gasoline. This approach was successfully implemented at the Gulf Oil Company's Philadelphia refinery. These methods are now employed worldwide in petroleum refineries [12].

6.2 Linear Programming and Early Computational Methods

The first computer-based simplex algorithm was programmed for the National Bureau of Standards' SEAC digital computer. It was used to solve a US Air Force programming problem with 48 equations and 71 variables in 18 hours [22].

1952 Researchers at Project SCOOP and the RAND Corporation independently explored linear programming problems in which the objective function coefficients or the right-hand side values are linear functions of a parameter, developing basic modifications to the simplex method to solve them [15].

6.3 Combinatorial Optimization

1954 George B. Dantzig, D. Ray Fulkerson, and Selmer M. Johnson demonstrated the effectiveness of cutting planes by solving a traveling salesman problem in 49 cities, consisting in finding the shortest route that a traveling salesman must follow to visit all cities and return to the starting city. This work established the importance of cutting plans for integer programs [15].

Selmer M. Johnson, an American mathematician and researcher at the RAND Corporation, developed an algorithm for sequencing n jobs on a two-machine flow-shop to minimize the maximum flow time, it was called the Johnson's algorithm [22].

6.4 Project Management and Expert Systems

1957 OR approaches to the management of complex and dynamic projects came into existence. These included the Program Evaluation and Review Technique (PERT), Critical Path Method (CPM), and the Metra Potential Method (MPM), which have been used globally in project management, notably in the construction industry [25].

1965 Early specific expert systems made use of stored knowledge and inference engines to provide advice on difficult problems. The DENDRAL system evaluated the overall structure of complex organic molecules based on chemical data using graph theory algorithms. This project also explored rule-based programs using if-then rules, which led to the creation of MYCIN, an expert system that could suggest the cause of a blood infection and appropriate antibiotics [5].

6.5 Decision Analysis

1966 Initial work in Decision Analysis included a study of how oil wildcatters made decisions, as well as the formalization of decision analysis as a research program. The term "decision analysis" was adopted by Ronald Howard at Stanford for his research program. Howard's 1966 article was the first publication to use the term "decision analysis" and describe its application. This work was followed by Peter C. Fishburn's paper, which presented a detailed discussion of the development of decision theory along with a list of related references. Consequently, decision analysis grew from practical uses, academic course development, and formal publications [31].

7 Modern Developments (1971-1990)

7.1 Multi-Criteria Decision Making

1970 Multi-criteria decision making (MCDM), also known as multi-criteria decision analysis (MCDA), addresses the problem of selecting the "best" alternative from a set of options based on how each alternative performs against a collection of criteria or goals. This research area has been a major focus of operations research and has roots in areas such as utility theory and vector maximization [22].

7.2 Complexity Theory and Approximation Algorithms

1971 The idea of NP-completeness is a key concept in complexity theory based on work by Stephen A. Cook and Richard M. Karp. The NP class consists of problems for which a proposed solution can be quickly verified, even if finding the solution may be computationally difficult. A problem is considered NP-complete if it is among the most difficult problems in NP, which means that if a polynomial-time algorithm were found for any NP-complete problem, it would imply that every problem in NP could also be solved in polynomial time. Karp's 1972 paper showed that a variety of combinatorial optimization problems are NP-complete, including the traveling salesman problem, establishing the great importance of this concept. This work offered a way to understand why certain problems are so difficult to solve and led to a focus on finding approximation algorithms as opposed to exact solutions for NP-complete problems [14, 34].

7.3 Queuing and Probabilistic Models

1974 The hypercube queuing model, created by Richard C. Larson, is used to address operational and planning issues related to police dispatching and emergency response. The model considers the fact that rapid response of emergency services, such as police cars and ambulances, depends on their location and the number deployed. It assists in the design of patrol areas, the placement of emergency service facilities, and the evaluation of dispatch policies. Building on research from the President's Crime Commission Science and Technology Task Force (1966-1967), the hypercube queuing model was put into use by several police departments. The model was developed to help address operational and planning issues relating to police dispatching and emergency response [25].

7.4 Heuristic Algorithms and Cognitive Biases

1975 Richard M. Karp thought that a probabilistic approach was the best way to understand why combinatorial heuristic algorithms worked effectively in practice. The initial result of this research was Karp's partitioning algorithm for the traveling salesman problem in a plane. This work had the effect of launching probabilistic analysis of algorithms as a research area. Another success for this approach was the later work by others on the probabilistic analysis of the simplex method. Karp received the 1977 Lanchester Prize for his traveling salesman paper [34].

Genetic algorithms are heuristic methods that mimic natural selection, where a random search advances the solutions. John H. Holland first developed these methods in 1962 while researching the evolution of complex adaptive systems. His computational research culminated in his book Adaptation in Natural and Artificial Systems. The use of genetic algorithms in combinatorial optimization has grown since the mid-1980s. To use this method, solutions are represented as binary strings, analogous to chromosomes. A population of solutions then develops through generations, with each new generation created in three stages: evaluation of the fitness of each solution, selection of solutions to create offspring based on their fitness values, and creation of new solutions through mutation or crossover. The evolutionary principle of "survival of the fittest" leads to high-quality solutions for the original optimization problem [25, 28].

1979 Daniel Kahneman and Amos Tversky, two pioneering psychologists today well-known for their groundbreaking work in cognitive biases and decision-making, began exploring the psychology of human judgment in the early 1970s, focusing on deviations from rationality and on judgment heuristics. These heuristics are "rules of thumb" that people use to resolve difficult judgment problems when they lack the cognitive mechanisms to quickly solve the problem accurately. Prospect theory, a result of their collaboration, explains deviations of decision-makers from standard normative expected utility theory: it suggests that people generally do not monitor a prospect's impact on their final position or overall wealth, but rather they evaluate results in terms of gains or losses relative to a point of reference. People are also very sensitive to how choices are presented or 'framed'. Prospect theory recognizes the asymmetry between gains and losses, where the pain of a loss usually outweighs the pleasure of an equal gain. The Kahneman-Tversky research on rationality has influenced research in medicine, law, public policy, international relations, decision analysis, and economics [33].

7.5 Optimization and Learning Techniques

1980 Constraint programming, also called constraint logic programming, originated in computer science and artificial intelligence. It has been shown to be effective for solving optimization problems, specifically those that involve sequencing and scheduling, and in general for problems with combinatorial structure, such as integer programming. Constraint programming involves defining the problem within a programming language and using logic-based methods to reduce the solution space in the search for an optimal solution. Constraint programming techniques have been used in conjunction with OR techniques to increase effectiveness. Constraint programming, in its initial design, aims to find a feasible solution that satisfies all stated constraints, such as job release dates and due dates, and it might not necessarily minimize the objective function. However, it is possible to include a constraint programming technique within a framework intended to minimize any due-date-related objective function. Optimization Programming Language (OPL) was created to address optimization problems through a combination of constraint programming techniques and mathematical programming techniques. Constraint programming can be traced back to constraint satisfaction problems explored in the 1970s [25].

1984 Neural networks were initially developed as a method to model the connectivity within the mammalian brain, in particular the connections between neurons. John J. Hopfield suggested that a network of nodes with random and symmetrical connections is similar to a magnetic material known as spin glass, which can store different patterns of spins. Hopfield used Donald O. Hebb's synaptic weight modification to stabilize the network activity and arrive at a stable configuration, resulting in what is known as the Hopfield network. In operations research, the design of a neural network algorithm involves choosing a network architecture and a training or learning process that determines and updates the weights on the connections between nodes to train the network to identify good solutions. To solve optimization problems, the weights are adjusted until a stable state is reached, which corresponds to a local minimum of the objective function. Hopfield and David W. Tank created a neural network for the traveling salesman problem, with the neurons representing whether a city should be visited at a specific point in the tour. In scheduling problems, neural networks can be applied to determine which machine a job should be assigned to upon each new release, where the network is trained to identify the most suitable machine. The knowledge of the network is stored in the weights, and learning algorithms are used to adjust these weights to achieve appropriate responses. One such method is the backpropagation learning algorithm [64].

7.6 Supply Chain and Financial Applications

1989 Supply chain management aims to integrate these components to ensure that products are available in the correct quantities at the right locations. This minimizes system-wide costs while meeting service-level requirements. Supply chain management addresses the flow of goods from suppliers to end-customer locations. This field came into existence through the integration of logistics, operations, inventory, and distribution management. A key contribution was the work done at Hewlett-Packard (HP), in collaboration with professor Hau Lee and colleagues at Stanford University, which was initially focused on inventory reduction for HP personal computers and printers. Later stages led to important supply chain strategies such as design for localization and postponement (delayed product differentiation) [22].

1990 Operations research has a long history of applying its techniques to financial problems, starting with early work in portfolio analysis. By 1990, the importance of Operations Research (OR) in solving complex problems in financial engineering was recognized. OR techniques were increasingly applied to tasks such as pricing derivatives and developing trading strategies, primarily utilizing mathematical programming methods and Monte Carlo simulations. Other methods, such as game theory, network analysis, and decision trees, have also been used in addressing financial issues [25].

8 Impact and Legacy

8.1 Transition from Military to Civilian Applications

Operations research has deep historical origins in various scientific studies, but it became particularly prominent during World War II. The British and US military experienced critical difficulties in distributing limited resources among military operations. This situation led them to gather teams of scientists to use scientific methods to address strategic and tactical challenges [22].

Following the end of the war, the success of OR in military environments led to its adoption in civilian areas. As businesses faced increasing complexity and specialization during the post-war industrial boom, former military OR consultants identified similar organizational challenges in civilian contexts. By the early 1950s, OR had spread to a variety of corporate, industrial, and governmental organizations [22].

8.2 Role of Computer Technology

Two key factors accelerated the development of OR during this time. First, substantial technical progress was made in the field, such as the development of the simplex method for linear programming by George Dantzig in 1947. By the late 1950s, many of the core tools of OR, including dynamic programming, queuing theory, and inventory theory, were firmly established. Second, the computer revolution substantially increased the practical application of OR. The creation of electronic digital computers, and later personal computers with specialized software, made the complex calculations of OR practical and available to a much larger group of users [15, 22].

8.3 Broad Application Across Industries

Regarding the nature of Operations Research, it fundamentally involves studying and coordinating operations within organizations. Its range of applications is remarkably wide, with uses in sectors such as manufacturing, transportation, construction, telecommunications, financial planning, healthcare, military operations, and public services. OR employs a scientific approach, starting with careful observation and data collection, followed by the construction of mathematical models that represent the key features of real-world problems. This systematic and research-driven methodology explains why OR is also sometimes referred to as management science, reflecting its use of scientific principles to address organizational challenges [22].

9 Algorithms and Methods

The discipline of Operations Research has progressed significantly due to the development and refinement of important methodologies and algorithms. These advancements have provided OR with the essential tools to tackle increasingly intricate problems across many different applications. The introduction of linear programming and the simplex method, the creation of project management techniques like CPM and PERT, the framework of queuing theory, the instruments of decision analysis and stochastic modeling, and the use of simulation have enabled operations researchers to solve complex problems across various industries and applications. These advancements continue to progress, making OR an essential discipline for improving decision-making processes and improving efficiency in a diverse range of systems and operations [43].

This section explores some of the most important methodological and algorithmic breakthroughs that have shaped OR, examining practical uses, and importance in detail.

9.1 Origins and Formulation

Dantzig, during his time at Project SCOOP, articulated the generalized mathematical representation of linear programs. This formulation enabled the representation of various optimization challenges in a standardized format. The fundamental concept behind linear programming involves finding the most favorable outcome (like maximizing profit or minimizing costs), given a defined set of limitations on available resources [15].

9.1.1 Linear Programming

One of the most revolutionary advancements in OR is linear programming (LP). This mathematical technique provides a structured approach for the optimization of a linear objective function, bound by linear equality and inequality constraints. The importance of linear programming lies in its ability to model a vast spectrum of real-world problems, including resource allocation, production scheduling, and logistics for transportation. The formalization of linear programming and the development of the simplex method are credited to George B. Dantzig, whose work in 1947 with the U.S. Air Force's Project SCOOP (Scientific Computation of Optimal Programs) was fundamental [15].

9.1.2 The Simplex Method

Dantzig also created the simplex method, a core algorithm designed to solve linear programming problems. The simplex method operates by iteratively progressing from one viable solution to another, enhancing the objective function at each step, until the optimum solution is discovered. This algorithm is remarkably efficient and capable of handling problems with numerous variables and constraints. Dantzig's work on the simplex algorithm, the transportation simplex algorithm, and the relationship between linear programming and two-person zero-sum games were formally documented in *Activity Analysis of Production and Allocation* in 1951 [15].

9.1.3 Early Applications

One of the initial and most significant applications of linear programming was in the optimization of aviation gasoline blending during the early 1950s. By using linear programming techniques at the Gulf Oil Company's Philadelphia Refinery, economists and mathematicians were able to pinpoint the most cost-effective way to combine various gasoline components to meet specific performance criteria. Currently, such methodologies and their extensions are employed for managing and operating the world's petroleum refineries [31].

In the 1950s, linear programming was also applied to transportation problems; it became a tool for optimizing resource allocation and production planning. Its core strength lies in handling situations with limited resources in various industries, particularly in determining the most efficient ways to transport products and schedule production activities [22].

9.2 Project Management Techniques

The creation of project management techniques has furnished OR with essential tools for planning, scheduling, and overseeing complex projects, thus improving efficiency and lowering costs. CPM and PERT appeared in the late 1950s and early 1960s, they have become essential in contemporary project management and are used across numerous industries [22, 43].

These techniques are vital for managing projects that are substantial in size and complexity, especially those with interconnected tasks. They have seen widespread adoption in many sectors, such as construction, aerospace, and manufacturing. These methodologies have assisted in:

- Enhancing project planning by highlighting critical activities and potential bottlenecks project managers can allocate resources effectively and optimize schedules.
- Reducing costs by minimizing delays and inefficiencies project management techniques contribute to the reduction of total costs.
- Assisting managers in monitoring project progress, identifying potential issues and implementing corrective measures.

9.2.1 Examples of Project Management Techniques

CPM

CPM is a deterministic project management technique that concentrates on identifying the critical path within a project network. The critical path constitutes the sequence of activities that dictates the minimum time possible to complete the project. CPM makes the assumption that the durations of activities are known precisely and focuses on the trade-offs between time and cost. It provides a structured method for project scheduling by determining which activities have the greatest effect on the overall project timeline and, therefore, require the most managerial attention [43, 63].

PERT

Similar to CPM, PERT is also employed for project scheduling; however, it is mainly utilized for projects in which activity durations are uncertain. PERT was initially created to plan research and development activities, specifically for the Polaris missile project, and it proved to be highly effective, completing the project two years ahead of schedule. PERT utilizes three estimates for each activity: optimistic, most probable, and pessimistic. These estimates are used to compute expected activity durations and to assess the probability of accomplishing project milestones [43, 63].

MPM

The Metra Potential Method helps plan and organize projects by mapping out tasks as a network diagram. By analyzing how different activities connect and depend on each other, it calculates how long the entire project will take to complete. The method identifies the most time-intensive sequence of tasks (the critical path), which determines the minimum duration of the project. This visualization of task relationships helps project managers understand and optimize their workflow [43, 63].

These techniques will be further explored and analyzed later in the thesis.

9.3 Queuing Theory

Queuing theory is a branch of OR that is dedicated to the mathematical modeling and examination of waiting lines or queues. It analyzes the behavior of systems where customers or items arrive for service, wait in a queue, and then receive service. Queuing theory is essential for optimizing systems that encounter congestion and waiting times [22].

The investigation of queues involves modeling the arrival process (how customers arrive), the service process (how customers receive service), and the queue discipline (the order in which customers are served), therefore enhancing the customer service and decreasing waiting times. These components combine to generate a complicated interaction that queuing theory seeks to understand and optimize.

Queuing theory has diverse applications across various sectors, which include:

- telecommunications, analyzing and optimizing network performance;
- transportation, managing traffic flow and airport operations;
- manufacturing, optimizing production lines and inventory management.

Using stochastic processes, queuing theory models the random nature of arrivals and service times, giving a comprehensive understanding of queue dynamics [43].

9.3.1 Pollaczek–Khintchine Formula

A fundamental result within queuing theory, the Pollaczek–Khintchine formula provides a means to compute the average waiting time within a single-server queue with a generalized service time distribution. The formula 2 is crucial for examining systems in which service times vary and do not adhere to a simple exponential pattern.

$$W_q = \frac{(\lambda E(S))^2 + \lambda^2 \mathrm{Var}(S)}{2\lambda\,(1 - \lambda E(S))} \tag{2}$$

The Pollaczek-Khintchine formula is a fundamental equation in queueing theory. It determines how long customers typically wait in line in a single-server system where arrivals follow a Poisson process but service times can follow any distribution (known as an M/G/1 queue). The formula calculates the mean waiting time using three key factors: how frequently customers arrive (the arrival rate), how long it typically takes to serve each customer (mean service time), and how much the service times vary (variance). This equation is valuable because it can predict waiting times in many practical situations, regardless of how service times are distributed [22].

9.3.2 Queue Networks

Queueing networks describe systems with multiple connected service points, where customers flow between different service stations. Unlike single-queue systems, customers in these networks may follow various paths through multiple service facilities and analyzing these networks helps determine important metrics like total system wait times and customer counts.

A fundamental principle in studying these networks is the equivalence property, which states that when a queue has Poisson arrivals and exponential service times, its output also follows a Poisson distribution [25, 32].

9.4 Decision Analysis

Decision analysis offers a structured methodology for making decisions under uncertain circumstances. It facilitates the assessment of different decision alternatives and the selection of the best course of action when information is either incomplete or probabilistic. Decision analysis helps decision-makers grasp the potential risks and benefits that are tied to various options [31].

A noteworthy early application was C. Jack Grayson's 1962 dissertation on how oil wildcatters made decisions. This research showcased how decision-making processes can be analyzed and formalized. Ronald Howard officially adopted the name "Decision Analysis" for his research program at Stanford, thereby establishing it as a distinct discipline [31].

Decision analysis relies on: decision trees, graphical representations of decisions, chance occurrences, and potential outcomes; probability distributions, utilized to model the uncertainty in the possible results of events; and expected value, a weighted average of potential outcomes, used for the evaluation of various decision alternatives [31].

Decision analysis is applied in various sectors, including strategic planning, investment decisions, risk management, policy evaluation and healthcare, to drive treatment decisions and resource allocation [22].

9.5 Stochastic Modeling

Stochastic modeling involves the analysis of systems that include random or probabilistic components. This approach is critical in OR for understanding and managing systems with considerable uncertainty.

Stochastic modeling is used across several different sectors: in financial modeling, it helps analyze the unpredictable nature of stock prices and market risks; for inventory management, it is used to determine optimal stock levels while accounting for uncertain demand patterns; in risk management, where it helps evaluating and so reducing potential risks [17].

9.5.1 Markov Chains

Markov chains are a fundamental stochastic modeling tool, used to analyze systems that transition between states over a period. These models assume that the likelihood of moving to a new state relies solely on the current state and not the prior history of the system, making them analytically manageable.

Markov chains are employed in many different applications, such as random walks to model the movement of a random variable, Leontief input-output model analyzing interdependencies within an economy and occupational mobility studying transitions in occupational careers [17].

9.5.2 Optimal Stopping

Optimal stopping represents a branch of stochastic modeling that deals with the challenge of determining when to cease a series of observations to maximize an objective function. This is applicable in areas like finance, where deciding when to stop trading (or use an option) is crucial. Key results in optimal stopping theory are linked to the theory of martingales. Early research by Arrow, Blackwell, and Girshick, along with the efforts of Chow and Robbins, contributed significantly to this field [17].

9.6 Simulation

Simulation is a methodology that involves constructing a model of a real-world system and subsequently performing experiments with the model to gain insights. It serves as a valuable instrument for the analysis of complicated systems that are challenging or impossible to study using other methods. Simulation starts with establishing a representation of the system that accurately reflects its behavior along with all of its essential aspects; this model can be a computerized simulation that mimics the system being studied. Once the model is built, experiments are performed to investigate various scenarios and evaluate the effects of different policies or parameters. Simulation makes it possible to repeat experiments within a controlled environment without disrupting the real-world system. The data generated by the simulation experiments are thoroughly evaluated in order to draw relevant conclusions. This evaluation is the main aspect of simulation, which provides insights and guides improvements. Simulation is widely used in manufacturing to examine the performance of production lines and scheduling; in logistics to optimize supply chain operations and transportation networks; and in healthcare to model patient flow, resource allocation, and medical operations [30].

Simulation offers important advantages in the analysis of complex systems that might be difficult to reproduce using traditional mathematical approaches. Through simulation, analysts can identify system bottlenecks, evaluate the impact of various changes and experiment with different system configurations and policies. Perhaps most importantly, simulation provides a cost-effective way to test new strategies before implementing them, helping organizations avoid expensive mistakes and optimize operations [30].

10 The Systemic Approach

The systemic approach provides a high-level perspective, focusing on understanding a system in its entirety rather than treating each specific part individually. This methodology emphasizes the development of models that are valuable for making informed decisions. These models, while not necessarily capturing every minute detail, strive to be sufficient and realistic representations of the actual system being examined.

A system can be defined as a collection of interacting components that together form a cohesive whole, bounded by both spatial and temporal constraints and is influenced by its external environment. A model, on the other hand, is an abstract depiction of a system that involves approximations, employing logical or mathematical relationships. Therefore, a model details a system in terms of its elements (agents and entities) along with the connections between them and any external forces that may have an impact. Those adopting a systemic approach operate under the assumption that the objectives of the agents and the limitations on their behavior are known. Additionally, it's presumed that a method can be devised to encourage agents to act in a desired way to achieve these objectives and that the model's accuracy can be validated by comparing its outputs with real-world observations [27].

10.1 Model Creation

Operations research can employ a methodical approach to solving problems, which includes creating and using models to analyze complex situations. This strategy allows decision-makers to gain valuable insights, assess potential alternatives, and implement effective solutions. The modeling process within OR can be divided into a number of steps, each one crucial in determining the overall success of a specific study. These steps, while not always strictly sequential and can involve some iteration, provide a structure for performing OR studies.

The development of an operations research model usually entails the following phases: Problem Definition, Model Formulation, Model Solution, Validation and Implementation [27].

10.2 Problem Definition

The initial step in the OR modeling process involves clearly and concisely defining the problem at hand. In practical situations, problems often lack clarity and precision, and this requires the OR team to conduct a thorough examination of the relevant system.

The OR team needs to acquire a comprehensive understanding of the system being studied, which includes its various components, the ways they interact with one another, and the overall environment in which they function. This may involve collecting data, watching processes in action, and conducting interviews with various stakeholders. The system as well as the model are defined, taking into consideration the goals of the decision-maker, the involved entities, the decision-making options, the system's current state, its parameters, external influences, and any constraints [27].

Ensuring that the model is clear, that it accurately reflects the real problem, and that it's free of logical errors or contradictions, is of great importance.

- The identification of objectives is a key aspect that involves clearly articulating what the organization is trying to accomplish. These objectives could involve profit maximization, cost reduction, waste minimization, productivity increase, efficiency improvement, or meeting customer demand. These goals need to be Specific, Measurable, Achievable, Relevant, and Time-bound, with a single acronym: SMART.
- The determination of constraints, in other words the identification of limitations and restrictions that affect the decision-making process, is crucial to the problem definition. These might be resource restrictions, such as limitations of labor, materials, equipment, or budget, as well as capacity restrictions, regulatory rules, or physical limitations. Understanding these constraints is essential for creating a practical model.
- Identification of interrelationships is also a crucial step that involves considering how the system in question relates to other

areas of the organization. It's vital to be aware of how choices in one area can impact others. Gathering relevant data is also part of the problem definition phase, and it involves identifying the information needed to build and solve the model.

- Upon understanding the entities involved, the relevant data must be gathered or synthesized. This data can come from many sources, and its collection and management can be expensive, requiring processes such as data ingestion, data fusion, data cleaning, data augmentation, and data validation. This data may have inconsistencies, errors, or missing information; therefore, a data exploration and cleaning phase is usually recommended. Collecting and reporting unnecessary data can lead to wasted resources, such as time and money, and might negatively impact the validity of the model's results.

Data collection and model building can often happen iteratively and in parallel, meaning that one could start by creating an initial model with limited or synthetic data, and then collecting real-world data to refine it; or perhaps while changing and improving the model simulating different possible scenarios, adjustments on the model or the data may be required, creating an ongoing cycle of feedback where each influences the other.

A properly defined problem statement is the base for the overall success of an OR study. It offers the essential direction for all subsequent steps. Consider, for example, the problem of improving the efficiency of a production line. The problem definition phase would encompass identifying key performance indicators, bottlenecks, available resources, and any particular requirements of the products being produced.

10.3 Model Formulation

Once the problem has been clearly defined, the next step is to develop a mathematical model that represents the core of the problem.

This model acts as a simplified version of the actual system, capturing the significant connections between variables and parameters.

The formulation of the model entails the following critical aspects.

- The selection of decision variables involves identifying which are the variables that can be modified and adjusted to achieve the objective; the model will determine the final value of these variables. For example in a production line, if the objective is to reduce the waste of resources, the decision variables could be the quantities of different products to produce, or the distribution of resources among the various stages of production.
- The definition of the objective function involves formulating a mathematical expression that describes what needs to be optimized, either minimized or maximized; this function should be written in terms of the decision variables. For example, in a production line the objective may be to minimize the total operating time of the machines or the processing of a job to be finished as early as possible.
- The formulation of constraints involves expressing the limitations and restrictions found during the problem definition phase as mathematical equations or inequalities; these constraints will limit the possible values of the decision variables, ensuring that the solutions found are feasible. For example, in a production line constraints may be related to the availability of raw materials, machine capacity and the number of available workers.
- The model type selection involves choosing the appropriate model type based on the nature of the problem, such as linear programming, integer programming, network models, dynamic programming, or simulation. The chosen model will depend on the complexity of the relationships between the variables, and the nature of the objectives and constraints. In the context of production line optimization, this might involve linear programming for resource allocation, integer programming for discrete decisions like machine setups, or simulation models for analyzing random variations in the system.

A well-formulated model should be clear, comprehensive, and easy to work with. It needs to accurately capture the essential aspects of the problem and make it possible to conduct an analysis using available mathematical tools. Mathematical models provide advantages over verbal descriptions by being more concise and easier to understand, making it easier to identify cause-effect relationships, and considering all interrelationships at the same time [18, 27].

10.4 Model Solution

Once the mathematical model has been developed, the next step involves creating a procedure to derive solutions from the model. This involves multiple phases:

- The selection of solution technique involves choosing a suitable method for solving the model. For instance, the simplex method is used for solving linear programming problems, while branch-and-bound techniques are suitable for integer programming problems. For some more complicated problems, heuristic and meta-heuristic approaches might be more appropriate, especially when finding a global optimum is computationally infeasible.
- The model might be solved by using specialized software that provides the required algorithmic tools.The computer-based procedure development involves implementing the chosen solution technique using software. This could mean developing custom applications or using commercially available software to solve the specific problem.
- The finding of optimal or near-optimal solutions involves using the software to find a solution that maximizes or minimizes the objective function, all while respecting all the constraints.

The solution process needs to be both efficient and effective, providing valuable information about the problem, as well as useful answers to the decision maker.

10.5 Implementation and Validation

Once the model has been solved and a solution has been obtained, the last steps consist of validating the model and implementing the solution. This includes:

- Model testing, to verify the model and to refine it as needed. Validation involves confirming if the model accurately represents the real-world system, as well as whether the solution is practical. Model testing includes examining the logic of the model, as well as its mathematical, predictive, and operational validity. Sensitivity analysis plays a key role in evaluating the operational validity of a model, and is used to check how the optimal solution changes when some parameters change.
- Model refinement, based on the validation results, involves refining the model to increase its accuracy. This could involve revising the model's assumptions, changing the objective function or constraints, or adding other variables.
- The preparation for ongoing application involves developing procedures to use the model as a decision-making tool on a continuous basis. The model should be easy to access and use by managers and other stakeholders.
- Model implementation involves putting the model into use as prescribed by management and to monitor its performance. The implementation phase is where the actual benefits of the study come into practice. The OR team should be involved in this phase to make sure that the model is correctly applied and that any flaws in the solution are corrected, and its results are translated into practical procedures. Implementation also involves making sure that all personnel have been trained and that the new system is properly managed and maintained. Model implementation should be seen as the beginning of continuous improvement, as the system and its environment may change.

In summary, the process of model creation in operations research is a systematic approach that includes problem definition, mathematical model formulation, development of a solution procedure, and the validation and implementation of the solution. This structured approach ensures that the models developed are not only robust and reliable, but that they are also directly aligned with the organization's specific objectives and needs [18, 27].

10.6 Computational Complexity

Computational complexity is crucial in operations research, specifically in areas such as production line optimization where NP-hard problems are common. By understanding the difference between P and NP problems, and recognizing NP-hard problems, one can choose the correct tools and techniques to find effective solutions. Having awareness of computational complexity provides the user the most basic tool to choose whether using approximation algorithms, or heuristics and metaheuristics.

Computational complexity is a field that measures the computational resources, like time and memory, required to solve a problem using a computer algorithm. It's a way to categorize problems based on their difficulty, and so it is helpful when choosing the appropriate solution method and the applicable techniques. This section provides an overview of computational complexity [21, 60].

10.6.1 P vs. NP

A common aspect of categorizing problems according to computational complexity involves making a distinction between P (polynomial time) and NP (nondeterministic polynomial time) problems.

P problems are those that a computer algorithm can address within polynomial time. Polynomial time means that the time required to solve the problem grows polynomially with the size of the input. Because these problems can be solved efficiently even with large inputs, they are regarded as "easy" or solvable. Examples of P problems include sorting lists of numbers and determining the shortest path in a graph.

NP problems are those where a solution can be verified in polynomial time, but finding a solution may not be possible within polynomial time. If you're given a possible solution to an NP problem, it can be quickly verified for correctness, but the actual process of finding the solution might take an exponentially increasing amount of time as the input size grows. Many real-world optimization problems fall into this category.

The P versus NP question: A core question in computer science is whether P is equal to NP. It is currently not known whether all NP problems can be solved in polynomial time, or if P=NP. Although a solution to this question would have significant implications for computer science, it's not a major concern for operations research practitioners, because NP-complete problems are considered unsolvable by current computing technology.

10.6.2 NP-Hard Problems

Within the NP class, there is the category of NP-hard problems, which are at least as challenging as the hardest problems within NP. If an NP-hard problem could be solved in polynomial time, then all problems within NP could also be solved in polynomial time, meaning that P would be equal to NP [60].

A problem is considered NP-complete if it is both in NP and NP-hard. These are the most difficult problems within NP. Many of the optimization problems that are often encountered in OR, including those related to production line optimization, are in the NP-complete or NP-hard categories [60].

The implications for OR are that if a problem is classified as NP-hard, it is unlikely that an algorithm exists that can find an optimal solution for all cases in a polynomial amount of time; if so, alternative approaches like approximation algorithms and heuristics are needed.

Some examples of NP-hard problems related to production line optimization are:

- **Job shop scheduling**, which involves determining the best sequence of operations for jobs on multiple machines to minimize makespan or other performance criteria, and which is NP-hard.
- **Project scheduling with resource constraints**, which involves scheduling project activities while considering limited resources in order to minimize project duration, is also an NP-hard problem.
- **Integer programming**, which involves finding the best solution to a mathematical program where some or all variables must be integers, is also NP-hard.
- **Knapsack problem**, which consists of maximizing the total value of items that can be carried in a backpack (with fixed capacity). Despite it is classified as NP-hard, efficient solutions can be found using techniques such as greedy approximation algorithms that offer practical approximations in polynomial time.

Recognizing that a problem is NP-hard helps avoid unnecessary attempts to find perfect solutions, and is a motivation to use alternative approaches. An exact algorithm should be used to solve the model when the problem is simple, stable, and an optimal solution is required with enough resources, because the exact algorithm enumerates all possible solutions. A heuristic algorithm is appropriate for complex and NP-hard problems, where the data is uncertain, or when speed is critical, because it either constructs a solution, or it enumerates a very small set of alternative solutions. Heuristics are also useful when exact methods are computationally impractical, and they provide reasonable solutions when an optimal solution is not necessary, or too costly to find. Hybrid methods, that combine exact and heuristic methods, are commonly used for practical problem solving [21, 60].

10.6.3 Approximation Algorithms

Because finding an optimal solution for NP-hard problems is computationally challenging, approximation algorithms try to find near-optimal solutions in a reasonable time. They are frequently used in OR to handle complicated optimization problems where finding the exact optimal solution is either impossible, or not worth the computational effort [38].

Approximation algorithms typically offer performance guarantees, specifying how close the approximate solution is to the optimal solution. For example, an algorithm might ensure that it will find a solution that is within 10% of the optimal value [38].

There are various approaches for designing approximation algorithms, such as greedy algorithms, which make locally optimal choices at each step; randomized algorithms, which use random decisions, and linear programming relaxations, which solve a simplified version of the problem [18].

10.6.4 Heuristics and Metaheuristics

Heuristics are problem-solving techniques that use rules or experience to find practical solutions, but without guaranteeing optimality. Metaheuristics are a class of higher-level heuristics that direct the search for suitable solutions [27].

Heuristic techniques are typically problem-specific and can be very effective for particular applications, such as production line optimization. Examples include dispatching rules, such as the shortest processing time rule, and other rules for sequencing and scheduling [27].

Metaheuristic techniques systematically explore the solution space to find better solutions, they are more general and less specific than heuristics [27].

Common metaheuristics used in OR include:

- Simulated annealing, which is inspired by the annealing process in metallurgy, explores the solution space by accepting worse solutions with decreasing probability, thereby avoiding local optima.
- Tabu search, a local search algorithm, prevents cycling by maintaining a list of recently visited solutions.
- Genetic algorithms, which are inspired by natural selection, maintain a population of solutions, and improve them through selection, crossover, and mutation.
- Ant colony optimization, which is inspired by the foraging behavior of ants, builds solutions iteratively, with each iteration adding an element to the solution probabilistically.

Advantages of heuristics and metaheuristics are that they can find reasonably good solutions for NP-hard problems in practical amounts of time. They are often easier to implement than approximation algorithms, but they don't have the same performance guarantees [27].

11 Production Line Optimization

11.1 Introduction to Production Planning

A crucial element of operations management is production planning, which involves allocating resources across a span of time to accomplish a set of tasks. This chapter will present the basic concepts of production planning, contrasting deterministic and stochastic models and outlining common objectives for planning. The goal of this chapter is to establish a basis for later discussions on particular planning models and algorithms in the following sections [27].

Based on the kind of data used, production planning models are generally divided into two categories: deterministic and stochastic. The selection between a deterministic and a stochastic model is dependent on the specific traits of the production environment and the degree of detail necessary. In practical terms, a combination of both types of models may be implemented to handle different aspects of production planning [18].

11.1.1 Deterministic Models

Deterministic models operate under the assumption that all parameters, such as processing durations, release dates, and due dates, are known with certainty. This indicates that there is no variation, and the data are regarded as fixed and predictable.

Deterministic models are commonly used when the number of jobs is finite and there are one or more objectives to minimize. Examples of deterministic scheduling problems include scheduling jobs on a single machine, across parallel machines, within job shops and in open shops.

Makespan, the total time to complete all jobs, is a frequent objective in many deterministic scheduling challenges. Manufacturing environments often utilize deterministic scheduling models [8, 16].

11.1.2 Stochastic Models

In stochastic models, one or more parameters are considered as random variables with a known probability distribution. This introduces both uncertainty and variability within the model. Random job processing times and machine breakdowns are examples of stochastic parameters.

Stochastic models aim to create schedules that perform well on average or are resilient to variability. Many real-world situations where uncertainty is unavoidable find relevance in stochastic models. Examples of stochastic scheduling problems include parallel machine, flow shop, job shop, and open shop models that involve random processing durations [16].

11.1.3 Planning Objectives

Planning objectives provide standards for evaluating the performance of a schedule. These objectives define what constitutes a good schedule, based on the priorities of the business. Common objectives in planning include:

- Makespan represents the total time required to complete all jobs within a given schedule and is a fundamental metric in both production and project scheduling. Minimizing makespan is often a key objective, as it directly correlates to maximizing productivity and resource utilization. This concept is applied in various environments, including single-machine, parallel machine, and job shop scheduling systems. In project scheduling, minimizing makespan similarly focuses on completing all tasks as efficiently as possible, ensuring swift project completion and enhanced operational performance.

- Tardiness measure the time by which a job is completed past its designated due date. Weights representing a different level of importance can be assigned to jobs in order to prioritize some over others. Total tardiness represents the cumulative delay across all jobs, and a common scheduling objective is to minimize both the total tardiness and the number of jobs completed past their due date.
- Earliness measures how early a job is completed before its due date and is often minimized to reduce inventory costs or meet just-in-time delivery requirements. In scheduling problems where both earliness and tardiness are considered, the objective functions are classified as non-regular, and are written as a balance of the two.
- Flow Time (or Completion Time) is the duration of time that a job spends in the system from start to finish. A primary goal is often to minimize the total completion times across all jobs.
- Throughput measures the rate at which a facility produces output. Maximizing throughput involves strategies such as ensuring bottlenecks remain active and reducing sequence-dependent setup times.

Other examples of objectives can include minimizing setup times, inventory costs, or resource utilization expenses.
The selection of planning objectives depends on the context and precise goals of the production setting. In real-world scenarios, a combination of several objectives, sometimes with conflicting priorities, is often involved [50, 51].

11.2 Single-Machine Scheduling

This part explores the specifics of single-machine scheduling, a basic area within production planning that deals with sequencing jobs on a singular resource. We will look into fundamental scheduling rules, examine the intricacies of earliness and tardiness, and evaluate the influence of sequence-dependent setup durations [36].

11.2.1 Basic Scheduling Rules (SPT, WSPT)

Single-machine scheduling problems focus on determining the optimal sequence for processing a series of jobs on a single machine. The primary goal is usually to minimize a specific objective function, such as the makespan (total time to complete all jobs), total tardiness (sum of late completion times), or total flow time (sum of completion times for all jobs). Basic scheduling rules, also known as dispatching rules, provide simple heuristics for ordering jobs and are frequently used as a starting point or for comparison with more complex methods. Two essential rules in single-machine scheduling are the Shortest Processing Time (SPT) rule and the Weighted Shortest Processing Time (WSPT) rule [50, 51].

The Shortest Processing Time (SPT) rule sequences jobs by ordering them in ascending order of their processing durations. This rule is known to minimize the total completion time, also referred to as total flow time, when all jobs are available at time zero. Additionally, the SPT rule minimizes the average flow time and the average number of jobs within the system. In single-machine environments, the SPT rule is optimal for minimizing the total completion time and is also optimal for minimizing the makespan. Therefore, the SPT rule offers an efficient and straightforward method for optimizing several performance measures in single-machine scheduling [50, 51].

The Weighted Shortest Processing Time (WSPT) rule sequences jobs by ascending order of the ratio of their processing duration to their weight ($\frac{p_j}{w_j}$). This rule extends the SPT rule by incorporating a weight for each job, reflecting its relative significance or cost. The WSPT rule is particularly effective when some jobs are more crucial than others, and the objective is to minimize the weighted total completion time, where the completion time of each job is multiplied by its associated weight. The WSPT rule is also known as the $c\mu$

rule or λw rule in some contexts, emphasizing the role of considering a job's weight in its prioritization [50, 51].

These basic rules, while simple, form the basis for more advanced techniques. The SPT rule is particularly valuable when all jobs hold equal importance, whereas the WSPT rule allows for prioritization based on job weights.

11.2.2 Earliness and Tardiness

In numerous real-world scheduling problems, jobs have due dates, and deviations from these due dates may result in expenses. Earliness assesses how much a job is finished prior to its due date, while tardiness measures how much a job is completed after its due date. Scheduling problems that factor in both earliness and tardiness are said to have non-regular objective functions [50, 51].

Completing jobs before their due dates, earliness, can lead to undesirable costs or penalties. In just-in-time systems, where inventory is minimized, storing completed jobs too early can increase holding costs. Therefore, earliness is often penalized as it represents a deviation from the desired delivery schedule. This highlights the importance of precisely timed operations, making completing jobs early as problematic as completing them late.

On the other hand, tardiness refers to the delay that arises for completing a work after its due date, can result in significant penalties: a loss of reputation with customers, potential breaches of contract, or other financial repercussions. Minimizing tardiness is therefore critical to meeting deadlines, meeting customer requirements, and maintaining a company's reputation. In many scheduling contexts, job delays can be weighted according to their importance, reflecting the different consequences of delaying customer orders.

Simultaneously minimizing both earliness and tardiness adds a layer of complexity to scheduling problems: these situations are common in environments where both earliness and tardiness are considered undesirable; addressing this combined objective usually involves minimizing a weighted sum of earliness and tardiness penalties, where different weights can be applied to each factor according to the production needs [50, 51].

Another common scheduling objective is to minimize the total number of jobs that are completed past their due dates, in other words to reduce the overall count of tardy jobs. This approach can be useful when a company aims to fulfill the majority of its commitments on time, rather than focusing only on reducing the total accumulated delay. This focus on the number of tardy jobs helps to maintain a better percentage of on-time deliveries, ensuring an overall greater customer satisfaction [50, 51].

11.2.3 Sequence-Dependent Setup Times

In numerous single-machine environments, the time required to set up the machine for a job is contingent upon the job that was processed immediately before it. These setup times are called sequence-dependent setup times. The presence of such setup times greatly complicates the scheduling procedure.

Sequence-dependent setup times significantly complicate scheduling problems. The order in which jobs are processed has a significant impact on both the makespan and the total completion time. Ignoring these setup times, which are delays incurred when preparing a machine or resource for a task, can lead to highly inefficient schedules that waste valuable time and resources. This complexity arises because the time needed for each setup is not constant, but varies with previous and subsequent jobs [37].

Examples of sequence-dependent setup times are common: in manufacturing, these setup times can be associated with activities like changing tools on a machine, cleaning a machine between the procession of different products or adjusting processes for a different material or color. The duration of these setups is not fixed, but it depends on the specific combination of sequenced jobs. For example, changing from a dark paint color to a light paint color may require a more extensive cleaning process than changing between similar colors.

Minimizing the makespan with tasks that have a sequence-dependent setup is a NP-hard problem. It's possible to solve these problems using approximation methods like simulated annealing, tabu search, genetic algorithms and ant colony optimization. These techniques don't guarantee finding the best schedule, but instead producing good-quality solutions and most importantly within a reasonable time [37].

A specific instance of sequence-dependent setup times occurs when jobs are categorized into different families. Within a family, jobs can be processed with minimal setup, but switching between families requires a significant setup duration. This scenario often leads to batch processing strategies, where jobs within the same family are processed consecutively to minimize the number of costly family transitions. These strategies group similar jobs together to reduce the overall time and resources spent on setups and maximize overall efficiency [37].

The addition of sequence-dependent setup times makes single-machine scheduling more difficult, necessitating specialized algorithms to find good solutions.

11.3 Parallel Machine Scheduling

This section explores the complexities of parallel machine scheduling, a frequent scenario in production settings where numerous machines can process jobs at the same time. We will examine different kinds of parallel machine environments, specifically identical machines, uniform machines, and discuss online scheduling [62].

11.3.1 Identical Machines

Parallel machine scheduling involves an environment with multiple machines that have the same processing speed, where any job can be processed on any machine. The core challenge in this setting is to assign jobs to specific machines and then sequence those jobs on their assigned machines to optimize a defined objective. Common objectives are the following.

The problem of minimizing makespan on identical parallel machines is NP-hard when all jobs are available at the start. A commonly employed heuristic for this purpose is the longest processing time (LPT) rule, which sequences jobs in descending order of their processing durations; instead when jobs arrive over time and complete information about all jobs is not known in advance, online scheduling techniques become necessary.

Another common objective is minimizing the total completion time, often referred to as total flow time, which is the sum of the completion times of all jobs. The shortest processing time (SPT) rule is optimal for minimizing total completion time if all jobs are of the same weight. When dealing with stochastic scheduling problems, involving exponential processing durations, the shortest expected processing time (SEPT) rule has been shown to be optimal for minimizing the total completion time. Sometimes, the total completion time and makespan are minimized simultaneously, presenting an additional scheduling challenge.

Finally, some parallel machine scheduling problems allow for preemptions, where the processing of a job on a machine can be interrupted and then resumed later, either on the same machine or a different one. These preemptions can sometimes lead to improved solutions and can be used to minimize makespan, for instance, through employing a longest remaining processing time rule [38, 39].

11.3.2 Uniform Machines

In a uniform parallel machine environment, each machine operates at its own unique processing speed, meaning the time it takes to process a job varies depending on the specific machine it is assigned to. Essentially, while all machines can process the same set of jobs, they do so at varying speeds. This setup represents a generalization of the simpler identical parallel machine environment, where all machines operate at the same speed.

Scheduling problems on uniform parallel machines are more difficult than those on identical machines. When scheduling on uniform parallel machines, the decision of which job to assign to which machine must consider the machine's individual processing speed. To solve these problems many methods have been studied: integer programming, which attempts to find the optimal solution through mathematical formulation; dispatching rules, which use simple heuristics to assign jobs; and local search algorithms, which iteratively improve solutions by exploring the neighborhood of existing solutions.

When the goal is make-span minimization, aiming to complete all jobs in the shortest possible time, the problem on uniform parallel machines is also NP-hard. Due to this computational difficulty, heuristics, often generalizations of the longest processing time (LPT) rule, are frequently employed to find near-optimal solutions. Furthermore, for minimizing makespan, online scheduling algorithms can be used when jobs arrive over time and full information is not known at the start, adapting to the changing environment dynamically.

Similar to the identical machine setting, minimizing total completion time remains a crucial objective when scheduling jobs on uniform parallel machines, particularly in scenarios where the jobs vary in importance. This objective aims to reduce the sum of the completion times of all jobs, giving greater weight to those completed earlier, and often requires different optimization strategies compared to makespan minimization [41, 42].

11.3.3 Online Scheduling

In online scheduling, jobs appear over a period of time, and the scheduler lacks comprehensive knowledge about all jobs in advance. This differs from offline scheduling, where all jobs are known in advance. Online scheduling is relevant in dynamic production environments that involve new jobs constantly appearing.

Online scheduling environments are characterized by the need to make decisions on the spot as new jobs arrive and their arrivals are not known in advance. This means that scheduling decisions must be made without complete knowledge of the entire workload, requiring solution algorithms that can adapt to the dynamically changing conditions. A competitive analysis evaluate the performance of the online algorithm comparing it to an optimal offline schedule which assumes full knowledge.
Online scheduling techniques are frequently applied to minimize the makespan on both identical and uniform parallel machine environments.

Various approaches are utilized to tackle the challenges of online scheduling. Dispatching rules, which apply simple heuristics to assign jobs to machines as they become available, are a common method for addressing the immediate decision-making required in online scenarios. List scheduling algorithms, which prioritize jobs according to a specific rule, are also widely used. These algorithms generate a list of jobs, and then assign them to available machines based on this prioritized list. A fundamental focus of online algorithms is achieving the best possible performance with the least amount of information about the future, requiring techniques that are robust, adaptable, and efficient under uncertainty [19, 52].

11.4 Job Shop Scheduling

This section delves into the complexities of job shop scheduling, a challenging area within production line optimization. Job shops are distinguished by flexible job routing, where each job could have its own unique series of operations that need to be performed on various machines. This great flexibility results in complex scheduling problems.

11.4.1 Disjunctive Programming

Disjunctive programming is a technique designed for modeling and solving job shop scheduling problems. This method is effective in managing the "either-or" constraints that arise when multiple jobs can be performed by the same machine. In the context of job shop scheduling, the order in which jobs are processed on each machine is not predetermined; therefore, the specific operations associated with the jobs must be sequenced for every machine, adding a layer of complexity. This requirement gives rise to disjunctive constraints enforcing that only one operation can be processed on a given machine at any particular time.

In disjunctive programming, binary variables are used to model the sequencing of operations: a variable is assigned a value of 1 if one task precedes another on the same machine and 0 otherwise, effectively encoding the precedence relationships between the two phases. The primary objective in job shop scheduling is typically to minimize the makespan, the total duration required to finish all jobs, and this involves determining the optimal order of operations while following both the disjunctive constraints, which dictate sequencing on shared resources, and the predefined precedence constraints between tasks.

Disjunctive programming formulates job shop problems as mixed-integer programs with disjunctive constraints. These formulations, while highly effective in representing the problem, often become complex, making it difficult to find optimal solutions for large problem instances. The disjunctive constraints impose a computational complexity making the problem NP-hard [54].

11.4.2 Shifting Bottleneck Heuristic

The shifting bottleneck heuristic is a decomposition technique specifically designed to tackle complex job shop scheduling problems. This heuristic operates by focusing on identifying and resolving bottlenecks within the system. A bottleneck machine is defined as a machine that restricts the overall productivity of the entire job shop, essentially limiting the rate at which jobs can be completed. The heuristic iteratively identifies the bottleneck machine by examining the workloads of all machines. In each iteration, the machine with the heaviest load, meaning the one that is the most utilized, is determined to be the current bottleneck.

Once a bottleneck machine is identified, the jobs that have operations on that specific machine are prioritized and sequenced first. Often, a single-machine scheduling algorithm, designed for the simplified problem of scheduling jobs on a single resource, is applied to determine the order of these operations. After the jobs on the bottleneck machine have been sequenced, the schedule for the other machines is then generated, taking into account the already established schedule for the bottleneck.

The shifting bottleneck heuristic follows an iterative refinement process, where bottlenecks are identified and sequenced one after another. Each time a machine is sequenced, the impact of this decision on the schedules of all other machines is considered, leading to a progressive improvement of the overall schedule. This iterative procedure is repeated until all machines in the job shop have been scheduled, resulting in a complete schedule that addresses the most constrained resources in the system first [2, 48].

11.4.3 Constraint Programming

Constraint programming (CP) is a declarative programming paradigm that is widely used to solve complex combinatorial problems, including job shop scheduling. Unlike procedural programming, which focuses on specifying the exact steps of a solution, constraint programming defines the relationships between variables through constraints. This approach provides a flexible and intuitive way to represent the diverse constraints present within a job shop environment.

In constraint programming, a job shop scheduling problem is modeled by defining variables, constraints and an objective function. The variables can represent quantities such as tasks start time and tasks end time, while constraints express the requirements of the problem. These constraints include the processing of each job, the capacity of each machine, and the precedence constraints between the different operations within each job.

A key characteristic of constraint programming is constraint propagation, a process which utilizes the defined constraints to reduce the domain of possible values for the variables. Constraint propagation involves maintaining a list of possible values for each variable and iteratively removing values that violate the constraints imposed. This process reduces the search space and speeds up the search for a feasible solution. Once constraint propagation has been completed, a search algorithm is utilized to further explore the remaining solution space and identify an optimal solution. Various search strategies, such as depth-first search and local search methods, may be used depending on the complexity of the problem. CP demonstrates notable advantages, including the ability to handle complex constraints and high flexibility in problem modeling, making it a powerful approach for modeling and solving job shop scheduling problems [53].

11.4.4 Local Search Methods

Local search methods are iterative improvement techniques that from an initial solution, repeatedly attempt to improve it by making small and incremental changes. In the context of job shop scheduling, local search is frequently used to discover effective schedules by exploring the neighborhood of the current solution, moving step by step toward better options.

The first step in implementing a local search algorithm is defining the neighborhood: the set of possible solutions adjacent to the current solution. In job shop scheduling, the neighborhood is typically defined by swapping the order of two operations on the same machine or by moving an operation from a point in time to another. The iterative improvement process involves evaluating solutions within the neighborhood and, when found, replacing the past best solution with the superior one. This process is repeated until the local optimum is reached, meaning no better solutions can be found in the neighborhood, or the current solution is satisfactory enough.

Several specific local search methods are often used:

- simulated annealing utilizes a probability function to accept less optimal solutions, particularly during the early stages of the search, to increase the chance of escaping local optima and finding better overall solutions;
- tabu search tracks the moves that have been made recently, prohibiting their repetition, which helps the algorithm to explore

different areas of the solution space and avoid getting trapped in cyclical patterns;

- genetic algorithms apply concepts from evolutionary biology, creating a pool of candidate solutions and improving the pool over time using techniques such as mutation and crossover;
- ant colony optimization employs the analogy of ants searching for food, allowing the algorithm to explore the solution space based on the artificial pheromone trails, representing the fitness of paths in the search space.

Local search methods are excellent at discovering good solutions relatively quickly, even for very large problems, and can be adapted to a wide range of optimization problems [1, 29, 47].

11.5 Project Scheduling

This section focuses on project scheduling, a crucial element of production line optimization, particularly when handling complex projects with many interdependent tasks. In contrast to job shop scheduling, project scheduling often assumes an unlimited quantity of machines but focuses on precedence constraints between activities. The primary objective is to minimize project completion time (makespan) while adhering to these constraints [3].

11.5.1 Critical Path Method (CPM)

The Critical Path Method (CPM) is a deterministic approach to project scheduling that provides a structured framework for managing project timelines. CPM's aim is to identify the critical path, which is the sequence of activities that determines the overall duration of the project. A network diagram is used to represent visually the project activities and their dependencies: activities are shown as either nodes or arcs, with arrows indicating the sequence in which they must be completed, providing a clear visualization of the project structure and helping identifying potential bottlenecks and critical areas.

The fundamental assumption of CPM is that activity durations are known with certainty. To each activity is assigned the specific time that it requires for completion, forming a deterministic structure. CPM uses a forward pass to calculate the earliest start time (EST) and earliest finish time (EFT) for every activity. An activity's EST is determined by the latest EFT of its predecessors, while the EFT is calculated by adding its duration to its EST. When the forward pass is completed for all activities from start to end of the workflow, a backward pass is performed to calculate the latest start time (LST) and latest finish time (LFT) for each activity. The LFT of an activity is determined by the earliest LST of its successors, while its LST is calculated by subtracting its duration from its LFT.

The critical path is defined as the sequence of activities that have zero slack, meaning their EST equals their LST and their EFT equals their LFT. The slack represents the amount of time an activity can be delayed without impacting the project's overall completion time, so activities that are not on the critical path have positive slack, indicating there is some flexibility in their scheduling, while activities on the critical path cannot be delayed without causing a delay to the entire project.
CPM is widely used to manage project scheduling effectively in construction, engineering, and other sectors that involve projects with well-defined structures and predictable activity durations [43, 63].

11.5.2 Program Evaluation and Review Technique (PERT)

The Program Evaluation and Review Technique (PERT) is a probabilistic approach to project scheduling, sharing similarities with the Critical Path Method (CPM) but differing significantly in the way it handles activity durations. Unlike CPM, which assumes deterministic activity durations, PERT accounts for the inherent uncertainty in these durations by using three time estimates for each activity: optimistic time (O), most likely time (M), and pessimistic time (P); these three estimates are then used to calculate an expected activity duration, formula 3, giving greater weight to the most likely time. In addition to expected duration, PERT also calculates the variance of activity duration, providing a measure of the uncertainty associated with each activity.

$$T_E = \frac{O + 4M + P}{6} \tag{3}$$

The expected project duration, under PERT, is calculated by summing the expected durations of the activities on the critical path. It's important to note that the critical path in PERT is identified based on these expected activity durations rather than actual durations which are unknown at the planning phase. This means the critical path may be different from the one observed in reality, particularly where activity times differ from their expectations.

A distinctive feature of PERT is its ability to estimate the probability of completing the project by a specific date. PERT uses the standard deviation of activity durations to offer a statistical estimate of the likelihood of finishing the project within the desired timeframe. This probability analysis offers insights into the risk and uncertainty associated with the project schedule, allowing managers to assess and plan for potential delays. That's the reason why PERT is particularly beneficial in projects with high levels of uncertainty, such as research and development, new product launches, and other projects where activity times are not accurately known, providing a more realistic solution that explicitly considers the variability in activity durations [43, 63].

11.5.3 Project Scheduling with Resource Constraints

Project scheduling with resource constraints addresses the complexities that arise when projects must be managed within the limitations of available resources. These resource constraints, such as manpower, equipment, and budget, add another layer of difficulty to traditional project scheduling, which primarily focuses on precedence relationships between activities. In many real-world projects, resources are not unlimited, and this limitation requires project activities to be scheduled in a way that ensures resource usage does not exceed defined limits. This introduces the necessity to consider not just the order in which tasks must be completed but also the availability of resources at any given time.

Project scheduling problems with resource constraints are NP-hard, so finding an optimal schedule can be computationally challenging, especially for large-scale projects. In this case, heuristic algorithms are often employed to find good, although not necessarily optimal, solutions. These heuristics prioritize activities based on factors such as the critical path, the amount of resources required and the earliest start times.
To reduce the risk of bottlenecks and shortages, firms try to minimize peaks and values in resource utilization, smoothing out efficiency and reducing risk. This is the case in manufacturing companies, where there are limits on the quantity of available workers, machines, resources, and it is necessary to plan an efficient schedule with time and resource allocation [3].

11.5.4 Time/Cost Trade-offs

Time/cost trade-offs are an important consideration in project management, acknowledging the relationship between the duration of a job and its costs. It is often the case that reducing the duration of a job, also said crashing an activity, is a common technique in project management that can lead to an increase in costs, and this relationship needs to be carefully analyzed. This is typically achieved by allocating more resources to an activity or modifying the way in which it is executed, for example by adding more workers or using faster equipment.

In some scenarios, the cost of crashing an activity is assumed to be linear, meaning that the cost increases at a constant rate for each unit of time that the activity duration is reduced. It's relatively easy to calculate and apply linear time/cost trade-offs, however, in many situations, the relationship between time and cost is nonlinear. Nonlinear costs more accurately capture the reality that the cost of crashing activities may not be constant and can become exponentially higher as the duration of activities are reduced further. Nonlinear time/cost trade-offs often lead to the need for nonlinear optimization models, which are more computationally challenging.

Trade-offs are evaluated to determine the best balance between a project's timeline and its total expenses. This analysis is critical when the rewards for accelerating the schedule is greater than the increased costs that arise by doing so, for example scenarios where there are financial penalties for delays or an early completion offers rewards, where it become important to avoid fines and it's possible to generate extra revenue [27].

11.6 Flexible Manufacturing Systems

This section explores Flexible Manufacturing Systems (FMS) that represents an important step towards automation and adaptability in manufacturing. In contrast to traditional production lines, FMS can handle a variety of products and production volumes.

The modeling and optimization of FMS are complex due to the inherent flexibility and multiple constraints that need to be considered. Real-world applications of FMS are in use in the automotive, aerospace, and electronics industries [6, 46, 49, 55].

11.6.1 System Characteristics

A Flexible Manufacturing System (FMS) is characterized by its inherent capacity to process a variety of different product types and dynamically adapt to shifts in demands. This level of adaptability is achieved through a sophisticated combination of several key components:

- partially manned or automated machines, which perform the actual manufacturing processes. These machines are of multiple types and therefore capable of performing different operations, allowing the FMS to handle a broad range of manufacturing tasks;
- material handling systems, such as automated guided vehicles (AGVs) or conveyors, are responsible to optimize the flow of material in the system and to guarantee an efficient movement of parts between machines;
- computerized control system, which manages all aspects of the system's operations: monitors production processes, adjusts operations in real-time and optimizes performances. This computerized control is the head of the FMS and enables it to quickly respond to any change in the production needs. [6, 55]

11.6.2 FMS Modeling

Modeling a Flexible Manufacturing System (FMS) requires careful consideration of the complex interactions between its various components, including the machines, material handling systems, and the overall workflow. Capturing these intricate relationships is critical for accurately predicting system behavior and optimizing its performance. To achieve this, various modeling techniques are employed. Mathematical programming models, such as mixed-integer linear programming (MILP), may be applied to formulate FMS problems and find optimal solutions by mathematically representing constraints and objectives.

Simulation models are used to analyze the behavior of the FMS under different scenarios. By replicating the different systems, simulation models allow researchers and engineers to test configurations, control strategies and production schedules. Queueing theory can be used to change the workflow of the system, analyzing waiting times at different machines and identifying potential bottlenecks. This analysis helps in understanding and optimizing the flow of parts within the FMS, ensuring a smoother and more efficient production.

The FMS's vital components, including machine capacities (which represent the limits on machine throughput), material handling constraints (which reflect restrictions on how materials can travel throughout the system), and process routes (which specify the precise order of operations needed to produce each product), are all taken into account by these various modeling methodologies. Models may offer a detailed description of the FMS by incorporating these elements, which promotes improved planning and management [6, 55].

11.6.3 Optimization Objectives

The objectives of Flexible Manufacturing System (FMS) optimization can vary significantly, depending on the specific goals and priorities of the production facility. These objectives often represent a trade-off between different aspects of system performance, requiring careful consideration and balancing.

One common objective is makespan minimization, which focuses on reducing the total time needed to complete all jobs. This aim is geared towards achieving the shortest possible production cycle.
Another key objective is throughput maximization, which seeks to increase the rate at which products are manufactured. This is often directly linked to identifying and addressing bottleneck machines, which limit the overall production output.

Another important objective in FMS optimization is minimizing work-in-progress (WIP) inventory: that is the amount of partially finished products within the system, reducing WIP can lower holding costs and improve overall efficiency.
Tardiness minimization is a critical objective, particularly in environments where meeting deadlines is essential, so it focuses on reducing the number of jobs completed after their due date, improving on-time delivery performance.

Finally, when there are sequence-dependent setup times within the FMS, setup time minimization becomes a relevant goal. These are setup times that change based on the sequence in which jobs are processed. Optimizing the order of jobs to minimize these setup durations can boost overall system productivity and decrease idle time. FMS optimization aims to improve the performance, efficiency, and flexibility of the manufacturing system by concentrating on these different goals [6, 55].

11.6.4 FMS Scheduling

Scheduling within a Flexible Manufacturing System (FMS) is considerably more complex than scheduling in traditional production lines. This increased complexity arises from the inherent flexibility of FMS and the numerous constraints that must be taken into account. The dynamic nature of FMS, with its multiple machines, diverse product types, and flexible material handling systems, creates a challenging scheduling environment. Therefore, effective scheduling strategies are crucial for realizing the full potential of an FMS.

Dispatching rules can be used to make decisions about which job should be processed next, these rules offer simple but effective ways to prioritize jobs based on specific criteria. For example the shortest processing time (SPT) rule prioritizes jobs with the shortest processing times, while the first-in-first-out (FIFO) rule processes jobs in the order they arrive.

Given the complex nature of the FMS environment, heuristic algorithms are frequently applied to find good solutions in a reasonable amount of time. Local search methods, such as simulated annealing or tabu search, are frequently employed to iteratively improve solutions by exploring the solution space. Constraint programming represents another approach that can be used to handle the intricate scheduling constraints present in an FMS. Additionally, shifting bottleneck heuristics, which focus on optimizing the performance of the most constrained resources, are used to discover good solutions in job shop environments, such as FMS with diverse routing needs [6, 46].

11.6.5 FMS Challenges

Several factors contribute to the complexity of scheduling within a Flexible Manufacturing System (FMS). Machine eligibility is one such constraint, where jobs may only be processed on a specific subset of the available machines, limiting the flexibility of job assignment. Material handling limitations also play a significant role, as the limited capacity and travel times of material handling systems can lead to bottlenecks and delays in the production process. Further complicating matters are sequence-dependent setup times, where the time required to set up a machine for a new job can vary based on the sequence of jobs processed, adding a layer of complexity to the scheduling decisions.

Routing flexibility, while offering benefits by allowing jobs to have alternative routes through the system, also introduces further scheduling challenges. The multiple possible paths a job might take increases the number of scheduling options and the complexity of finding the optimal routes and sequences. Limited buffer capacities between machines can also lead to blocking, when a machine is ready to process a job, but the next machine is not available and starvation, when a machine is ready but no jobs are available to be processed. These buffer constraints impact the overall flow of jobs within the FMS.

Finally, dynamic events, such as machine breakdowns or unexpected changes in demand, can occur frequently in an FMS and often require real-time rescheduling to mitigate their impact. These unexpected disruptions force the system to adapt and re-optimize schedules quickly, adding to the already complex nature of FMS scheduling [6].

11.6.6 Cyclic Scheduling

In specific Flexible Manufacturing System (FMS) environments, a cyclic scheduling approach may be adopted to manage production. Cyclic schedules are characterized by their repetition of a defined production pattern, with a fixed sequence of operations. This means that the system produces the same sequence of products in a recurring cycle, simplifying scheduling decisions once the cycle has been defined.

A key element in developing a cyclic schedule is the Minimum Part Set (MPS), which helps to determine the production ratios for different products within the cycle. The MPS specifies the number of each type of part that needs to be produced within each cycle, ensuring that production meets the required ratios. The cycle time represents the duration required to complete a single cycle of production. This is the time it takes to produce one complete set of parts defined by the MPS, which, once established, becomes the recurring unit of production [49].

Part II
A Study on Project RefAIne

This part delves into the methodology of constructing models for Project RefAIne, employing various techniques from the field of Operations Research (OR).
The second part of the thesis presents a description of the project itself, the formulation of the model and its solution.
The objective is to analyze the project using the principles of production scheduling and optimization that were discussed in earlier chapters.

12 Problem Description

As the name suggests, Project RefAIne provides an artificial intelligence integration that refines the production line of a major Italian multinational company, making the current management system of the workshops evolve in a more intelligent way, through a transition process towards a digital management system of the factory. More simply put, artificial intelligence will intervene directly on the production lines.

This section provides a description of the manufacturing environment in which Project RefAIne is being applied and the specific challenges that it addresses.
It is essential to have a clear understanding of the context to develop effective models, as emphasized in the systemic approach to operations research, so the production system, with its various components and constraints, is described to provide a solid foundation for the subsequent model formulation and implementation phases.

The project is still in its early stages, but already includes many of the elements analyzed in the thesis.

12.1 Glossary of Terms

The following terms are defined within the context of Project RefAIne:

Scheduling

A structured timetable that organizes activities in a specific timeframe and machine. It includes details such as start and end times, deadlines and sequences of operations, to ensure efficiency, coordination and most of all to meet objectives within the set constraints.

Work cell

A work cell is defined as a group of multiple machines, typically three or more, that are managed by a single operator. The grouping of multiple machines into work cells affects how workload is distributed and how operators manage their tasks.

Line

The plant utilizes five main assembly lines. These lines operate with a takt time, which is different from the cycle time used in work cells, and involve a larger number of operators. The lines are primarily focused on the assembly process, and their demand drives the production in the work cells. Understanding the relationship between lines and work cells is crucial for coordinating overall production and ensuring that assembly needs are met. As noted in the introduction to production planning, understanding the production rate and required throughput is fundamental for developing appropriate production schedules.

Order

Every order needs to be satisfied in the produced schedule. An order contains details such as the requested part number that needs to be produced and in which quantity it needs to be produced. Every order has a due date, the order needs to be successfully completed before that date.

Part number

Product that needs to be produced. Some products can be produced in multiple ways, with different costs associated and using different machines. More than 400 different part numbers are present.

Machine

A machine, or workstation, is an individual machine within the production system; each machine is associated with an OEE.

OEE

The Overall Equipment Effectiveness (OEE) is a metric that for month evaluates efficiency with respect to time for each workstation. During the planning phase the average of the last three months is considered.

Cycle

To produce a specific part number one or multiple cycles are available. Secondary cycles are alternative processes (to primary) for working on raw or semi-finished products, the main differences between them are in terms of processing time, output quantity and machine used. These provide flexibility in production planning, allowing for alternate routes if machines or other resources are not available, and this aligns with the principles of Flexible Manufacturing Systems.

Phase

Every cycle is composed of phases. These phases need to be executed in the given order to successfully produce a part number, either on the same machine or across different machines. The need for multiple phases impacts the scheduling of the parts, requiring that all the phases are completed in sequence, as discussed in the section on job shop scheduling. The number of phases for a given cycle can vary from one to five, every phase has a defined machine and processing time.

Tool

Tools are used to remove material from products. The availability and management of tools are crucial for ensuring continuous production. Tools are a critical resource that needs to be considered when planning the operations, as was noted in the discussion of production constraints.

Equipment

Equipment refers to the elements that hold products in the machine. Some equipment includes quick-release attachments. This information is not recorded in a centralized system, but the presence of quick-release attachments reduces setup time. Setup times, as discussed in previous chapters, play an important role in production scheduling.

12.2 General Context

This subsection provides further context to the operations of Project RefAIne.

12.2.1 Workload Balancing

Workstations must have a balanced workload among machines, so that no machine is overloaded or underutilized. This is important for optimizing the use of resources, and aligns with objectives discussed in the introduction to production planning and is often considered in parallel machine scheduling.

Operators need specific skills and qualifications to work in the stations, meaning that labor must be considered as a limited resource, impacting the scheduling of different jobs.

12.2.2 Types of Machines

Mono-pallet

A mono-pallet machine operates on one part number at a time. This limitation affects the planning of the operations, since it reduces flexibility.

Bi-pallet

A bi-pallet machine can accommodate two tools to work on multiple part numbers. However, it can only process one part number at a time. The advantage lies in the ability to set up the next part or use different tools. Workload between the two part numbers must be balanced. This type of machine introduces a scheduling challenge, and the workload between the different part numbers must be considered, similar to the challenges discussed in the section on parallel machine scheduling.

Multi-pallet/FMS

Multi-pallet machines, similar to bi-pallet machines but with multiple pallets (usually up to three), are used to balance workload and optimize space. This type of machine has elements similar to flexible manufacturing systems (FMS).

12.2.3 Multi-phase Workflows

The presence of multi-phase workflows highlights the need to consider sequence-dependent setup times and the requirement to follow the order of phases in the sequence for a product to be successfully processed, from raw material to final product.

12.2.4 Code and Tool Management

Multiple pieces of the same code can be processed on a machine, but different codes cannot share the same tool (with some exceptions). This limitation is similar to the challenge of having sequence-dependent setup times, as discussed in single-machine scheduling.

Multi-pallet management is currently handled only during the scheduling phase. This places an additional burden on the scheduling process, since it must explicitly consider multi-pallet management.

12.3 Current Software Tools

Cyberplan

Cyberplan is used for assembly line planning. It manages primary cycles but excludes alternative cycles. Cyberplan also assumes infinite workshop capacity. The program is slow and has issues with updates. The assumption of infinite workshop capacity can lead to unrealistic production plans.

SAP

SAP is used to register part numbers and production cycles. It handles set-up and machine time management and automates raw material requests.

SAD

SAD provides real-time monitoring of machine status and manages scheduling loading.

Excel

Excel is used for manual planning and simulation. It is also used for information collection on machine load, OEE, and part numbers. Excel uses pivot tables for sequence and quantity management.

Current Workflow

The current workflow involves manual data integration between these systems.

12.4 Analysis of the Current System

Disjointed Systems

The current approach uses multiple software tools: Cyberplan, SAP, SAD and Excel. These tools do not communicate between each other so a manual data integration is required. This is time-consuming, error-prone and inefficient.

Limited Planning Capabilities

Cyberplan's limitations, such as assuming infinite workshop capacity and excluding alternative cycles, can lead to inaccurate production plans. This makes it difficult to optimize the production process and adapt to changing conditions.

Manual Processes

The use of Excel, for simulation and to adjust the final schedule, indicates a lack of automation. These manual processes are slow and not scalable, and so they limit the ability to quickly adjust the scheduling in case of disruptions or required changes.

OEE Tracking

The Overall Equipment Effectiveness (OEE) is calculated for each workstation monthly, and the average of the past three months is used in planning. OEE is a key consideration in planning.

Limited Visibility

The system does not show the exact number of pallets that are actively working, which limits fine-grained optimization. Single phases are also sometimes excluded from planning, which can limit the level of detail available.

12.5 Required Objectives

The primary objective of Project RefAIne is to develop a program that can integrate the company's data and the user inputs to generate an effective and efficient production plans and schedules. By automating the planning and scheduling process, the company aims to reduce manual work, improve the accuracy of the plans and improve the overall efficiency of production, thus increasing throughput and reducing costs.

This project is a great example of an operation research application that aims at enhancing organizational decision-making and efficiency addressing complex scheduling challenges in flexible manufacturing systems (FMS), job shop scheduling and project scheduling, by providing a strategic planning tool with the goals of makespan minimization and throughput maximization.

The main objective set by the company follows a client-first perspective, that aims at satisfying the client needs in the first place, even if an outsourcing of production is required in order to meet the due date of the order.

This section has provided a detailed description of the production environment and the objectives of project RefAIne. The next section will discuss the specific model formulation used to address the challenges identified in this section.

13 Data Collection and Analysis

This section outlines the data collection process and analysis. The quality and relevance of the data are crucial for the accuracy and reliability of the models, that must be grounded in real-world observations and accurately represent the production system. This is the second phase of the systematic approach, which emphasizes the importance of gathering and analyzing data before building the model, as detailed in earlier sections of this study.

13.1 Data Sources

The primary data source for Project RefAIne is an Excel file containing multiple sheets with information such as production cycles, machine characteristics, orders, and the factory calendar. This file serves as the central repository of information for the model development and implementation.

13.2 Data Preprocessing

Before using the data for model implementation, several preprocessing steps were taken to ensure its quality and suitability.

Data Selection

Data was selected from the Excel file based on the required columns and relevant sheets. This involved filtering out unnecessary information and focusing on the most relevant fields. The selection process is also necessary to make sure that the data is consistent and that the data types are correct.

Data Cleaning

Missing values are removed, ensuring that the models operate with complete and reliable information. A selected interval of time is considered in the production, for instance all orders in the month of October were selected.

Data Transformation

The raw data is transformed into structured formats, such as Python dictionaries, for an easier retrieval and manipulation. For instance, the cycles data is organized into a dictionary and, given a specific part number, all the possible cycles that can be used to produce it are returned.

Data Filtering

In the model implementation, orders are filtered to include only those due in October 2024, and only those with a due date later than the 10th of the month, ensuring that all selected orders fall within the specified timeframe. This is done to focus the analysis on a specific planning horizon.

Data Aggregation

The daily machine availability is calculated based on the factory calendar and the daily working hours for each machine. Therefore weekends are automatically removed from the work days of each machine in October, and this information is stored in a dictionary. This step is important to model the available capacity of the machines realistically, since machine availability depends on the factory calendar.

The data analysis phase provides the basis for the model formulation and implementation, ensuring that the models are built with a comprehensive understanding of the system. The data is selected, cleaned and transformed before being fitted to the mathematical model.

The data collection and analysis phase is a critical step in the development of the production planning models. It ensures that the models are based on reliable data, and that the models accurately represent the production environment. The model implementation already automates all these steps so that a model can be implemented and optimized. The data analysis is essential for the successful application of the operations research principles.

13.3 Integration with Data

All company data, including numbers and names, has been anonymized to ensure confidentiality and no identifiable information is included in the data presented. This process guarantees the privacy and protects sensitive business information of the company.

The preprocessed data are ingested in the model, by loading the data from the dictionaries defined in the data processing phase, and using this data as input to the optimization model. This integration ensures that the model is dynamic and can be easily updated with the most recent production data. The model is designed to be able to automatically load data from the excel file, and to transform the data into the format that the model requires. This also means that the model is able to easily process new orders, or new machine information. This integration is a key aspect of the implementation phase, as it ensures that the model is not static and can adapt to the changes of a production environment.

13.3.1 Processed Data

At the current state, the model takes data from four different dictionaries:

- **cycle_dict** is used to access all cycles information, the main keys are the different part numbers, each of them leads to all the possible cycles to process it, furthermore each of these cycles may be diveded in multiple phases (that have already been ordered following their processing order requirements), finally three values are associated to each phase, namely the number of operation, the machine that is required and the execution time, that doesn't account for the machine OEE.
- **oee_dict** is used to pair each machine (keys) to its current OEE (values).
- **orders_dict** contains all the orders that the factory has to process, associated to three values: the part number requested, the quantity to produce and the due date of the order.
- **hours_dict** matches a new dictionary for each machine, for this example the former contains the list of all days in the chosen time window and the amount of hours that the parent machine is free. The time window and the granularity (days, weeks) of the schedule may change.

From the Excel it was possible to extract a total of 156 different part numbers and 336 different cycles, table 1 is an example of the formatted data of part number 169, derived using cycle_dict.

Cycle	Phase	Machine	Time
1	1	19	15.01
	2	22	12.00
	3	5	3.75
2	1	27	8.22
	2	49	14.53
	3	5	3.75
3	1	19	15.01
	2	49	12.50
	3	5	3.75
4	1	19	15.01
	2	36	12.76
	3	5	3.75
5	1	27	8.22
	2	36	12.76
	3	5	3.75

Table 1: Cycles for PN 169. The time to execute an operation is provided in minutes and it already accounts for the OEE of the machine.

14 Model Formulation

This section details the formulation of the model for Project RefAIne, incorporating principles of production scheduling and optimization discussed in the preceding chapters of this study. The model aims to integrate data and user inputs to generate efficient production plans and schedules, optimizing workflow and automating processes, as per the company's objectives.

In the following phase, the mathematical model is formulated upon the problem definition and data analysis steps, as highlighted in the systemic approach to operations research. This implementation focuses on the practical application of the previously developed theoretical model to ensure that it can be used to generate feasible and optimal production schedules.

14.1 Mathematical Programming Model

A mathematical programming model is developed to solve the scheduling problem. This model employs decision variables, an objective function and constraints to formulate the problem as a mathematical optimization. The model is based on the principles of linear programming, a technique that has been at the basis of OR since its discovery. The use of linear programming is motivated by its capacity to model a wide array of real-world scenarios, including resource allocation and production scheduling, in a very short time.

14.1.1 Decision Variables

The primary decision variables in the model represent a way to map every possible decision in the system, these variables determine the production schedule. Taking as time window the entire month of February, weekends excluded, this mapping creates a total of 27566 variables to store all possible choices.

Decision variable V depends on each possible *Order_id*, *Part_number_id*, *Cycle_id*, *Phase_id*, and *Day_id*.

To save on memory only the possible days are considered, meaning that for each order only the days until the due date of the order are taken into account.

The variables are indexed to reflect the various components of the production process, such as the specific order, part number, cycle, phase and day.

14.1.2 Objective Function

The model seeks to minimize the weighted sum of production costs. These weights can reflect different priorities like the cost of holding inventory, production delays or setup operations; in this case the model implementation uses a cost vector c that is multiplied by the decision variables and it is biased using an exponential smoothing function. The objective function is designed to ensure that the production plan not only meets demand but does so in a cost-effective way: prioritizing jobs that are closer to their due dates and total quantity. This objective function usually tries to balance multiple factors to achieve an optimal outcome, so this function will surely be modified in the future.

$$\min \sum_{o,p,c,d} x_{o,p,c,d} * C_{opd} * I_{op} \tag{4}$$

The objective is to minimize the sum for each order o, the quantity of product p, produced on day d, using cycle c, multiplied by coefficient C that implements exponential smoothing and by coefficient I.

Coefficient I is used to normalize orders: it depends on the quantity of product p for the order o and it's the inverse of the sum of part numbers required by the order.

For every order o the quantity of product p produced on day d is biased using a coefficient calculated as an exponential smoothing with α=0.6. The first days values will be near zero and they will exponentially increase the closer is the order deadline, an example is provided by table 2.

Day	C Value
1	0.006144
2	0.015360
3	0.038400
4	0.096000
5	0.240000
6	0.600000

Table 2: **Exponential Smoothing** If the deadline is the 6th of the month: every day after the date the coefficient will take the value 0, while for all the previous days, it will take the corresponding value.

14.1.3 Constraints

The mathematical model includes several constraints that need to be carefully designed in order to generate feasible production schedules and respect the requirements of the company:

Order Quantity Constraints

These constraints ensure that the total quantity of each part number produced meets the required order quantity before the due date. This constraint is crucial for meeting customer demands and avoiding shortages.

$$\sum_{o,p} X_{opd} \geq X_{opD}, \quad \forall d \in D \tag{5}$$

where:

- X_{opd} is the quantity of product p of order o produced on day d.

- For each order o, day d is a any day before day D, the deadline of the order; therefore $d \leq D$.

Temporal Constraints

The model follows the temporal constraints imposed by the factory calendar, ensuring, for example, that no production is scheduled on non-working days or non-working hours. This is achieved by considering only the factory calendar and the available shifts directly excluding all other moments. Also, the model checks for previously assigned working time for each machine in the time window taken into consideration, in order to directly exclude them for that machine, so that jobs do not overlap.

For every machine m and for every day d, the following constraint must hold:

$$\sum_{o,p,c} X_{opcd} \cdot t_{cpm} \cdot \mathrm{OEE}_m \leq t_{dm}, \quad \forall m, \forall d \tag{6}$$

where:

- X_{opcd} is the quantity of product p of order o on machine c for day d.
- t_{cpm} represents the processing time per unit of product p on machine m.
- OEE_m is the Overall Equipment Effectiveness of machine m.
- t_{dm} denotes the available time for machine m on day d.

Precedence Constraints

These constraints guarantee that operations within the same cycle are performed in the correct sequence, making sure that a phase with x material can start only after x material has been processed by the previous phase. This is achieved by constraining the value of the decision variable associated with the current phase to be less than or equal to the variable associated with the previous phase, thus implementing the precedence constraints of each phase and guaranteeing

that the correct sequence is always respected.
These constraints are necessary to ensure the multi-phase workflow of each product and cycle are satisfied.

x_{opcfd} is the quantity of product p, of order o, produced on day d using cycle c, during phase f.

For every phase f and for every day d, the following must hold for phase $f+1$:

$$x_{opcfd} >= x_{opcf+1d} \tag{7}$$

Continuing until the last phase and the last day:

$$x_{opcfd} + x_{opcfd+1} >= x_{opcf+1d+1} \tag{8}$$
$$... \tag{9}$$
$$x_{opcfd} + ... + x_{opcfd+n} >= x_{opcf+1d+n} \tag{10}$$
$$x_{opcfd} + ... + x_{opcfd+n} == x_{opcf+1d+1} + ... + x_{opcf+1d+n} \tag{11}$$

This must be applied to every pair of consecutive phases in order to ensure that all phases of a chosen cycle are executed correctly and products are processed as required.

Implementing this constraint greatly slows down the time it takes the model to load, and it also puts a heavy weight on the RAM of the computer.

Machine Capacity Constraints

These constraints ensure that the total processing time of all jobs assigned to a machine does not exceed its available capacity on any given day. The available time is based on the factory calendar and the Overall Equipment Effectiveness (OEE). The model browse the data to determine how many minutes a machine can work on a given day, and the time each job occupies in a machine. These constraints also incorporate the average OEE from the past three months, as discussed in the problem description. The machine capacity constraint is a common constraint in production scheduling problems, ensuring that the available resources are not over utilized, however in this problem no minimum or maximum limit of resources per machine were set, other than the available time of each machine for the given day.

Material Availability

The model assumes raw materials are available at the scheduling stage but may consider material availability during the planning phase. The implementation assumes unlimited availability of materials, in real-world situations, this constraint should be implement for the correct formulation of the problem.

Production Constraints

Specific constraints related to equipment and production, such as the ability of a machine to process multiple parts on different pallets, are also included. The model implementation does not explicitly model this constraint, but it can be modeled using additional variables for each pallet and their respective constraints. These constraints are essential for representing the specific operational rules of the production environment.

14.2 Software and Tools

The model was implemented using Python, leveraging several libraries for data manipulation, mathematical programming and optimization. Plots were produced to favor readability of the data and conduct analysis, some of these plots are presented in the next section.

Pandas

This library is used for data preprocessing, data manipulation, and data analysis, including reading data from Excel files and transforming them into data structures that can be used by the optimization models. Pandas is also used for data cleaning and for integrating the different data sources. This library is essential for all data-related operations, from data selection to data aggregation, as described in the previous section on data collection and analysis.

NumPy

This library is used to handle numerical operations: handling arrays and performing matrix operations.

SciPy

The SciPy library's linprog function is used to solve the linear programming problem. This function implements the simplex algorithm and allows for constraints to be defined using matrices and vectors.

Matplot and Seaborn

These two libraries are used to plot and analyze the data. All plots are in black and white to favor accessibility. All the data presented has been anonymized.

15 Model Results

The implemented model is based on a mathematical programming approach and aims to generate the optimal production scheduling for the company. The model is structured to consider the specific characteristics of the production environment, including the workstations, machine types, and multi-phase workflows. The model also considers the need to optimize a specific objective function: to minimize the number of parts produced, weighted by the due date of the order. This allows the model to prioritize orders that are due sooner. Taking into account the various constraints, including resource constraints, production constraints and temporal constraints the results are extracted to a data frame, which includes the optimal values of the decision variables, as well as all other information associated with each decision variable (such as the order, part number, cycle, phase, and day). This data frame is used to show the results of the model.

In Colab, the generation of the entire model takes 63 seconds not taking into account the creation of the data ingested, while the resolution of the model takes about 21.64 seconds.

The output generated from this model provides all the information that the company requires for production planning, as described earlier in this study. In fact, it's possible to augment the data frame generated: by matching each pair of cycle and phase with the machine they require and the time that processing each of these execution take, then multiplying this operation time by the quantity produced. The resulting data frame, for each decision variable, would have access to: the total operating time required to produce a certain quantity of the given part number, for the given order, using the given cycle and phase, for the given machine, at the given day. This is presented in table [3], what follows can be obtained from it.

Production Schedule

The primary output is a detailed production schedule that specifies the number of parts to be produced for each order, part number, cycle, phase and day. This schedule is derived directly from the values of the decision variables obtained from the optimization process. This output can be visualized as a Pandas data frame or exported in Excel to maintain alignment of the current practices of the company and not disrupt the workflow.

Index	Order	Day	Machine	PN	Cycle	Phase	Q	TpO	TOT
0	1	1	1	1	1	1	28	31	868
1	1	1	2	1	1	2	28	0.65	18.2
2	1	1	3	1	2	1	0	28	0
3	1	1	2	1	2	2	0	0.65	0
4	1	1	1	1	1	1	28	31	868
5	1	1	2	1	1	2	28	0.65	18.2
6	1	1	3	1	2	1	0	28	0
..	...	...	...	...	...	...	...	...	...
27565	229	30	2	156	336	4	0	1.6	0

Table 3: Augmented schedule: Q is the quantity produced by the index decision variable, TpO is the time required by the operation (it takes into account the machine OEE) that depends on Cycle and Phase, and it's multiplied by Q to obtain the total operating time (TOT). The latter is the actual processing time that the index decision variable scheduled. All time values are in minutes.

Machines Utilization

The machines utilization can be derived by analyzing the production schedule, figures 1 and 2. This will help identify bottlenecks and imbalances within the production system, figure 3.

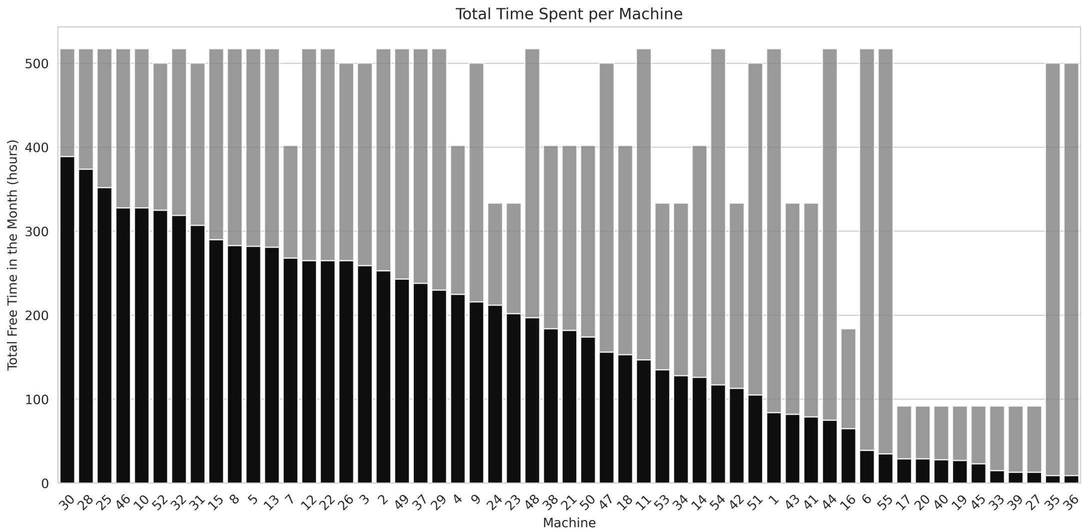

Figure 1: The stacked bar chart shows the total operating time per machine during the entire month, with operational time in black drawn on top of its free time in gray. Machines are sorted in descending order based on operational time.

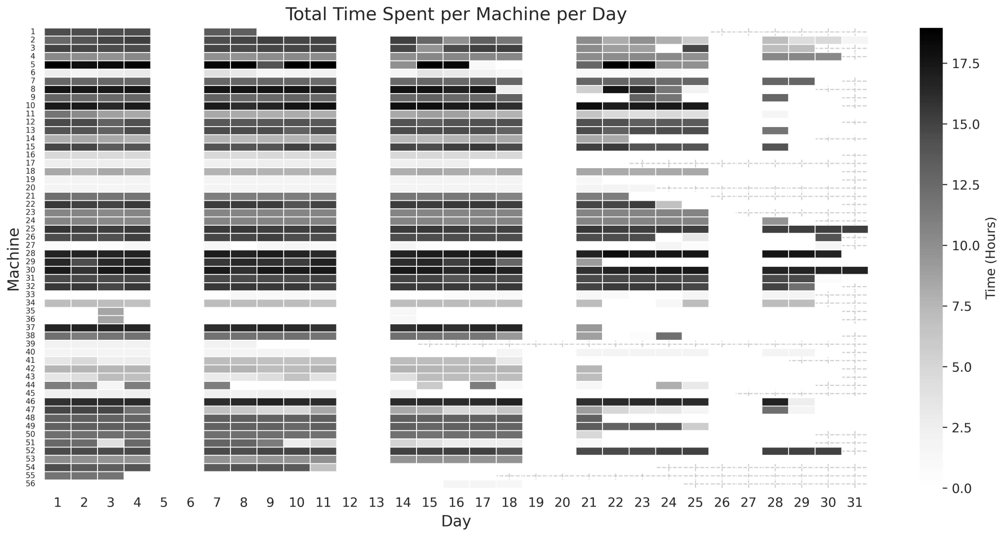

Figure 2: This heatmap shows the operating time per machine every day, with darker shades indicating higher usage. The X-axis represents days, the Y-axis represents machines and the color bar indicates time in hours.

Order Completion Times

The model returns to the user all orders' completion times based on the produced schedule, useful for assessing tardiness and meeting deadlines.

Taking order 42 as an example, accessing the orders_dict is possible to see its specifications: it requires a quantity of 144 of part number 37 by the 29th of the month, so, filtering the table by order 42 and considering only rows with a quantity produced greater than zero, it's possible to access all information about how the order has been scheduled, such as on which machine it's going to be processed and its the completion date.

Shift Plan

The model can provide a shift plan, by analyzing the production schedule, indicating how many shifts are needed to fulfill the planned production. This information can be used to ensure the proposed schedule is feasible with the available shifts.

Let us take machine 25 as an example since from figure 1 it is known to be the one with the most operating time: it operates for 477 hours a month, almost 20 days.

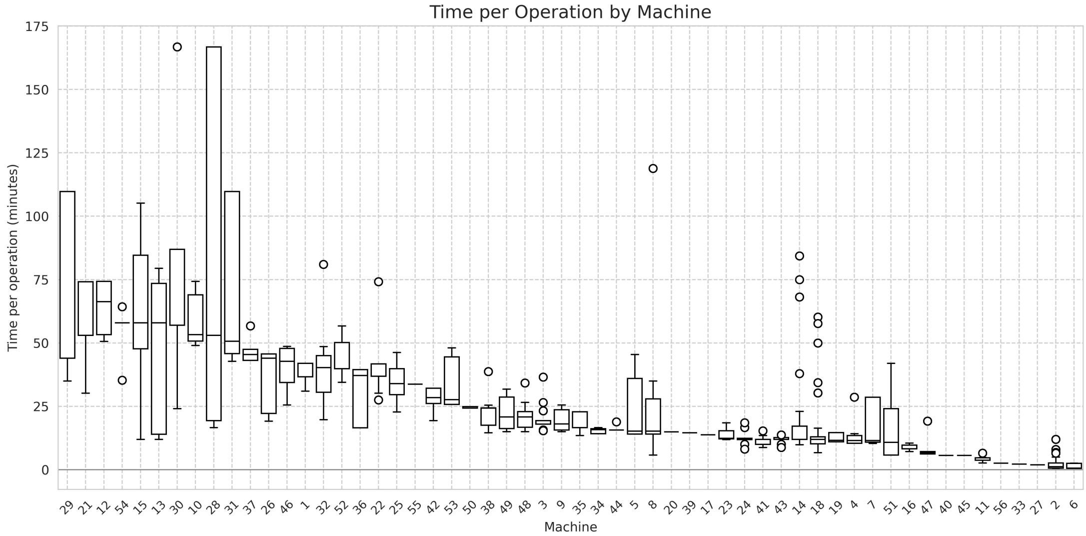

Figure 3: The boxplot shows the distribution of operation times for different machines, with some machines having significantly higher median times and variability, while others operate consistently faster or on operations that require less time, with outliers indicating occasional long operations.

Figure 4 shows the machine, where it is possible to see that the machine during the month works on eleven different tasks for almost sixteen hours every day.

15.1 Model Limitations

The main model limitation is due to the fact that at the current state, assigning tasks at any given day only considering the total operating time that a machine can process that day, is not enough to ensure the feasibility of the schedule, in fact that doesn't account for the order of the scheduled tasks. Considering the precedent constraint ensures that in subsequent days all processes have the required materials to be executed, but in a single day that constraint may doesn't hold, meaning that from the current schedule in the real-world multiple tasks may result assigned to the same machine at the same time.

To overcome this issue, a finer, more granular scheduling approach may be required. For example, conducting the schedule at an hourly

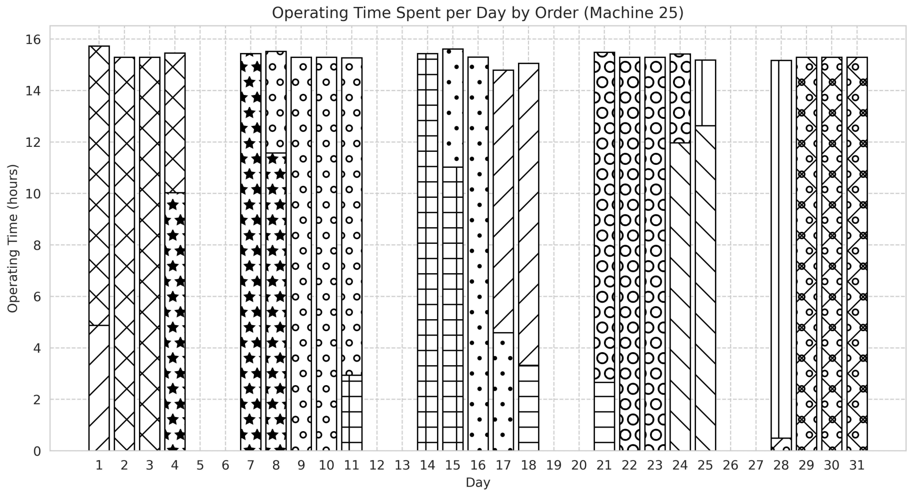

Figure 4: The stacked bar chart shows the amount of working time of machine 25 for every day of the month, with days on the x-axis and working time in hours on the y-axis. Different orders are distinguished by hatch patterns.

level instead of a daily one. However, this will significantly increase the time required to generate and solve the model.
If this is still not sufficient, a more complex model may be necessary, moving beyond linearity and introducing a variable to order all operations. As explained in the first part of the thesis, various techniques can be implemented to solve the new model. For instance, a local search algorithm could be used to refine the order of tasks by starting from the schedule of the presented model and evaluating neighboring solutions.

16 Conclusion & Future Work

Project RefAIne is still in its early stages, future integrations regard the industrialization of the procedures and a further alignment of the model with the current technologies of the company. Most importantly, the limitations discussed in the previous section must be addressed and new constraints integrated for the model to resemble the real-world scenario. New metrics can be implemented in the objective function and simulations can be produced to analyze different KPIs such as the idle time of machines, or how long a resource remains unused in the inventory of the company.

On this latter point, the greater innovation that Project RefAIne brings to the company is in fact, not the sole scheduling tool, but its integration with metrics and KPIs. The model and its produced schedule serve as a means to simulate and therefore analyze in a very convenient time and easy way any possible scenario, to provide quick adjustments in the production workflow reacting to an unseen event, such as the sudden malfunction of a machine and in that case re-balance the workload of all the machines, without disrupting the production. These responsive adjustments are going to enhance the client-first politic of the company, offering a better and very agile service to the customer.

The scheduling aspect of the model is going to help automate a daily process that at the moment takes at least 30 minutes of work to at least two specialized employees. This time is going to be reduced to about 6-7 minutes a day. Furthermore, this produced schedule is rarely optimal, since its computation isn't optimized and is somewhat done by guesswork lead by experience, rather than through precise measurements or automation; in fact, it's also partial, since it is currently only possible for short terms, while the model is capable of producing an optimized one over longer time windows and in a highly responsive manner.
The simulation aspect instead, is going to analyze the factory and, through what-if scenarios, predict and detect issues and then present them to the user so that it can act to solve them. Such issues may be potential bottlenecks that can be resolved by acquiring a new tool, unlocking the opportunity of putting in parallel multiple machines, reducing the tardiness of operations. Simulations can be helpful to test a new configuration of the machines around the factory, to test the workflow when deciding to prioritize an order over another, identifying potential raw material shortages, and much more.

Based on the company's market analysis, no system exists like the one presented: a system that not only meets all the constraints and perfectly aligns with their needs and themes but also addresses the requirements of many other companies in the manufacturing sector.

Currently, OR is applicable to any administrative process of a company and, thanks to it, the planning, organization, direction and control of all its processes and activities are facilitated. Moreover, over the years its application has been recognized and developed in different areas of possible use: operations research methods are now increasingly used in many other fields such as social sciences, biological sciences, environmental sciences and others as they offer a series of advantages that improve the operations and success of a company.

First reviewing the theoretical principles and then with the practical case study of Project RefAIne, the thesis has demonstrates how OR is indispensable to solve the challenges that a firm may face, furthermore giving the opportunity of enhancing its decision-making capabilities, efficiency and adaptability. By developing a mathematical programming model that integrates the data of the company, the project automates the creation of the production scheduling while adhering to the constraints imposed by the company such as machine capacities, multi-phase workflows and order due dates.

Today more than ever, OR is a tool available to everyone and is indispensable for organizations that want to solve complex real-world problems and be competitive in the market.

This thesis reaffirms OR's enduring relevance in an era of technological advancement and industrial complexity. By tracing the historical insights with modern computational tools, it offers a blueprint for leveraging OR to drive innovation, efficiency and resilience in production systems. Project RefAIne stands as a proof to OR's potential, encouraging more firms to take a step toward the application of artificial intelligence and autonomous integration.

References

[1] Local search for planning and scheduling, revised papers of ecai 2000 workshop in berlin, germany. In A. Nareyek, editor, *Local Search for Planning and Scheduling*, volume 2148 of *Lecture Notes in Computer Science*. Springer Verlag, Berlin, 2001.

[2] J. Adams, E. Balas, and D. Zawack. The shifting bottleneck procedure for job shop scheduling. *Management Science*, 34:391–401, 1988.

[3] E. Balas. Project scheduling with resource constraints. In E. M. L. Beale, editor, *Applications of Mathematical Programming Techniques*, pages 187–200. English University Press, London, 1970.

[4] R. M. Barnes. *Motion and Time Study: Design and Measurement of Work*. John Wiley & Sons, New York, 1968.

[5] A. Barr and E. A. Feigenbaum. *The Handbook of Artificial Intelligence, Vol. 2*. Addison-Wesley, Reading, 1982.

[6] C. Basnet and J. H. Mize. Scheduling and control of flexible manufacturing systems: a critical review. *International Journal of Computer Integrated Manufacturing*, 7:340–355, 1994.

[7] R. E. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, 1957. Dover reprint 2003.

[8] J. Blazewicz, W. Cellary, R. Slowinski, and J. Weglarz. Scheduling under resource constraints - deterministic models. *Annals of Operations Research*, 7, 1986.

[9] George Boole. *The Mathematical Analysis of Logic*. Macmillan, London, 1847.

[10] C. B. Boyer. *A History of Mathematics*. John Wiley & Sons, New York, 1968.

[11] Jr. C. H. Edwards. *The Historical Development of the Calculus*. Springer-Verlag, New York, 1979.

[12] A. Charnes, W. W. Cooper, and B. Mellon. Blending aviation gasolines – a study in programming interdependent activities. In A. Orden and L. Goldstein, editors, *Proceedings: Symposium on Linear Inequalities and Programming*, pages 115–145. Headquarters, USAF, Washington, 1952.

[13] G. Chartrand. *Graphs as Mathematical Models*. Prindle, Weber & Schmidt, Boston, 1977.

[14] S. A. Cook. An overview of computational complexity. In *ACM Turing Award: The First Twenty Years Lectures 1966–1985*, pages 411–431. ACM Press, New York, 1987.

[15] G. B. Dantzig. Linear programming. *Operations Research*, 50(1):42–47, 2002.

[16] M. A. H. Dempster, J. K. Lenstra, and A. H. G. Rinnooy Kan, editors. *Deterministic and Stochastic Scheduling*. Reidel, Dordrecht, 1982.

[17] M. Dror, P. L'Ecuyer, and F. Szidarovszky. *Modeling Uncertainty: An Examination of Stochastic Theory, Methods, and Applications*. Springer, 1994.

[18] Francisco Facchinei, Carlo Mannino, Stefano Lucidi, and Massimo Roma. Appunti dalle lezioni di ricerca operativa, 2003. Anno Accademico 2003-2004, Università di Roma "La Sapienza" - Sede di Latina (Università Pontina), Cap. 1,2,4,5,6,7,8 (Facchinei, Mannino), Cap. 3 (Lucidi, Roma).

[19] R. Fleischer and M. Wahl. Online scheduling revisited. *Journal of Scheduling*, 3:343–355, 2000.

[20] J. Franklin. *The Science of Conjecture: Evidence and Probability before Pascal*. The John Hopkins University Press, Baltimore, 2001.

[21] Michael Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, San Francisco, CA, 1979.

[22] Saul I. Gass and Arjang A. Assad. *An Annotated Timeline of Operations Research: An Informal History*. Springer, New York, 2005.

[23] III H. Everett. Generalized lagrange multiplier method for solving problems of optimum allocation of resources. *Operations Research*, 11:399–417, 1963.

[24] A. Hald. *History of Probability and Statistics and Their Applications Before 1750*. John Wiley & Sons, New York, 1990.

[25] C. M. Harris. Center for naval analyses. In S. I. Gass and C. M. Harris, editors, *Encyclopedia of Operations Research and*

Management Science, pages 79–83. Kluwer Academic Publishers, Boston, 1 edition, 2001.

[26] G. C. Heyde and editors E. Seneta. *Statisticians of the Centuries*. Springer-Verlag, New York, 2001.

[27] Frederick S. Hillier and Gerald J. Lieberman. *Introduction to Operations Research*. McGraw-Hill, New York, 9th edition, 2014.

[28] J. Holland. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, 1975.

[29] H. Hoos and T. Stützle. *Stochastic Local Search - Foundations and Applications*. Morgan Kaufmann Publishers, Elsevier, Amsterdam, 2005.

[30] S. V. Hoover and R. F. Perry. *Simulation: A Problem-Solving Approach*. Addison-Wesley, Reading, 1989.

[31] R. A. Howard. Decision analysis: Applied decision theory. In *Proceedings of the Fourth International Conference on Operational Research*, pages 55–71, Boston, Mass., 1966.

[32] J. R. Jackson. Job-shop like queueing systems. *Management Science*, 10(1):131–142, 1963.

[33] D. Kahneman and A. Tversky. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–291, 1979.

[34] R. M. Karp. Combinatorics, complexity, and randomness. In *ACM Turing Award: The First Twenty Years Lectures 1966–1985*, pages 433–453. ACM Press, New York, 1987.

[35] M. W. Kirby. *Operational Research in War and Peace*. World Scientific, London, 2003.

[36] E. L. Lawler. Optimal sequencing of a single machine subject to precedence constraints. *Management Science*, 19:544–546, 1973.

[37] Y. H. Lee, K. Bhaskaran, and M. L. Pinedo. A heuristic to minimize the total weighted tardiness with sequence dependent setups. *IIE Transactions*, 29:45–52, 1997.

[38] J. Y.-T. Leung, H. Li, and M. Pinedo. Approximation algorithms for minimizing total weighted completion time of orders on identical machines in parallel. *Naval Research Logistics*, 53:243–260, 2006.

[39] J. Y.-T. Leung and M. Pinedo. Minimizing total completion time on parallel machines with deadline constraints. *SIAM Journal of Computing*, 32:1370–1388, 2003.

[40] D. G. Luenberger. *Linear and Nonlinear Programming.* Addison-Wesley, Reading, 1984.

[41] C. U. Martel. Scheduling uniform machines with release times, deadlines and due times. In M. A. Dempster, J. K. Lenstra, and A. H. G. Rinnooy Kan, editors, *Deterministic and Stochastic Scheduling*, pages 89–99. Reidel, Dordrecht, 1982.

[42] S. T. McCormick and M. Pinedo. Scheduling n independent jobs on m uniform machines with both flow time and makespan objectives: a parametric analysis. *ORSA Journal of Computing*, 7:63–77, 1995.

[43] J. G. Moder and C. R. Philips. *Project Management with CPM and PERT.* Van Nostrand Reinhold, New York, 1970.

[44] R. J. Wilson N. L. Biggs, E. K. Lloyd. *Graph Theory 1736–1936*, pages 157–190. Oxford University Press, Oxford, 1976.

[45] J. F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36:48–49, 1950.

[46] S. Park, N. Raman, and M. J. Shaw. Adaptive scheduling in dynamic flexible manufacturing systems: A dynamic rule selection approach. *IEEE Transactions on Robotics and Automation*, 13:486–502, 1997.

[47] E. Pesch. *Learning in Automated Manufacturing - A Local Search Approach.* Physica-Verlag (A Springer Company), Heidelberg, Germany, 1994.

[48] M. Pinedo and M. Singer. A shifting bottleneck heuristic for minimizing the total weighted tardiness in a job shop. *Naval Research Logistics*, 46:1–12, 1999.

[49] M. Pinedo, B. Wolf, and S. T. McCormick. Sequencing in a flexible assembly line with blocking to minimize cycle time. In K. Stecke and R. Suri, editors, *Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems*, pages 499–508. Elsevier, Amsterdam, 1986.

[50] Michael L. Pinedo. *Planning and Scheduling in Manufacturing and Services.* Springer Science & Business Media, New York, 2009.

[51] Michael L. Pinedo. *Scheduling: Theory, Algorithms, and Systems*. Springer, New York, 3rd edition, 2016.

[52] K. Pruhs, J. Sgall, and E. Torng. Online scheduling. In J. Y.-T. Leung, editor, *Handbook of Scheduling*, chapter 15. Chapman and Hall/CRC, Boca Raton, Florida, 2004.

[53] J.-C. Régin and M. Rueher, editors. *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, volume 3011 of *Lecture Notes in Computer Science*. Springer, 2004.

[54] B. Roy and B. Sussmann. Les problèmes d'ordonnancement avec contraintes disjonctives. Note D.S. 9, SEMA, Paris, 1964.

[55] M. J. P. Shaw and A. B. Whinston. An artificial intelligence approach to the scheduling of flexible manufacturing systems. *IIE Transactions*, 21:170–183, 1989.

[56] S. M. Stigler. *The History of Statistics*. Harvard University Press, Cambridge, 1986.

[57] S. M. Stigler. Daniel bernoulli, leonhard euler, and maximum likelihood. In *Statistics on The Table: The History of Statistical Concepts and Methods*, pages 302–319. Harvard University Press, Cambridge, Mass., 1999.

[58] F. W. Taylor. *The Principles of Scientific Management*. Harper & Brothers, New York, 1911.

[59] C. W. Thornthwaite. Operations research in agriculture. *Journal of the Operations Research Society of America*, 1(2):33–38, 1953.

[60] Craig A. Tovey. Tutorial on computational complexity, 1991.

[61] H. M. Wagner. *Principles of Operations Research*. Prentice-Hall, Englewood Cliffs, 1969.

[62] R. R. Weber. Scheduling jobs with stochastic processing requirements on parallel machines to minimize makespan or flow time. *Journal of Applied Probability*, 19:167–182, 1982.

[63] J. D. Wiest and F. K. Levy. *A Management Guide to PERT/CPM*. Prentice-Hall, Englewood Cliffs, 1977.

[64] R. Wilson and R. Sharda. Neural networks. *OR/MS Today*, 19(4):36–42, 1992.