Ing. Dominik Zehetner, BSc.

Scheduling and Optimization of Resources in Maintenance Operations

Master's Thesis

Graz University of Technology

Institute of Engineering and Business Informatics
Head: Univ.-Prof. Dipl-Ing. Dr.techn. Siegfried Vössner

Supervisor: Dipl-Ing. Dr.techn. Nikolaus Furian

Graz, October 2016

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Graz, ____________________________ _________________________________

Date Signature

Eidesstattliche Erklärung[1]

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am ____________________________ _________________________________

Datum Unterschrift

[1] Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008; Genehmigung des Senates am 1.12.2008

Foreword and Acknowledgement

To begin with, I would like to express my gratitude to the people who either made this thesis possible or supported me while writing it, and to those who accompanied me during my years at university.

First, I want to thank Univ.-Prof. Dipl.-Ing. Dr.techn. Siegfried Vössner for giving me the opportunity to write this thesis at the Institute of Engineering and Business Informatics and use its infrastructure. I would also like to express my sincere gratitude to my supervisors Dipl.-Ing. Dr.techn. Nikolaus Furian and Dipl.Ing. Dietmar Neubacher. I immensely enjoyed the exciting and extensive discussions with both of you. It is mostly due to you that this thesis was not only interesting, but also a lot of fun.
Most importantly, I want to thank my family and friends for their never ending support and encouragement during my years at university. I would like to particularly thank my parents Herlinde and Josef and my sister Elisabeth. A special thank also goes to my friend Manfred, whom I shared an office with during the writing process of this thesis. Long hours of work are not disturbing but indeed rather enjoyable when one is in your company.
Finally, I want to thank my love Christin for her understanding and patience, her commitment and support with all of my decisions, and her cheering me up at all times.

Thanks to all of you!

Abstract

Industry 4.0 drives forward the trends of automation and digitization. Hence, the reliability of production machines is becoming ever more important, as is the field of maintenance. Due to new requirements, the maintenance department of an automotive manufacturer will change from a decentralized organization to an organization with a high degree of centralization. This leads to new challenges regarding management efforts when it comes to the scheduling and assignment of maintenance activities and employees. The present thesis investigates the structure of one company's maintenance department in detail, providing a scheduling algorithm for the needs of this specific company. In order to provide sound knowledge of the requirements of the factory, the theory deals with the fundamentals of maintenance and organizational structures as well as optimization and scheduling which are key factors for developing a suitable algorithm. Due to their different advantages and disadvantages, four different types of algorithms are developed. Three of them are based on linear programming and one on a heuristic solving method. The latter serves as a reference for the present means of manual planning. The best linear solver and the greedy algorithm perform a case-study which is inspired by the maintenance department of the thesis's partner company. The final case study discusses three forms of organization with different degrees of centralization. The results are then quantitatively compared to each other in order to rate the improvements that emerged due to the sophisticated algorithm and the change in organizational structures. The final conclusion states that centralization leads to better scheduling results. The sophisticated solver continuously performs better than human-like planning algorithms, although it reaches the biggest gain with regard to the organization that has an average degree of centralization.

Kurzfassung

Industrie 4.0 treibt den Trend von Automation und Digitalisierung weiter voran. Aus diesem Grund wird die Anlagenverfügbarkeit als immer wichtiger erachtet und folgedessen auch der Bereich der Instandhaltung. Aufgrund der neuen Anforderungen wird die Instandhaltungsabteilung eines produzierenden Automobilkonzerns von einer dezentralen Organisation, zu einer Form mit höherem Zentralisierungsgrad restrukturiert. Dies führt zu neuen Herausforderungen, wenn es um die Einteilung und Terminierung der Instandhaltungsaktivitäten und Mitarbeiter geht. Diese Arbeit widmet sich detailliert der Struktur der Instandhaltungsabteilung, und umreißt Scheduling Algorithmen, die nach den Bedürfnissen des Werks entwickelt worden sind. Um eine genaues Verständnis für die Anforderungen des Unternehmens zu erhalten, enthält der Theorieteil Grundlagen der Instandhaltung, der Organisationstheorie, sowie der Optimierung und des Scheduling, welches ein zentrales Kriterium für die Entwicklung eines passenden Algorithmus ist. Aufgrund der verschiedenen Vor- und Nachteile wurden vier unterschiedliche Algorithmen entwickelt. Drei Scheduler basieren auf linearer Programmierung und einer auf einer heuristischen Methode, welcher als Referenz für die momentane manuelle Planung verwendet wird. Der beste lineare und heuristische Löser bearbeiten eine Case-Study, die an das Partnerunternehmen angelehnt ist. Der Case behandelt drei Organisationsformen mit unterschiedlichen Zentralisierungsgraden. Die Resultate werden quantitativ verglichen, um die Verbesserungen aufzuzeigen die mit einem High-Level Algorithmus und dem Wechsel der Organisationsstruktur erzielt werden können. Eine Erkenntnis der Arbeit ist, dass Zentralisierung zu einem besseren Planungsergebnis führt. Eine weitere ist, dass der lineare Solver in allen Fällen besser arbeitet, als der heuristische, und er die größte Verbesserung bei der Organisationsform mit mittlerer Zentralisierung erreicht.

Contents

List of Figures

List of Algorithms

List of Abbreviations

CPF Corner Point Feasible

DFS Depth First Search

ERP Enterprise Resource Planning

GA Genetic Algorithm

GCP Greedy Choice Property

GLA Global Linear Algorithm

HGA Human Greedy Algorithm

KPI Key Performance Indicators

LLA Local Linear Algorithm

LP Linear Programming

MILP Mixed Integer Linear Programming

MSSP Maintenance Shift Scheduling Problem

OEE Overall Equipment Effectiveness

OPE Overall Production Effectiveness

OR Operations Research

PLA Pyramid Linear Algorithm

TPM Total Productive Maintenance

TS Tabu Search

Chapter 1

Introduction

The advance in technology in the last years has dramatically changed everyday life. Ever since Apple introduced the iPhone in 2007, smartphones have not only become a booming branch, but most people can in fact no longer imagine life without them. Moreover, intelligent devices are connected to the Internet and thereby generate an immense amount of data. Researchers even expect the overall created data volume will double every two years (Chen et al., 2014, p.1). As a result, numerous data-driven business models have been created in recent years and classic businesses try to adapt to the requirements of the new digital society. It is in this context, that Industry 4.0 – an initiative of the German government – plans to combine these new trends with conventional means of production technology in order to mark the next Industrial Revolution.

After the steam machine, assembly lines, electronics and IT, intelligent factories are about to launch the fourth Industrial Revolution. The latter is based on digital networks with a self-organizing production and direct communication between humans, machines, facilities, logistics and, finally, the products themselves. Production and logistics processes are combined so as to create more efficient and flexible production systems. Not only will this lead to highly intelligent value chains and life-cycle-phases of tailored products, but the costs of production will also decrease even with regard to customized production. (Plattform Industrie 4.0, 2016)

A PWC (2015) survey states that companies expect an improvement in efficiency of 20% within the next five years, due to the implementation of Industry 4.0. Thus, the Austrian industry intends to invest almost 4% of its revenue by 2020. Biedermann (2008, p.9) noticed that one of the most effective factors of increasing efficiency in manufacturing today can be reached by investing in maintenance. He realized that many production departments have almost reached their production capacity and high personnel costs decrease the productivity. However, with an increase in performance and automation levels, the investment in equipment also rises which subsequently leads to higher downtime costs of the facilities. Furthermore, the more complex machinery becomes, the more increases the need for maintenance. Hence, the area of responsibility for maintenance is growing and, compared to other business segments, it can make a considerable contribution to the indirect creation of value.

It is the management which needs to master the new challenges of production, maintenance and IT. The new degree of complexity requires new forms of organizational structures that can provide more efficient communication between internal departments and external service providers. This applies both to communication between employees and that between computer systems. This form of communication even calls for new competences on all levels of employment. The optimum maintenance strategy is also closely linked to the organizational structure, which changes due to the current trend of digitalization. (Guntner et al., 2016, p.14)

In cooperation with a partner company from the automotive manufacturing industry, the Institute of Engineering and Business Informatics of University of Technology Graz initiated the project "Total Productive Maintenance (TPM) 4.0". The objective of this project is to link the influence of Industry 4.0 with a modern state-of-the-art maintenance system. The project aims to analyze the maintenance department of the above mentioned company and verifies whether it is still suitable for modern and agile processes of maintenance. An intermediate result of this investigation was that a change from a decentralized maintenance organization to a centralized department enables numerous advantages. Indeed, it increases efficiency and better fits the recent needs of maintenance, although centralizing usually leads to a larger group size which, generally speaking, is more difficult to manage. Due to the higher number of resources and tasks that have to be assigned, the complexity of planning such maintenance actions increases. However, the problem is not only the increased time effort for scheduling this system, but it becomes more difficult to find good or feasible solutions for such an extensive timetable.

The present thesis analyses the needs and requirements of the new organization and in detail investigates the degree of centralization. In order to handle increasing management efforts at the central maintenance department, a scheduling tool needs to be developed to reduce the planning effort at the head of the unit. This should be done by an algorithm the purpose of which is to schedule TPM shifts and assign suitable employees to them. This process is called Maintenance Shift Scheduling Problem (MSSP). This thesis is structured into six chapters. The first covers theoretical aspects by describing the fields of maintenance, the organizational structure and scheduling. The following part defines the problem, representing the mathematical model of this specific scheduling problem. In chapters 4 and in 5, three algorithms which are able to solve the MSSP are described in more detail and are evaluated. The best algorithm will be taken up in chapter 6, more precisely in a case that was inspired by a real life scenario from the business partner. The solution of this case will be compared to an algorithm which imitates human behavior in order to measure the increase in solution quality when using a more sophisticated method. The last section includes the conclusion and an outlook on future milestones of this project.

Chapter 2

Theory

The following chapter summarizes the existing literature. To begin with, fundamental aspects of maintenance as well as the latter's definitions, objectives, strategy and procedures will be explained. Subsequently, the theory of structuring organizations is treated. The last part of this section deals with the theory of personnel scheduling, taking into account mathematical formulas, solving techniques and applications.

2.1 Maintenance

This section briefly outlines maintenance in the context of scheduling. In order to understand the specific terms of this field, definitions of maintenance are described first. The damage model is the foundation of all considerations in the field of maintenance and it is described in the next section. Moreover, the objective and tasks of maintenance are explained in more detail. Subsequently, crucial information about the costs of maintenance is obtained, in order to optimize maintenance timetables according to the indicators. The information of the prior sections is obtained in order to choose a suitable mix of maintenance strategies, which are described in section 2.1.5. The next section explains TPM as an extensive maintenance system based on preventive maintenance strategy. In order to develop a scheduling algorithm, a sound knowledge of capacity planning and scheduling in maintenance is essential. The last section thus gives a short overview of these two subjects.

2.1.1 Terms and Definitions

There exist numerous definitions of maintenance; however, in Europe, the definitions according to DIN31051:2012-09, 2012 are usually the ones that are cited in a standardized context:

> "The combination of all technical and administrative actions as well as actions of management in the lifetime of a unit in order to be in a fully functional state or to recover in this one, so that this unit can fulfill its requirements."

Furthermore, the specific terms of maintenance are described as follows:

- **Wear** is the reduction of the wear margin, due to chemical or physical processes
- **Wear-limit** is the defined minimum value of the wear margin
- **Wear margin** is the possible reserve function capacity under defined circumstances which a unit possesses.
- **Unit** is a component, device, subsystem, functional unit, equipment or a system which can be described and considered as an entity
- **Service** includes all activities delaying degradation of the remaining wear margin. The activities include: cleaning, conservation, greasing, oiling, complementing, changing and readjusting
- **Inspection** refers to all activities used to determine and evaluate the actual condition of facilities, machines, assemblies or components. Inspection refers to collecting data, and related activities can be measuring, verifying and monitoring
- **Repair** covers activities for retrieving the nominal condition, such as renewing, patching and adjusting
- **Improvement** is the combination of all technical and administrative activities as well as activities of the management to increase the reliability, maintainability or safety of a unit without changing its initial function.

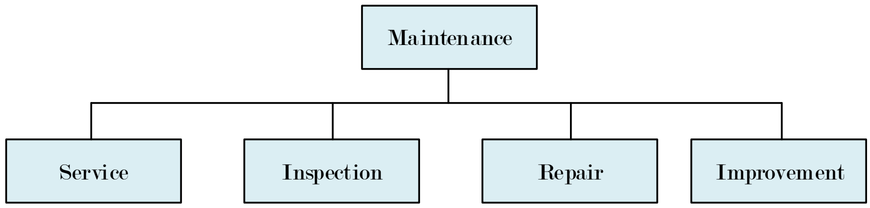


Figure 2.1: Divisions of maintenance, adapted from DIN31051:2012-09 (2012, p.4)

The decrease in operating capability of equipment, which occurs during its life cycle makes maintenance necessary as it can both prevent and improve this state. The four fundamental areas of maintenance are **service**, **inspection**, **repair** and **improvement** (see fig. 2.1).

2.1.2 Damage Model

Fig. 2.2 shows a dismantling curve of a component. The component owns a certain wear reservoir. This can be described as the amount and thickness of the component's wear material. Due to operating conditions, the wear material becomes thinner and the wear reservoir decreases. When it reaches zero, there is no wear material left and the component breaks. The reservoir can be filled up again, e.g. by overhauling or changing the component. Afterwards, the wear margin can vary from what it was like at the beginning to the shape of the function. It should be noted that the dismantling curve can be a discontinuous function as well. (Biedermann, 2008, p.12; Strunz, 2011, p.40)

In order to choose the right maintenance actions, failure characteristics of components must be taken into account. These are defined as **early**, **random** and **wear-out failures**. Fig. 2.3 illustrates a typical failure characteristic: the vertical axis describes the failure probability and the abscissa is the operating time of the component. The component passes three phases in its life cycle, the **burn-in period**, the **working period** and the **wear period**. These phases are defined by the gradient of the curve. The probability to fail is high in the burn-in period (e.g. due to design faults, material defects or operator errors). This kind of failures occur shortly after putting the component into operation. The probability of failure decreases constantly in this phase. In the second phase, the

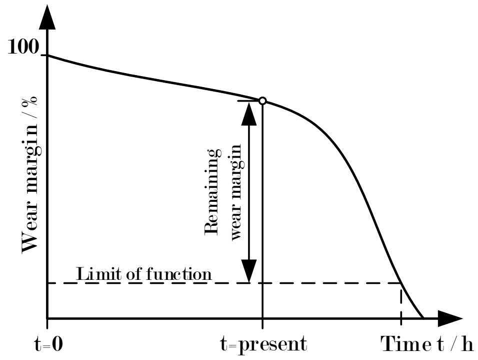


Figure 2.2: Dismantling curve, adapted from Schenk (2010, p.30)

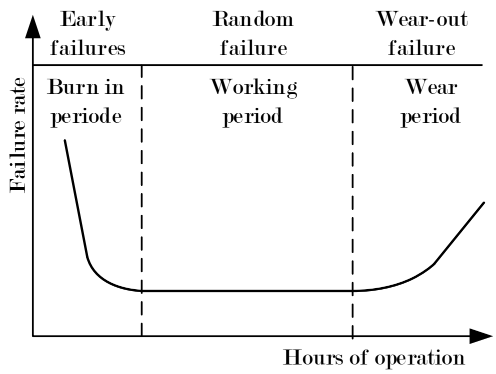


Figure 2.3: Capacity balancing, adapted from REFA (1991, p.445)

probability remains stable and faults can occur e.g. because of overload. Failures in this phase are independent of the operating hours and cannot be predicted properly. The wear-out phase is the last in the life cycle of a unit where fatigue, erosion and wear may lead to its failure. The probability of failure rises with the ongoing operating hours. Due to this consideration, it is clear that preventive maintenance should only be performed in the wear-out phase. Only in this phase, the root of failure is known and it can be influenced by preventive measures. In the early and random failure phases, a higher reliability cannot be reached by preventive maintenance actions. (REFA, 1991, p.446; Matyas, 2016, pp.42-43)

2.1.3 Objectives and Tasks of Maintenance

There are two contrary main objectives with regard to maintenance. On the one hand, it aims at maximizing reliability and safety, while, on the other hand, it seeks to minimize operational costs. Hence, an equilibrium between the costs of maintenance and the costs of machine breakdowns needs to be found by trying to take into account both indirect costs that are hard to quantify (e.g. delivery, quality, risk, etc.) and overall operational costs in order to minimize both. Unwanted downtime and accidents leading to environmental or even personnel damages must be avoided. Thus, maintenance has to be seen as an integrated part of production and it should be implemented in the logistics system in order to minimize costs. (Matyas, 2016, p.32) The main objectives can be broken down into technical, economic and other objectives (see table 2.1)

Subtarget	
Technical	• Improvement of the equipment's technical condition • Reduction of consequential damage • Reduction of machine breakdowns • Reduction of maintenance effort • Standardisation of structural and process organisation • Improvement of communication to other departments
Economic	• Reduction of personnel costs • Reduction of costs of material • Reduction of damage and consequential costs • Increase of machine availability • Conservation of equipment value
Others	• Increasing the level of protection • Decreasing personnel fluctuation

Table 2.1: Subtargets of maintenance, adapted from Matyas (2016, p.32)

2.1.4 Costs of Maintenance

Experts estimate that German companies invest some 140 billion Euro for maintaining machinery and facilities, which shows essential saving potential. The effort of maintaining must always be compared to the value it creates. This view leads to new concepts that aim at increasing availability while simultaneously decreasing overall costs. Hence, the borders of production, logistics, quality management and maintenance become rather blurred. (Matyas, 2016, p.27)

According to REFA (1991) the overall costs consist of **maintenance costs**, **breakdown costs** and the **subsequent costs of breakdown** (see fig 2.5).

- **Maintenance-costs** consist of the maintenance staff salaries, the costs for maintenance equipment and the downtime costs. With regard to preventive maintenance, the costs for spare parts must also be added.

- **Breakdown costs** are the expenses for repairing a damage or a malfunction, e.g. the salary of staff, necessary maintenance equipment and costs for auxiliary material. Costs for downtime are excluded within this calculation.
- **Consequential costs** are all expenses caused by occurrence of a damage but repairing. A precise illustration of these costs is provided 2.4.

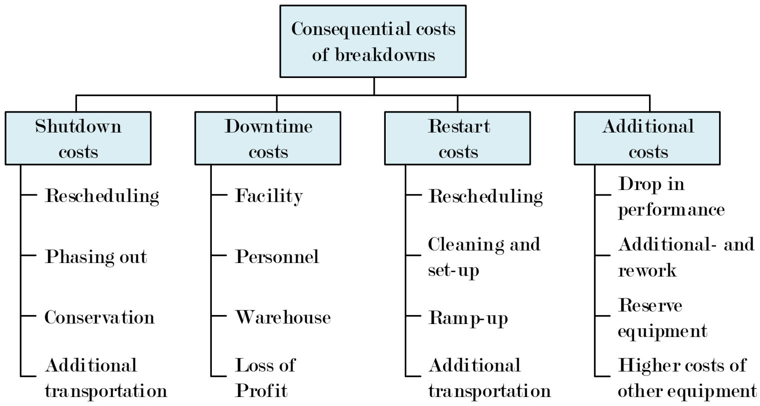


Figure 2.4: Consequential costs of breakdown, adapted from REFA (1991, p.433)

Fig. 2.5 below shows that the maintenance costs are linear to the intensity of the maintenance. Both the breakdown and the subsequent costs show a hyperbolic trend. It is also noteworthy that the consequential costs are higher than the damage costs – independently of the intensity. The overall costs (in green) are the sum of the other functions. Matyas recognizes that the optimum level of maintenance intensity can be determined by finding the minimum of the overall costs. If there are too few maintenance activities, damage and consequential costs will increase and the overall costs will rise accordingly. If the intensity is high, costs caused by breakdowns will decrease, but the maintenance costs and with them the overall costs rise. In REFA, it is stated that the intervals of maintenance must be adjusted with regard to the operational time and operating conditions such as temperature, pressure or oil levels. Furthermore, it is important to distinguish maintenance due to corrosion from maintenance due to wear. In case of maintenance due to corrosion, the operational time does not influence the maintenance interval. However, maintenance due to wear is directly influenced by the operational time. Leidinger claims that there is no correct intensity of maintenance for the whole maintenance system. This relation can behave differently at every maintained unit and for every unit there can be another optimum level of maintenance intensity. (Matyas, 2016, pp.48-49; REFA, 1991, pp.448-449; Leidinger, 2014, pp.24-25)

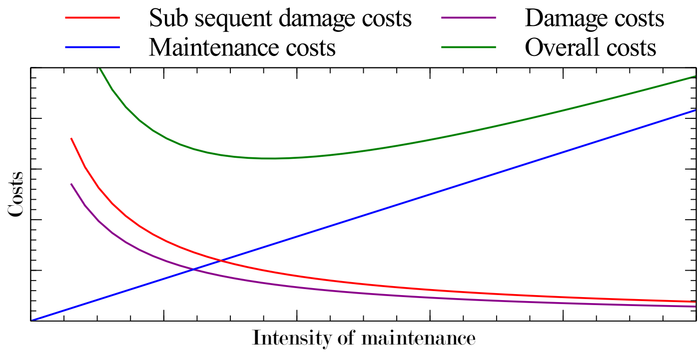


Figure 2.5: Influence of intensity of maintenance to costs, adapted from REFA (1991, p.448)

2.1.5 Maintenance Strategies

Maintenance strategies are approaches to the solution of maintenance problems. They are rules that define which maintenance activity is executed on a certain unit at a certain time. Also, they should guarantee a high reliability of the units at minimum costs. The right strategy depends on the failure behaviour and the damage costs of the maintenance unit. (REFA, 1991, p.444)

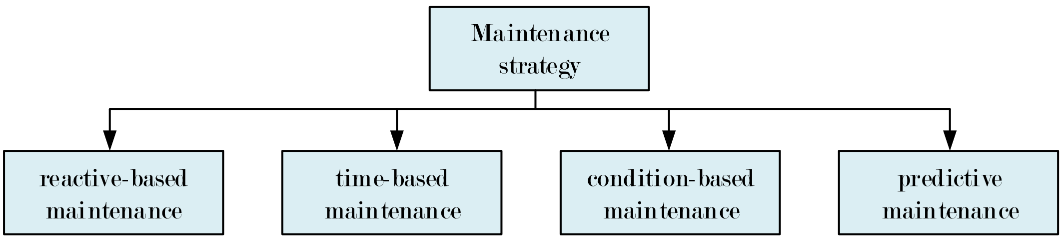


Figure 2.6: Maintenance strategies, adapted from Matyas (2016, p.120)

Matyas (2016) distinguishes four types of strategies which are displayed in fig. 2.6: **reactive-based**, **time-based**, **condition-based** and **predictive maintenance**.

The **reactive based** strategy is a system which accepts that damages and failures are conscious. The advantage of this strategy is that the wear margin of a unit can in fact be fully used. This reduces the effort for inspection as well as the effort for changing the unit. The problem of this strategy is, however, that the machine decides the date of failure, instead of the operator. If this happens at the wrong time, damage and consequential cost can be significantly higher than the yield of the full usage of the wear margin. This concept is useful in case production machines are redundant or the production process is insignificant. (Leidinger, 2014, p.20; Matyas, 2016, p.121)

The second strategy of maintenance, the **time-based** maintenance method, performs all maintenance activities after a certain amount of operating hours or following a defined schedule. This strategy assumes that, after a specific operating time, the wear margin is worn out. The objective of this procedure is to change or overhaul the part before the wear margin is used up. Matyas (2016, p.122) rates this as the most suitable procedure when safety or the environment can be harmed or when the lifetime is known. In order to keep the costs of maintenance and downtimes (including subsequent costs) low, it is necessary to adjust the maintenance interval to the actual lifetime of the component. A good way to minimize costs would be to change parts shortly before they fail. In this case, the service life is almost exhausted but there are no damages on other parts of the machine yet. Although this strategy includes high efforts, it is expected that planned maintenance is cheaper and faster than an unplanned maintenance activity. For this method, it is crucial to find the optimum operating period of a component, which is very complex mainly due to statistical reasons. (Leidinger, 2014, p.17)

The method of **condition-based** maintenance uses the degree of wear in order to trigger a maintenance activity. Appropriate monitoring and diagnose technology make it possible to obtain information on performance deviations. This method usually uses five techniques: vibration monitoring, thermography, tribology, process parameter, and visual inspection. This data allows to adapt the maintenance interval to the actual life time of the component. Usually, a failure can be noticed before the actual destruction takes place but the interval between the notice and the failure might take either milliseconds or years. (Mobley, 2002, p.4)

The three classical maintenance strategies described before are in many cases no longer sufficient due to increasing complexity of facilities. Higher availability can only be established through high maintenance intensity, which leads to waste of resources. Forecasting wear and the impact of it are central issues in **predictive** maintenance. The majority of existing models uses historical data of long-term-studies in order to forecast machine breakdowns. Based on these datasets, it is possible to derive statistical methods in order to determine typical wear processes of components. (Matyas, 2016, pp.136-137)

All of the strategies described above have their advantages and disadvantages, which is why none of them can be applied universally. However, a good mix of the methods allows for an optimized maintenance approach. (Leidinger, 2014, p.21)

A well-mixed maintenance strategy is a necessary basis for a maintenance system, but it alone does not achieve success. There are many factors for success which are not taken account in maintenance strategies, e.g. that machines are maintained by individuals with human needs. Total Productive Maintenance (TPM) implies social factors, amongst many other important issues.

2.1.6 Total Productive Maintenance

The roots of TPM go back to the American statistician Dr. Deming who imported his idea to Japan. The company Nippondso (today Denso) introduced a maintenance program in 1960, which was based on preventative maintenance for automated processes. This program needed a large amount of maintenance engineers, so instead of hiring more personnel, the management decided to use operators that were currently working on the machines to perform maintenance actions. Due to this strategy, the company did not only save additional staff, but they also ensured the operators' thorough understanding of the machines. These employees were able to detect problems at an early stage or whether the quality of the components was decreasing due to the machinery. The cooperation between maintenance engineers and operators allowed continuous improvements of the machines. In fact, there exist records that the goods' quality increased and that there were fewer levels of defect parts. The communication and the group effort built the foundation for working on preventive maintenance, prevention of maintenance and maintainability improvement. The involvement of all employees made this program well-established and included even the top management level. (Augstiady and Cudney, 2016, pp.9-10)

Objective and Definition TPM

Nakajima (1988) defined the framework of TPM and casted it to the book *Introduction to TPM* which is standard literature in this field of research. The following sections are based on his work.

TPM is often defined as “productive maintenance involving total participation”. This is often misunderstood by management because it assumes that preventive maintenance activities are executed autonomously on the shop floor. A more precise definition of TPM includes the following elements:

1. The objective of TPM is to maximize Overall Equipment Effectiveness (OEE)
2. TPM establishes a detailed system of preventive maintenance for the equipment’s entire life span
3. It is implemented by various departments
4. Every single employee is involved, from a worker on the shop floor to the top management
5. Motivation management through autonomous small group activities

The word “total” in TPM has three meanings: **total effectiveness** (see point 1 above), **total maintenance system** (includes maintenance prevention, point 2) and **total participation of all employees** (points 3,4,5).

In general, the procedures of TPM must be tailored to the individual company. The needs and problems vary which is why each enterprise must develop its own action plan. However, there are basic elements for the development of TPM which are suitable for most situations. Successful implementation requires:

1. Elimination of the six big losses to improve OEE
2. Autonomous maintenance programs
3. Scheduled maintenance programs for the maintenance department
4. Increased skills of maintenance and operations personnel
5. An initial equipment management program

The main objective of production improvement activities is to increase productivity, which can be done by decreasing input and/or increasing output. According to the definition of TPM, output consists of improving quality, reducing costs, and meeting delivery dates while increasing morale and improving safety and health conditions. Input includes labor, machine and materials. Due to increasing automation, the production process shifts from workers to machines and the equipment becomes more relevant to the output. TPM maximizes output by running equipment effectively and maintaining ideal operating conditions. In order to maximize OEE, TPM defines six big losses which are obstacles to OEE. These six losses that need to be eliminated are:

1. Breakdown caused by equipment failure
2. Unnecessary adjustments and set-up
3. Minor stops and idling
4. Reduced speed
5. Start-up losses
6. Scrap and rework

Overall Equipment Effectiveness

The six losses described in the previous section, influence the three factors of the OEE, which are stated in the following equations:

$$Availability = \frac{total\ available\ time - actual\ downtime}{total\ available\ time}$$

$$\begin{aligned} Performance &= operating\ speed\ rate \cdot operating\ rate = \\ &= \frac{Ideal\ cycle\ time}{Actual\ cycle\ time} \cdot \frac{actual\ cycle\ time \cdot total\ output}{operating\ time} \end{aligned}$$

$$Quality = \frac{total\ output - number\ of\ defects}{total\ output}$$

$$OEE = Availability \cdot Performance \cdot Quality$$

The *Availability* is affected by breakdown and adjustment losses. The *total available time* is total attendance time less downtime allowances (i.e. breaks). The *actual downtime* is the sum of all unplanned breakdowns. Performance is affected by idling and minor stoppage losses, as well as its reduced speed. The *Ideal cycle time* is the cycle-time the machine was designed to achieve at standard conditions. *Total output* describes the number of produced units (including defect parts). The *operating time* is the availability time. For detailed information regarding the calculation of the OEE see Willmott and McCarthy, 2001, pp.82-83.

2.1.7 Maintenance Capacity Planning and Scheduling

Planning can be defined as the process of determining future decisions and actions necessary to accomplish goals and targets. Planning helps to achieve goals in the most efficient and effective way. When applying this method properly, it can minimize costs, reduce risks and decrease the number of missed opportunities. In general, planning includes the determination of tasks as well as the resources required for them. Scheduling is the process of putting those tasks into a time frame. (Ben-Daya et al., 2015, p.237) Thus, before scheduling can be performed, planning must be accomplished – which is displayed in the following sections.

Maintenance Capacity Planning

Maintenance capacity planning uses the forecasted maintenance load to determine the optimal level of resources such as crafts, skills, spares, inventory, equipment and tools. A fundamental element of capacity planning is the determination of employees and their properties, e.g. skills or overtime capacity. Due to random and varying workload, the optimal allocation of resources is a complex and challenging task. A major issue of capacity planning is to determine the optimal mix of skills from the available employees. A proper way to determine this mix is to use costs and availability measures. (Ben-Daya et al., 2015, p.36)

The influences of the maintenance strategy on the capacity can be tremendous. When determining the necessary capacity for all preventive maintenance actions, an additional fluctuating demand can be expected (e.g. unplanned breakdowns). The maintenance capacity plan has to consider these peaks if other maintenance activities should be done on time. (Kroesen, 1983, p.82)

The objective of capacity balancing is to compare the demand for capacity and the capacity limit. The comparison and harmonization of these two quantities is called capacity balancing (see fig. 2.7). (Wiendahl, 2014, p.326)

Fig. 2.8 presents two methods for capacity balancing. Measures for capacity adjustment in the short term is only possible by increasing or decreasing the available operating time of employees or equipment, which can be seen on the left hand side of the figure. On the right hand side, the method of capacity alignment is illustrated. (Schuh, 2006, p.48)

- **Capacity adjustment** increases or decreases the capacity limit, e.g. by directing overtime
- **Capacity alignment** moves the demands of a bottleneck in other areas,e.g. jobs are shift to other dates or to other technological capacities.

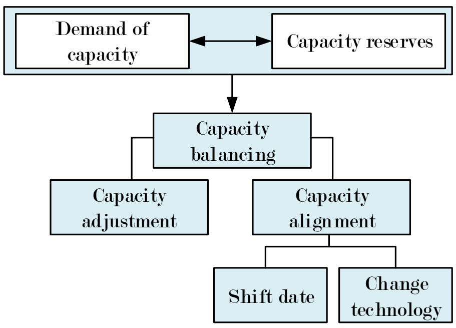


Figure 2.7: Capacity planning, adapted from REFA (1991, p.192)

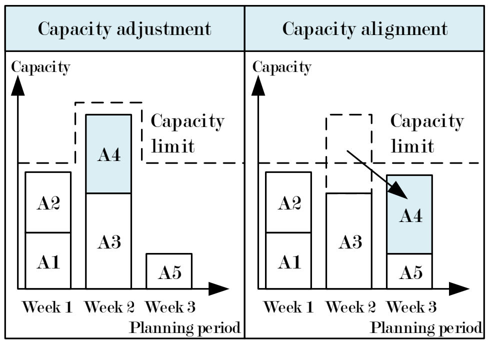


Figure 2.8: Capacity balancing, adapted from Schuh (2006, p.49)

Although the need for predictable maintenance activities like inspection or service can be determined, the demand for unplanned activities like breakdowns can only be considered in a stochastic way, i.e. on the basis of estimations and experience. This leads to inaccurate and uncertain capacity planning as only a range can be specified in which the capacity is expected. Although bottlenecks can be moved at scheduled maintenance activities, for unplanned measures, this is only possible to a limited extent. As a result, the capacity demand fluctuates throughout the year. One way to master this challenge is to fit the size of the personnel to the base load and to cover bottleneck situations with outsourced workforce. Another approach would be to rethink the ratio of centralized and decentralized maintenance staff. Due to the small size of decentralized units, the share of unplanned activities is usually higher and the number of high skilled employees lower. (Rasch, 2000)

Maintenance Scheduling

Maintenance scheduling answers the question of which jobs are done at a certain point in time with specific resources. There are many points to consider when scheduling tasks according to a time frame, intended goals, interrelations between the different tasks, availability of resources, overtime and other external and internal limitations and constraints. A quality indicator is usually measured by performance in relation to the intended goals or tasks. In maintenance, performance can be measured with different types of costs, by meeting due dates, with times of completion or the utilization of resources. (Ben-Daya et al., 2015, p.11)

A very important point with regard to maintenance scheduling is the determination of priority. The work load in efficiently manned maintenance organizations usually exceeds the availability of men and/or equipment. Therefore, the priority of each task has to be taken into account. In small plants with one central maintenance department, priorities would be defined in casual discussions between the maintenance and production departments. However, if there is more than one production department involved, the priorities are often set on the basis of human relationships or geographic location. This can lead to negative effects on overall efficiency. A solution to this problem could be introducing one individual who is able to estimate the effect of decisions on the overall plant performance. In large-scale plants, it has proven effective to handle these problems at a lower level of management. (Higgins, 1995, pp.1-11)

Scheduling can be a stumbling block when it comes to maintenance. It can drive the efficiency of an organization, but it is important to know which things to prioritize. This is important for developing an solver as these priorities must be the driver of the objective function which can be optimized by a suitable mathematical optimization method.

However, it is not possible to optimize the OEE directly by a scheduling algorithm; rather, influencing factors of the OEE have to be found in order to create a suitable and favorable solution for maintenance scheduling.

2.2 Theory of Organizational Structures

This section briefly outlines the theory of organizational structures with a special focus on maintenance organizations. To ensure clarity about the term organization, a short definition is stated in the first section. Second, the design process of an organizational structure is outlined, in order to understand the procedure of restructuring which happens in the maintenance department of the partner company. The last section translates the theory of organizational structure into an applied context of maintenance.

2.2.1 Definition of Organisation

Schreyögg (2008, p.4) gives two answers to the question “What is organization?”. When talking of an organization, one may think of companies, churches, or schools. This assumption is usually based on having in mind special characteristics, e.g. that a company is in “reorganization” or that it is “organized” in a central way. Hence, the term organization is divided into two groups: the first is called institutional organization, and the second instrumental organization.

Vahs (2012, p.11) considers organization as the opposite of chaos and claims that every target-oriented cooperation is based on order. Without order, there is chaos, which he interprets as confusion. In a state of confusion, it is impossible to master complex tasks in a systematic way. Hence, there is a need for organization, which is defined as the implementation of regulations so as to create order in organizations.

This order can be described as organizational structure. Since these rules are mostly based on formal, official rules, it can be qualified as the formal structure of an organization. The mission of regulations is not only to ensure an efficient working procedure, but also to resolve conflicts, create room for new ideas and define the appearance outside of the organization. These examples clarify that rules are always directed at a system’s members and that their purpose is to make the members' manners and behavior predictable. They also limit their scope of actions, which means that structures narrow down the amount of possible actions to one or only a small number of possibilities. (Schreyögg and Koch, 2007, pp.289-290)

So far, rules are discussed as general regulations that create order and structure. This rules can be classified into **general regulations** and **occasional regulations**. The latter are important for individual cases. General rules replace occasional rules, thus making them unnecessary. It can be assumed that it is effective to substitute general regulations with occasional ones when it comes to frequent and uniform operations since they show an optimal ratio of occasional and general regulations. If flexible procedures are handled with general regulations, over-organization may occur. On the other hand, if frequent and similar actions are handled with occasional rules, this often leads to under-organization (see fig. 2.9). (Gutenberg, 1966, pp.237-240)

It has by now become clear that regulations or, in other words structures, are necessary to accomplish complex tasks and create an efficient working procedure. The next chapter describes how to effectively create such structures.

2.2.2 Design Process of an Organizational Structure

The main objective of the structure of organizations is to solve the dual problem (see fig.2.10). On the one hand, coordination and efficiency increase due to the creation of structures and subsystems, but, on the other hand, the overall system thus becomes more complex. The more departments and subsystems are created, the more challenging it becomes to maintain an overview and sustain cooperation between these groups. Thus, the integration of parts becomes increasingly problematic. This is called the dual problem of the structuring of organizations, which is further described in the following sub-chapters. (Schreyögg and Koch, 2007, p.92)

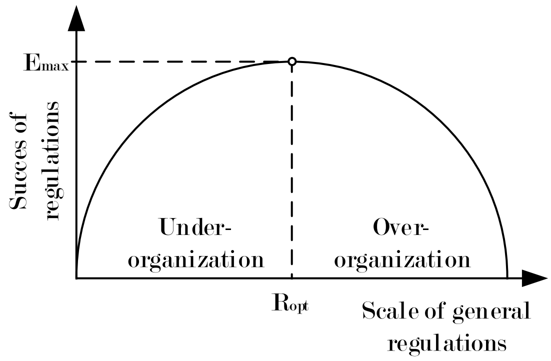


Figure 2.9: Success of regulations, adapted from Steinmann and Schreyögg (2005)

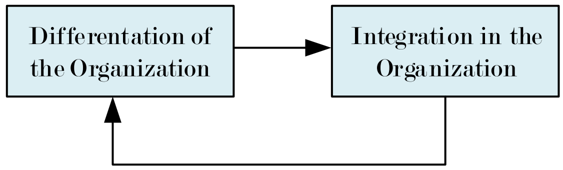


Figure 2.10: Dual problem of structuring of organizations, adapted from Schreyögg (2008, p.92)

Differentiation

Differentiation is the first problem. The overall task of organization is normally more demanding than a task for just one person. Hence, the question that needs to be answered is how to divide this overall task into assignments suitable for several people under consideration of the appropriate degree of specialization. (Vorbach, 2015, p.310)

This problem is called task analysis and covers the systematic separation of the overall tasks into subtasks. The latter can then be again divided into even smaller subtasks and so on, whereby the logic follows a top-down approach. It ends when a subtask can be handled by a single person. (Vahs, 2012, p.52)

According to Kosiol (1976), it is necessary to extensively collect all tasks and order them in a systematic way in order to be able to appropriately distribute them. Task analysis is the groundwork for task synthesis and the structuring of an organization. The overall task can be divided into five dimensions in order to obtain a high amount of subtasks. Vahs (2012) divides those five dimensions into the following:

- **Type of Execution**: e.g. purchasing, production, sales
- **Object**: e.g. motors, gears
- **Rank**: e.g. procedural decisions, motor-assembly
- **Phase**: e.g. production planning, production controlling

- **Primary / secondary Tasks**: value-adding Tasks, Administration

When the overall tasks are broken down into subtasks of the appropriate workload, the task synthesis begins. In this step of the process, tasks are grouped together whereby the smallest unit is called job. This is a bundle of activities which can be handled by one person. A greater number of jobs is usually combined and subordinated to a unit with the authority to issue directives. This means that one can decide between executing and leading units. However, organizations can have staff-units or central units as well. Staff units are used for relieving the management, i.e. for support, coordination, and consulting, but are not allowed to give directives to other units. This is the main difference to centrally organized units whose technical expertise allows them to issue directives. Centralized units could be for example Research and Development, Procurement, or Financing and Controlling. (Vorbach, 2015, p.312)
For a graphical interpretation of task-analysis and synthesis see fig. 2.11.

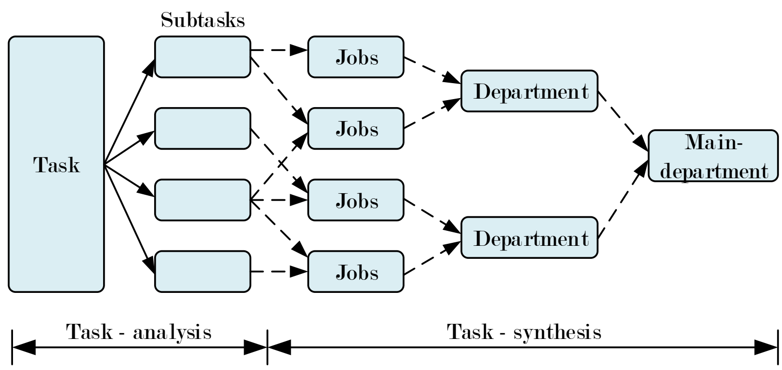


Figure 2.11: Building of departments, adapted from Steinmann and Schreyögg (2005, p.444)

Dividing labor creates complexity as the process forms numerous units which all are focus on their specific tasks. When taking into consideration the overall organization, this separation disrupts the performance process. The tasks are executed at different departments at different times, which makes it necessary that the results of the subtasks have to be combined. This is the goal of organizational integration. (Schreyögg, 2008, p.129)

Integration

There are three approaches for the problem of integration (Steinmann and Schreyögg, 2005, p.457):

- Hierarchy
- Programs/Plans
- Rules for self-coordinating

Hierarchy creates a system of rank-related competence. Each unit is classified in a rank so that there is no doubt which one is superordinate and which is subordinate. There exists an explicit procedure for each coordination problem. The subordinate unit reports the problem to the superordinate unit until a unit is found which covers both units in between which the coordination problem occurs. Coordination by **programs** in large organizations is by far the most common additional tool of integration. Programs are binding guidelines, i.e. general rules with the purpose of integration and they lead to standardization of the working process. In contrast to other ways of integration, certain events trigger mandatory activities or decisions. With regard to **self-coordination**, the units involved contact each other directly in case there is a problem between two of them. This kind of connection becomes more and more common and the horizontal connection is on the border of occasional and general regulations. Benefits of this kind of organization is the relief of management units, growing motivation of employees due to flexibility, and fast and efficient problem solving. A disadvantage may be that some employees do not want to use this kind of freedom due to social or technical incompetence and increased time requirements because of group work. (Vorbach, 2015, p.316; Schreyögg, 2008, p.131)

Apart from the measures written above, the second way to establish integration is to reduce the necessary level of integration in an enterprise. In general, the dimension of necessary integration is defined by the degree of division of labor and the sum of measures for reducing integration. These reducing actions can be (Vorbach, 2015, p.314, see also Kieser and Walgenbach, 2010):

- Decentralization of units (e.g. Profit-Center)
- Create buffers which loosen the coupling (e.g., interim storage in logistics)

- Use of flexible resources (e.g. universal machine, cross-skilled-staff)
- Sustaining standards and tolerances (coordination is necessary if tolerance is exceeded)
- Management by exception
- Reduction of standards of results and efficiency (the more efficient, the more coordination is necessary)

Decentralization is one possibility to reduce the measures of integration but it isolates the decentralized unit from the overall organization. Centralization is the tightest means of coordinating decision-making in an organization. In fact, all decisions are made by one person and then implemented through direct supervision. Thus, decision-making is very easy and what remains is only the need for coordination. So, why should an organization give power to decentralized units? In fact, many decisions are complex and cannot be understood by one individual alone as human beings have limited cognitive capacity and are unable to process every information relayed to them. As organizations face many complex conditions, decentralization is thus a widespread organizational phenomenon. Here, decision-making powers are combined among individuals who are able to understand the requirements and intelligently respond to them. Decentralization is also considered as a factor for motivation and the advancement of the system's flexibility. (Mintzberg, 1979, pp.182-183)

2.2.3 Organizational Structures in Maintenance

In most traditional organizations, maintenance is still part of the production department. An additional form of maintenance commonly found in organizations is the integration of a central maintenance department. In recent years, an increasing number of companies has also introduced decentralized maintenance departments which operate as profit centers. Another change in maintenance organizations is caused by increasingly complex tasks. Maintenance employees have become specialized in mechanical and electrical skills and now share tasks with the production department which deals with tasks such as cleaning and greasing. With the growing size of manufacturing facilities, central maintenance departments are expanded by decentralized support bases in order to provide shorter response times. The tasks of maintenance were specified by the production department so as to optimize them. Thereby, preventive and condition-based maintenance was largely ignored which lead to high costs. (Matyas, 2016, p.74)

Three factors must be considered, when designing a maintenance organization: the **capacity** of maintenance, **centralization vs. decentralization** and **in-house vs. outsourcing**. Furthermore, numerous criteria like responsibilities, effective span of control and good supervision can also be taken into account for designing. There are three common organizational models in maintenance which can take various shapes (Ben-Daya et al., 2015, p.6):

- Centralized maintenance: all maintenance employees report to a central maintenance manager
- Decentralized maintenance: maintenance employees are assigned to a specific area or unit
- Matrix structure: a hybrid form of centralized and decentralized structure

There are three common types of decentralized maintenance organization: **location-oriented**, **object-oriented** and **task-oriented**. In location-oriented maintenance organizations, every production sector gets assigned one maintenance workshop. It is possible to fully implement maintenance into the production department, in which case production managers are also maintenance managers. Complex maintenance activities are mostly executed by the central maintenance department or an external service provider. The object-oriented approach focuses on specific maintenance objects such as turning machines, gears, etc. With this kind of specialization, it is possible to develop competence centers. In a task-oriented organization, the maintenance members focus on special maintenance tasks, including inspection, oil change or cleaning. Thus, it becomes possible to create specialist groups and execute maintenance actions more efficiently. (Strunz, 2011, pp.543-545)

For the present thesis, the discussion of centralization vs. decentralization is crucial as it can be one factor to rise flexibility and reduce bottleneck times. Therefore, this issue is discussed in more detail in the following chapter.

Centralization vs. Decentralization

Higgins (1995, p.1.15) and Ben-Daya et al. (2015, p.7) discuss the topic centralization vs. decentralization extensively. The next section is a summary of both authors regarding this subject. To begin with, there exist numerous advantages of both centralization and decentralization of maintenance. Advantages of centralized maintenance are listed below:

- Better dispatching possibilities due to a more diverse craft group
- More and higher quality equipment
- Easier interlocking of craft effort
- Possibility of specialized supervision
- Better equipped training facilities
- More flexibility and improvement of utilization of resources.

The advantages of decentralized maintenance units are also listed below:

- Shorter response times due to reduced travel time to the workplace
- Better knowledge of equipment due to repeated experience
- Closer alliance with the objectives of a smaller unit improve applications
- Greater interest, therefore better preventive maintenance
- Improved relationship between maintenance-and production

It is remarkable that the last three points of the advantages of a decentralized maintenance department go along with the advantages of TPM policy (cf. section 2.1.6).
Higgins notices that neither one is the solution for each and every difficulty arising in maintenance. It has been proven in many organizations that implementing both centralized and decentralized maintenance systems is the most effective approach. A centralized maintenance group can provide a pool of craftsmen when major work is handled. In a completely decentralized maintenance unit, this would mean that the employees are located far away from the optimum needs of the area; furthermore, coordinating a bigger job within separate maintenance units would cause more difficulty. On the other hand, downtime can be minimized by a decentralized group which can give immediate attention to smaller maintenance problems. A smaller sphere of production equipment improves the performance of the employees as they area more familiar with the machinery. Higgins

also claims that good overall efficiency can be reached through decentralization of a certain number of less specialized maintenance workers who are supported by a small number of specialized craftsmen in order to provide emergency service in their area. He suggests that, for determining the number of maintenance employees in a decentralized area, a comprehensive study of the required services needed to sustain production has to be made. The preventive maintenance program can provide a reservoir of work during periods of fewer breakdowns. Ben-Daya et al. claim that, in decentralized maintenance organization, departments are assigned to specific locations. This reduces the flexibility of the overall organization due to reduced skill range and less efficient utilization. He suggests a hybrid form of centralized and decentralized systems, which he calls a cascade system.

The number of employees in a group usually rises with centralization (i.e. if no additional hierarchy levels are implemented). Hence, it seems reasonable to briefly discuss the effects of group size in the next section.

Group Performance

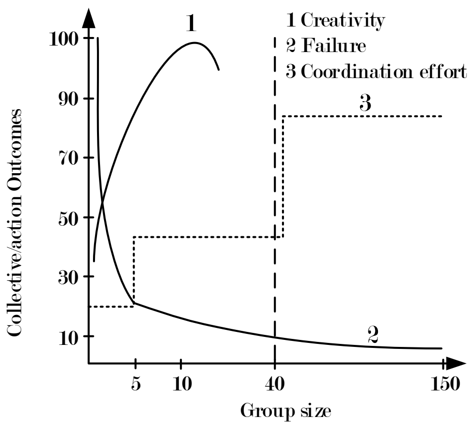


Figure 2.12: Group performance dependent of group size, adapted from Grunwald (1996) (see also Clemm, 1985)

A group is more than a collection of individuals, but rather a relatively closed, social and dynamic organism with own laws. When considering the group as a system with interaction possibilities dependent of the number of people n, the number of interactions equals $\frac{n\cdot(n-1)}{2}$. Hence, there are ten interactions in a group of five, 21 interactions in a group of seven, and 36 interactions in a group of nine. It becomes obvious that the complexity increases with each additional employee entering the system. The above-described interactions can be illustrated as losses which reduce the efficiency and the work performed in the team. Hence, if the losses increase, the performance is reduced drastically, which makes it necessary to find an optimum number of group members. A group has the best output when it has seven members. Fig. 2.12 outlines the group performance depending on the number of group members. It is obvious that the failure rate decreases dramatically until the group size reaches five individuals. The peak of creativity is reached at about ten group members. Another remarkable point is the effort for coordination which rises stepwise, at five and then shortly after 40 people. (Grunwald, 1996, pp.742-743)

Due to centralization it may be possible that the number of people in a group is higher than 40. To avoid this step-wise rising effort, scheduling algorithms can be the tool of choice to reduce the management effort to an acceptable degree.

2.3 Optimization Methods and Scheduling

This section focuses on the importance of scheduling in maintenance. Firstly, the fundamentals of optimization methods, with its modeling process and solving techniques is described. Subsequently, a short overview about scheduling is provided, discussing its numerous definitions and classifications. Lastly, similar applications are described, in order to adapt the ideas and solutions to the own present problem.

2.3.1 Optimization Methods

Generally, mathematical optimization can be described as finding the ideal solution to mathematically defined problems, for instance in the field of physics or management systems. In the first case, solutions often have to be found that correspond to minimum energy configurations. The second case deals with the economic importance in regard to industry and society and it needs to make decisions which ensure, for example, minimum cost or maximum profit. (Snyman, 2005, p.2)

The Modeling Process

The foundation of optimization is the problem defined in a model. The question that now arises is what is a model and why is it used? Dym (2002) describes is as follows:

> "**model** (n): a miniature representation of something; a pattern of something to be made; an example for imitation or emulation; a description or analogy used to help visualize something (e.g., an atom) that cannot be directly observed; a system of postulates, data and inferences presented as a mathematical description of an entity or state of affairs" (Dym, 2002, p.3)

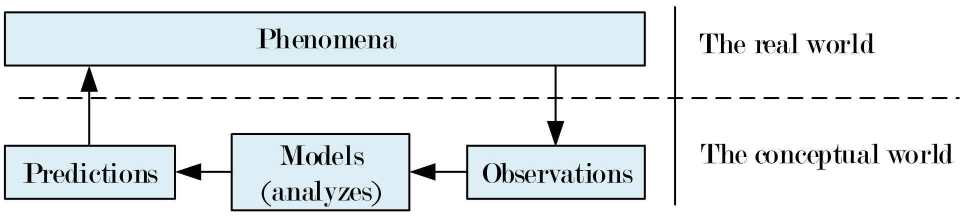


Figure 2.13: Conceptual Model, adapted from Dym (2002, p.7)

Furthermore, a mathematical model can be defined as a representation of behavior of real devices and objects in mathematical terms. From a scientific point of view, there is the "real" and the "conceptual world". In the external world, which is the equivalent of the real world, various phenomena and behaviors can be observed. The conceptual world, where one tries to understand what is going on in the real world, is the world of the mind. This can be illustrated with three stages: **observation**, **modeling**, and **prediction** (see fig. 2.13). In the observation part, the real world is investigated and measured and empirical data is gathered. The modeling part is about analyzing the observations and developing models which

- describe that behavior,
- explain why that behavior occurred, or
- allow to predict the future behavior.

In the prediction part of the method, the models are exercised in order to illustrate what happens in a set of events in the real world. These predictions are followed by observation, with which the model can be validated or rated as inadequate. (Dym, 2002, pp.4-5)

Models are idealized representations expressed in terms of mathematical symbols. Laws of physics such as $E = mc^2$ and $F = ma$ are well-known examples. In a similar way, mathematical expressions and systems of equations can model a business problem. Hence, if there are n possible decisions to be made, they are represented as **decision variables** $(x_1, x_2, ..., x_n)$ whose values have to be determined. With this variables, a measure of performance can be expressed as their mathematical function $(e.g, P = 3x_1+2x_2+...+5x_n)$. This equation is also called **objective function**. Restriction on the values can be assigned with inequalities or equations (for example, $x_1 + 2x_1 + x_2 + 3x_3 \leq 10$). Such mathematical expressions are named **constraints**. The constant values in the objective function and constraints are called **parameters**. The model's problem may be to vary the decision variable to minimize or maximize the objective function. (Hillier and Liebermann, 2000, p.11)

The modeling process of a real-world problem requires a four-step cyclic performance which is outlined in fig. 2.14. The four steps are (Snyman, 2005, p.5):

1. Observe and study the real-world situation associated with the practical problem
2. Abstract the problem by mathematics, that is described in fixed model parameters $\mathbf{p}$, and variables $\mathbf{x}$, the latter has to be determined, so the model performs in an acceptable manner
3. Solve the mathematical model that requires an parameter dependent solution $\mathbf{x}^*(\mathbf{p})$
4. Evaluate the solution $\mathbf{x}^*(\mathbf{p})$ and its practical usage; if necessary, adjust parameters and refine the model and run through loop again

Linear Programming

Linear programming is a special type of an optimization problem. It deals with the minimization or maximization of a linear objective function in consideration of linear defined constraints (Künzi et al., 1967, p.11).
Basically, there are two challenges in linear programming. The first is to obtain a model in a correct, understandable form, and the second is to reformulate it so that the problem can be solved. Although this formulation has strict limitations, it has a broad field of applications due to powerful algorithms which can solve linear problems in a fast and

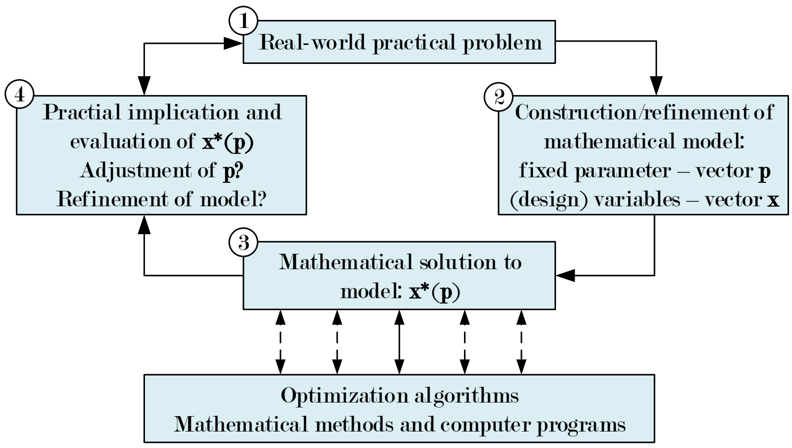


Figure 2.14: Modeling process, adapted from Snyman (2005, p.5)

easy way. Thus, if it is generally possible to formulate a problem in a linear way, it is generally worth the effort. (Wallace, 1989, pp.1 - 2)

The mathematical definition of a linear program in standard form is to find values of $x_1 \geq 0$, $x_2 \geq 0$,...$x_n \geq 0$ and min z satisfying (Dantzig and Thapa, 1997, p.7):

$$\begin{array}{ccccccccc} c_1x_1 & + & c_2x_2 & + & \dots & + & c_nx_n & = & z(min) \\ a_{11}x_1 & + & a_{12}x_2 & + & \dots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \dots & + & a_{2n}x_n & = & b_2 \\ \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ a_{m1}x_1 & + & a_{m2}x_2 & + & \dots & + & a_{2m}x_n & = & b_m \end{array} \tag{2.1}$$

In matrix notation the equations 2.3.1 can be written as:

$$\begin{aligned} min \quad & \mathbf{c}^T\mathbf{x} \\ \text{s.t.} \quad \mathbf{A}\mathbf{x} \geq \mathbf{b}, & \quad \mathbf{A} : m \times n \\ \mathbf{x} \geq 0 & \end{aligned} \tag{2.2}$$

In many practical applications, the optimum of the model must be represented in integer values in the variables x_n. It is often necessary to assign machines, people, or vehicles, which can be accomplished by using integer values for variables. The mathematical models together with all variables as integer are referred to **integer programming**. If only some variables have to be integer values, this model is referred to as **Mixed Linear Integer Programming** (MILP). (Hillier and Liebermann, 2000, p.576)

Solving Techniques for Linear Programming

In the last section, the rules of the statement for the mathematical problem were defined. However, it is impossible to find an optimal solution with this regulation only. Indeed, there exist several solving strategies for linear problems. The **Simplex method** was developed by Georg Dantzig in 1947 and it is still a very efficient method to solve huge linear problems on computers. Although it is an algebraic procedure, its concept is based on a geometric foundation. Fig. 2.16 outlines a linear problem. Each constraint boundary forms the boundary of the feasible region. The points of intersection are corner point solutions. The five points that lie on the feasible regions, are the corner point feasible (CPF) solutions. In order to solve the solutions, a certain procedure has to be followed which is presented in algorithm 1. (Hillier and Liebermann, 2000, pp.110-111)

Algorithm 1: Simplex algorithm

```
Choose 0 as the initial CPF solution;
while Solution is not optimum do
    if Adjacent CPF solutions do have a higher objective value then
        Solution is not optimum;
        Choose adjacent CPF solution with higher objective value;
    else
        CPF is solution;
        Solution is optimum;
    end
end
```

The problem with this algorithm, is that it mostly does not return the solution as integer values. Integer problems need more sophisticated solvers, one of them being the **Linear Cutting Plane** solver (fig. 2.15). The latter uses a linear programming solution as a starting point (e.g. determined by a Simplex algorithm). Afterwards, it tries to reduce the feasible region by adding functional constraints (cutting planes) until the optimum integer solution is found. The very first algorithms developed by Ralph Gomory for integer programming used this cutting plane method. However, such algorithms provided unsatisfactory results for most problems. Nevertheless, new methods combine cutting plane and branch-and-bound techniques in order to provide powerful algorithms for solving large-scale binary integer problems. (Hillier and Liebermann, 2000, pp.628-629)

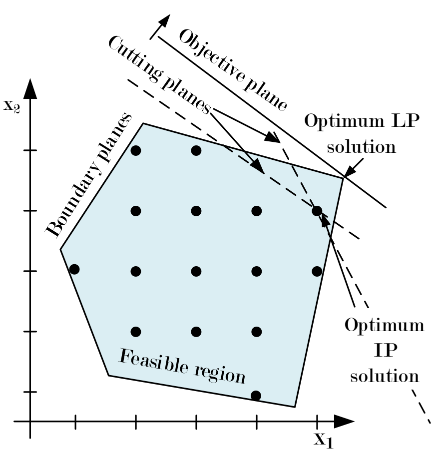


Figure 2.15: Linear cutting plane strategy, adapted from Gurobi (2016)

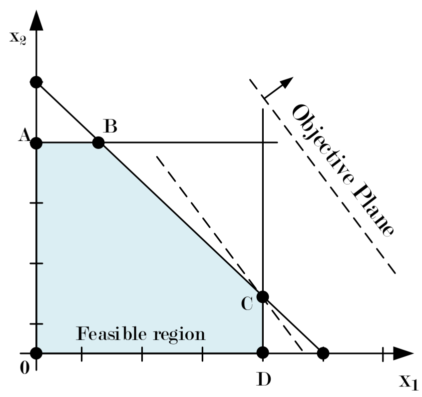


Figure 2.16: Simplex algorithm, adapted from Dantzig and Thapa (1997, p.65)

The basic idea of **Branch-and-Bound method** is to work through a structured decision tree in a systematic way. The algorithm proceeds by repeatedly partitioning all feasible solutions into smaller subclasses until the optimal solution is reached. The partitioning (branching) is carried out so that each subclass of solutions is mutually exclusive and all inclusive, so that each feasible solution belongs to exactly one class. For each subclass, the solver computes an upper bound. If this is lower than the objective value of a previously found solution, this branch will no longer be considered. Upper bounds are the basis for the next stage of branching, and new upper bounds for the next subclasses are computed. This procedure is repeated until the optimal feasible solution is found: it has an objective value higher than the upper bounds of all subclasses and higher than all other feasible solutions found (Kolesar, 1967).

This method has a tremendous advantage over a simple brute-force algorithm. It is not necessary to compute all solutions when whole branches can be terminated and removed from the scope, thus it is much faster than a brute force algorithm and still able to compute optimal solutions.

The methods described above are always able to provide an optimum solution, if there is one. In many cases heuristic algorithms provide good solution, although they are usually not optimal. The **greedy algorithm** is a very common approach based on heuristics, which can be used for a linear model, but is not restricted to it. Optimization algorithms typically go through a sequence of steps, with choices for each one. A greedy algorithm always decides for the choice that currently seems best. It thus makes a locally optimal choice, hoping that it will lead to a globally optimal solution. In every step of the sequence, the solver makes a certain choice. The greedy algorithm prioritizes this choices with the **greedy-choice property** (GCP). When considering which choice to make, it makes the one that seems best for the current problem – without taking into account results from future sub-problems. Even if this does not always yield the ideal solution, many problems can be solved optimal in this way. (Cormen et al., 2009, pp.414-424)

2.3.2 Scheduling

Scheduling is concerned with the allocation of scarce resources to activities with the goal of optimizing one or more objectives. Depending on the situation, resources and activities can take various different forms. Resources can be processors in a CPU, nurses in a hospital, machines in a workshop, or maintenance employees. Tasks may be executions of computer programs, duties of nurses, operations in a production process, or maintenance activities. The objectives can have different forms as well: e.g. minimizing lead time, or maximizing the number of tasks completed after their due date. Scheduling is a decision-making process that plays an important role in many disciplines. (Leung, 2004; Pinedo, 2002, p.1)

A schedule is a tangible plan, e.g. a class schedule, and it normally informs about when things are supposed to happen. A schedule provides a plan for timing certain activities and, assuming that the questioner is interested in the answer, it can respond to the question of “When will a particular event take place (if all goes well)?” Hence, answering when-questions usually includes information about timing. An appropriate answer may be "Class starts at 9:00 o'clock". However, another useful answer may be that the class starts when the prior one has finished. Thus, the when-question can be answered in two ways: either with a focus on timing, or with a focus on the sequential information of the schedule. Actually, these two answers are often the approach taken when generating the schedule: first its activities are sequenced and in the second step, they are scheduled. (Baker and Trietsch, 2009, p.1)

Pinedo (2004, pp.8-9) thus distinguishes planning and scheduling in an enterprise operating in the field of manufacturing and services. In the manufacturing environment, orders have to be transformed into jobs with due dates. Often, these jobs have to be processed on machines in a specific sequence or order. However, these machines may be busy and jobs might be delayed; moreover, unexpected events such as machine breakdowns may take place. A detailed schedule in such an environment can help to maintain control of operations and efficiency. The scheduling process on the shop floor also impacts the long-term production planning process as well as the inventory system. Decisions at the higher planning level may directly affect the detailed scheduling process. Describing the planning and scheduling system of a generic service organization is normally not as easy as describing a generic manufacturing system. In fact, service organizations often face different problems as they may have to reserve resources (e.g. classrooms, trucks, or others), or allocate, assign and schedule workforce or equipment – hence, the algorithms may be incompatible with methods of manufacturing.

Modeling a Scheduling Problem

Baker and Trietsch (2009, p.2) structure scheduling problems as follows: they consist of a set of tasks which has to be executed, and a set of resources to perform these tasks. The general problem relates to two aspects: timing the tasks while recognizing the capability of resources. To determine whether the tasks can be feasibly finished, one needs to know the type and amount of each resource. When specifying the resources, the boundary of the scheduling problem is effectively described. Furthermore, the tasks must be described with information such as the resource requirement, the duration, and the possible time window. This information about tasks and resources defines the scheduling problem. The objective function should include all costs that influence scheduling decisions even if they can often not be measured and are hard to identify. However, there are three common types of decision-making goals in scheduling:

- **Turnaround** measures the time necessary for completing a task
- **Timeliness** measures the conformance of a successfully accomplished task to a given deadline
- **Throughput** measures the amount of completed work in a defined period of time

The scheduling models are categorized by specifying the configuration of resources and tasks. For instance, a model may consist of one or several machines which would differ between single- or multi-stage models. Additionally, if the set of jobs varies over time, the system is called dynamic; if the jobs remain constant, it is called static. In scheduling problems, there are two types of feasibility constraints: on the one hand, there is limited capacity of resources and, on the other, there are technological restrictions such as the sequence of tasks or resources. Therefore, solved scheduling problems answer two questions:

- Which resource should be allocated to each task?
- When should the tasks be performed?

Although the formulation seems rather simple, solving the problem with an appropriate model is a fairly complex procedure.

Classification of Scheduling Problems

The field of scheduling is large and many problems which are termed as scheduling do not have much in common with the actual problem definition of this thesis. Due to this, it pays off to classify the field of scheduling into categories in order to facilitate the literature research.

Wren (1996) classified the problems of scheduling in **Scheduling**, **Timetabling**, **Sequencing** and **Rostering**.

Scheduling is the allocation of resources to objects, placed in space-time, with subject to constraints. Objective is to minimize the total cost of some set of the resources used.

Common examples are problems of delivery vehicle routing or transport scheduling, which minimize the number of drivers or vehicles, in order to minimize costs. Another example is the job shop scheduling problem, which may minimize the number of time periods used.

Timetabling is the allocation of given resources to objects, place in space-time, subject to constraints, in such a way to satisfy as possible a set of desirable objectives. Common examples are examination and class scheduling, and forms of personnel allocation.

Sequencing is the construction of an order, subject to constraints, in which activities have to be carried out, or objects are placed in some representation of a solution. Examples are travelling salesman problem, and the flow shop scheduling.

Rostering is the placing of resources into slots in a certain pattern, with subject to constraints. One may simply obtain a feasible allocation, or minimize some objective. Often resources rotate through a roster at this problem

2.3.3 Related Work

This section reviews various problems of literature which are related to the MSSP regarding modeling technique or solving procedure. The latter can be classified into three different methods. For many mathematical models, there are algorithms which lead to an exact solution, which means that the solution is optimal. Thus, they are called **exact** algorithms. **Approximative** methods normally do not lead to an optimum solution but deviations of the ideal solution can be estimated. However, there are problems for which the solution by exact or approximative methods is very complex or even impossible. **Heuristic** methods can be one way to avoid this dilemma. In contrast to the other two algorithms – where exact solutions or estimations to them are given – heuristic algorithms cannot grant the certainty that the delivered solutions are good or even acceptable. Although this is a huge disadvantage, heuristic algorithms are in fact suitable for various applications. (Hauke and Opitz, 2003, pp.12-13)

Exact Solution Methods

Breslaw (1976) formulates the faculty assignment problem with a linear programming model. This assigns faculties to courses with certain constraints like a maximum workload for each faculty. A benefit matrix is described in order to find the optimal solution. Although integer values are necessary for this kind of formulation, the problem is solved by an ordinary Simplex algorithm as the structural formulation of the constraint set guarantees that the extreme points of the feasible region will be described with integer values.

Junginger (1986) states that a timetable is a schedule of meetings of classes (or students) and their teachers over a definite period of time. It requires certain resources (e.g. rooms) and fulfills a number of requirements. Timetables are designed to meet restrictions like

- certain subjects can only be taught in specific rooms
- the availability of teachers or classes is dependent on time
- no teacher or class is allowed to be allocated at the same time more than once
- classes or teachers can be combined for certain lessons
- there may be objects which require two consecutive hours

Junginger presented an assignment problem. He implemented the variable x_{cth} which is 1 when class c is given lesson by teacher t in hour h. A linear objective function 2.3 was introduced, in which d_{cth} can be assigned to a large value when the assignment of teacher t, to class c, at hour h is less desirable.

$$z = \sum_{c \in C} \sum_{t \in T} \sum_{h \in H} d_{cth} \cdot x_{cth} \tag{2.3}$$

Gelinas and Soumis (2005) use a Dantzig-Wolfe decomposition for a job-shop problem with a sup-problem for each machine. Each single machine represents a sub-problem as a sequencing problem with time windows. The algorithm presented is of the branch-and-bound variety, the lower bound is obtained using a Dantzig-Wolfe decomposition. Gelinas and Soumis use a primal approach and precedence constraints, mostly due to faster convergence. The problem is formulated as a non-linear one with disjunctive constraints. Relaxing the precedence constraint presents a problem that is individual to each machine.

Approximative Solution Methods

Garey et al. (1978) introduce efficient algorithms that find schedules to be almost optimal. They present an introduction to approximation methods for scheduling, outlining several algorithms that can provide a worst-case performance ratio for multiprocessor scheduling problems. Their approach is to use a performance guarantee defined by $A(I) \leq r \cdot OPT(I) + d$, where d and $r \leq 1$ are specified constants, $OPT(I)$ is the optimum objective value for instance I, and $A(I)$ denotes the objective value found within a feasible solution. Garey et al. analyze the algorithms and find the smallest r for each of them. With this information, they are able to define bounds which each is at least able to reach.

Lenstra et al. (1990) outline a scheduling problem with m parallel machines and n independent jobs. Each job must be assigned to one of the machines and requires a certain processing time. The objective is to find the schedule with the shortest makespan. Therefore, they present a polynomial algorithm which guarantees being within factor 2 of the optimum makespan. This is almost as good as possible, in the sense that no polynomial algorithm can prove a factor less than 3/2. Before this paper was published, the best algorithm was able to deliver solutions with $2\sqrt{m}$ optimum makespan, where m is the number of machines.

Heuristic Solution Methods

Colorni et al. (1994) introduce a genetic algorithm (GA) to solve the timetable problem. The algorithm can construct a timetable of classes where teachers are assigned one or more subjects to teach in two or more classes for a total of eighteen hours a week. The objective is to minimize the infeasibility rate and the didactic, organizational and personal costs by following constraints: every teacher and class must be given a number of hours, there is a maximum of one teacher per class at the same time, no teacher can be in two classes at the same time, and every class must be assigned to a teacher. In order to avoid infeasible solutions, the following corrections are applied to the GA:

- The crossover can be changed in a way that it can be applied consistently
- New crossovers and mutation operators must be defined to generate only feasible solutions
- Apply crossover and mutation operators and make some kind of genetic repair afterwards, so that infeasible solutions change to feasible ones through the use of a filtering algorithm

The fitness function is a linear formulation of the terms of the objective function. The algorithm was implemented at a high school in Milan and the solutions were compared to the manual timetables from the preceding year. The total cost of the hand-designed timetable was 234, while the GA generated a timetable with the objective value of 91.

Nowicki and Smutnicki (1996) outline a fast approximative Tabu Search (TS) algorithm for finding a minimum makespan in a job-shop. In the introduced algorithm, a substantially small neighborhood was defined using so-called blocks of operations. A block is a subsequence of consecutive operations that enable the algorithm to significantly reduce the number of allowed neighborhood moves. Furthermore, a move is only executed if it directly benefits the objective function. The initial solution for the taboo-search is provided by an insertion technique which is primarily applied to flow-shop problems. Due to these implementations, the algorithm is not only capable of finding a shorter makespan than the best approximation approaches, but it also runs for a shorter period of time.

Wildeman (1997) developed another interesting approach by demonstrating that grouping maintenance activities saves costs because of the reduced number of set-ups. His approach considers a multi-component system with n components. On each component, a preventive maintenance activity can be carried out that comes with an ideal execution date. If maintenance activities are grouped, the set-up costs only have to be paid once. Due to grouping to the same date, it may be that maintenance activities are moved from the ideal execution date. For this reason, a penalty function with a parabolic form is introduced. Costs would arise when a maintenance activity is executed after or before the ideal date.

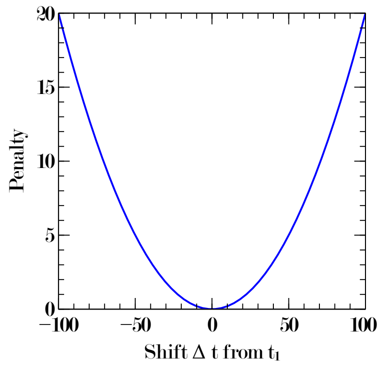


Figure 2.17: Penalty curve for moving maintenance activities, adapted from Wildeman (1997)

Hojati and Patil (2011) demonstrate a way to schedule skill-based, part-time employees with an integer linear programming-based heuristic. They applied their method to the service industry, where every employee possesses a special set of skills. They used a soft constraint approach for the skill: basically, every employee is allowed to do all tasks if they are available, but if the employee does not have the skill required for the task, he has to pay a penalty. The objective of the model was to minimize the overall working hours and the employees' deviation from their weekly work hours. They combined this approach with a Pareto principle. Due to the high amount of constraints and variables, this model is hard to solve optimally, i.e. for a reasonable example, there are about 40,000 binary and integer variables and 4,000 constraints, and they reached a computation time of nine hours. Their solution to this problem is thus a decomposition of this model: each day is solved separately and joined to a global solution. This approach reduces the computation time for each day to approximately a second and is thus solvable by Excel's standard solver.

Sauppe et al. (2015) deal with a problem to assign a significant amount of meetings to meeting rooms. The meetings have attributes including the number of participants, necessary equipment like beamers, and a meeting date. The meeting rooms which are allocated to the meetings have to at least fulfill these requirements. The objective of this algorithm is to schedule as many panel meetings as possible. If it does not succeed in scheduling all panels, it is permitted to move the date to the near future or to the past. The authors implemented three different algorithms and tested them with eight different datasets. For implementation, they chose a depth-first search (DFS) in a time-constrained branch-and-bound approach, a greedy approach and a random based approach. In both historic and randomly generated datasets, the greedy algorithm was the solver with the highest scheduling rate. Although it was not the fastest, its speed was comparable to the random solver, whose scheduling rate was significantly lower. In all cases, the time-constrained DFS was the slowest algorithm. Another remarkable point is that it reduced the complexity of its problem dramatically by breaking down the time horizon to several days instead of months.

Chapter 3

Problem Definition

This chapter defines the MSSP. The first section outlines the requirements determined from the needs of the TPM 4.0 project partner. The second section uses these requirements to formulate a mathematical model that can then be solved with algorithms.

3.1 Requirements

In order to formulate the mathematical conception, the requirements of the MSSP are defined in this section. The task of the algorithm is to define a maintenance schedule which combines both efficient workload distribution among maintenance employees and an optimum allocation of maintenance tasks to production lines in the space of time.

In fact, tasks have certain requirements regarding qualification. For instance, some of them can only be performed by specific employees possessing certain certifications. As described in section 2.1, most maintenance employees have the necessary specialization in the field of mechanics or electrics. Some of them may have more than one specialization; for example, a long-time employee might be able to do more tasks although his or her education originally focused on only one field. There may also be employees who supervise the staff performing maintenance actions. Hence, there can be several complex combinations of skills and specializations. Which employee is able to do which task can usually by deducted by a so-called qualification matrix, a means that is indeed very common in enterprises. With regard to allocating tasks, the optimum case would be that an employee performs tasks for which he or she is barely qualified. In this case, another, better qualified, employee is not assigned and has free capacity for unpredictable maintenance actions.

As described in chapter 2.2.3, there are numerous benefits of performing work in teams, such as increasing creativity or motivation. Although it would be technically possible to assign each task separately, it is usually much better to use teams in order to jointly accomplish tasks by linking them and performing them at the same time. Most of these task bundles usually take about eight hours of time, which corresponds to the length of one shift. In the present case, such a bundle is called a TPM shift. With this structure, team members of the shift could arrange themselves and help each other if necessary. Thus, planning whole task bundle shifts generates a scheduled preventive maintenance plan, but it also creates freedom for the team and its individuals. TPM shifts are performed once a week and should always be scheduled on (or approximately on) the same weekday in order to provide a consistent maintenance period.

The maintenance employees have limited availabilities. The partner company of the present project TPM 4.0 operates the production and maintenance departments in three shifts. Thus, in the model, the staff is only partially available during the timeframe and the algorithm must be capable of handling limited availability of the employees. The latter can be part of the decentralized department or of the centralized maintenance department, which allows the worker to only work on the line he or she is assigned to or to perform tasks on all production lines.

The production line has limited availabilities as well: there may be production batches with a high priority and, if such batches are produced, no maintenance must be allowed at the production line in order to be able to deliver on time. The best date for maintenance would be when no production is scheduled on the line, i.e. it is shut down anyway. However, there are also timeslots which are unfavorable with regard to the maintenance or production departments in a TPM shift. In order to handle these, the algorithm must process the production plan and match it to the maintenance schedule.

In order to schedule the maintenance actions, the algorithm needs the following input:

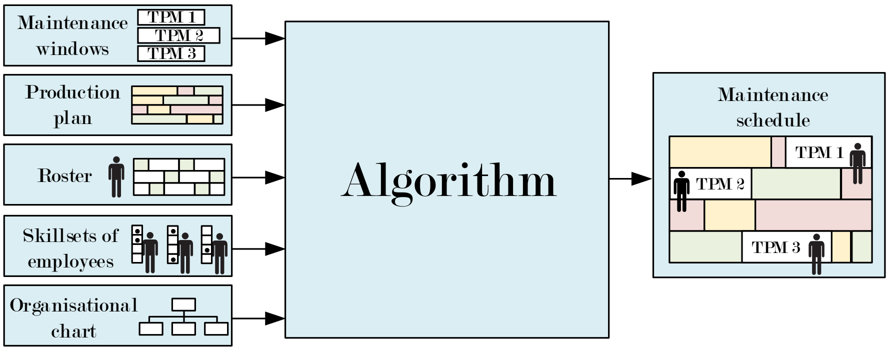


Figure 3.1: Input-output algorithm

- TPM windows, with skill requirements, ideal date, and tasks
- Production priority plan for each line
- Roster of employees
- Skill matrix of employees
- Organizational chart

The requirements are quite similar to a scheduling problem as described in section 2.3.2. The algorithm must allocate resources to objects and to a certain date. In the present case, resources can be represented as employees, while objects represent the tasks of the TPM shift. The objective function must consist of several terms: it must penalize over-qualification of employees, moving shifts to another date than the desired one, or undesirable dates with regard to the production system.

3.2 The Problem Formulation

The MSSP can be formulated as described in the following. Each TPM shift j possess a set of task T_j. The task-bundle j is allocated to a production line l and must be assigned to suitable timeslots k. Fig. 3.2 outlines an example with five employees who are assigned to three TPM shifts, which are allocated to three different lines. The timeframe is represented by four timeslots. Each task t must be assigned to a resource i and to consecutive timeslots k. Resource i can only be assigned if the resource is fulfills the requirements of the task. Fig. 3.3 represents the task allocation to resources from the same example as fig. 3.2. Each employee is allocated to the task on the same row, e.g. task 4 to employee 2. The dots in the boxes represent requirements, i.e. qualifications or skills. It becomes obvious that all employees fulfill the requirements; only employee 1 is overqualified with regard to skill 2. The figure also illustrates the allocation of tasks to TPM shifts, e.g. TPM1 owns task 1 and task 4.

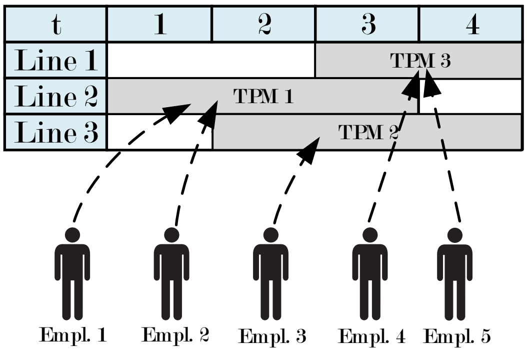


Figure 3.2: Scheduling of TPM shifts to timeslots

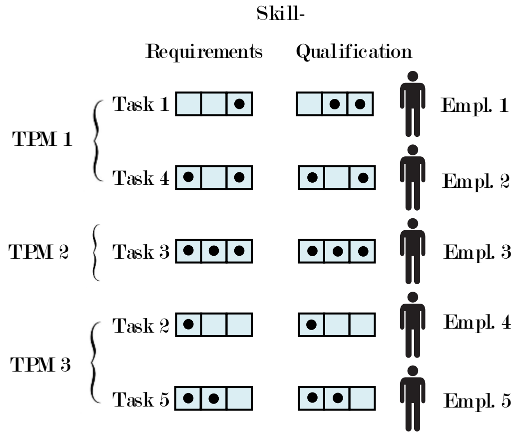


Figure 3.3: Allocation of employees to tasks, considering skills

The assigning of shifts to production lines and workers has two further restrictions: the production line as well as the employees are not always available. Fig. 3.4 and 3.5 show availability matrices of production lines and employees. Each dot in the matrix represents an available production line or employee at a specific timeslot, e.g. line 3 is available at all four timeslots in 3.4 and employee 2 is only available from timeslots 1 - 3. This means that TPM shifts can only be assigned to production lines or employees when they are available (cf. fig. 3.2)

t	1	2	3	4
Line 1			•	•
Line 2	•	•	•	
Line 3	•	•	•	•

Figure 3.4: Availability of production lines

t	1	2	3	4
Empl. 1	•	•	•	
Empl. 2	•	•	•	
Empl. 3	•	•	•	•
Empl. 4			•	•
Empl. 5		•	•	•

Figure 3.5: Availability of employees

As described in the prior section, the maintenance department is hierarchically structured and not all employees are allowed to work on all lines. Fig. 3.6 shows the organizational structure of the above example, illustrating that employees 3 and 4 can execute tasks on all lines, while employees 1, 2 and 5 are only allowed to work on the lines they are allocated to. This form of organizational structure can be transformed to a matrix (fig. 3.7), each dot representing a permission of an employee to work on the line stated in the column.

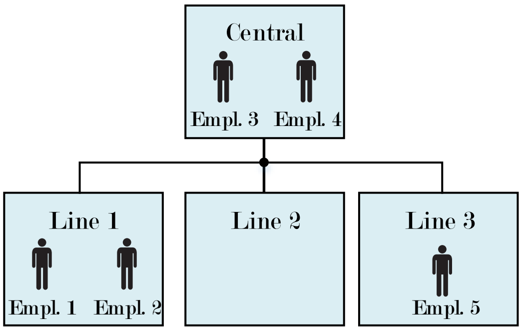


Figure 3.6: Organizational structure

	Line 1	Line 2	Line 3
Employee 1	•		
Employee 2	•		
Employee 3	•	•	•
Employee 4	•	•	•
Employee 5			•

Figure 3.7: Organizational structure in matrix form

3.2.1 The Linear Model

In the prior section, the constraints of the MSSP were defined with graphical tools. In order to find optimal solutions, these problems have to be formulated in a mathematical way. In this section, the model is formulated as a MILP assignment problem.

Sets and Indices

$i \in I$: set of resources i

$l \in L$: set of production lines l

$j \in J$: set of TPM shifts j to be scheduled

$t \in T$: set of task

$k \in K \equiv \{1, 2, .., \kappa\}$: set of consecutive time slots

$p \in P$: set of skills

Decision Variables

$U_{ikt} \in \{0,1\}$, unity if resource i is scheduled to task t on timeslot k
$V_{jk} \in \{0,1\}$, unity if TPM shifts j is scheduled on timeslot k
$X_{jk} \in \{0,1\}$, unity if TPM shifts j starts at timeslot k
$Y_{it} \in \{0,1\}$, unity if resource i is allocated to requirement t
$W_j \in \{0,...,T\}$, start timeslot k of TPM shift j

Parameters

$A_{ik} \in \{0,1\}$, unity if resource i is available at timeslot k
$B_{jl} \in \{0,1\}$, unity if TPM shifts j is allocated to line l
$C_{jt} \in \{0,1\}$, unity if task t is allocated to TPM shifts j
$F_{ip} \in \{0,1\}$, unity if employee i possesses skill p
$F_{tp} \in \{0,1\}$, unity if task t requires skill p
$G_{it} \in \{0,1\}$, unity if $\quad F_{pi} \geq F_{pt} \quad \forall i, p, t$
$M_{il} \in \{0,1\}$, unity if resource i is allowed to work on production line l
$Z_{lk} \in \{0,1\}$, unity if line l is available at timeslot k
$d_j \in \mathbb{N}$: duration of TPM shift j in timeslots

Constraints

$$\sum_{l \in L} (M_{il} \cdot B_{jl}) = S_{ij} \quad \forall i, j \tag{3.1}$$

$$\sum_{j \in J} (S_{ij} \cdot C_{jt}) = O_{it} \quad \forall i, t \tag{3.2}$$

$$\sum_{i \in I} Y_{it} = 1 \quad \forall t \tag{3.3}$$

$$\sum_{k} X_{jk} = 1 \quad \forall j \tag{3.4}$$

$$W_j = \sum_k k \cdot X_{jk} \quad \forall j \tag{3.5}$$

$$-W_j - \kappa \cdot (1 - V_{jk}) \leq -k + d_j \quad \forall k, j \tag{3.6}$$

$$W_j - \kappa \cdot (1 - V_{jk}) \leq k + 1 \quad \forall k, j \tag{3.7}$$

$$\sum_{k \in K} V_{jk} = d_j \quad \forall j \tag{3.8}$$

$$V_{jk} \leq \sum_{l \in L} (Z_{lk} \cdot B_{jl}) \quad \forall j, k \tag{3.9}$$

$$Y_{it} \leq G_{it} \quad \forall i, t \tag{3.10}$$

$$Y_{it} \leq O_{it} \quad \forall i, t \tag{3.11}$$

$$U_{ikt} \leq Y_{it} \quad \forall i, k, t \tag{3.12}$$

$$U_{ikt} \leq \sum_{j \in J} (C_{jt} \cdot V_{jk}) \quad \forall i, k, t \tag{3.13}$$

$$\sum_{k \in K} U_{ikt} \geq Y_{it} \cdot \sum_{j \in J} (C_{jt} \cdot d_j) \quad \forall i, t \tag{3.14}$$

$$\sum_{i \in I} U_{ikt} \leq 1 \quad \forall k, t \tag{3.15}$$

$$\sum_{t \in T} U_{ikt} \leq A_{ik} \quad \forall i, k \tag{3.16}$$

Some input variables have to be transformed to a different form of matrix in order to create easier formulations for the linear model. M_{il} represents the organizational structure of the department. Equation 3.1 enables resources to do tasks in certain TPM shifts j. Furthermore, 3.2 uses S_{ij} from the preceding equation to unlock resources i to tasks t. Equation 3.3 expresses that only one resource i can be allocated to a task t. Constraint 3.4 ensures that a TPM shift can only be executed once during the whole time frame. The set of equations from 3.5 to 3.8 is needed to model the decision variable V_{jk}, and to ensure that the TPM shift is allocated to a consecutive number of timeslots k. 3.5 transfers the binary start-timeslot of X_{jk} to the integer value W_j. Equation 3.6 sets an upper bound to V_{jk}, i.e. all V_{jk} of which timeslots k are larger than the end-timeslots of TPM shift j are 0. Constraint 3.7 sets all values of V_{jk} to 0 of which timeslots k are lower than the start-timeslot of TPM shift j. 3.8 sets all values of V_{jk} between start and end-timeslot of TPM shift j to 1. Formula 3.9 guarantees that production lines can only be maintained on dates they are available on. 3.10 ensures that only resources i, with suitable skill levels, can be allocated to task t. The next equation 3.11 guarantees that resources i can only work on lines l they are allowed to work on (see eq.3.1 and eq.3.2). 3.12 ensures that U_{ikt} can only be 1, when employee i is assigned to task t. Constraint 3.13 is needed to set all timeslots of U_{ikt} to 0, if the allocated TPM shift j of task t is not executed on timeslot k. Equation 3.14 sets U_{ikt} to 1, if the matching TPM shift j of task t is assigned to timeslot k and employee i. 3.15 avoids double booking of employees to the same task task t. 3.16 is needed, to allocate only resources, when they are available, and it also ensures, that every resources can only be assigned to one task at the same time. For a better understanding of the data structure of the in- and output data see section 8.1 in the appendix.

Costs and Objective Function

Every maintenance task has an optimal execution date. This assumption is derived from the damage model described in section 2.1.2. In case of changing a component, this could be a date at the end of the working-period of the bathtub curve. When looking at the time after the optimal date, probability of a component failure rises rapidly (cf. wear period 2.3) and the chances of higher financial costs due to subsequent damage costs increase. On the other hand, if a component is changed too early, there may be a rest-margin (see fig. 2.2) and a component with residual value might be disposed of. Thus, it is obvious that costs arise, independent of whether the component is changed earlier or later. The next question to address is what the penalty function looks like. Possible forms of penalty functions are displayed in fig. 3.8. Wildeman (1997) uses a parabolic such as curve b). Another approach would be to create a discontinuous function a), or an asymmetric curve c), which is directly derived from the wear margin or the bathtub curve. In the present thesis, the penalty function is formulated in a discontinuous way as a) due to little knowledge of actual tasks and TPM shifts in the partner company. A more sophisticated function seems reasonable only if the probability of failure is exactly known for each component. Equation 3.17 formulates the penalty matrix H_{jk}. For each task bundle j and timeslot k the costs can be determined by the optimum timeslot b_{opt} the actual timeslot k and the cost coefficient m_j

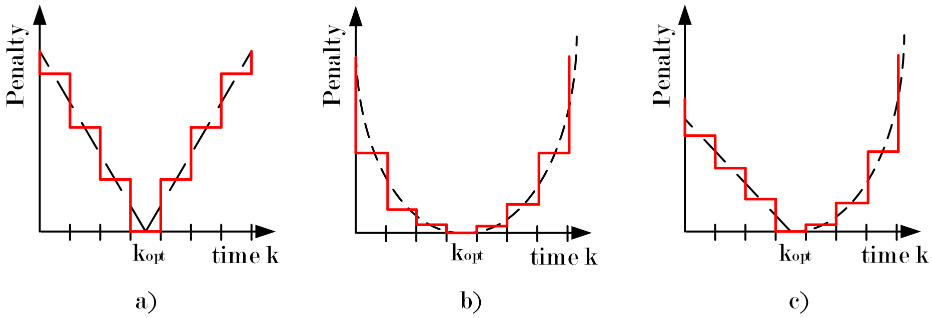


Figure 3.8: Different characteristics of penalty functions

$$H_{jk} = |b_{opt} - k| \cdot m_j \tag{3.17}$$

The second term of the objective function is used for qualification of the employees for each task. Every additional skill of an employee makes him or her capable of doing more tasks, hence its value increases. The most resource saving scenario would be to allocate a task to that employee who has those qualifications that correspond to the task requirements.

A high qualified maintenance worker doing a low-level task would seem irrational because the time could be used more efficiently to do high-level tasks instead. In order to penalize an allocation like this, every skill has a certain value. It would make sense that scarce skills have a high value and low level skills cost only little or nothing. Another way to model costs of skills would be to determine them from actual salaries of employees. Fig. 3.9 represents two possible examples for the allocation of employees to a task. In the upper case, the allocation is possible but not optimal due to over-qualification of the employee with regard to this specific task. Thus, this over-qualification must be penalized. The lower case shows another infeasible allocation due to a missing skill of the employee. Equation 3.18 subtracts the required skills of tasks in the same way as in fig. 3.9 with the available skills of resources and multiplies it with the length of the actual task.

$$Q_{it} = \sum_{p \in P} (q_p \cdot (F_{ip} - F_{tp})) \cdot \sum_{j \in J} (C_{jt} \cdot d_j) \quad \forall i, t \tag{3.18}$$

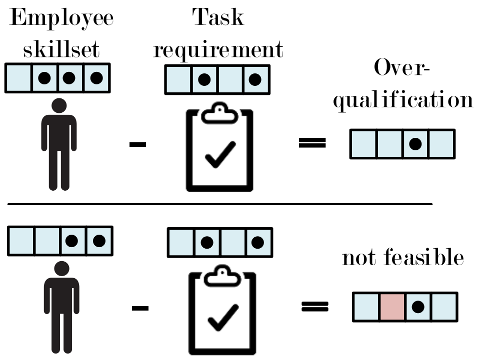


Figure 3.9: Qualification of employees

Equation 3.19 determines the downtime costs for each production line to equivalent costs of a TPM shift in order to find a linear combination in the objective function. The costs for each shift are outlined in the matrix E_{jk}. Every line l has a priority factor for each timeslot k which is derived from the production plan. If the priority factor D_{lk} is high, no maintenance should be performed in this time window. If the factor is 0, the line stands still, which means that maintenance would now be optimal. This downtime costs depends could be determined, by the lost revenue because of not producing any products.

$D_{lk} \in \mathbb{Z}^+$: Priority value for line l over timeslots k

$H_{jk} \in \mathbb{Z}^+$: Costs for maintenance window j if it starts on timeslot k

$m_j \in \mathbb{Z}^+$: Cost factor for moving maintenance window to another timeslot k

$n_l \in \mathbb{Z}^+$: Downtime cost factor for each maintenance window

$q_p \in \mathbb{Z}^+$: Cost factor for skill level p

$$E_{jk} = \sum_{l \in L} (B_{jl} \cdot (D_{lk} \cdot n_l)) \quad \forall j, k \tag{3.19}$$

The actual objective function consist of the terms of equations 3.17, 3.18 and 3.19. The objective is to minimize the sum of the costs:

$$min(Z) = \sum_{j \in J} \sum_{k \in K} (H_{jk} \cdot X_{jk} + V_{jk} \cdot E_{jk}) + \sum_{i \in I} \sum_{t \in T} (Y_{it} \cdot Q_{it}) \tag{3.20}$$

Chapter 4

Scheduling Algorithms for MSSP

This chapter describes four different algorithms for the MSSP. The first three algorithms were developed one after the other, which is why they have essential features in common and do only differ slightly from each other. Having been the starting point, the global algorithm is described first. In the second section, the local linear algorithm is described, which decomposes the global problem to several local ones that are separately solved. The pyramid algorithm is the most sophisticated algorithm and it is discussed in the third subchapter. The last algorithm outlined in the final section is the benchmark algorithm used for the case-study. Its intention is to imitate human behavior in order to answer the following question: To what extent would the schedule quality increase if the scheduling process would be switched from manual to automatic procedure?

4.1 Global Linear Algorithm

The global linear algorithm (GLA) is not only the first that was developed, but it is also the simplest possible with regard to the programming effort. It uses the linear model as described in section 3. This method can be classified as an exact algorithm. For solving the linear model, the commercial software package Gurobi (2016) was used. This is a state-of-the-art optimization tool for linear and non-linear problems, which provides the latest optimization algorithms on multi-core processors. With regard to MILP, it provides algorithms as cutting planes or heuristic and supports several interfaces such as C#, Python or AMPL. In fact, this makes it convenient to use on many platforms (Gurobi, 2016). C# is used as a framework for the algorithm and deals with data handling. The project's long-term objective is to obtain input data from the Enterprise Resource Planning (ERP) system, process it, and give the solution back to the system. Due to missing links, the first version of the algorithm uses text files as an interface. Its procedure can be seen in algorithm 2 below.

Algorithm 2: Global Linear Algorithm

Data: Read input files
Preprocess input matrices;
Solve global linear problem;
Postprocess solution to readable format;
Data: Write output files

4.2 Local Linear Algorithm

As described in section 2.3.2, Hojati and Patil (2011) demonstrate that one way of decreasing complexity and solving time in a model is to decompose it. Thus, they split their model's scope of scheduling into several shifts. When it comes to the maintenance problem, such a solution is not feasible because there are two kinds of scheduling. First, the TPM shift must be fixed to a certain date, and, second, the employees must be allocated to a maintenance window. This problem is more complex, and necessitates a decomposition of a higher degree.

Fig. 4.1 outlines the decomposition scheme of the local linear algorithm (LLA), showing a scenario with three lines, six timeslots, two TPM shifts and five employees. The decomposition scheme considers each TPM shift separately. In this example, TPM shift 1 is allocated to line 2. Thereby, the scope can be reduced to these lines, which allows for line 1 and 3 to be excluded. The further away the TPM value is shifted from the ideal start-timeslot, the higher the objective value gets. Therefore, it seems only reasonable to reduce the scope to timeslots near the ideal one. It may be that some employees are not available at these timeslots, or are even unable to accomplish the task. These employees can then be excluded from the scope without having to change the actual solution. It is obvious that the global model can be reduced to a smaller one: the exclusion of all gray elements helps to reduce the complexity of the problem and makes large problems solvable in a shorter time than with the GLA of section 3. Due to the fact that each sub-problem can only be considered consecutively, an appropriate order is necessary so as to generate a near-optimum global solution. This can be accomplished by a greedy algorithm.

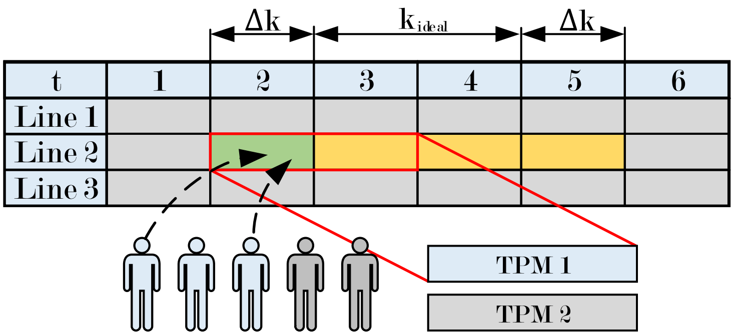


Figure 4.1: Local linear decomposition scheme

It is a difficult task to find the right greedy choice property as it needs to be quantitatively significant and must be able to determine the algorithm itself. A straightforward way is to use the objective value. Here, one can argue that the maintenance bundle with the highest objective value is the hardest to allocate. In order to obtain this value, all maintenance-bundles have to be solved locally. Afterwards, the maintenance windows can be ordered according to their difficulty. Another approach is to use the maintenance-bundles' potential: it would not make sense to allocate a high objective value first, if the objective value of all possible solutions of the maintenance window does not vary (or varies only a little). Thus, it seems reasonable to use the windows' potential. Therefore, it is necessary to determine all minimum and maximum objective values of the maintenance windows and compute their difference. This magnitude can be used as a greedy choice property which must be a one-dimensional magnitude. In order to generate a one-dimensional comparison value out of two values, a pareto analysis can be performed. For this reason, Hojati and Patil, 2011 introduced the pareto factor α. By varying α, one can generate solutions which are optimized on each case. For greedy algorithm the GCP G_{prop} would be

$$G_{prop} = \alpha \cdot O_{max-obj} + (\alpha - 1) \cdot O_{pot-obj} \tag{4.1}$$

where $O_{max-obj}$ is the locally determined objective value for the TPM shift, $O_{pot-pot}$ is the so called potential, and $\alpha = [0, 1]$ is the pareto value.

The solution for the MSSP is a procedure illustrated in algorithm 3 that uses a greedy method to choose the hardest TPM shift and afterwards solve it by modeling a linear problem. Hence, it is a mix of an exact and a heuristic method as it uses a heuristic approach in the global scope but eventually solves the local problems optimally. Although a global optimal solution cannot be proved with this kind of algorithm, solutions near the optimum are very likely, hence the overall algorithm can be classified as a heuristic approach. The linear formulation changes its form due to decomposition, and the local model is formulated in the next section.

Algorithm 3: Local linear algorithm

```
Data: Read input files
foreach TPM shift do
    Determine T_j, K_j, I_j;
    Determine and save GCP;
end
Put TPM shifts in order of GCP;
for Each TPM shift with highest GCP to TPM shift with lowest GCP do
    Update I_j while Solution not OK do
        Solve local linear problem;
        if Solution not feasible then
            Expand planning horizon;
            Update K_j
        else
            Assemble local solution to global solution;
        end
    end
end
Data: Write output files
```

4.2.1 The Local Linear Model

Due to decomposition, the problem of section 3.2 can be reduced to a local linear problem. This formulation only considers one line l and one TPM shift j, hence these indexes disappear in the local problem. Furthermore, only the tasks t in the formulation, which are included in the shift, and only the employees who are able to perform these tasks must be considered. This leads to a tremendous reduction of the linear problem, as stated below.

Subsets

$K_i \subseteq K$: set of timeslots k, where resource i is available

$K_l \subseteq K$: set of timeslots k, where line l is available

$K_j \equiv \{K_l : j \in J_l\}$: set of timeslots k, where TPM shift j is allowed to be executed

$I_j \subseteq I$: set of employees i, which are suitable for TPM shift j

$T_j \subseteq T$: set of tasks t, which are allocated to TPM shift j

Decision Variables

$$
\begin{aligned}
U_{ikt} &\in \{0,1\}, \text{unity if resource } i \text{ is scheduled to task } t \text{ on timeslot } k \\
V_k &\in \{0,1\}, \text{unity if TPM shift is scheduled on timeslot } k \\
X_k &\in \{0,1\}, \text{unity if TPM shift starts at timeslot } k \\
Y_{it} &\in \{0,1\}, \text{unity if resource } i \text{ is allocated to requirement } t \\
W &\in \{0,...,T\}, \text{start timeslot } k \text{ of TPM shift}
\end{aligned}
$$

Parameter

$A_{ik} \in \{0,1\}$, unity if resource i is available at timeslot k

$F_{ip} \in \{0,1\}$, unity if employee i possesses skill p

$F_{tp} \in \{0,1\}$, unity if task t requires skill p

$G_{it} \in \{0,1\}$, unity if $\quad F_{pi} \geq F_{pt} \quad \forall i,p,t$

$Z_k \in \{0,1\}$, unity if line l is available at timeslot k

$d \in \mathbb{N}$: duration of TPM shift j in timeslots

Constraints

$$\sum_{i \in I_j} Y_{it} = 1 \quad \forall t \in T_j \tag{4.2}$$

$$\sum_{k \in K_j} X_k = 1 \tag{4.3}$$

$$W = \sum_{k \in K_j} k \cdot X_k \tag{4.4}$$

$$-W - \kappa \cdot (1 - V_k) \leq -k + d \quad \forall k \in K_j \tag{4.5}$$

$$W - \kappa \cdot (1 - V_k) \leq k + 1 \quad \forall k \in K_j \tag{4.6}$$

$$\sum_{k \in K_j} V_k = d \tag{4.7}$$

$$Z_k \geq V_k \quad \forall k \in K_j \tag{4.8}$$

$$Y_{it} \leq G_{it} \quad \forall i \in I_j, t \in T_j \tag{4.9}$$

$$Y_{it} \leq O_{it} \quad \forall i \in I_j, t \in T_j \tag{4.10}$$

$$U_{ikt} \leq Y_{it} \quad \forall i \in I_j, k \in K_j, t \in K_j \tag{4.11}$$

$$U_{ikt} \leq V_k \quad \forall i \in I_j, k \in K_j, t \in T_j \tag{4.12}$$

$$\sum_{k \in K_j} U_{ikt} \geq Y_{it} \cdot d_j \quad \forall i \in I_j, t \in T_j \tag{4.13}$$

$$\sum_{i \in I_j} U_{ikt} \leq 1 \quad \forall k \in K_j, t \in T_j \tag{4.14}$$

$$\sum_{t \in T_j} U_{ikt} \leq A_{ik} \quad \forall i \in I_j, k \in K_j \tag{4.15}$$

Costs and Objective Function

$$H_k = |b^{opt} - k| \cdot m \quad \forall k \in K_j \tag{4.16}$$

$$Q_{it} = \sum_{p \in P} (q_p \cdot (F_{ip} - F_{tp})) \cdot d \quad \forall i \in I_j, t \in T_j \tag{4.17}$$

$$E_k = D_k \cdot n \quad \forall k \in K_j \tag{4.18}$$

$$min(Z) = \sum_{k \in K_j} (H_k \cdot X_k + V_k \cdot E_k) + \sum_{i \in I_j} \sum_{t \in T_j} (Y_{it} \cdot Q_{it}) \tag{4.19}$$

4.3 Pyramid Linear Algorithm

The pyramid linear algorithm (PLA) is an improvement of the LLA that was outlined in the prior section. In the local algorithm, the order of TPM shifts is based on a determination of the greedy choice property of each. This determination is executed before the first TPM shift is assigned. Due to the fact that the TPM shifts are part of one global solution, they affect each other. Hence, it may be that if a TPM shift is fitted to a global solution, the objective values of the TPM shifts that are not assigned change. This can lead to new bottlenecks and a new distribution of objective values to TPM shifts. As stated in the global formulation in 3.2.1 it can be seen that the problem is dependent of timeslots k, employees i, tasks t and TPM-shifts j. The connection of all four indexes broke up due to elimination of j in the constraints of the local problem. This effect can also be seen when comparing the two objective functions 3.20 and 4.19 due to the missing index j in the local model.

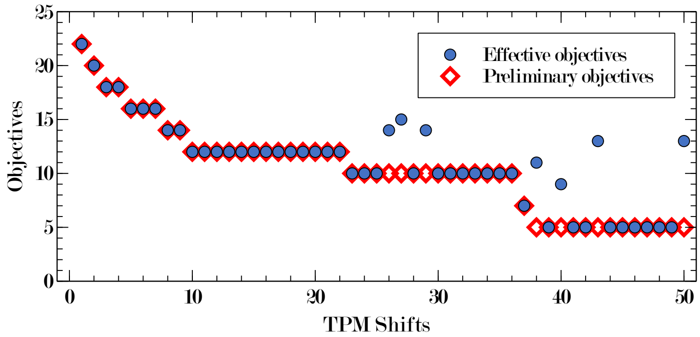


Figure 4.2: Order of shifts through objective values

The horizontal axis of fig. 4.2 represents the order of the allocation of TPM shifts, while the vertical axis shows their objective values. The latter, which were determined at the beginning of the solving process, are outlined as red diamonds; the blue circles illustrate the effective objectives of each TPM shift. The figure shows seven TPM shifts that do not reach the predicted objective value. The probability of deviations increases at the end of the sequence because of the allocation of the previous TPM shifts in which less qualified employees are already allocated and better timeslots are blocked. This can lead to worse or even infeasible global solutions.

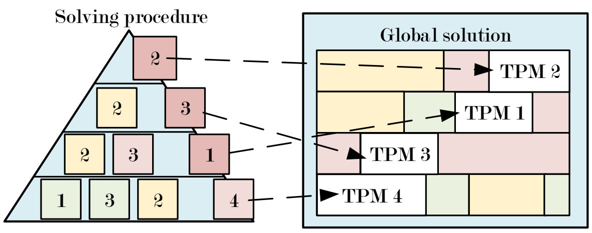


Figure 4.3: Solving procedure for pyramid algorithm

A solution for this problem is to order the TPM shifts again after assembling a local solution to the global solution. The safest way to do so is to order them after each assignment. Fig. 4.3 shows the procedure of the pyramid algorithm: each step of the pyramid represents one assignment sequence. The order of the blocks represents the order of the TPM shifts regarding the greedy choice property. The first row shows that TPM 4 is the hardest shift and is thus allocated to Line 4. Afterwards, TPM 4 is excluded and the remaining shifts are re-evaluated. It can be seen that the order changes and TPM 1 is now the most difficult shift. Again, the shift is assigned to Line 2 and the sequence recommences. This procedure cannot solve the problem completely but it decreases deviations. Due to the evaluation of TPM orders after each assignment, the computation time increases tremendously, which is a major disadvantage of the method. Algorithm 4 represents the procedure in form of a pseudo code.

Algorithm 4: Pyramid linear algorithm

```
Data: Read input files
Generate list of TPM shifts;
while not all TPM shifts are scheduled do
    foreach TPM shift do
        while Solution not feasible do
            Determine T_j, K_j, I_j;
            Solve local linear problem;
            if Solution not feasible then
                Expand planning horizon;
            else
                Save solution;
                Save GCP;
            end
        end
        Order TPM shifts regarding GCP;
        Assemble local solution of TPM shift with highest GCP to global solution;
        Remove TPM shift from list;
    end
end
Data: Write output files
```

4.4 Human Greedy Algorithm

The human greedy algorithm (HGA) is used for evaluating the quality of solutions of the linear solver of the last chapter. This algorithm follows simple rules and tries to imitate a human-like planning procedure. It is structured like a greedy algorithm, which means that TPM shifts, tasks and employees are ranked on the basis of their difficulty or skills, respectively. The algorithm starts with the most difficult TPM shift at the ideal starting timeslot, the difficulty of the shift j is determined by the GCP Λ_j which is defined by equation 4.20. If there are enough skilled employees, the algorithm assigns them to the shift and goes on with the next one. Both task t and employee i own a GCP in order to accomplish the best combination of both. Equation 4.21 defines the GCP for employee i, which is high, if the worker is high qualified and vice versa. Formula 4.22 defines the GCP of the task t. However, if there are not enough resources, it tries to move the shift to the future or the past. If there are still not enough employees, the algorithm stops and the solution is infeasible. The procedure of the method is illustrated in algorithm 5.

$$\Lambda_j = \sum_{t \in T} \left(\sum_{p \in P} (F_{tp} \cdot q_p) \, C_{jt} \cdot d_j \right) \tag{4.20}$$

$$\Phi_i = \sum_{t \in T} (O_{it} \wedge G_{it}) \tag{4.21}$$

$$\Theta_t = \sum_{p \in P} (F_{tp} \cdot q_p) \sum_{j \in J} (C_{jt} \cdot d_j) \tag{4.22}$$

Algorithm 5: Human Greedy Algorithm

```
Data: Read input files
for TPM Shift j with highest Λ to TPM Shift j with lowest Λ do
    Set timeslots to ideal start timeslot of TPM shift;
    while TPM shift is unplanned do
        for Task t with highest Θ to Task t with lowest Θ do
            for Employee i with lowest Φ to Employee i with highest Φ do
                if employee is able to do task then
                    Allocate employee to task;
                    break;
                end
            end
        end
        if All tasks are allocated successfully then
            Next TPM shift;
        else
            Check adjacent timeslot;
        end
    end
end
Data: Write output files
```

Chapter 5

Comparison of MSSP Algorithms

The four algorithms outlined in the previous chapter are benchmarked by four datasets of different complexity. The first section outlines the datasets which are used for the benchmark test, while the second section deals with the evaluation of the results of each algorithm. The solutions of each dataset are compared to each other, with a special focus on solution quality and computational time.

5.1 Datasets

The datasets represent input data of the algorithm. Table 5.1 represents basic information for each of them. Dataset 1 is a low level set with the purpose to test basic functions of the algorithms. The complexity increases steadily from set 1 to 4. The complexity of dataset 4 is almost comparable to a real-life example, and the results of this dataset will thus be most important with regard to the numeric evaluation of the algorithms.

5.2 Numerical Evaluation of Algorithms

The evaluation of the algorithms is carried out with two indicators: the objective function of the result, and the computation time. Although the latter can be a powerful indicator

	Timeslots	Lines	TPM shifts	Tasks	Employees	Skills
Dataset 1	20	3	4	6	5	2
Dataset 2	40	10	10	20	10	4
Dataset 3	80	20	20	30	20	5
Dataset 4	120	20	30	100	40	5

Table 5.1: Cornerstones of datasets

for benchmarking an algorithm, it may not be so important in this case. Hence, the computation time is considered less important than the objective value – as long as it lies within the range of a few minutes. In order to accomplish comparable results, all tests were performed on the same computer system which is described in table 5.2.

CPU	Intel Core 2 Duo E 6550 @2.33GHz
Memory (RAM)	6.0 GB
Harddisk	250GB Seagate ST3250310AS SATA
Operating System	Windows 7 64 bit
Gurobi Version	6.5 64 Bit
.net Framework	4.5

Table 5.2: Computer system

Table 5.3 illustrates the objective values of each dataset. The columns represent different algorithms, while the rows show the datasets from the prior section. The gap is outlined on the right hand side of the table. The optimum objective value for dataset 4 is unknown, thus the gap is determined in reference to the GLA from dataset 1-3 and in reference to the LLA of dataset 4. The global algorithm performed best in all cases except for dataset 4 which it could not solve. The computation was suppressed after a few hours of computational time. The gap of the LLA and PLA to the global algorithm is in a range between 0% and 8.6%. The gap of the HGA to the best algorithm is rather high between 9.4% and 55.3%. A further remarkable point is that the gap is highest in the dataset with the lowest complexity. It seems that there is no correlation between the complexity and the quality of the solution with regard to the HGA. One could argue that this algorithm rather randomly finds good solutions.

	Objective values				Gap			
Dataset	GLA	LLA	PLA	HGA	GLA	LLA	PLA	HGA
1	57	57	57	103	0%	0%	0%	55.3%
2	374	409	409	413	0%	8.6%	8.6%	9.4%
3	679	690	690	791	0%	1.6%	1.6%	14.2%
4	-	1259	1259	1816	-	0%	0%	30.7%

Table 5.3: Objectives for test-datasets

Fig. 5.1 outlines the normalized objective value. The highest objective of each dataset was used as a reference to normalize the values and render them comparable. It can be seen that the human-based algorithm generates the worst solution in all cases. Dataset 1 and 4 show a larger gap of about 30% to 50% compared to the other algorithms, while on dataset 2 and 3 it is actually small. The GLA generates the best solutions except at set 4, as stated above. The objective value of the local linear and the PLA are equal on all datasets, with little or no gap to the global algorithms. In dataset 4, these algorithms obtained the best result of all methods.

	Computation time / sec				Relative computation time			
Dataset	GLA	LLA	PLA	HGA	GLA	LLA	PLA	HGA
1	0.93	3.48	8.98	0.19	10.4%	74.1%	100.0%	2.1%
2	2.44	8.67	14.26	0.25	17.1%	60.8%	100.0%	1.8%
3	58.74	48.04	226.0	0.67	26.0%	21.3%	100.0%	0.1%
4	-	195.15	1675.15	1.31	-	11.7%	100.0%	0.1%

Table 5.4: Computational times for test-datasets

Table 5.4 outlines the computational time of the datasets for each algorithm. The relative computation time is referenced with regard to the maximum of each dataset. It becomes obvious that the computation time of the HGA lies in a range of 0.19 to 1.31 seconds, which leads to an increase of factor 6.9. The increase at the PLA is equal to factor 186 from the first to the forth dataset, as compared to factor 56 at LLA. One can conclude from this data that the computation time at the HGA has a much lower dependency on complexity than the LLA or even the PLA. The computation time with the PLA in dataset 4 is 1675.15 seconds, which is equal to approx. 28 minutes. This means that the operator of a scheduling tool would have to wait for about half an hour to check the solutions, which is definitely not acceptable.

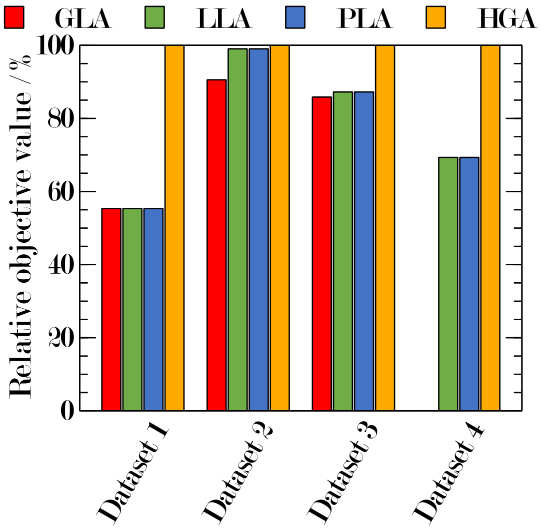


Figure 5.1: Normalized objective values of algorithms

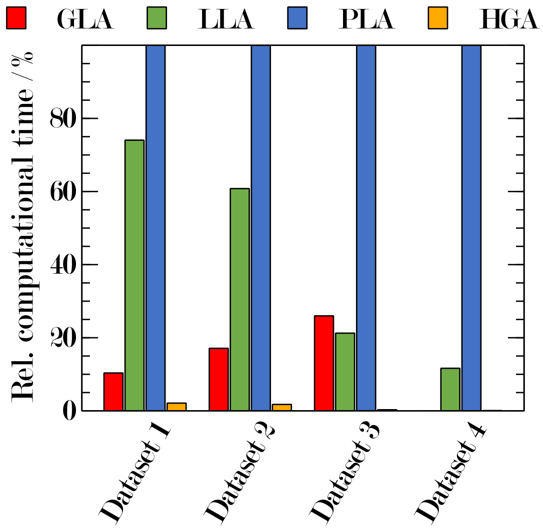


Figure 5.2: Normalized computational time of algorithms

Fig. 5.2 represents the normalized computation time from table 5.4. The reference of the normalization is the maximum computation time of each dataset, which results in the slowest algorithm having a computation time of 100% and the others being measured in relation to it. It is obvious that the pyramid algorithm is the slowest and the human greedy algorithm is by far the fastest of all solvers. The least complex datasets are preferable for the global algorithm compared to the local linear algorithm. It is fastest in datasets 1 and 2, and the GLA is the fastest in dataset 3. The computation time is about 9 seconds lower than the one of the LLA and needs only 20% of the time of the pyramid procedure. The global algorithm was not able to solve dataset 4. The LLA needs 195 seconds which corresponds to 12% of the pyramid solving time.

Fig. 5.3 and 5.4 allow a more precise comparison between the GLA, LLA and PLA algorithms. Each point represents one solution for the dataset; the horizontal position depends on the computation time, while the vertical depends on the objective value. Similar symbols represent the same dataset, and the colors represent the applied algorithm for a specific solution. The further right one solution is, the faster it was generated. The solution quality increases (i.e. the objective value decreases) towards the top of the

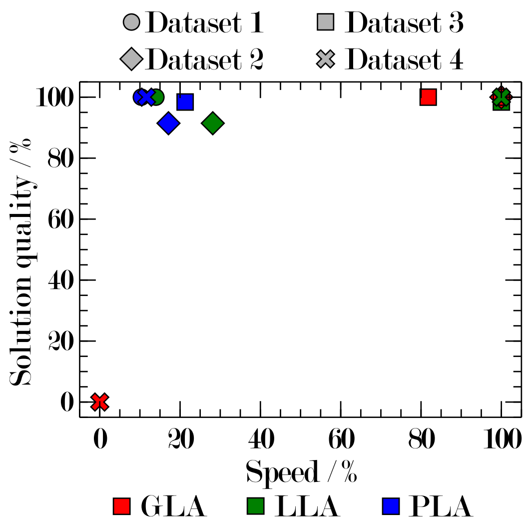


Figure 5.3: Scatter plot of algorithm results

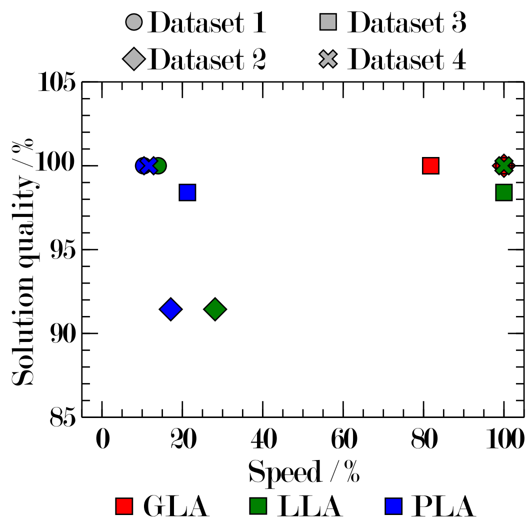


Figure 5.4: Scatter plot of algorithm results - details

chart. The most obvious element is that there is no algorithm that is best with regard to all criteria. Although the global linear algorithm creates fast and good solutions, it was unable to solve dataset 4. This is critical because this is exactly the dataset with a real-life-application complexity – if the algorithm is not able to solve this example, it is not suitable for a real case. Another important point is that all solutions (except the infeasible one) have a solution quality over 90%. Thus, all algorithms deliver useful results. Moreover, the solutions of the local linear method are always better than the pyramid linear procedure. Although they have the same solution quality, the results are generated faster than with the pyramid algorithm.

Summing up, the HGA is excluded due to poor results and the GLA is eliminated because it was not able to solve real-life cases. The PLA delivered solutions of the same quality as the LLA, but was slower in every case. The best algorithm seems to be the LLA, which is why this method is chosen to perform in the case study in the next chapter.

Chapter 6

Case Study

In this chapter, the selected local linear algorithm is used for a case-study that is inspired by a real life example of the partner company. For evaluating the performance of the algorithm, the results are compared with the human greedy algorithm. Due to the fact that this algorithm follows simple rules, it can simulate a planning procedure that is similar to that of a human. By comparing these two algorithms' results, it is possible to measure the solution's quality improvement that might exist if the linear algorithm were implemented in a maintenance department.

The case focuses on the maintenance department of a manufacturing company. At the moment, the plant operates with a decentralized maintenance department. It is therefore interesting to investigate whether a centralization would make sense. There are currently 20 production lines; on each of them, maintenance workers must perform one TPM shift once a week. Every two weeks, a more complex shift is executed, which means that more personnel is necessary to accomplish the tasks (see table 6.1 for the exact number of resources per shift). The workers are hierarchically arranged, i.e. level 2 workers can do level 1 tasks but not vice versa. Level 3 employees may do all tasks. Both the production and the maintenance department operate in three shifts and the operators' shifts are allocated equally. An employee is allocated to the same shift every day, except for Saturday and Sunday where no employees are available. The staff consists of 144 employees, which makes 48 employees per shift. These includes eight level 3, 20 level 2 and 20 level 1 employees. Urgent production batches make it impossible to maintain the lines for 16 hours per week. It is not optimal to maintain the line 12 hours before and after this time frame, so a penalty for executing a TPM shift in these slots is higher than during the rest of the week. See table 6.1 for a small summary of this dataset.

Line complexity	Duration	Number of lines	Demand of workers Level 1	Level 2	Level 3	
Low	4h	5	1	1	1	Week 1
			1	2	2	Week 2
Medium	8h	10	2	1	2	Week 1
			2	2	3	Week 2
High	12h	5	2	2	2	Week 1
			2	2	4	Week 2

Table 6.1: Demand of employees

6.1 Scenarios

The following sections describe three scenarios of possible organizational structures based on the case outlined in table 6.1. The scenarios differ concerning the allocation of certain employees to decentralized or centralized departments. However, the workload remains stable over all three scenarios.

6.1.1 Scenario 1: Decentralized

Scenario 1 is a decentralized scenario: there is a central maintenance department as well as decentralized units directly at the production lines. According to the literature described in section 2.2.3, this scenario is an effective way to organize maintenance employees. As shown in fig. 6.1, there are eight level 3 maintenance employees in the central maintenance department, who can be allocated to all production lines as long as they are available. Each of the production lines has one level 1 and one level 2 employee. They can only be allocated to these production lines as it is not possible to execute tasks on another line. In this scenario, 16.6% of the employees are allocated at the centralized unit and 83.3% to the production lines.

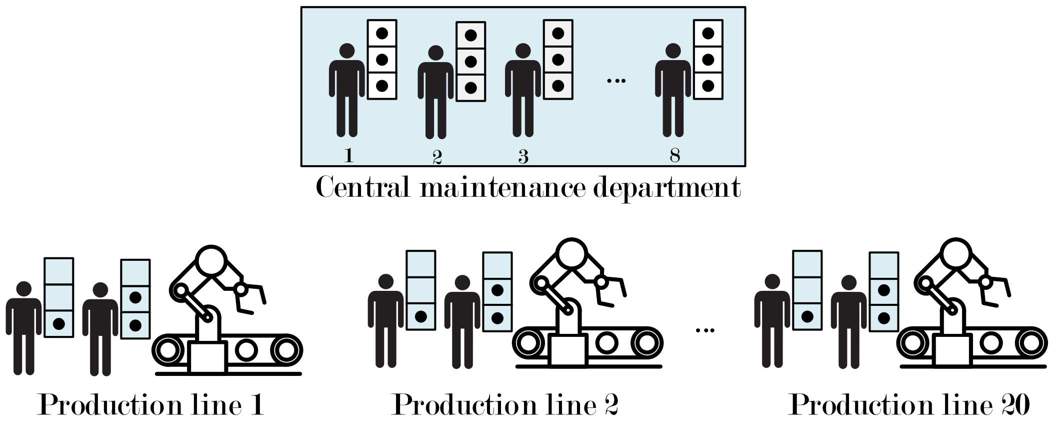


Figure 6.1: Organizational structure of decentralized maintenance

6.1.2 Scenario 2: Hybrid

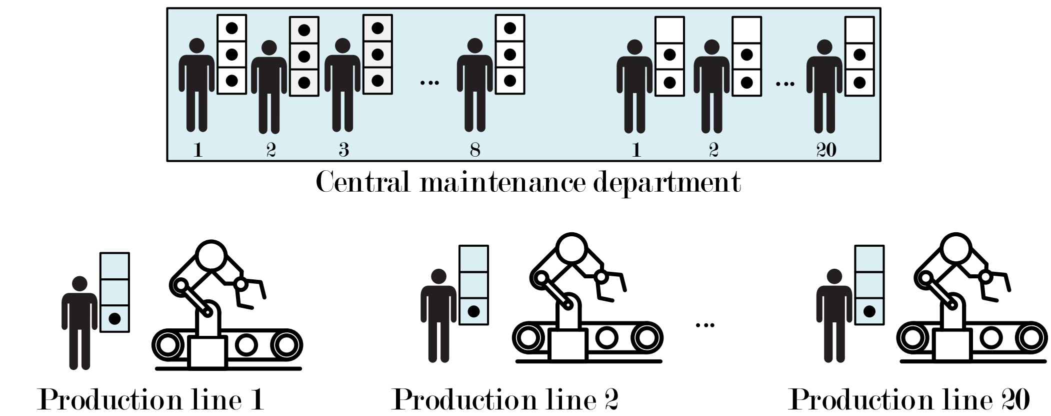


Figure 6.2: Organizational structure of mixed maintenance

The second scenario is a hybrid model of the decentralized and a centralized scenarios. This means the level 2 employees are moved to the central maintenance departments. All other parameters remain similar to in Scenario 1. For the sake of clarity, this means that in every shift, there is one level 1 worker on each production line, as well as 20 level 2 and eight level 3 employees in the central maintenance department. This leads to a staff share of 58.3% at the centralized and 41.6% at the decentralized units. Fig. 6.2 illustrates this scenario.

6.1.3 Scenario 3: Centralized

The last scenario (fig. 6.3) is fully centralized, i.e. all employees work in the central maintenance department. Thus, the 20 level 1, 20 level 2 and eight level 3 workers for each shift can work on all production lines, resulting in 100% of the workers being centralized.

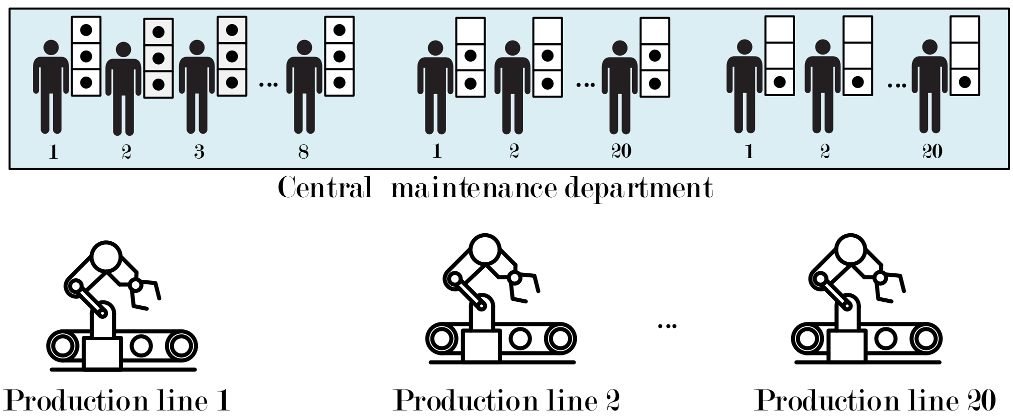


Figure 6.3: Organizational structure of central maintenance

6.2 Graphical User Interface

In order to facilitate the handling of the scheduling tool, a graphical user interface is developed, as can be seen in fig. 6.4. This tool makes it possible to choose sets of data, solve them and display the solution via a Gantt chart. Each bar represents an employee working on a task; tasks within the same TPM shift are displayed in the same color. In order to obtain get detailed information for each task, a mouse-over function is implemented which displays the exact start and end-time of each task, as well as the line the employee must work on, and the ID number of the TPM shift. The data can be displayed with regard to the order of employees or the order of production lines. The first option is useful and necessary for the maintenance staff themselves, e.g. to quickly find dates and tasks they must accomplish. The second option is targeted at production managers in order to enable them to check if the production plan matches the maintenance schedule.

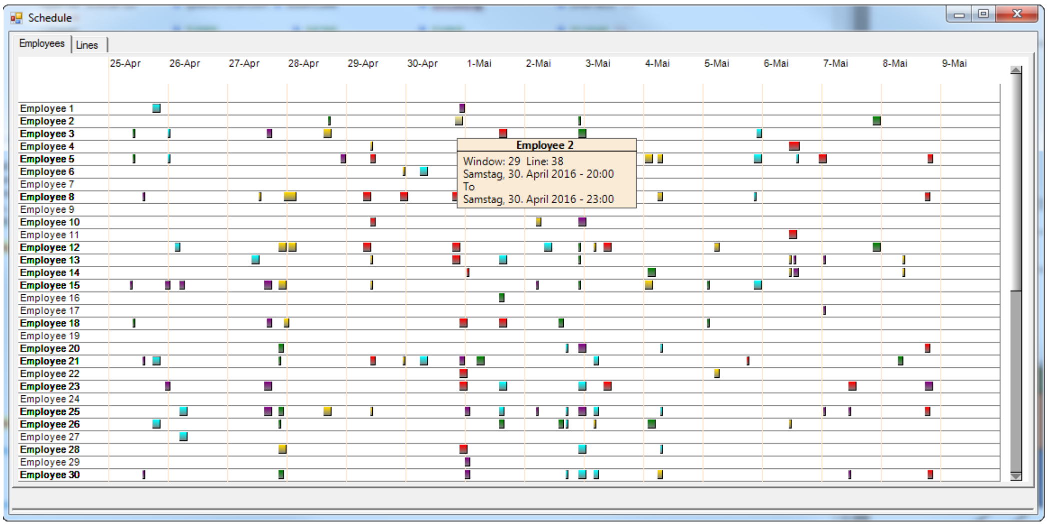


Figure 6.4: Graphical user interface of the scheduling tool

6.3 Numerical Evaluation

This section describes the numeric evaluation of the case study, including all three scenarios of both algorithms. The evaluation of this case is based on the objective value and its individual shares. The computation time is not considered here because of two reasons: the scheduling time can only be measured by the computational time of the algorithms but the evaluation should compare the real-life situation, which is planned manually, with a newly implemented algorithm. Usually, the manual planning effort may range from mere minutes to several hours, and it thus takes far longer than the human greedy algorithm. A direct comparison of the computation time of the local-linear and the human greedy algorithm would create a false statement and it is thus not considered here.

		Qualification	Downtime	Move shift	Overall
Decentral	Local	1030	900	176	2106
	Human	1030	1020	140	2190
	Difference	0	120	−36	86
	Relative	+0%	+13.3%	−20.5%	+4.0%
Hybrid	Local	284	860	136	1280
	Human	364	1040	72	1478
	Difference	80	180	−64	198
	Relative	+28.8%	+20.9%	−47.1%	+15.5%
Central	Local	0	860	116	976
	Human	0	1040	74	1114
	Difference	0	180	−42	138
	Relative	0%	+20.9%	−36.2%	+14.1%

Table 6.2: Results - Human/Linear

Table 6.2 shows the objective value of the decentralized scenario. The overall objective value of the HGA is 2392 compared to LLA of 2532. This results in the linear solver providing a 5.5% better result than a human schedule. Especially the downtime share of the objective function is improved by 15.5%. Although the HGA performed 4.2% better than LLA, the linear solver is able to look for suitable timeslots for the TPM shift. The qualification share of the objective value is identical, due the fact that there is just minimum choice of the personnel in this scenario.

Due to the linear solver an overall performance increase of 15.5% is accomplished in the hybrid case. Fig. 6.6 illustrates that the major share of the objective function is generated due to downtime. This value did not decrease from the decentral to the hybrid case. The biggest improvement is generated in the field of over-qualification: due to more central resources, it is possible to allocate less skilled employees to tasks and free qualified resources for troubleshooting. The share of "move shifts" decreases about 400 points when comparing the decentral to the hybrid scenario, which indicates that most shifts are done at the optimal date.

The centralized case has the lowest objective functions of all scenarios. The downtime share is almost similar to the hybrid case. Over-qualification is reduced to a minimum in

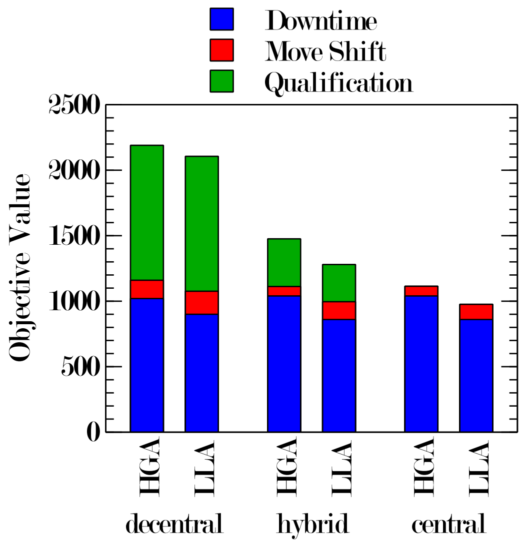


Figure 6.5: Objective values of cases

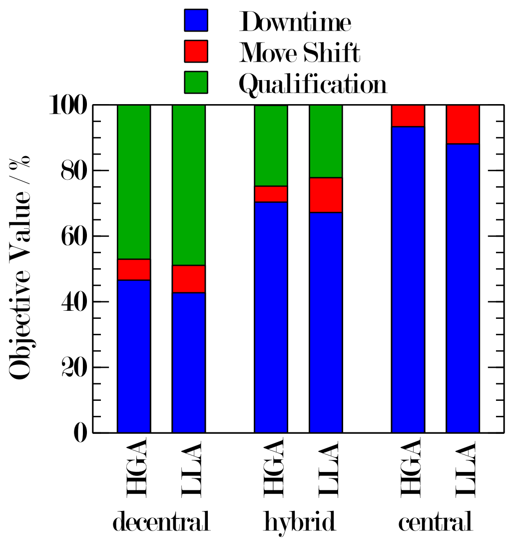


Figure 6.6: Relative objective shares

the linear greedy function, which shows that every employee does exactly what he or she can do. The human greedy method has a good score as well. The penalty caused by moving shifts is relatively similar to the hybrid case. A remarkable point is that in this scenario, the advantage of the linear to the human greedy algorithm again decreases to 14.1%. An explanation for this effect is provided in a later paragraph.

Fig. 6.5 shows the absolute objective value of each scenario, with each share of downtime, shift movements and over-qualification. Fig. 6.6 subsequently represents the objective value share relative to the overall objective value of each case. It becomes obvious that the biggest share in the decentral case relates to over-qualification – which is zero in the central case in both methods. The share of downtime is most dominant in the hybrid and central case, although the share of shift movements remains rather constant.

In fig. 6.7, the relative objectives of LLA with a reference to the HGA are represented. A notable point is that there is no decrease of the qualification objective in the decentral and central organization although there is an improvement of 28.8% in the hybrid case. The reason for this is twofold. In the decentral scenario, there are few possibilities to allocate other employees, hence both algorithms deliver a similar employee assignment. At the centralized scenario, however, so many resources are available for all lines that there is no longer a shortage of resources. Thus, the HGA performs the same objective value as the LLA. The advantage of the local linear algorithm in the downtime share of the

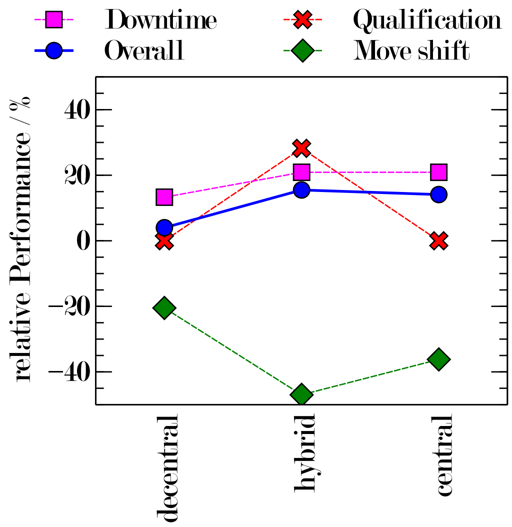


Figure 6.7: Objective value with reference human algorithm

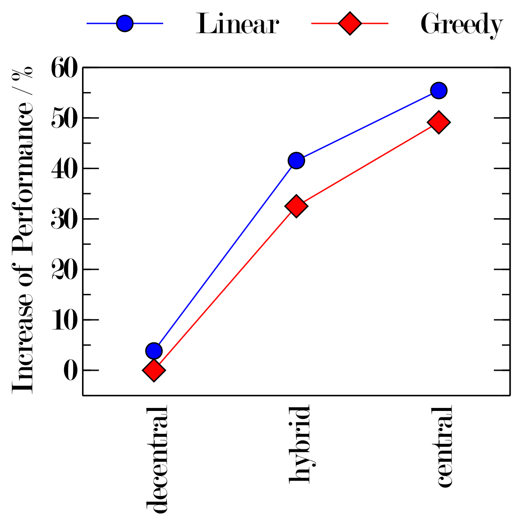


Figure 6.8: Objective value with reference to actual state

objective value is relatively constant, with a slight increase towards centralization. These benefits are optimized through losses in the shift-moving share. The linear algorithm can improve the overall function due to flexible arrangements of TPM shifts. The overall performance is 4.0% better in the decentralized organization, compared to 15.5% in the hybrid version and 14.1% in the central organization. The strong increase from the central to the hybrid scenario is due to more complex constraints and a larger solution space. The small decrease from the hybrid to the central scenario may be caused by a high number of good possible solutions, that make assigning easier than in the hybrid case, and the lead decreases.

Fig. 6.8 shows the relative improvement of the present state. The latter is defined by a decentralize organization with the human greedy scheduling method. The decision quality improves by some 4% when changing the scheduling method the LLA. When changing the structure to hybrid, the objective value decreases 32.5% when using HGA, and 41.5% with the LLA. Changing from the hybrid to the central form increases performance again to 55.4% with the LLA and to 49.1% with the HGA. This figure shows that scheduling quality improves by 50% in both methods. The best enhancement is established by changing the structure from decentral to hybrid organization. A smaller, but still remarkable improvement of 13.9% points (LLA) and 16.6% points (HGA) is

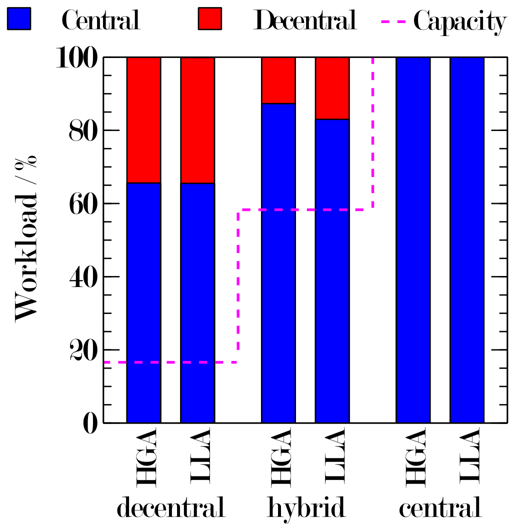


Figure 6.9: Share of workloads in local and centralized units

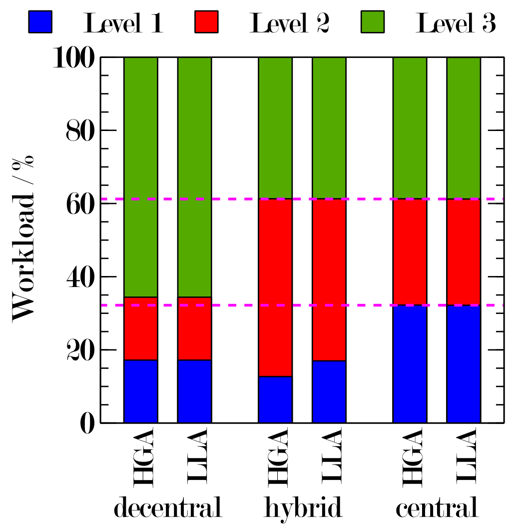


Figure 6.10: Distribution of workload to different skill levels

established by changing from hybrid to central form of organization.

Fig. 6.9 shows the distribution of the workload on the centralized and decentralized units. The dashed line in magenta shows the ratio of the decentralized/centralized capacity. Both methods allocate resources very similarly, although a better productivity can be measured with regard to the linear solver in the hybrid organization structure. In the decentral organization, 34.4% of the workload is executed by decentralized units, although 83.3% of the workforce is allocated them. The hybrid scenario leads to a higher centralization and a better ratio of centralized workforce and demand. 58.8% of the workforce is allocated in the centralized maintenance department, and 87.3% of the tasks get done when the schedule is generated with the HGA – compared to 83% that are accomplished when scheduling with the LLA. Due to the fact that there are no decentralized employees any more, all the work is done by centralized employees in the fully centralized organization. Fig. 6.10 shows the workforce distribution represented in skill levels: the dotted line represents the demand of the workload, hence 32.2% of the tasks have level 1, 29.03% level 2 and 38.7% level 3 requirements. It emerges that 65.6% is done by highly skilled level 3 workers in the decentral organization. The rest of the work is equally shared between level 1 and level 2 employees. When centralizing the organization to the hybrid structure, a major share is pushed down to level 2 employees. Level 3 employees are optimally assigned with both HGA and LLA algorithms, while the workload for level 1 employees decreases. In the fully centralized organization, staff is assigned optimally with both algorithms.

Chapter 7

Conclusion and Outlook

This chapter summarizes the thesis in three sections. The main work was divided between two parts: the first was intended to investigate the behavior of a maintenance department in different forms of organizational structures, the result of which is presented in the first section 7.2. The subsequent part dealt with the development of an algorithm and section 7.1 outlines the conclusion of this task. The last section in this chapter provides an outlook, as well as possible future milestones and possibilities of project TPM 4.0 that can be deducted from this thesis.

7.1 Scheduling Algorithm

As described in section 6, the developed linear greedy algorithm is a suitable tool for scheduling the maintenance department. All solutions of the LLA are significantly better than those of the HGA. Due the fact that the latter imitates human planning behavior, it can be assumed that the LLA results in better scheduling than solutions that may be accomplished by a human. It levels out the various terms of the objective function and is able to move shifts more flexibly than the human greedy algorithm. Moreover, it better allocates resources in complex problems, e.g. in the hybrid scenario, and increases the flexibility of the department concerning unplanned maintenance activities. Linear programming is a rather old concept for optimization problems; however, its robustness, simple solving algorithms and modeling complexity justify modern applications.

The combination of a heuristic and an exact method can provide satisfying results. Although the solution may not be optimal, the linear part finds optimum solutions in the local problem. The gap between the LLA and the GLA never exceeds 8.6% and the first was able to solve datasets which the GLA could not solve. A good decomposition technique allows to reduce the complexity of any case so as to reach fast computation time as well as good solution quality.

In theory, the PLA should provide better results than the LLA. In fact, the objective values were equal to the LLA, but computation time was much higher. The evaluation allowed to disprove this approach and save computing power by using a simpler algorithm.

The HGA algorithm was the worst in terms of solution quality, but by far the fastest with regard to computation time. This solution could eventually be a good start for a NVS to improve results even more and close the gap to the optimum. With such a feature, the heuristic method could become competitive and still be faster than the algorithms based on linear programming.

Summing up, a scheduling algorithm in production or maintenance can definitely and significantly reduce management efforts. Not only is manual scheduling time-consuming and complex, but a scheduler can also decrease efforts and provide better results. If other scheduling problems arise in the partner company, one should consider using a scheduling algorithm, as problems are very likely to be rather similar from a mathematical point of view.

7.2 Degree of Centralization

If the defined objective function benchmarks the efficiency of an organization, it can be stated that in this case study, the best structure is the centralized one. Although the LLA improved the results of the HGA (4% - 15.5%) it has to be noted that the biggest gain can be reached by changing the form of the organization. By merely changing the organizational form from decentralized to fully centralized, the scheduling quality increased by 49.1%. It emerged that the improvement from decentral to a hybrid organization is bigger than the step from a hybrid to a central organization.

In fact, centralizing leads to a better assignment of skilled workers. When comparing a decentralized maintenance structure with a highly skilled central department to a centralized structure employing all levels of workforce, it turned out that the workforce is distributed more equally with regard to all qualification levels. This means that highly skilled employees are relieved and can be used either for improvement projects or be on idle for unplanned maintenance. The workforce of lower level workers can be used more often, so productivity rises in this group. Generally, the hybrid scenario radically relaxed bottleneck situations. Furthermore, over-qualification provides the biggest room for improvement when centralizing an organizational structure – in the present case, it even makes up the major share of the decentral organization, but would be zero if the structure were fully centralized.

The share of moving shifts is rather small in all scenarios. Due to the fact that the intervals of TPM shifts on each line should be rather constant, this can be rated as positive. One could argue that if shifts should be on the same date each week, moving them could be eliminated completely, but the LLA shows that small movements can improve the overall scheduling quality. This can be rated as a useful and important feature.

The share of downtime remains almost constant when centralizing the organization. There, it becomes the largest share and should thus be investigated more closely in the future in order to further increase scheduling quality.

It is crucial to note that the present one is a conceptual model with various assumptions and simplifications. Even though the modeled workload only consists of tasks for preventive maintenance, the majority of work in maintenance occurs through unplanned maintenance. For this part of work, response times can be crucial and should thus not be neglected. As described in section 2.2.3, there are several advantages which advise a decentralized organization. One of the most important ones may be response time which is covered better in this form of decentralization. Although the central maintenance department is the best based on the objective value, if response time and social aspects are taken into account, the best solution could be a lower degree of centralization, e.g. a hybrid form.

7.3 Outlook

At the moment, in- and output is implemented via text files. All of the necessary input data should be available in a common ERP system. There, scheduling could be triggered in the back-end, without any user interaction. This is immensely time saving and frees the maintenance management for other activities. If scheduling would be done manually, the resulting schedule would be rigid with regard to short term modifications. A small change can disrupt the whole schedule and everything might be reconstructed by hand. An automatic planning procedure takes several minutes on a standard computer, and it is necessary to plan several times a week in case unpredicted changes, e.g. concerning the production plan, occur. This advantage can increase the flexibility of the whole production system.

As centralization proceeds, steps for further improvement can be discussed. According to the objective function, the biggest share of it is downtime. To shorten it, the TPM shifts could be split into smaller timeslots. These could then be distributed over the week while production proceeds. Certain machines in a line provide a shorter cycle time than the production line, so it is possible to shut down these faster machines for maintenance without decreasing the OEE of the production line. At the moment, a project is being tried for developing an algorithm to determining so-called “Flex-TPM” windows. An interface between the scheduling algorithm of this thesis and the Flex-TPM algorithm could increase the OEE of the production system even further.

In general, the possibilities of structuring an organization should be discussed. In section 2.2.2, the design process of an organizational structure was outlined, with the main parts differentiation and integration. A task analysis defines jobs that are combined by departments. Also, integration is necessary to establish cooperation of the departments. The organizational structure in the partner company is usually split between several strictly divided groups. Especially in maintenance, it seems reasonable to consider a new form of integration. One could define areas of responsibility, as outlined in fig. 7.1 that

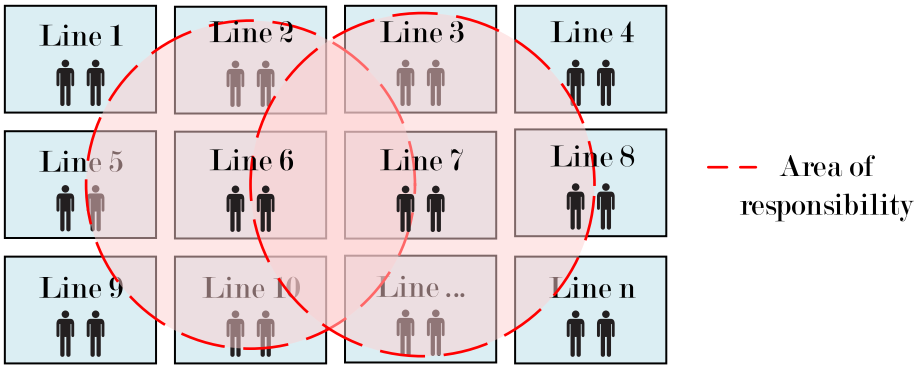


Figure 7.1: Area of responsibilities

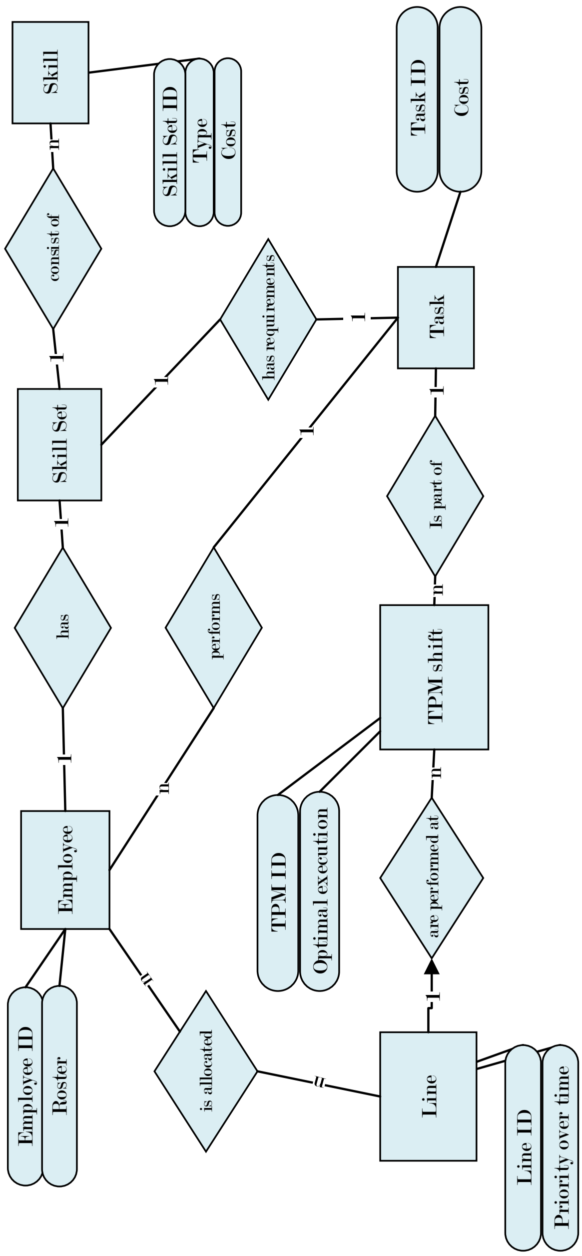
represents an abstract production line layout. There are several lines with decentralized employees working on them. As presented in chapter 6, a decentralized organization can have significant disadvantages, but also possess major benefits, as described in section 2.2.3. An area of responsibility can be defined in a way that assigns employees to a home base, but they are allowed to be assigned to adjacent production lines by the scheduling tool (this feature is already implemented in the algorithms). This leads to several benefits: the tool creates better solutions due to more possibilities in scheduling. The maintenance activities lead to on-the-job training for the employees of other production lines as they become familiar with them. Thus, in case of reactive-unplanned maintenance activities, these employees could work on adjacent production lines and support their colleagues.

The scope of this thesis excludes several points which could be investigated in more detail in order to find the best solution for the organizational structure. First, unplanned actions have a stochastic behavior and play a major role in the maintenance department. This part may have a larger influence in the optimum degree of the centralization than the planned maintenance activities. A second point that is neglected in this thesis is travel time. The plant of the partner company has a large number of production lines, which means that travel times between the lines represent a significant part of working time. Employee tracking in combination with route optimization could help to reduce travel times to a minimum. In order to find the most suitable organizational structure for the department, an organizational simulation should be done, including travel times and unplanned maintenance activities.

Chapter 8

Appendix

8.1 A-1


Skill
Skill Set ID
Type
Cost
consist of
Skill Set
has requirements
Task ID
Cost
Task
Is part of
has
performs
TPM shift
TPM ID
Optimal execution
are performed at
Employee
Employee ID
Roster
is allocated
Line
Line ID
Priority over time

Bibliography

Augstiady, Tina Kanti and Elizabeth A. Cudney (2016). *Total Productive Maintenance - Strategies and Implementation Guide* (cit. on p. 12).

Baker, Kenneth R. and Dan Trietsch (2009). *Principles of Sequencing and Scheduling* (cit. on p. 34).

Ben-Daya, Mohamed et al. (2015). *Planning and Control of Maintenance Systems* (cit. on pp. 15, 17, 24–26).

Biedermann, Hubert (2008). *Ersatzteilmanagement* (cit. on pp. 2, 6).

Breslaw, Jon A. (1976). "A Linear Programming Solution to the Faculty Assignment Problem." In: *Socio-Economic Planning Sciences* (cit. on p. 37).

Chen, Min et al. (2014). "Big Data: A Survey." In: *Springer Science Business Media New York* (cit. on p. 1).

Clemm, H.L. (1985). "Kosten u. Nutzen von Einzel- u. Teamrbeit." In: *Der Arbeitsmethodiker* (cit. on p. 26).

Colorni, Alberto et al. (1994). "A Genetic Algorithm to Solve the Timetable Problem." In: *Computational Optimization and Applications* (cit. on p. 38).

Cormen, Thomas H. et al. (2009). *Introduction to Algorithms* (cit. on p. 33).

Dantzig, George B. and Mukund N. Thapa (1997). *Linear Programming 1: Introduction* (cit. on pp. 30, 32).

DIN31051:2012-09 (2012). *DIN 31051:2012-09* (cit. on pp. 5, 6).

Dym, Clive (2002). *Principles of Mathematical Modeling* (cit. on p. 28).

Garey, M. R. et al. (1978). "Performance Guarantees for Scheduling Algorithms." In: *Operations Research* (cit. on p. 38).

Gelinas, Sylvie and Francois Soumis (2005). “Dantzig-Wolfe Decomposition for Job Shop Scheduling.” In: *Column Generation* (cit. on p. 37).

Grunwald, Wolfgang (1996). “Psychologische Gesetzmäßigkeiten der Gruppenarbeit : über die Grundbedingungen erfolgreicher Zusammenarbeit.” In: *Personalführung* (cit. on pp. 26, 27).

Guntner, Georg et al. (2016). *Roadmap der Instandhaltung 4.0*. URL: `http://www.salzburgresearch.at/wp-content/uploads/2015/05/IH40-Roadmap-final.pdf` (cit. on p. 2).

Gurobi (2016). *Gurobie Corporate Brochure*. URL: `http://www.gurobi.com/pdfs/Gurobi-Corporate-Brochure.pdf` (cit. on pp. 32, 52, 53).

Gutenberg, Erich (1966). *Grundlagen der Betriebswirtschaftslehre - Band 1 die Produktion* (cit. on p. 19).

Hauke, Wolfgang and Otto Opitz (2003). *Mathematische Unternehmensplanung: Eine Einführung* (cit. on p. 36).

Higgins, Lindley R. (1995). *Maintenance Engineering Handbook* (cit. on pp. 17, 25).

Hillier, Fredericks S. and Gerald J. Liebermann (2000). *Introduction to Operations Research* (cit. on pp. 29, 31, 32).

Hojati, Mehran and Ashok S. Patil (2011). “An Integer Linear Programming-based Heuristic for Scheduling Heterogeneous, Part-time Service Employees.” In: *European Journal of Operational Research* (cit. on pp. 40, 53, 54).

Junginger, Werner (1986). “Timetabling in Germany - A Survey.” In: *Interfaces* (cit. on p. 37).

Kieser, Alfred and Peter Walgenbach (2010). *Organisation* (cit. on p. 22).

Kolesar, Peter J. (1967). “A Branch and Bound Algorithm for the Knapsack Problem.” In: *Management Science* (cit. on p. 32).

Kosiol, Erich (1976). *Organisation der Unternehmung* (cit. on p. 20).

Kroesen, Alfred (1983). *Instandhaltungsplanung und Betriebsplankostenrechnung* (cit. on p. 16).

Künzi, H.P. et al. (1967). *Numerische Methoden der mathematischen Optimierung* (cit. on p. 29).

Leidinger, Bernhard (2014). *Wertorientierte Instandhaltung* (cit. on pp. 9–12).

Lenstra, Jan Karel et al. (1990). “Approximation Algorithms for Scheduling Unrelated Parallel Machines.” In: *Mathematical Programming* (cit. on p. 38).

Leung, Joseph Y-T. (2004). *Handbook of Scheduling* (cit. on p. 33).

Matyas, Kurt (2016). *Instandhaltungslogistik* (cit. on pp. 7–12, 24).

Mintzberg, Henry (1979). *The Structuring of Organizations* (cit. on p. 23).

Mobley, R. Keith (2002). *An Introduction to Predictive Maintenance* (cit. on p. 11).

Nakajima, Seiichi (1988). *Introduction to TPM* (cit. on p. 13).

Nowicki, Eugeniusz and Czeslaw Smutnicki (1996). "A Fast Taboo Search Algorithm for the Job Shop Problem." In: *Management Science* (cit. on p. 39).

Pinedo, Michael L. (2002). *Scheduling* (cit. on p. 33).

Pinedo, Michael L. (2004). *Planning and Scheduling in Manufacturing and Services* (cit. on p. 34).

Plattform Industrie 4.0 (2016). *Was ist Industrie 4.0.* URL: `http://www.plattform-i40.de/I40/Navigation/DE/Industrie40/WasIndustrie40/was-ist-industrie-40.html` (cit. on p. 1).

PWC (2015). *Industrie 4.0 Osterreichs Industrie im Wandel.* URL: `https://www.pwc.at/publikationen/industrie-4-0-oesterreichs-industrie-im-wandel-2015.pdf` (cit. on p. 2).

Rasch, Alejandro Alcalde (2000). *Erfolgspotential Instandhaltung* (cit. on p. 17).

REFA (1991). *Methodenlehre der Betriebsorganisation, Planung und Steuerung Teil 2* (cit. on pp. 7–10, 16).

Sauppe, Jason J. et al. (2015). "Assigning Panels to Meeting Rooms at the National Science Foundation." In: *Interfaces* (cit. on p. 40).

Schenk, Michael (2010). *Instandhaltung technischer Systeme* (cit. on p. 7).

Schreyögg, Georg (2008). *Organisation* (cit. on pp. 18, 20–22).

Schreyögg, Georg and Jochen Koch (2007). *Grundlagen des Management* (cit. on p. 19).

Schuh, Günther (2006). *Produktionsplanung und -steuerung* (cit. on p. 16).

Snyman, Jan A. (2005). *Practical Mathematical Optimization* (cit. on pp. 27, 29, 30).

Steinmann, Horst and Georg Schreyögg (2005). *Management* (cit. on pp. 20–22).

Strunz, Mathias (2011). *Instandhaltung* (cit. on pp. 6, 24).

Vahs, Dietmar (2012). *Organisation - Ein Lehr- und Managementhandbuch* (cit. on pp. 18, 20).

Vorbach, Stefan (2015). *Unternehmensführung und Organisation* (cit. on pp. 20–22).

Wallace, Stein W. (1989). *Algorithms and Model Formulations in Mathematical Programming* (cit. on p. 30).

Wiendahl, Hans-Peter (2014). *Betriebsorganisation für Ingenieure* (cit. on p. 16).

Wildeman, R.E. (1997). "A Dynamic Policy for Grouping Maintenance Activities." In: *European Journal of Operational Research* (cit. on pp. 39, 40, 49).

Willmott, Peter and Dennis McCarthy (2001). *TPM - A Route to World Class Performance* (cit. on p. 15).

Wren, Anthony (1996). "Scheduling, Timetabling and Rostering - A Special Relationship?" In: *Lecture Notes in Computer Science* (cit. on p. 35).