euspen's 24th International Conference & Exhibition, Dublin, IE, June 2024
www.euspen.eu

Automated scheduling system for parallel gear grinding machines

Christopher Janßen[1], Melina Kamratowski[1], Mareike Davidovic[1], Thomas Bergs[1,2]

[1]Laboratory for Machine Tools and Production Engineering (WZL) of RWTH Aachen University, Campus-Boulevard 30, 52074 Aachen
[2]Fraunhofer IPT, Steinbachstraße 17, 52074 Aachen, Germany

c.janssen@wzl.rwth-aachen.de

Abstract

In gear manufacturing at the Gear Department of the WZL of RWTH Aachen University the machine scheduling approach has predominantly been manual, creating potential inefficiencies. This study presents the development of an automated scheduling system designed for the simultaneous operation of two parallel grinding machines. Starting with a detailed definition of machine scheduling requirements and an in-depth analysis of the machine environment, a mathematical formulation of the scheduling problem was established. This formulation was subsequently translated into the Python programming language for efficient implementation. To further aid planners, the system generates visual outputs in the form of Gantt charts. Based on a color-coding scheme, these charts transparently indicate which job is being processed on which machine. Upon validation, the scheduling system demonstrated that an optimized allocation plan, depending on the number of jobs, can be derived in seconds. This system significantly enhances planning processes and augments transparency and adaptability in manufacturing workflows.

Scheduling, Operations Research, Gear Manufacturing, Gear Grinding

1. Introduction

Gears are complex mechanical components that play a central role in a wide range of industrial applications. They are indispensable for the precise transmission of movements and forces in machines and technical systems. To ensure the quality and performance of gears, continuous research and further development of the load capacity and manufacturing processes of gears and gear components is necessary. The Gear Department of the Laboratory for Machine Tools and Production Engineering of RWTH Aachen University is working intensively on this scientific issue.

The organization of the manufacturing processes in the gear department of the WZL poses a major challenge due to the different types of jobs, process times and variable batch sizes. The gear manufacturing process chain includes several sequential operations, whereby gear grinding as part of hard finishing is particularly important for the quality of the final gears. Implementing a job scheduling system can increase the efficiency and reliability of planning and reduce potential bottlenecks.

The introduction of a scheduling system also leads to better traceability in planning. In addition, it is possible to react more flexibly to disruptions in the planned process, for example due to tool failures.

The primary objective of this paper is to develop and implement a job scheduling system for the gear department of the WZL of RWTH Aachen University. The specific requirements and boundary conditions of the various jobs and machines are analyzed and a suitable model is developed. The successful implementation of this system will relieve the responsible planner and at the same time increase scheduling reliability and efficiency. As a result, machine utilization could be optimized and throughput times reduced, which would lead to an overall increase in productivity in the gear department. At present, there is no adequate scheduling model that meets the requirements of gear production under the current boundary conditions and could therefore be used or adapted.

2. State of the Art

Scheduling is a decision-making process that is used in many manufacturing and service industries to optimize the distribution of resources and tasks [1]. The resources and tasks can be machines in a workshop or runways at an airport, for example [1]. Other examples include applications in the semiconductor or paper industry, optimizing catering services in a hospital or optimizing machine utilization in a factory. [2–4]

Each task can have a priority, an earliest possible start time and a due date. The goals of optimization can be, for example, to minimize the completion time or the number of delayed tasks. It is also possible to ensure optimum utilization of machines by distributing tasks. [1]

In the following, the focus is on scheduling for planning machine assignments. Job scheduling takes place directly before the jobs are released and production is carried out [5]. In machine scheduling problems, the assignment of jobs to work carriers or machines or vice versa is considered, taking into account targets and restrictions [6].

It is generally assumed in job scheduling problems that the number of jobs and machines is finite. The number of jobs is denoted by n and the number of machines by m. Usually, the index j refers to a job, while the index i refers to a machine. If a job requires a number of processing steps or operations, then the pair (i, k) refers to the processing step or operation of job j on machine k. [7]

The processing time of a job j on machine i is represented by p_{ij}. The release date of a job j is represented by r_j and can also be referred to as the provisioning time. It is the time at which job j arrives in the system, i.e. the earliest time at which job j can be processed. The actual starting time of a job is represented by t_j. [8]

The latest completion date or due date of job j is referred to as d_j and represents the binding dispatch or completion date. Completing the job after the completion date may be allowed, but will result in the imposition of a penalty. If a completion date for job j must be met, it is referred to as a deadline and is defined as $\overline{d}_j$. [1]

The makespan C_{max} is the total time required to complete a set of jobs on one or more machines. The goal in many scheduling problems is to minimize the makespan C_{max}, especially when it comes to making optimal use of the capacity of machines or workstations and minimizing idle times. An overview of these parameters is given in **Figure 1**.

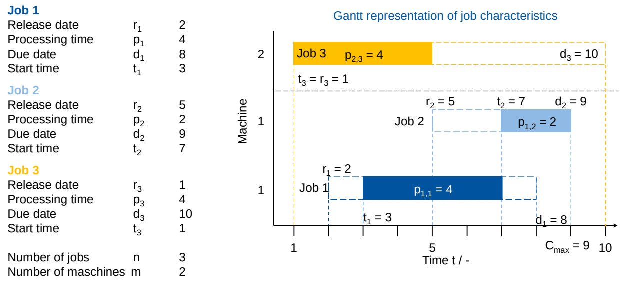

Figure 1: Representation of job characteristics

3. Objective and Approach

The aim of this report is to create a job scheduling system for optimizing production planning of gear grinding machines in a research institute's gear department. This system aims to simplify planning, improve visualization, and enhance planning reliability for both research and industrial projects. The approach involves analyzing production requirements, including existing machinery, orders, and time constraints.

Based on the requirements analysis, a mathematical formulation of the scheduling problem is derived. This involves modeling machines, orders, and time constraints while aiming for a practical yet adaptable model.

Next, the model is implemented in the programming language Python and solved using appropriate software. Validation is conducted using test data sets to assess correctness and solver performance. Both, commercial and open-source solvers are employed for this evaluation.

4. Modeling of the Scheduling Problem

This chapter elucidates the current state of machinery, clamping systems, and procedural factors impacting production order planning. It derives boundary conditions from available data and formulates assumptions for modeling the scheduling problem, culminating in its mathematical description.

4.1. Manufacturing process in the Gear Department

The gear department of the WZL of RWTH Aachen University currently possesses two machines dedicated to the hard finishing of gears. These machines are the KX 500 Flex by Kapp Niles (referred to as KX 500) and the Viper 500 by Klingelnberg (referred to as Viper).

Both the KX 500 and Viper machines offer versatility in gear production, and are capable of performing operations such as profile, internal profile, and generating grinding. They can handle gears with a maximum tip diameter of d_a = 500 mm. However, their technical specifications differ, particularly in terms of module range m_n and maximum gear width b.

The Viper machine primarily serves research purposes, predominantly within a university setting, where it is employed for investigating the profile grinding process. In contrast, the Kapp KX 500 is assigned to research purposes in regard to generating grinding and the manufacturing of test gears.

In addition to these primary distinctions, various influencing factors come into play, including clamping methods, interfering contours, and tool availability, which can impact the choice between the two machines. However, the machine selection may also be swayed by factors such as the current workload and other constraints.

Current scheduling processes cover a timeframe of 3-5 months and typically involve managing approximately 10-15 jobs. Nevertheless, the ability to adapt flexibly to personnel or material shortages, whether arising from workforce constraints or tool and machinery availability, is essential. The frequency and extent of these adjustments can vary, occurring on a weekly or monthly basis depending on the specific circumstances.

4.2. Problem formalization

Considering the machine environment and the constraints or specifications of the production job, the following boundary conditions and assumptions for the model are delineated.

Boundary conditions

(1) Due dates: Each job has a fixed end time by which it must be completed at the latest.

(2) Availability: The jobs have an availability from which they can be started. They cannot be processed before this time.

(3) Sequence: There is no predefined sequence. The sequence is determined based on the availability of machine capacity and the due dates.

(4) Process times: Each job has a specific process time, which specifies how long this job takes on a machine to be fully completed.

(5) Mandrels: There is a limited number of mandrels for part clamping. If two jobs on different machines require the same mandrel, they cannot be processed at the same time. The same applies to shaft grinding.

(6) Two machines: Two gear grinding machines are available, on each of which only one job can be carried out at a time.

(7) Prioritization: The jobs are not prioritized. Prioritization is defined implicitly via the due dates of the respective job.

(8) Machine assignment: Some jobs must be executed on a specific machine, while other jobs can be processed on both machines.

(9) Machine speed: Both machines work at the same speed and have the same performance capacities.

(10) Maintenance and repair: The maintenance and repair of the machines can be specified as a job with a start and end date and an equivalent duration. This blocks other jobs at the relevant time.

Assumptions

The following assumptions are made in order to simplify the scheduling problem and map it realistically:

(1) Tools and Workpieces: It is assumed that tools and workpieces are readily available and prepared for each job as soon as the job becomes available. All requisite preparations have been executed in advance.

(2) Set-up time: The set-up time for each job is variable and cannot be standardized. It is therefore taken into account in the production time or process time of the respective job.

(3) Set-up process: The set-up process for the machines can take different lengths of time. This variance is also taken into account in the process time of the respective jobs.

(4) Quality recording: The recording of the quality of the manufactured gears during the production process is already included in the process time of the respective job.

(5) Process times: The process times are identical on both machines for all jobs. However, since a change is conceivable, the following problem is modeled with variable process times depending on the respective machine.

(6) Interruption: Jobs cannot be interrupted. Jobs with large quantities can be divided into smaller jobs.

Taking these boundary conditions and assumptions into account, the scheduling problem is modeled and a suitable strategy for optimizing production planning is developed.

4.3. Mathematical formulation

Certain indices, parameters and decision variables are required for modeling.The indices, sets and variables are shown in **Table 1**.

Table 1 Variables required for modelling

Variables	Description
C_{max}	Makespan
$i,j \in J = \{1, 2, \ldots, n\}$	Amount of jobs
$k \in = \{1, 2\}$	Amount of machines
r_i	Release date of job i
$\bar{d}_i$	Due date of job i
t_i	Start time of job i
p_{ik}	Process time of job i on machine k
x_{ik}	Allocation of job i on machine k
y_{ij}	Processing of job i before job j
w_{ij}	Usage of Ressources of job i and j
M	A big number e.g. 10,000

Based on the selected indices, parameters and decision variables, the scheduling problem can be defined as depicted in Figure 1 and Figure 2. The objective of the optimization is described by Eq. (1) to minimize the makespan C_{max} or the completion time, see **Table 2**. Eq. (2) defines that the start time t_i of a job i plus the respective process time p_{ik} must be less than or equal to the completion time $\bar{d}_i$. This equation prevents a production delay of the jobs. Eq. (3) specifies that the start time t_i of each job must be greater than the respective release time r_i. Provided that all release times r_i are greater than or equal to 0, it is also ensured that the start time t_i is not shifted into a negative time range.

Table 2 Equations 1-7

Description	Mathematical Formulation	
1) Objective	$\min C_{max}$	
2) Deadline	$t_i + p_{ik} \leq \bar{d}i \quad \forall i \in J, \forall k \in K$	
3) Start time	$t_i \geq r_i \quad \forall i \in J$	
4) Sequence	$t_i + p_{ik} - M \cdot (1 - x_{ik}) \leq t_j + M \cdot (1 - y_{ij}) + M \cdot (1 - x_{jk})$	*
5) Sequence	$t_j + p_{jk} - M \cdot (1 - x_{jk}) \leq t_i + M \cdot (y_{ij}) + M \cdot (1 - x_{ik})$	*
6) Resources	$t_i + p_{ik} - M \cdot (1 - x_{ik}) \leq t_j + M \cdot (1 - y_{ij}) + M \cdot (x_{jk}) + M \cdot (1 - w_{ij})$	*
7) Resources	$t_j + p_{jk} - M \cdot (1 - x_{jk}) \leq t_i + M \cdot (y_{ij}) + M \cdot (x_{ik}) + M \cdot (1 - w_{ji})$	*
*	$\forall i, j \in J, \forall k \in K, i \neq j$	

Eq. (4) and Eq. (5) establish a direct sequential relationship between two jobs i and j. Both constraints specify that jobs processed on the same machine cannot occur in parallel. The formulation guarantees that the start time t_i and its corresponding processing time p_{ik} must be less than or equal to the start time of the succeeding job j.

Eq. (6) and Eq. (7) introduce the variable w_{ij} to consider mandrel usage and shaft grinding. Job start times t_i and t_j are determined as in Eqs. (4) and (5). However, a sequencing constraint is applied only when jobs i and j are processed on different machines, requiring limited mandrel availability.

Further mathematical formulations are depicted in **Table 3**. Eq. (8) ensures that when switching between the profile and generating grinding process for jobs i and j, an additional setup time is taken into account.

The sum of the decision variables x_{ik} across all machines k for each job i in Eq. (9) ensures that each job is assigned to a machine. If a job is already assigned to a machine by a user, the secondary condition is ignored.

Eq. (10) states that the sum of all start times and process times for all machines is less than or equal to the makespan C_{max}. The factor x_{ik} ensures that the respective process time is only taken into account if a job is actually executed on the corresponding machine.

Eq. (11) and Eq. (12) define the partial machine assignments or the mandrels used and the necessity using a shaft clamping system. If a decision variable is part of a table specified by the user for the machine assignment L_{pred} or the mandrel assignment W_{pred} (pred for predetermined), the value of the decision variable is assigned according to the specification.

Eq. (13) defines the type of decision variables for x_{ik}, y_{ij} and w_{ij}. All three decision variables are defined as binary variables. This condition means that the variables can only assume the values 0 and 1.

Eq. (14) ensures that the makespan C_{max} is greater than or equal to zero. This prevents the makespan C_{max} from being optimized into the negative range.

Table 3 Equations 8-17

Description	Mathematical Formulation
8) Process	$t_i + p_{ik} - M \cdot (1 - x_{ik}) + \text{setup} \cdot (z_i - z_j) \leq t_j + M \cdot (1 - y_{ij})$
9) Allocation	$\sum_{k \in K} x_{ik} = 1 \quad \forall i \in J$
10) Def. C_{max}	$\sum_{k \in K} t_i + p_{ik} \cdot x_{ik} \leq C_{max} \quad \forall i \in J$
11) Allocation x_{ik}	$x_{ik} = 1, \qquad \forall i, k \in L_{pred}$
12) Allocation w_{ik}	$w_{ij} = 1, \qquad \forall i, j \in W_{pred}$
13) Def. Variables	$x_{jk}, y_{ij}, w_{ji} \in [0, 1]$
14) Limitation C_{max}	$C_{max} \geq 0$
*	$\forall i, j \in J, \forall k \in K, i \neq j$

5. Validation of the scheduling system

To ensure the functionality of the scheduling system with regard to the boundary conditions and the target function, various scenarios are analyzed using test data sets. This is followed by a comparison of the functionality of the commercial Gurobi solver with the freely available PuLP solver.

The first test data set is used to verify whether the boundary conditions, see section 4.1, are met. These include the limitations that no jobs may be processed in parallel on a machine, the release time r_i of the jobs must be met and the due date $\bar{d}_i$ must not be exceeded. Six different jobs are scheduled for optimization in the first test data set.

All jobs have the release time $r_i = 1$, but have different process times p_{ik} and individual end times $\bar{d}_i$ at which the jobs must be completed. The results show that parallel processing on one machine can be ruled out, see **Figure 2**. Therefore, it was possible to assign a sequence relationship to all jobs.

Furthermore, all jobs were evenly distributed between the two machines in order to minimize the makespan C_{max}. Each machine was assigned a job with a process time $p_{ik} = 1$, 2 and 3 respectively. The makespan is $C_{max} = 7$, which corresponds to the latest end time $\bar{d}_i$ of jobs 1 and 4. All specified end times were met.

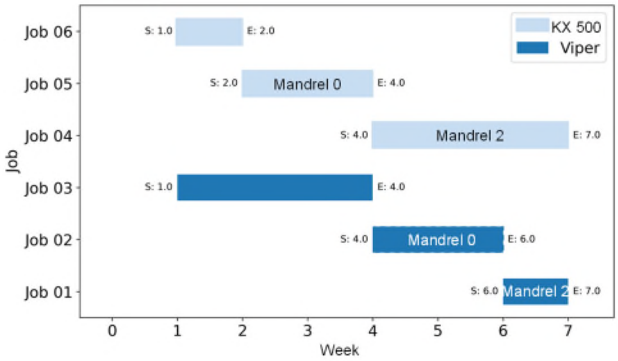

Figure 2. Results of a validation test

Due to the availability of two mandrels of the corresponding size 2, parallel machining on two machines is possible without any problems. The results of the optimization show the functionality of considering the availability of different mandrels. Jobs that require a mandrel that is only available once are not scheduled in parallel. When using mandrels that are available twice, however, jobs can be scheduled in parallel.

For the second test, the machining processes, profile grinding (P) and generating grinding (W) were specified. Based on the specified process, it is checked whether a sequence of identical processes leads to reduced setup times and the avoidance of penalty costs. The results are shown in **Figure 3**.

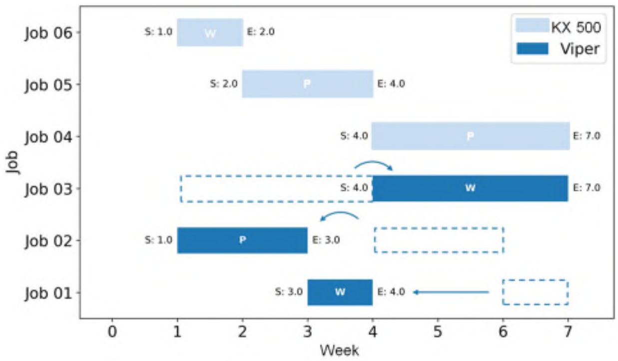

Figure 3. Results of the second validation test

There are no changes to the assignment of jobs on the KX 500 Flex, as there is already an optimum here. However, there are significant changes in the sequence on the Viper. Job 1 was scheduled to start in week 3. Job 2 starts on the same day as week 1, while the start of job 3 has been moved to week 4. This postponement ensures that the grinding process is not interrupted by another process.

6. Performance comparison

In a comparison, the solving speed of the commercial solver Gurobi and the open source solver PuLP is analyzed, see **Figure 4**. The scheduling system was implemented using the commercial solver Gurobi. To analyze the difference with an open source solution, a comparison was made. The results show that there is a critical threshold of 19 jobs, beyond which the solution time increases significantly from seconds to minutes and beyond. For example, with a job count of 20, the Gurobi solver requires 770 seconds (equivalent to approx. 13 minutes), while finding a solution with PuLP takes 5,500 seconds (approx. 1.5 hours). With a job count of 50, these times increase to 4,500 seconds (75 minutes) for Gurobi and 53,500 seconds (approx. 15 hours) for PuLP. These data illustrates the exponential influence of the number of jobs on the increasing complexity of the problem and the associated solution time. A duration of 770 seconds can be considered acceptable in the context of optimization, as it represents a significant time saving compared to manual optimization methods. This assessment also applies to a duration of 4,500 seconds with a job count of 50. However, the comparison suggests that with increasing complexity, a commercial solver such as Gurobi is preferable.

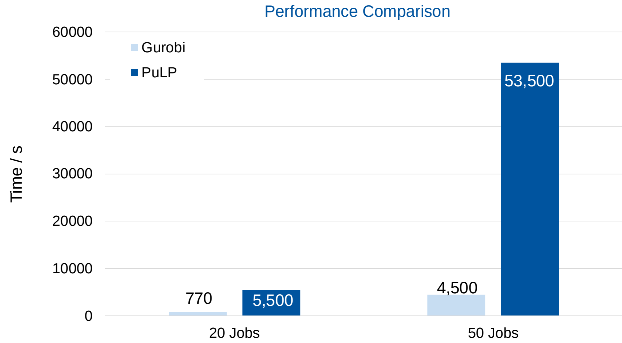

Figure 4: Results of a performance comparison

7. Summary and outlook

This report outlines the development of a tailored scheduling system for hard finishing at the gear department of WZL of RWTH Aachen University. The main goal was to create an efficient and adaptable production schedule for two specialized machines: the KX 500 Flex from KAPP NILES and the Viper 500 KW from KLINGELNBERG.

Through analysis, a flexible mathematical model for machine occupancy and clamping systems was formulated and implemented using Python and the Gurobi solver. Validation with two data sets confirmed functionality and adherence to specifications. A comparison with the PuLP solver showed significant time savings and efficiency gains as dataset complexity increased.

The development of a job scheduling system at WZL Gear Department not only enhances operational efficiency within this academic setting but also holds significant potential for broader industrial application. This research, while specific to the WZL's unique needs, offers a scalable and adaptable framework that can inform scheduling practices in various manufacturing sectors. This work provides a methodological blueprint for industries facing similar production challenges, emphasizing the broader applicability and potential of academic research to improve industrial production planning and efficiency.

References

[1] Pinedo ML. Scheduling. Boston, MA: Springer US; 2012.

[2] Dugardin F, Chehade H, Amodeo L, Yalaoui F, Prins C. Hybrid Job Shop and Parallel Machine Scheduling Problems: Minimization of Total Tardiness Criterion. In:. Multiprocessor Scheduling, Theory and Applications: I-Tech Education and Publishing; 2007. doi:10.5772/5227.

[3] Chen JC, Wu C-C, Chen C-W, Chen K-H. Flexible job shop scheduling with parallel machines using Genetic Algorithm and Grouping Genetic Algorithm. Expert Systems with Applications. 2012;39:10016–21. doi:10.1016/j.eswa.2012.01.211.

[4] Abderrabi F, Godichaud M, Yalaoui A, Yalaoui F, Amodeo L, Qerimi A, Thivet E. Flexible Job Shop Scheduling Problem with Sequence Dependent Setup Time and Job Splitting: Hospital Catering Case Study. Applied Sciences. 2021;11:1504. doi:10.3390/app11041504.

[5] Kistner K-P, Steven M. Produktionsplanung. Heidelberg, sPhysica-Verlag HD; 2001.

[6] Domschke W, Scholl A, Voß S. Produktionsplanung: Ablauforganisatorische Aspekte. Berlin: Springer; 1993.

[7] Brucker P. Scheduling Algorithms. Berlin: Springer; 2004.

[8] Baker KR, Trietsch D. Principles of sequencing and scheduling. Hoboken, NJ, USA: Wiley; 2019.