THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Mathematical Optimization in Flexible Job Shop Scheduling

Modelling, Analysis, and Case Studies

Karin Thörnblad

CHALMERS | UNIVERSITY OF GOTHENBURG

Department of Mathematical Sciences
Division of Mathematics
Chalmers University of Technology and University of Gothenburg
Göteborg, Sweden 2013

This work was supported by:
GKN Aerospace (formerly Volvo Aero Corporation), Vetenskapsrådet (The Swedish Research Council), and NFFP (National Aviation Engineering Research Programme).

Cover illustration: An example of an optimal schedule for the multitask production cell found by the iterative procedure (see Paper III in the thesis for similar instances).

Mathematical Optimization in Flexible Job Shop Scheduling
Modelling, Analysis, and Case Studies

Karin Thörnblad
ISBN 978-91-7385-883-0

Doktorsavhandlingar vid Chalmers Tekniska Högskola
Ny serie Nr 3564
ISSN 0346-718X

Department of Mathematical Sciences
Chalmers University of Technology | University of Gothenburg
SE – 412 96 Göteborg, Sweden
Phone: + 46 (0)31–772 1000
Fax: + 46 (0)31–161973

karin.thornblad@gknaerospace.com
karin@thornblad.com

Printed in Göteborg, Sweden 2013

Mathematical Optimization in Flexible Job Shop Scheduling

Modelling, Analysis, and Case Studies

Karin Thörnblad

Department of Logistics Development
GKN Aerospace Engine Systems, Trollhättan, Sweden
and
Department of Mathematical Sciences
Chalmers University of Technology and University of Gothenburg

Abstract

The major theme of this thesis is the mathematical optimization modelling of the flexible job shop scheduling problem. This problem is present in the so-called multitask cell—a production cell at GKN Aerospace's facility located in Trollhättan, Sweden—which has served as a case study during the work with this thesis. The thesis consists of two parts, of which the first considers the major theme which runs through the first three appended papers. The second part of the thesis corresponds to the fourth appended paper and deals with the computational complexity of flow shop scheduling problems with deteriorating jobs. The basic flow shop scheduling problem is a special case of the flexible job shop scheduling problem.

In the first paper, two mathematical optimization models are formulated for the general flexible job shop scheduling problem. One of the models—a so-called time-indexed model—is incorporated into an iterative procedure, which is then shown to solve instances of the problem much faster than when the model is directly solved. The iterative procedure outperforms two other mathematical optimization models regarding the computation time required to find the best solution. We have studied two main types of objective functions: the makespan and a weighted average of completion times and tardiness of the jobs; we observed a large difference in the performance of the scheduling models between the objectives. This implies that any evaluation of scheduling algorithms must be done with respect to an objective that is relevant for the application for which they are intended. The makespan objective is the most widely used within research, however, it is seldom relevant in real applications. We recommend as a suitable objective to minimize the weighted average of the completion times and the total tardiness where the tardiness weight decreases with the job's due date.

In the second paper, we study the scheduling of the multitask cell and propose a means to schedule the cell including fixture availability and preventive maintenance activities. A suitable definition of the tardiness weights is proposed, and the quality of the resulting schedules, when applied in a dynamic environment, is discussed. A time-indexed model is formulated including three variants of modelling the preventive maintenance activities. Real instances collected from the multitask cell are solved by a version of the iterative procedure employing the time-indexed model. We are able to produce near-optimal schedules for the real data instances within 15 minutes, which is regarded as an acceptable time frame.

The third paper continues the work presented in the second paper: the time-indexed model is further developed and the possibility to schedule operations with unmanned processing during night shifts is explored. We are able to substantially increase the output of the cell with the inclusion of unmanned processing during night shifts. For a number of real scenarios for the coming shift collected from the multitask cell we compare the schedules produced by our iterative procedure—with and without the night shift possibilities included—with those constructed by the FIFO (first–in, first–out) and by a built-in (in the multitask cell) critical ratio (CR) priority dispatching rule. The objective values obtained after running our iterative procedure for 15 minutes were on average 0.71 times the corresponding values computed by the FIFO or CR rules.

The fourth appended paper establishes a complexity result for a flow shop scheduling problem in which the jobs' processing times are functions of the jobs' starting times.

Keywords: Flexible job shop scheduling, Mixed integer linear programming (MILP), Time-indexed formulation, Makespan, Tardiness, Fixture availability, Preventive maintenance, Night shift, Unmanned time window, Dynamic scheduling, Priority rules, Dispatching rules, Critical ratio.

List of Publications

This thesis is based on the research presented in the following articles.

Paper I K. Thörnblad, A.-B. Strömberg, M. Patriksson, and T. Almgren, *A competitive iterative procedure using a time-indexed model for solving flexible job shop scheduling problems*, Available at `http://www.optimization-online.org/DB_HTML/2013/08/3991.html`, August 2013.

Paper II K. Thörnblad, A.-B. Strömberg, M. Patriksson, and T. Almgren, *Scheduling optimization of a real flexible job shop including fixture availability and preventive maintenance*, Revised for publication in European Journal of Industrial Engineering, (2013).

Paper III K. Thörnblad, A.-B. Strömberg, M. Patriksson, and T. Almgren, *Scheduling optimization of a real flexible job shop including side constraints regarding maintenance, fixtures, and night shifts*, Available at `http://www.optimization-online.org/DB_HTML/2013/08/3995.html`, August 2013.

Paper IV K. Thörnblad and M. Patriksson, *A note on the complexity of flow-shop scheduling with deteriorating jobs*, Discrete Applied Mathematics, **159** (2011) 251–253.

In addition to the above, the following eight publications have been written by the author within the project, but are not included in this thesis:

K. Thörnblad, A.-B. Strömberg, T. Almgren, and M. Patriksson, *Optimization of schedules for a multitask production cell*, In: 22$^{\text{nd}}$ NOFOMA conference proceedings, Kolding, Denmark, June 10–11, 2010, 13 pages.

K. Thörnblad and L. Kjellsdotter Ivert, *A comparison of schedules resulting from priority rules and mathematical optimization for a real production cell*, In: Proceedings of PLANs forsknings- och tillämpningskonferens, Skövde, Sweden, August 18–19, 2010, pp. 99–110.

K. Thörnblad, A.-B. Strömberg, M. Patriksson, and T. Almgren, *Mathematical modelling of a real flexible job shop in aero engine component manufacturing*, In: Proceedings of the 10th Workshop on Models and Algorithms for Planning and Scheduling Problems, Nymburk, Czech Republic, June 19–24, 2011, pp. 171–173.

K. Thörnblad, T. Almgren, A.-B. Strömberg, and M. Patriksson, *Optimering av scheman för en verklig produktionscell: tidsdiskretisering reducerar lösningstiden utan att lösningarnas kvalitet* *försämras*, In: Proceedings of PLANs forsknings- och tillämpningskonferens, Norrköping, Sweden, Aug. 31–Sep. 1, 2011, pp. 141–155 (in Swedish).

K. Thörnblad, *On the Optimization of Schedules of a Multitask Production Cell*, Licentiate thesis, Department of Mathematical Sciences, Chalmers University of Technology and the University of Gothenburg, Göteborg, Sweden, September 2011, 54 pages.

K. Thörnblad, A.-B. Strömberg, M. Patriksson, and T. Almgren, *A time-indexed formulation of a flexible job shop problem with a limited number of fixtures*, In: Proceedings of the 13th International Conference on Project Management and Scheduling, Leuven, Belgium, April 1–4, 2012, pp. 335–338.

K. Thörnblad, T. Almgren, M. Patriksson, and A.-B. Strömberg, *Mathematical optimization of a flexible job shop problem including preventive maintenance and availability of fixtures*, In: Proceedings of the 4th World P&OM Conference/19th International Annual EurOMA Conference, Amsterdam, Netherlands, July 1–5, 2012, 10 pages.

K. Thörnblad, A.-B. Strömberg, M. Patriksson, and T. Almgren, *An efficient algorithm for solving the flexible job shop scheduling problem*, In: 25th NOFOMA conference proceedings, Göteborg, Sweden, June 3–5, 2013, 15 pages.

Karin Thörnblad is the main author of all the publications above, including Papers I–IV.

Acknowledgements

First of all, I would like to start by expressing my gratitude towards my academic supervisors Michael Patriksson and Ann-Brith Strömberg for their generous help and guidance throughout the work with this thesis. I would also like to thank my industrial supervisor Torgny Almgren for taking the initiative to this project and for his constant support and encouragement. I have felt privileged sharing the interest in my research with all three of you.

Many thanks to my past and present colleagues and managers of the department of Logistics Development at GKN Aerospace (Anne, Susanne, Maria, Fredrik, Sture, Ulla, Andreas, Magnus, Håkan, Martin, and Joakim to name but a few) for your support, and for creating a good working environment. I would also like to thank one of the managers of the multitask cell, Christina Kann, and Nicklas Gödebu, manufacturing engineer, as well as the project leader of the former multitask project, Fredrik Johansson, for providing useful information and for putting up with all my questions. I would further like to thank Tomas Lindsta, manager of the operational profit center Machined Structures, for your encouraging support as a member of the steering committee.

Thanks to all past and present members of the optimization group at the Department of Mathematical Sciences at Chalmers (Christoffer, Peter, Anna, Adam, Mehdi, Emil, Magnus, Mahmoud, Zuzana, and Kin) for interesting discussions and for making my time at Chalmers a very pleasant memory.

A special thanks goes to Thomas Ericsson at the Department of Mathematical Sciences at Chalmers for teaching me programming in Matlab and high performance computing in C and, above all, for your interest in and support regarding problems encountered during the computational testing and the postprocessing of results.

Furthermore, I would like to gratefully acknowledge the financial support from GKN Aerospace (formerly Volvo Aero), Vetenskapsrådet (The Swedish Research Council, grant no. 621-2007-4716), NFFP (National Aviation Engineering Research Programme, grant no. 2009-01281), and VINNOVA through Chalmers Transport Area of Advance.

Gabriel, my wonderful husband! Thank you for you constant support, both as a supporting husband taking care of me and our children and for your IT support regarding the Linux environment and programming issues. Thank you for everything you are, and everything you help me to be when I am with you. I love you.

Elias, Isak, and Simon, my beautiful sons! Thank you for dragging me around in the forest when you exercise orienteering, and for being such a wonderful and intensive distraction from work. Thanks for bringing happiness and laughter wherever you go.

Karin Thörnblad
Rommele, August 2013

Contents

1 Introduction

The work with this thesis has been a journey from the first intuitive so-called engineer's model, which was not at all able to solve instances of the size needed for the practical application, to the development of an iterative solution procedure implemented with a so-called time-indexed model that, within a reasonable time frame, produces near-optimal schedules for the coming shift in the production cell studied. I hope that you will enjoy the ride!

1.1 Background

GKN Aerospace Engine Systems (formerly Volvo Aero Corporation) has invested in a complex production cell in Trollhättan, Sweden, containing a set of five multipurpose machines, with the aim to reduce the product costs, shorten the lead times, and increase the quality level and the delivery precision. The planning and control of this so-called *multitask cell* may be modelled and solved as a complex combinatorial optimization problem, namely a *flexible job shop scheduling problem* (FJSP). The control system of the multitask cell contains a built-in scheduling algorithm, which is based on a simple priority function. In the master's thesis [39], it was shown that the outcome of this algorithm is not satisfactory when considering the minimization of the total tardiness, since it is not adapted to the production of complex structures such as the aircraft engine components produced by GKN Aerospace. Therefore, the built-in scheduling algorithm is not in use and the production of components in the multitask cell is manually planned at present. The inferiority of the existing decision support leads to unnecessarily long product lead times and an inefficient use of the multipurpose machines in the multitask cell.

Since the results presented in [39] revealed that the scheduling of the multitask cell defines a challenging combinatorial problem this PhD project was born, with financial support from Volvo Aero Corporation (now GKN Aerospace), The Swedish Research Council, and later also by NFFP (Swedish National Aeronautics Research Programme) and VINNOVA (through Chalmers Transport Area of Advance).

1.2 Scope, purpose, and objectives of the thesis

The purpose of the research project is to contribute to the goal of enabling the construction of optimal, or near-optimal, schedules for flexible job shops, similar to the multitask cell at GKN Aerospace. A scheduling procedure which is fast and appropriate needs to be developed for the problem of scheduling such production cells, since the conditions are unceasingly changing with new jobs continuously arriving at the queue to the cell. Although the scheduling of the existing multitask cell is the main focus of this work, a set of constructed benchmark test instances has been utilized in some of the tests, with the purpose of obtaining more general results. These benchmark test instances are more complex than the real instances from the multitask cell in the sense that in the benchmark instances the processing times are resource-dependent. They are, however, less complex than the real instances in the sense that the benchmark instances contain no side constraints regarding, e.g., a limited fixture availability or preventive maintenance activities.

This thesis primarily focuses on developing a scheduling procedure—utilizing mathematical optimization—in order to produce optimal or near-optimal production schedules for the multitask cell in a reasonable computation time (≤ 15 minutes clocktime). This goal is achieved through

- the development of an iterative solution procedure, in which a time-indexed model is solved repeatedly employing shorter and shorter time steps, i.e., with an increasingly better accuracy;
- the development of a squeezing procedure which post-processes a discrete solution using the original non-discrete data in order to improve the results obtained by employing any specific length of the time steps in a time-indexed model;
- a benchmark comparison between our iterative solution procedure, the best (to our knowledge) mathematical model available in literature, and our so-called alternative model, which is a slightly improved version of the latter model;
- the study of these models' interrelations, from both a modelling and a computational perspective;
- a discussion on suitable objective functions including differentiated objective weights and their impact on the logistic performance of the production cell;
- the inclusion of side constraints based on a limited fixture availability, the scheduling of preventive maintenance activities, and the possibility to perform unmanned processing in the multitask cell during unmanned night shifts, in order to reflect the reality around the multitask cell.

Apart from during the unmanned night shifts, the number of staff available in the job shop is assumed to be sufficient for performing the manual work associated with all the operations scheduled, and staff scheduling is therefore not included in the work behind this thesis. Other areas not taken into account in this thesis is the availability of machining tools and limitations with respect to the storing of components between the operations in the job shop. All of these areas are subjects for future research. Limited staff availability and a limited availability of tools have a lot of similarities with the limited availability of fixtures, and could propably be included in any of our time-indexed models with little effort. The storage capacity inside the multitask cell is vast and has not yet been a limiting factor in production; hence it would be of economical reasons if limitations in inventory should be included.

1.3 Thesis outline

Chapter 1 ”Introduction” presents the background to this PhD project, including the scope, the purpose, and the objectives of the thesis. Chapter 2 ”Frames of reference” describes the multitask cell, the products, and the current production planning practice at GKN Aerospace.

Further, the flexible job shop scheduling problem (FJSP) is described, and the methods currently available to solve this problem are presented. Then, different objective functions proposed in the literature are discussed, followed by a presentation of the concept of dynamic scheduling. Existing literature concerning side constraints, similar to those included in our problem of scheduling the multitask cell, is reviewed. The chapter ends by a discussion on the computational complexity of the flexible job shop problem and the flow shop problem with deteriorating jobs.

In Chapter 3 ”The development of the mathematical models: A work diary” we summarize our work, from the first mathematical model developed to the final time-indexed model including several side constraints and our iterative procedure. The development of a problem decomposition, the choice of the time horizon and the length of the time steps for time-indexed models, and regarding side constraints and objective weights, is described. Finally, some additional (to the results presented in the appended papers) computational results are presented.

Chapter 4 ”Summary of the appended papers” presents a short summary of the appended papers, focusing on the papers' results and contributions. In Chapter 5 ”Conclusions” we conclude the results of this thesis work, and in Chapter 6 ”Future research” we suggest some areas of further work.

2 Frames of reference

This chapter describes the multitask cell at GKN Aerospace and the main surrounding activities, as well as relevant elements from operations research/mathematical optimization. In Section 2.1 the products, the multitask cell, and the production planning at GKN Aerospace are described. The general flexible job shop problem (FJSP) is described in Section 2.2, in which also a literature review over exact methods, metaheuristics, and priority dispatching rules for solving the FJSP is presented. In Section 2.3 some objective functions utilized for the FJSP are presented, and in Section 2.4 the concept of dynamic scheduling is described. Section 2.5 reviews the existing literature concerning the side constraints applicable to the FJSP that are present in the multitask cell. Finally, in Section 2.6 the computational complexity of the scheduling problems studied in this thesis is discussed.

2.1 Production planning at GKN Aerospace

GKN Aerospace Engine Systems, with its head office in Trollhättan, Sweden, develops and manufactures components for aircraft, gas turbines, and rocket engines with a high technology content, in cooperation with the world's leading producers. The main production site is situated in Trollhättan with about 2000 employees. Other production sites are situated in Kongsberg (Norway), Linköping (Sweden), and Newington (CT, USA). The planning and control of the supply chain of an aero engine component are affected by flight regularity conditions issued by flight safety authorities. As an example, all suppliers have to be approved by the authorities, and GKN Aerospace has to keep track of which mine the raw material for each final product comes from, due to the flight safety requirements on traceability.

Since the production of GKN Aerospace mainly concerns the aero engine industry, the quality requirements are high. Therefore, the processing machines are often very large and expensive, while the product volumes are typically small. Because of the logistics situation, where large and expensive machines process high-valued products in small volumes, the production at GKN Aerospace is mostly performed in workshops with functional layouts. One such workshop is the multitask production cell that is studied in this thesis.

2.1.1 The products

The products currently processed in the multitask cell are mainly combustor structures, such as diffuser cases and compressor rear frames (Figure 1), which are integrated cast structures made of a nickel-based material. The manufacturing process of these products is a complex combination of turning, milling, and drilling operations, which are mixed with NDT (Non Destructive Testing) procedures, and in some cases also welding operations and heat treatment. Other components processed in the multitask cell are the inner and outer rings of a turbine exhaust case, which are made in a martensitic stainless steel.

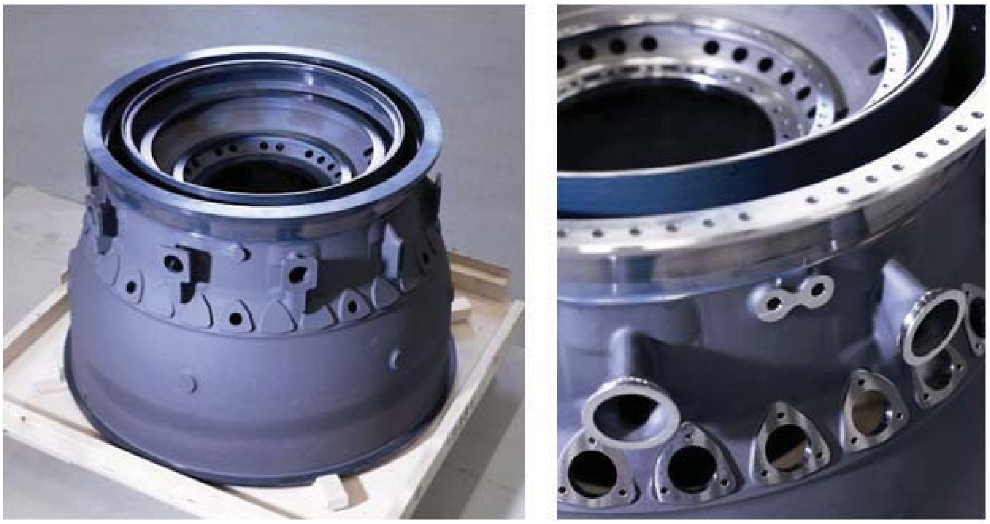

Figure 1: To the left is shown a semi-finished compressor rear frame—one of the parts processed in the multitask cell. To the right is shown a close-up of the final product.

2.1.2 The multitask production cell

The multitask cell consists of ten processing resources along with a central tool storage and an input/output conveyor; see Figure 2. The production cell is designed to carry out a large variety of jobs; five of the cell's resources are multipurpose machines that are able to perform three types of processing tasks (turning, milling, and drilling).

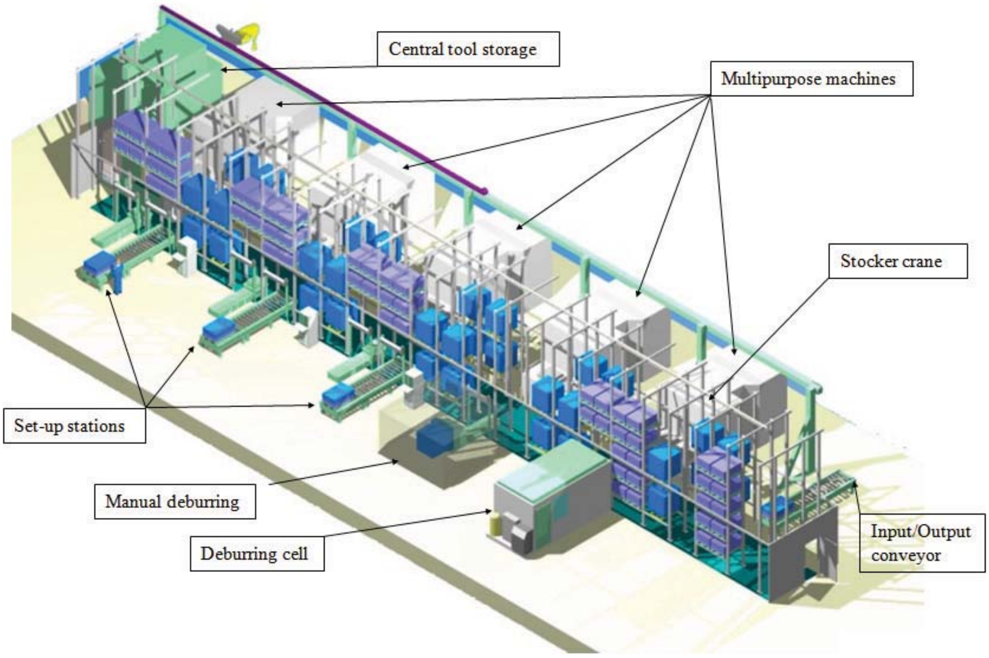

Figure 2: An overview of the multitask cell.

The multitask cell was built with the aim of achieving a high degree of machine utilization, reducing product lead times, and being flexible with regard to both the product mix and the type of processing. Presently, the multitask cell executes about 30 types of operations on eight products. Each part typically visits the multitask cell multiple times on its way to completion; see Figure 3. In this theses, one such visit to the multitask cell is called a *job*.

2.1.3 The production planning of the multitask cell

An overview of the supply chain for the civil aero engine components manufactured by GKN Aerospace is illustrated in Figure 3. The box denoted ”GKN Production” illustrates a typical path through the factory of one of the eight products that are processed in the multitask cell.

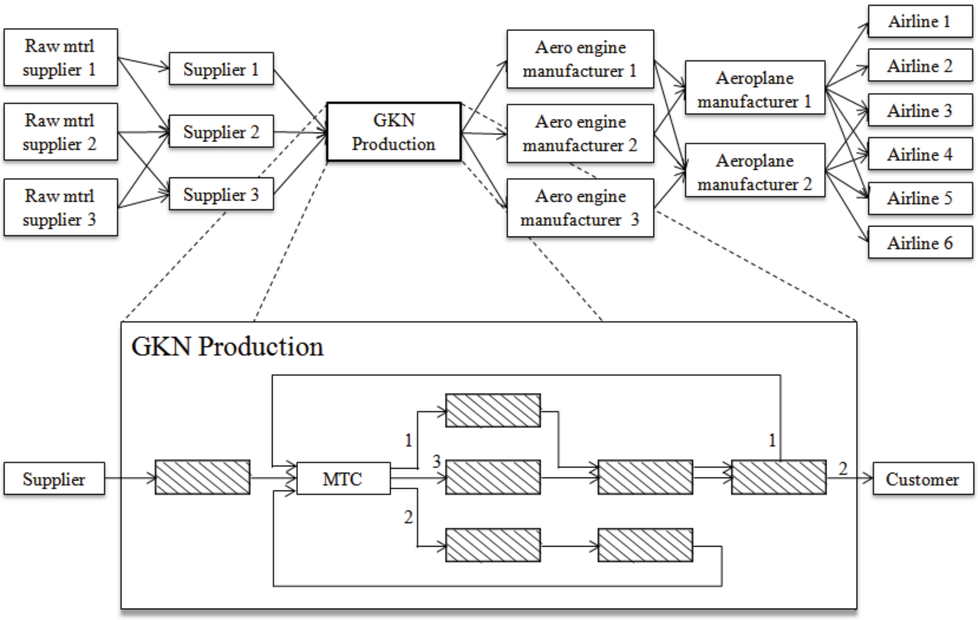

Figure 3: An overview of the supply chain for the aero engine components produced by GKN Aerospace. The box denoted ”GKN Production” illustrates a typical path of a component on its way to completion, with three visits to the multitask cell (denoted MTC). The striped boxes represent operations performed outside the multitask cell and the numbers on the arcs indicate the order in which the routes are to be taken.

The parts that are ready to be processed in the multitask cell are those that are checked-in at the input conveyor but not yet mounted into a fixture at a set-up station. After the check-in, the parts are transported by a stocker crane to special storage locations inside the multitask cell. There are also storage areas inside the cell for parts that are already mounted into fixtures. Each part to be processed in the multitask cell follows a specific routing, which consists of three to seven so-called *route operations* through the set of resources, starting and ending by the mounting into and demounting out of a fixture at one of the three set-up stations. In most cases, the second route operation consists of the processing in one of the five multipurpose machines.

The planning system at GKN Aerospace—the ERP system (Enterprise Resource Planning)—provides two reports, which propose the job priorities. One report is based on the earliest due date (EDD) priority rule and the other is based on the first–in, first–out (FIFO) priority rule.

Despite recent organizational changes among the planning staff in the multitask cell, the current production planning of the cell still is manual, as described in [60, 84]. As each job is allowed to be processed only in a subset of the multitask machines, the planning is not a simple task: For example, even though the processing machines are of the same type, they are not identical, and some of the machines have been rebuilt in order to be able to process larger components. Also, some machines are better able than others to repetitively produce components with very small deviations from target values, such as measurements of rotundity and thickness. For some products, the requirements on these measurements are very high, due to flight safety issues. Another reason why some jobs can only be processed in a subset of the machines is that some products are made of a different material, and the price paid by scrap dealers is much lower for mixed metal chips (the metal scrap from the turning process) than for sorted materials. This reason differs from the others, since it can be eliminated by either scheduling a cleaning operation in a machine between any two operations comprising parts made of different materials, or selling the metal chips at a lower price. As a consequence of the low product volumes and the immobile expensive machines, most of the components take different routes through the factory, such as the one illustrated in the box in Figure 3. For a manual planner it is therefore almost impossible to obtain a good overview of the planning situation.

As decision support, the planning personnel may use one of the two reports from the ERP system, or the job priorities based on a Critical Ratio (CR) supplied by the built-in control system of the production cell. Since the built-in scheduling CR priority dispatching rule has the benefit of being integrated into the control system, there is currently an interest to employ it in the multitask cell; in [39] it was, however, found to perform badly. Therefore we compared, in [86] (Paper III), the resulting schedules constructed by the CR and FIFO priority dispatching rules with our proposed scheduling method employing mathematical optimization.

The critical ratio at time t of a job j with due date d_j—as defined in the built-in control system of the multitask cell—is given by

$$\mathrm{CR}_j(t) := \begin{cases} \min\limits_{\text{all operations}} \left\{ \dfrac{1 + (d_j - t)(\texttt{\#machines})}{1 + \mathrm{TRPT}} \right\}, & t < d_j, \\ \min\limits_{\text{all operations}} \left\{ \dfrac{1}{1 + (t - d_j)(\texttt{\#machines})(1 + \mathrm{TRPT})} \right\}, & t \geq d_j, \end{cases} \tag{1}$$

where ”TRPT” denotes the total remaining processing time of an operation of the job j, and `#machines` denotes the number of machines allowed to process the operation; see [86] (Paper III) for a precise definition. The job is given the minimum critical ratio among its operations, and the job possessing the lowest critical ratio at time t is given the highest priority. When a job is late, i.e., when $t \geq d_j$, the job is given a higher priority, since the critical ratio is then decreased as the factor $\texttt{\#machines} > 1$ is moved from the numerator to the denominator in (1).

The FIFO priority dispatching rule process the jobs in the order in which they arrive at the machine. For the case of the multitask cell, it is not, however, that simple, since the jobs arrive to the entire production cell, and not to a single machine. In [86] (Paper III) the FIFO schedules were constructed such that at any time, the operation possessing the highest priority according to FIFO—and which is available for scheduling, regarding, for example, precedence and fixture constraints—was scheduled in an allowed resource.

The problem of scheduling the operations in the multitask cell is classified as a *flexible job shop scheduling problem* (FJSP) in the operations research nomenclature. In the following subsection, this problem is described in more general terms.

2.2 Flexible job shop scheduling

The job shop scheduling problem is one of three classic *shop scheduling* problems, the other two being the *flow shop* and the *open shop* scheduling problems; see [16]. The *job shop problem* (JSP) is defined as that to find the optimal sequences for a given set of jobs on a given set of machines. Each job consists of a number of operations, which must be processed in a given order. The constraint indicating that one specific operation must precede another specific operation is called a *precedence constraint*. Associated with each operation is thus a job, a machine, and a processing time. The FJSP is an extension of the JSP, in the sense that an operation is allowed to be processed by any machine from a given subset of the machines; see [18, Chapter 4].

Since the JSP and the FJSP fall into the class of NP-complete problems (see Section 2.6), they are among the most difficult (in terms of the computational effort required) to solve. Therefore, a large number of solution approaches have been proposed in the operations research literature; see [13, 40] for surveys of job shop scheduling techniques. In Section 2.2.1 we describe exact scheduling techniques, such as mathematical optimization. In Sections 2.2.2 and 2.2.3 we describe the approximate scheduling techniques of metaheuristics and the simpler, however widely used, methods of priority dispatching rules, respectively.

2.2.1 Exact methods

Mathematical models of problems similar to the JSP have been proposed ever since the birth of mathematical optimization and *integer programming* (IP) in the middle of the 20th century, starting with Wagner ([92]), Bowman ([14]), and Manne ([49]). These formulations model the dimension of time in three different ways, which reflects their respective definitions of the decision variables.

In [49], Manne studied the problem of sequencing jobs with precedence constraints on a single machine. The jobs' starting times are represented by integer variables [which may be represented by continuous variables, but Manne applied IP instead of *mixed integer linear programming* (MILP) (see [55] for an introduction to MILP)]. Another set of decision variables is defined as: $y_{jq} := 1$, if job j precedes job q, and 0 otherwise. There are many examples of *Manne models* (i.e., using this type of variables) of the FJSP in the operations research literature, i.e., models that use this type of variables; see [43, 47, 51, 58, 69, 100]. In [86] (Paper I), we present a Manne model of the general FJSP, and in [80, Section 5.2] we formulate a Manne model for the problem of scheduling the multitask cell.

Wagner ([92]) considers the classical job shop problem, although calling it the *machine-scheduling problem*. Since solely the ordering of the jobs on each machine is considered, the time dimension is implicitly treated. In this model, the binary decision variable equals 1 if the corresponding job is scheduled in a specific resource at a specific order-position, and 0 otherwise. See [45] for an example of a Wagner model of an FJSP.

A third way of modelling time is considered by Bowman ([14]), who denotes the classical job shop problem the *schedule–sequencing problem*. The planning period is divided into an integer number of time periods of equal length. The binary decision variables used by Bowman equal 1 if the corresponding job is processed by a specific resource during a specific time period, and 0 otherwise. A Bowman model for the FJSP can be found in [32] with an objective function including three types of costs derived from failing to meet the due dates, in-process inventory costs, and costs related to not fully completed orders. In [80], we present a Bowman model for the problem of scheduling the multitask cell.

The Bowman model is a so-called *time-indexed formulation*—it is based on a time discretization of the planning horizon. There are many other variants of time-indexed formulations of scheduling problems proposed in the operations research literature. For example, Pritsker et al. [63] propose a model for a multiproject scheduling problem, in which the binary decision variable equals 1 if the corresponding job is *completed* in a specific time period, and 0 otherwise. Another time-indexed formulation—in which a decision variable equals 1 if the corresponding job *starts* in a specific time period, and 0 otherwise (called *nail variables* in this thesis)—is found in [72], and more generally in [96] for production planning and scheduling problems. The formulations employing variables for each discrete time period lead to very large models in terms of numbers of variables and constraints, but formulations using nail variables typically yield better lower bounds than other MILP formulations of scheduling problems; see [90]. (This lower bound of a MILP model is the optimal objective value of the continuous relaxation of the model, i.e., in which the binary requirements of the variables are relaxed to a requirement that the variable value should be in the interval $[0, 1]$.)

In [59], the number of variables and constraints resulting from the *Manne model*, two variants of the *Wagner model*, and two variants of the *Bowman model* are studied for the JSP and two variants of flow shop problems. From the results of this investigation the author suggests that—in terms of the model sizes—the Manne model is the best formulation. The objective considered in ibid. is the minimization of the *makespan*, i.e., the minimization of the total processing time required, from the start of the first to the completion of the last job. In the evaluations performed in [24] for the FJSP, the Manne model of [58] was found to be the best in test. The objective studied was (again) the minimization of the makespan and the comparisons were made with one Wagner model, two other Manne models, and one Bowman model.

In previous work ([80]) we studied a Manne, a Bowman, and a time-indexed model employing nail variables for a special case of the multitask cell scheduling problem—scheduling the operations on the five multipurpose machines only. The time-indexed model employing nail variables (also presented in Section 3.3) outperformed by far the Manne and Bowman models with respect to both computation times and the sizes of the instances that could be solved to optimality using standard optimization software.

Other evaluations including time-indexed models with nail variables for closely related scheduling problems are [9] and [3]. In [9] a time-indexed model with nail variables outperformed two Manne models for a parallel machine scheduling problem with the objective of minimizing the sum of the weighted completion times. A principally opposite result was obtained in [3], where a Manne model outperformed a time-indexed model with nail variables for a JSP with non-fixed resource availability constraints and with the minimization of the makespan as objective. One explanation of the different outcomes may be that in [80] as well as in [9] a great effort is put on determining a suitable value of the time horizon, which has a huge impact on the numbers of variables and constraints in the model, and hence on its solvability. Even if a suitable value is set on the time horizon (and a suitable time step chosen), a time-indexed model often comprises many more variables and constraints than does the Manne models. Despite this, the time-indexed models employing nail variables are shown to be competitive—and even outperforming—the Manne models, especially when the objective includes tardiness and/or completion times; see [86] (Paper I).

Instead of solving the entire time-indexed model, the approach of *column generation* can help reducing the size of the problem (see [89]). It is, however, complicated to treat side constraints, such as precedence constraints between jobs and constraints for limited fixture availability, using column generation. Such restrictions are present in the problem of scheduling the multitask cell. The article [11] presents a solution approach based on column generation for solving a time-indexed formulation of a single-machine scheduling problem with the objective of minimizing the total weighted tardiness. In ibid. the time horizon is divided into subperiods, each of which is associated with a subproblem which is then solved using column generation.

In [86, 87, 88] (Papers I–III) we have developed a new methodology to tackle the problem concerning the possibly huge number of variables and constraints of the time-indexed model: an iterative solution procedure is applied, wherein the length of the time steps in the time-indexed model is decreased between iterations. In this way, the FJSP in the first iteration—with long time steps—is relatively easy, but the discretized representation of time implies that a feasible but often non-optimal solution is found. In each iteration, the FJSP is solved with an increasingly better accuracy, and the solution from the previous iteration is both used to determine the length of the time horizon and to serve as a starting solution for the coming iteration.

In [71] Sadykov and Wolsey combine the column generation approach with another exact method, namely constraint programming, for solving a multimachine scheduling problem without precedence constraints. Constraint programming seems to yield good results: in [7] constraint programming was employed to solve the FJSP with the objective of minimizing the makespan; the best known objective values of many benchmark test instances were then found after 10 minutes of computation time.

A way to improve the lower bounds of the optimal objective value sought is by employing Lagrangian relaxation; see [5, 56, 79], who study Lagrangian relaxations of a JSP, a hybrid flow shop scheduling problem, and an FJSP, respectively. In [42] a time-indexed formulation with nail variables is investigated in order to derive efficient lower bounds through column generation and Lagrangian relaxation for a parallel machine scheduling problem. The Lagrangian relaxation of the resource constraints was found to yield the best lower bounds in the study.

The exact methods described above are often combined with so-called *metaheuristics*, in order to achieve good solutions to the scheduling problems in less time than what is typically required by exact methods. We describe a few such methods in the next section.

2.2.2 Metaheuristics

A lot of research is currently concentrated on obtaining an approximate solution to FJSPs through the application of heuristic methods. The references [2, 4, 15, 22, 41, 65, 77, 93, 97, 98] propose metaheuristics and are all published within the last three years. The algorithms proposed are typically hybrid algorithms, i.e., they combine different previously developed metaheuristics, such as *genetic*, *tabu search*, and *artificial immune algorithms*, to name but a few. According to the survey [38] the development of metaheuristics for the JSP started by the development of the *shifting bottleneck heuristic* by Adams et al. in 1988 [1]. This heuristic is still popular; see [52] for a variant for the FJSP, and [19] which combines the shifting bottleneck heuristic with a MILP model for the JSP.

Another approach for finding good solutions to the FJSP is to combine simulation models with metaheuristics; see, e.g., [75, 76], which are especially interesting since the case study in these articles is the multitask cell at GKN Aerospace. A more general description of the above-mentioned combination of methods is found in [74].

The major disadvantage of metaheuristics is that there is often no other stopping criteria than a maximum allowed number of iterations, or a maximum computation time. Further, no quality measure for a solution obtained, such as a lower bound on the optimal value, which is yielded by mathematical optimization methods (see Section 2.2.1), is provided. Therefore, the quality of any solutions obtained becomes unknown. Another weakness of the metaheuristics is that often there are several parameters that have to be carefully calibrated in order for the algorithms to produce good solutions in a reasonable time (see [38]). Comparisons between different metaheuristics often become invalid since they are usually performed from an unbalanced perspective: Typically, the approaches taken are presented with a set of very well calibrated parameter values, while other approaches included in the study for comparisons are employed using a standard parameter setting. The comparisons between metaheuristics are often performed employing public benchmark test instances. Such common instances of the FJSP are found in, for example, [7]. One way of measuring the quality of a metaheuristic is to compute the average objective value found over a number, say N, independent runs together with ”best of N runs”, as done in [97]. This provides a measure of how repetitive the algorithm is.

As we have encountered instances which could not be solved to optimality, we have compared our results for public benchmark test instances in [86] (Paper I) with the best results obtained by constraint programming ([7]) and by an artificial immune algorithm ([4]). Within the computational testing of the instances collected from the multitask cell, we have compared our results with schedules constructed by a few dispatching rules; see [84, 82, 80], and [88] (Paper III). The comparison with more elaborate metaheuristics is a subject for future research.

2.2.3 Priority dispatching rules

In the ERP system at GKN Aerospace reports containing the job priorities according to the EDD (earliest due date) and the FIFO (first–in, first–out) priority dispatching rules are available (see Section 2.1.3). Therefore, in [84, 82] we chose to compare our mathematical optimization methods with these rules. In [80, pp. 42–44] we also compared with the SPT (shortest processing time) priority dispatching rule.

In the scheduling literature over 100 dispatching rules are proposed (see [36, Chapter 15]). For an extensive summary of and a discussion on priority dispatching rules, see [12, 35]. See also [48] for some recently proposed dispatching rules. The dispatching rules are popular as they are easy to employ, but they consider only local and current conditions of a machine ([36, Chapter 15]). Since the best choice of which job to process at the current time in a given machine depends not only on the future jobs, but also on when the other machines are available, there is a lot to gain by employing methods that may take such information into account, such as metaheuristics or exact methods. Dispatching rules can be classified in various ways; a distinction can, for example, be made between *static* and *dynamic* rules ([61, Chapter 14]). Static rules are not time dependent as the FIFO priority rule. Dynamic rules are time dependent, i.e., the priorities change over time. The variant of the CR (Critical Ratio) rule that is built-in in the control system of the multitask cell, as defined in (1), is a dynamic rule.

In [6], the *mean tardiness* obtained by each of five dispatching rules are compared for FJSP instances with four levels of machine flexibility. The dispatching rules are the shortest processing time (SPT), the earliest due date (EDD), the *most work remaining time* (MWRT), the *process time/total remaining time* (PDR), and the *minimum slack* (MS).

In the recently published article [20], an algorithm using dispatching rules is developed for the FJSP with transfer batches and with the objective of minimizing the average tardiness. In [78] composite dispatching rules are generated using genetic programming and compared with known dispatching rules for the FJSP. In order to improve the results generated by metaheuristics, dispatching rules are often included, especially for generating initial solutions ([21]).

2.3 On the choice of objective function

The objective that is the most studied for scheduling problems is the minimization of the makespan; see [38]. The early work of Wagner ([92]), Bowman ([14]), and Manne ([49]) considers the minimization of the makespan. Other common objectives are related to the jobs' earliness/tardiness and completion times, and/or inventory holding costs associated with the jobs. The objective in [32] is to minimize the sum of the costs associated with the in-process inventory, with the earliness/tardiness, and with orders not fully completed at the end of the scheduling horizon.

Eight out of the first ten recent references proposing metaheuristics for the FJSP listed in Section 2.2.2 consider the minimization of the makespan, and the remaining two ([22, 65]) consider a multiobjective approach for minimizing the makespan, the total workload, and the critical machine workload. In the FJSP instances solved in [22] and [65], the processing times are resource-dependent, and therefore the total workload depends on to which resource each operation is assigned.

Another objective is the minimization of the *weighted quadratic tardiness*, which is employed in [79] for the FJSP. The benefit of using the square of the tardiness is that the penalties for the jobs that are the most delayed then become much larger than the penalties for the jobs that are only a little delayed.

Within the modelling framework of MILP, any linear objective that can be formulated using the variables defined may be employed. In [86] (Paper I), we noted, however, a large difference in the performance of the scheduling methods depending on which objective that was considered. It is therefore of great importance that the evaluation of scheduling models are made with respect to objective functions that are well suited for the real applications for which they are intended. In ibid., we argue that the minimization of the makespan is an unsuitable objective when considering a dynamic environment, in which there is a need to repeatedly reschedule the job shop, which is the case in most industrial applications. When employing the makespan objective, some jobs even risk never being processed, since a job may repeatedly be scheduled as the last job in a set of subsequent schedules, if the rescheduling is done before the last job in the previous schedule has started processing. Another inconvenience of the minimization of the makespan is revealed if not all of the jobs are available from the start; then, if the makespan is due to a very late release date of one job, there will be a large set of optimal solutions, since there are many ways of scheduling the remaining jobs without any impact on the objective value. Probably, some of these optimal solutions will be regarded as better than others by practitioners. Hence the makespan objective is indeed not a suitable objective for most real applications.

In most of our work, for example [86, 87, 88] (Papers I–III), we propose an objective to minimize the weighted sum of the completion times and the total weighted tardiness, which works well in a dynamic environment. The main objective is the minimization of the total weighted tardiness. The term of the weighted sum of the completion times is included such that the set of optimal solutions is decreased, and serves also as main objective when all jobs can be processed in due time. In [87] (Paper II), we propose a means to compute appropriate weights for the tardiness terms as functions of the due dates of the jobs. With this definition, a delayed job is assigned a higher weight than a job that is less delayed, such that the jobs that are the most delayed will preferably be scheduled as early as possible.

Few definitions of tardiness weights are more advanced than a constant function or a randomly selected value from a predetermined interval, as done in [9, 62]. In [50], the values 4, 2 and 1 are assigned to the tardiness weights of the jobs whose associated customers are considered to be of high, average, or low importance, respectively.

2.4 Dynamic scheduling

In most real manufacturing systems, the conditions in the workshops are unceasingly changing, with, for example, new jobs continuously arriving and unexpected machine breakdowns. The problem of scheduling in the presence of real-time events is called *dynamic scheduling* ([57]). The unexpected real-time events are classified into three categories ([73]), according to

- *resource-related*: e.g., machine breakdowns, operator illness, material deficiencies, and unavailability of tools,
- *job-related*: e.g., job cancellation, rush jobs, and early or late arrival of jobs, and
- *disturbances related to the measurement of data*: e.g., differences between actual and pre-calculated processing time, and capacity efficiency.

Examples of expected events are new jobs arriving, preventive maintenance of the resources, and holiday periods. Most of these expected and unexpected events are inevitable, and therefore a rescheduling policy is needed. In the survey [57] of dynamic scheduling, three dynamic scheduling policies are described: *periodic*, *event-driven*, and *hybrid*. The periodic and hybrid policies are also known as *rolling time horizon* policies. In the periodic policy, schedules are generated at fixed, regular intervals, such that the dynamic scheduling problem is decomposed into a series of static problems. In the event-driven policy, a rescheduling is triggered as a response to each unexpected event. In a hybrid policy, rescheduling is periodically performed with respect to a rolling time horizon, but whenever there is an urgent event (e.g., a machine break-down), a rescheduling is immediately performed. The rescheduling policy that we propose for the scheduling of the multitask cell is the hybrid policy; see [87, 88] (Paper II–III).

Apart from our work on dynamic scheduling for the FJSP, the only work found in the literature concerning this problem is Fattahi and Fallahi [25], in which the problem is named DFJSP. The objective employed in ibid. is the weighted sum of the makespan and a so-called ”stability factor” originally proposed in [67]. The first additive term of the stability factor is the total deviation of the jobs’ starting times in a new schedule from those corresponding to the previous schedule. The second (and last) term of this factor is a penalty associated with the total deviation of the jobs’ starting times from the current time. In [25] a mathematical model is presented and the optimal objective values including the stability factor obtained when solving the mathematical model are compared with the those obtained by a proposed genetic algorithm.

2.5 Flexible job shop scheduling with side constraints

In order to reflect the reality of the multitask cell, some side constraints that are not present in the standard version of the FJSP may need to be included. As such side constraints usually make the scheduling problem even more complex, we have included them one at a time, as our scheduling procedure has evolved. The side constraints included in the scheduling of the multitask cell so far regard (with our earliest reference within parentheses):

a) the machine relase dates ([84]),

b) precedence relations between jobs—with a time lag ([84]),

c) a limited fixture availability ([85]),

d) the possibility to schedule operations with unmanned processing during night shifts ([88], Paper III), and

e) the scheduling of preventive maintenance (PM) activities simultaneously with the processing jobs ([81]).

The inclusion of machine release dates (Item a) are due to the fact that the ongoing production in the multitask cell occupies some of the resources at the start of the schedule.

Regarding Item b, the literature regards typically precedence constraints between operation, possibly with time lags, and not between jobs as our modelling of the multitask cell. In [23], a parallel genetic algorithm is presented for an FJSP incorporating sequence-dependent setup times, attached or detached setup times, machine release dates, and time lag requirements between operations. In ibid., a Wagner MILP model is also presented, though not solved. The PhD thesis [99] concerns scheduling with time lags between operations. If a number of jobs having precedence relations are considered as one longer job, the time lags are between operations. However, our early decomposition approach of the problem of scheduling the multitask cell (see Section 3.3) made such a union of jobs impossible, and later, our constraints for limited fixture availability (Item c) are constructed such that one job occupies a fixture throughout all its operations, and hence the precedence constraints between jobs are convenient.

To our knowledge, the only published work concerning an FJSP including a limited fixture availability are [64] and [75]; both describe simulation-based scheduling approaches; the latter studies the multitask cell at GKN Aerospace and includes also unmanned night shifts (Item d).

We are the first to propose a mathematical model for an FJSP including the opportunity to schedule unmanned processes during night shifts ([88], Paper III). A similar problem arises in the scheduling of personnel with varying skills working in shifts; see [91] for a recent survey over this field of research.

The constraints required for the scheduling of PM activities (Item e) are often called *availability constraints* in the literature. In [28], the availability constraints are categorized into two types: fixed and non-fixed. Fixed availability constraints correspond to scheduling the maintenance activities beforehand and blocking the respective resources during the corresponding time window; see [53]. In ibid., two objectives are simultaneously considered: the minimization of the makespan for the production part and the minimization of the system unavailability for the maintenance part. For our problem of scheduling the multitask cell, non-fixed availability constraints are considered, i.e., the starting time of the unavailable period is flexible within a time window and is determined during the scheduling procedure ([81], [87] (Paper II), and [88] (Paper III)). Other work considering the FJSP with non-fixed availability constraints are [28, 66, 94]; three critera are employed in each of the three papers: the makespan, the total workload, and the critical machine workload, and a hybrid genetic algorithm, a filtered beam search algorithm, and a GRASP algorithm, respectively, is proposed.

2.6 Computational complexity

Scheduling problems such as flow shop, job shop, and open shop problems have been known to be NP-complete since the mid seventies ([30]). An NP-complete or NP-hard problem is such that no algorithm exists (unless P=NP) that in polynomial time is able to solve all possible instances of the problem ([29]). Hence, the solution time risks to increase exponentially with the number of jobs. The difference between the notions NP-hard and NP-complete is that NP-hard problems are at least as hard as NP-complete problems; see [29, p. 109].

2.6.1 On the complexity of the FJSP

Let us consider the problem of scheduling a single machine with the objective to minimize the weighted tardiness. Using the $\alpha|\beta|\gamma$-notation introduced in [34], this problem can then be described as $1||\sum w_jT_j$. The first field, α, specifies the machine environment, where *F*, *J*, and *O* denote flow shop, job shop, and open shop, respectively. These notations can be combined with, e.g., *P* and *MPM*, which denote identical parallell machines, and multipurpose machines, respectively, and a number k, representing the number of machines. The β-field is used to describe the job characteristics. There can be at most six elements in this field, for example *prec*, r_j, or d_j for precedence constraints, release dates, and due dates, respectively, to name a few elements relevant to the problems studied in this thesis. The last field in this notation is the γ-field which is used to describe the optimality criterion. The most common objective functions are the makespan ($\gamma = C_{\max}$), the total completion times ($\gamma = \sum C_j$), the weighted completion times ($\gamma = \sum w_jC_j$), and the weighted tardiness ($\gamma = \sum w_jT_j$). Note that the summation indices are left out. For further explanation of the $\alpha|\beta|\gamma$-notation; see, e.g., [16]. The tardiness is defined as $T_j := \max\{0, C_j - d_j\}$, where C_j denotes the completion time, and d_j the due date of job j. The problem $1||\sum w_jT_j$ is NP-complete as it can be reduced to a knapsack problem ([46]). Further, the problem $1||\sum w_jT_j$ as well as $1|r_j|\sum w_jC_j$ are NP-hard in the strong sense ([16, Chapter 4]).

The single machine scheduling problem is a special case of the flow shop scheduling problem, which is, in turn, a special case of the JSP, since the ordering of the operations is the same for all the jobs in a flow shop problem. Finally, the JSP is a special case of the FJSP, as the set of available machines for each job in the FJSP is reduced to one machine in the JSP.

The problem of scheduling the multitask cell is, using the $\alpha|\beta|\gamma$-notation described as *J MPM*$10|$*prec*$; r_j; d_j|\sum(w_jC_j + w_j^{'}T_j)$. This is, however, not a description of the complete problem, since none of the side constraints mentioned in Section 2.5 are included in this notation. Since both the single machine problems $1||\sum w_jT_j$ and $1|r_j|\sum w_jC_j$ are special cases of the scheduling problem studied in this thesis, we can conclude that the problem of scheduling the multitask cell is NP-hard. In [17], it is shown that the FJSP (there denoted the *multipurpose machine job shop problem*) with the objective of minimizing the makespan is NP-hard for instances with at least three jobs and at least two machines.

2.6.2 On the complexity of flow shop problems with deteriorating jobs

In [30], the flow shop problem with the objective of minimizing the makespan was proven to be NP-complete for instances with three or more machines. It was also shown that the flow shop problem with the objective of minimizing the mean flow time, i.e., the sum of the jobs' completion times, and the job shop problem with the makespan criterion, are NP-complete for instances with two or more machines.

In [83] (Paper IV), we present a proof of NP-hardness for flow shop problems for which the processing time of each job equals a deterioration rate times the job's starting time. It is a note on an article by Mosheiov in 2002 [54], in which the proof regarding the complexity of a flow shop problem with deteriorating jobs is incorrect. We provide a correct proof of the statement that this problem is NP-hard for instances with three or more machines, when the objective is to minimize the makespan.

During the work with this note, it came to our notice that a correct proof, of an even stronger result, was published already in 1996 by Kononov, however, in Russian; see [44]. This proof is summarized in [83] (Paper IV). We noted, however, while tracing the citation history of the respective proofs by Mosheiov and Kononov, that the incorrect proof by Mosheiov [54, Theorem 2] was by far the one most often cited. Furthermore, it is questioned in [70] whether or not the result by Kononov is stronger than the one published in [83] (Paper IV); see [70] in which it is stated that the complexity status of the *4P product problem*, which is used in the proof by Kononov, is still an open issue.

3 The development of the mathematical models: A work diary

The problem of scheduling the multitask cell is described in Section 3.1. The notation used in the problem formulations is summarized in Table 1. At the beginning of this project, we formulated a Manne model of the FJSP defined by the scheduling of the multitask production cell, without any knowledge of the original Manne model from 1960 ([49]). Therefore, we denoted this model ”the full engineer's model” since the variable definitions are somewhat intuitive and the fact that we modeled the whole (that is, full) problem. This model is presented in Section 3.2.

Since the computation times for the full engineer's model were too long for practical utilization, the scheduling problem was decomposed into two subproblems; this decomposition is described in Section 3.3, where a time-indexed model is formulated for the problem of scheduling the multipurpose machines. In Section 3.4, we discuss the choice of the time horizon and the lengths of the time steps used in the time-indexed model. The inclusion of side constraints and their impact on the computation times are discussed in Section 3.5. A discussion on objective weights and schedule robustness is presented in Section 3.6, and finally, in Section 3.7, some additional computational results are presented.

Table 1: Nomenclature

Sets	*Descriptions*
$\mathcal{N}_j$	set of operations of job j; $i \in \mathcal{N}_j$
$\mathcal{J}$	set of jobs; $j \in \mathcal{J}$
$\mathcal{J}^{\text{maint}}$	set of preventive maintenance tasks; $j \in \mathcal{J}^{\text{maint}}$
$\mathcal{Q}$	set of pairs of jobs with precedence constraints; $(j, q) \in \mathcal{Q}$
$\mathcal{K}$	set of machines; $k \in \mathcal{K}$
$\mathcal{K}^{\text{MPM}}$	set of multipurpose machines; $\mathcal{K}^{\text{MPM}} \subset \mathcal{K}$
$\mathcal{K}^{\text{setup}}$	set of set-up stations; $k \in \mathcal{K}^{\text{setup}}$
$\mathcal{M}_{ij}$	set of machines allowed to process operation i of job j
$\mathcal{T}$	set of time steps; $u \in \mathcal{T}$
$\mathcal{F}$	set of fixture types; $f \in \mathcal{F}$
$\mathcal{S}_f$	set of jobs that use fixture of type f; $j \in \mathcal{S}_f \subset \mathcal{J}$
$\mathcal{P}$	set of night shifts; $n \in \mathcal{P}$

Parameters	*Descriptions*
n_j	total number of operations of job j, ($n_j = \lvert\mathcal{N}_j\rvert$)
λ_{ijk}	$= 1$, if operation i of job j can be processed on resource k $= 0$, otherwise
p_{ij}	processing time of operation i of job j
p_j^{pm}	the sum of the processing time of the post-machining operations, internal transportations times included
$\tilde{d}_j$	real original due date (time) of job j
d_j	due date of job j expressed in time steps for the time-indexed model
$\tilde{a}_k$	real time when resource k becomes available
a_k	time step when resource k becomes available
r_j	release date (time) of job j
r_j^{MPM}	release date of job j in the machining problem
w	time to transport a product between any resources inside the multitask cell
v_{jq}	the time lag that needs to elapse between the completion of job j and the start of job q, where $(j, q) \in \mathcal{Q}$
v_{jq}^{pm}	time lag for the machining problem
ℓ	length of the time steps
T	time horizon
M	a sufficiently large positive number ($M > T\ell$)
α_j (β_j)	objective weights for the completion time (tardiness) of job j

3.1 The problem description

The problem of scheduling the multitask cell is also described in detail in [87] (Paper II) and in [80, Chapter 5]. See Section 2.1 for a description on the products, the machines, and the current production planning of the multitask cell. The conditions for scheduling the multitask cell without the side constraints regarding PM activities, fixture availability, and unmanned night shifts, are summarized as follows:

1. Job j has n_j operations that must be processed according to a predefined sequence.
2. The operations $i \in \mathcal{N}_j = \{1, \ldots, n_j\}$ of each job $j \in \mathcal{J}$ are non-preemptive, i.e., the operations have to be completed once started processing.
3. Only when $\lambda_{ijk} = 1$, operation i of job j is allowed to be processed in resource k (i.e., when $k \in \mathcal{M}_{ij} \subseteq \mathcal{K}$).
4. Resource $k \in \mathcal{K}$ is available for processing at time $\tilde{a}_k$ and onwards in time.
5. A job can not start processing before its release date r_j.
6. Some jobs $(j, q) \in \mathcal{Q} \subset \mathcal{J} \times \mathcal{J}$ are subject to precedence constraints with a time lag v_{jq}, i.e., job q may not start before v_{jq} time units after job j is finished.

The parts that are ready to be processed from start are those that are checked-in but not yet mounted into a fixture; the release dates r_j of the corresponding jobs are set to zero. The release date of each remaining job to be scheduled corresponds to the time of the expected arrival of the corresponding part at the multitask cell. In the models presented in this section, the release dates are given for entire jobs only. In the models developed later, i.e., those in [86, 87, 88] (Papers I–III), the release dates for all operations within a job are utilized, since this showed to shorten the computation times. The release date of job j is then assigned to operation 1 of job j, as $r_{1j} := r_j$, and, accordingly, $r_{ij} := r_{i-1,j} + p_{i-1,j}$, $i = 2, \ldots, n_j$, since this is the earliest possible starting time of the operation. In Papers I–III, the time w for transporting any product between the resources inside the multitask cell is neglected in this formula, since w is only a few minutes long and hence small in comparison with most processing times, which range from 0.5 hours to approximately 23 hours. Similarly, a parameter δ_{ij} representing the total remaining processing time of job j at the start of operation i is introduced in the models presented in Papers I–III. Then, $\delta_{n_j j} := p_{n_j j} + v_{jq} + \sum_{r \in \mathcal{N}_q} p_{rq}$ for $(j, q) \in \mathcal{Q}$ (see Item 6 above) and $\delta_{n_j j} := p_{n_j j}$ if no job precedes job j on the same component, and $\delta_{ij} := \delta_{i+1,j} + p_{ij}$, $i = n_j - 1, \ldots, 1$.

The precedence relations between jobs with a time lag in Item 6 above arise due to the fact that the products typically visit the multitask cell multiple times on their way to completion. In Figure 3, the processing performed outside of the multitask cell (illustrated by the striped boxes) corresponds to the time lag v_{jq}. For example, after the completion of processing a job j in the multitask cell, the corresponding part may have to be welded and heat treated during v_{jq} time units before returning to the multitask cell for processing job q.

3.2 The full engineer's model

The full engineer's model is published in [80, model (4) in Chapter 5]. In order streamline the notation in this thesis with that in [86, 87, 88] (Papers I–III), it differs slightly from the notation used in [80]. The variables used in our formulation of the full engineer's model are thus:

$$\begin{aligned}
t_{ij} &:= \text{the starting time of operation } i \text{ of job } j,\\
C_j &:= \text{the completion time of job } j\ (C_j = t_{n_j,j} + p_{n_j,j}),\\
T_j &:= \text{the tardiness of job } j\ (T_j = \max\{0; C_j - \tilde{d}_j\}),\\
z_{ijk} &:= \begin{cases} 1, \text{ if operation } i \text{ of job } j \text{ is allocated to resource } k,\\ 0, \text{ otherwise,}\end{cases}\\
y_{iji'j'k} &:= \begin{cases} 1, \text{ if operation } i \text{ of job } j \text{ is processed before operation } i^{'} \text{ of job } j^{'} \text{ on resource } k,\\ 0, \text{ otherwise,}\end{cases}
\end{aligned}$$

for $i \in \mathcal{N}_j,\ i^{'} \in \mathcal{N}_{j^{'}},\ j, j^{'} \in \mathcal{J}$, such that $(i, j) \neq (i^{'}, j^{'})$, and $k \in \mathcal{K}$. Given the parameters and sets defined in Table 1, the full engineer's model for the problem to schedule the multitask cell is formulated as that to

$$\begin{aligned}
&\text{minimize} && \sum_{j\in\mathcal{J}} \left(\alpha_j C_j + \beta_j T_j - \varepsilon t_{1j}\right), && && \text{(2a)}\\
&\text{subject to} && \sum_{k\in\mathcal{K}} z_{ijk} = 1, && i \in \mathcal{N}_j,\ j \in \mathcal{J}, && \text{(2b)}\\
& && z_{ijk} \leq \lambda_{ijk}, && i \in \mathcal{N}_j,\ j \in \mathcal{J},\ k \in \mathcal{K}, && \text{(2c)}\\
& && y_{iji'j'k} + y_{i'j'ijk} \leq z_{ijk}, && i \in \mathcal{N}_j,\ i^{'} \in \mathcal{N}_{j^{'}},\ j, j^{'} \in \mathcal{J}, && \text{(2d)}\\
& && && (i, j) \neq (i^{'}, j^{'}),\ k \in \mathcal{K}, &&\\
& && y_{iji'j'k} + y_{i'j'ijk} + 1 \geq z_{ijk} + z_{i^{'}j^{'}k}, && i \in \mathcal{N}_j,\ i^{'} \in \mathcal{N}_{j^{'}},\ j, j^{'} \in \mathcal{J}, && \text{(2e)}\\
& && && (i, j) \neq (i^{'}, j^{'}),\ k \in \mathcal{K}, &&\\
& && t_{ij} + p_{ij} - M(1 - y_{iji'j'k}) \leq t_{i'j'}, && i \in \mathcal{N}_j,\ i^{'} \in \mathcal{N}_{j^{'}},\ j, j^{'} \in \mathcal{J}, && \text{(2f)}\\
& && && (i, j) \neq (i^{'}, j^{'}),\ k \in \mathcal{K}, &&\\
& && t_{ij} + p_{ij} + w \leq t_{i+1,j}, && i \in \mathcal{N}_j \setminus \{n_j\},\ j \in \mathcal{J}, && \text{(2g)}\\
& && t_{1j} \geq r_j, && j \in \mathcal{J}, && \text{(2h)}\\
& && t_{ij} \geq \tilde{a}_k z_{ijk}, && j \in \mathcal{J},\ k \in \mathcal{K}, && \text{(2i)}\\
& && t_{1q} \geq C_j + v_{jq}, && (j, q) \in \mathcal{Q}, && \text{(2j)}\\
& && C_j - t_{n_j j} = p_{n_j j}, && j \in \mathcal{J}, && \text{(2k)}\\
& && T_j \geq C_j - \tilde{d}_j, && j \in \mathcal{J}, && \text{(2l)}\\
& && T_j \geq 0, && j \in \mathcal{J}, && \text{(2m)}
\end{aligned}$$

$$t_{ij} \geq 0, \qquad i \in \mathcal{N}_j,\ j \in \mathcal{J}, \tag{2n}$$

$$z_{ijk} \in \{0,1\}, \qquad i \in \mathcal{N}_j,\ j \in \mathcal{J},\ k \in \mathcal{K}, \tag{2o}$$

$$y_{iji'j'k} \in \{0,1\}, \qquad i \in \mathcal{N}_j,\ i^{'} \in \mathcal{N}_{j'},\ j, j^{'} \in \mathcal{J},\ (i,j) \neq (i^{'},j^{'}),\ k \in \mathcal{K}. \tag{2p}$$

The constraints (2b) ensure that each operation is processed exactly once, and the constraints (2c) make sure that each operation is scheduled on an allowed resource. If operation i of job j and operation $i^{'}$ of job $j^{'}$ are to be performed on the same machine, then the constraints (2d) regulate that at most one of the variables $y_{iji'j'k}$ and $y_{i'j'ijk}$ *may* possess the value 1, while the constraints (2e) ensure that at least one of the variables $y_{iji'j'k}$ and $y_{i'j'ijk}$ *must* possess the value 1. Hence, if operation i of job j precedes operation $i^{'}$ of job $j^{'}$ in resource k, then operation $i^{'}$ of job $j^{'}$ cannot precede operation i of job j, and vice versa.

Furthermore, the constraints (2f) make sure that the starting time of operation $i^{'}$ of job $j^{'}$ is scheduled after the completion of the previous operation performed in the same resource. The parameter M is a big number, whose purpose is to relax the constraints whenever $y_{iji'j'k}$ is valued 0, i.e., when the operations are not scheduled for processing in the same resource. In [80, Chapter 6], an analysis on the impact of different values of M on the computation time is given. Although somewhat ambiguous, the results indicates that M should be chosen as small as possible, yet at least as large as the planning horizon.

Generally in scheduling problems, the constraints $t_{i'j'} + p_{i'j'} - My_{iji'j'k} \leq t_{ij}$, being symmetric to the constraints (2f), are required, as e.g., in [58], but these become redundant here since the variables $y_{iji'j'k}$ and $y_{i'j'ijk}$ are regulated by the inequalities (2d) and (2e). However, in [58], Özgüven et al. defined the variables $y_{iji'j'k}$ only for $j < j^{'}$. This restriction in the definition of the variables $y_{iji'j'k}$ is not possible to employ simultaneously with the constraints (2d) and (2e). While developing our so-called *alternative model* in [86] (Paper I), which is a slightly improved version of the model in [58], preliminary tests with benchmark test instances indicated that the computation times needed when using the two symmetric constraints were shorter than when employing the constraints (2d)–(2f).

The constraints (2g) ensure that an operation is not scheduled until the previous operation within the same job is completed and the corresponding part is transported to the current resource. The constraints (2h) regulate the starting time of the first operation of every job, so that no job is scheduled prior to its release date. Moreover, the constraints (2i) make sure that no operation is scheduled in any resource that is not yet available. Any pair (j, q) of two jobs belonging to the set $\mathcal{Q}$ must be separated by at least the planned inter-operation time, v_{jq}; this is indicated by the constraints (2j). The completion time and the tardiness of each job j are defined by the constraints (2k)–(2m). Note that, whenever the release date r_j is nonnegative, the nonnegativity constraints (2n) for the starting times are redundant due to the equations (2g)–(2i). Indeed, r_j is always nonnegative for the case of the multitask cell; see Section 3.7.2, which describes the generation of the test instances.

The computational testing of the full engineer's model was somewhat discouraging: the computation time (clock time) needed to solve a test instance with 15 jobs to optimum was about three months. This test was carried out on a 4Gb quad-core Intel Xeon 3.2 GHz system using AMPL-CPLEX 12.1.0 [27, 37] as optimization software. Now—three years later—and with access to hardware with more memory and more parallel threads, namely two 2.66GHz Intel Xeon X5650 processors, each with six cores (24 threads), and a total memory of 48 Gbytes of RAM, we rerun this test instance: it took as long as ten minutes for CPLEX to find the first feasible solution, and after 15 minutes, the `mipgap`[1] was 10.6%. We stopped the calculations after four hours, when the `mipgap` had only decreased to 7.2%. Obviously, it is still not possible to use this model for practical purposes.

3.3 The problem decomposition and the time-indexed machining model

Since the computation times for the full engineer's model (2) were too long for practical utilization, the scheduling problem was decomposed into two subproblems. The first subproblem, henceforth called the *machining problem*, is to find an optimal sequence of operations for each of the five multipurpose machines, i.e., for all $k \in \mathcal{K}^{\mathrm{MPM}}$. The main reason for this choice of decomposition was of a practical nature: that the work load on these resources is much higher than that on the others. Another reason was that, at the time, the second operation of each job was always performed in a multipurpose machine, and that there was only one visit to a multipurpose machine per job. Hence all the precedence constraints between jobs remained the same, and the whole problem could be reduced to the machining problem by simply leaving out the index for the operations (i) since $\mathcal{N}_j = \{2\},\ j \in \mathcal{J}$, in the machining problem. The second subproblem, henceforth called the *feasibility problem*, is to generate a feasible schedule for all ten resources, with an optimal (from the solution to the machining problem) sequence for each of the five multipurpose machines as input data.

In [80], we developed and compared, for the machining problem, a Manne model, a Bowman model, and a time-indexed model employing nail variables. The feasibility problem was then solved by a Manne model (almost identical to the full engineer's model). Since the sequence of jobs on the multipurpose machines was provided by the solution to the machining problem, the computation time needed to solve the feasibility problem was not an issue—at least in the beginning. The time-indexed model outperformed all the other models developed and tested regarding the computation time required to solve the machining instances and also regarding which instance sizes they were able to solve within a reasonable computation time.

Since the Manne model for the machining problem is similar to the full engineer's model (2), and the corresponding Bowman model turned out to perform as well (or bad) as the Manne model, we formulate here only the time-indexed model with nail variables.

The planning horizon for the schedule is divided into $T + 1$ time steps, each of length ℓ hours; see Figure 4. We let $\mathcal{T} := \{0, 1, \ldots, T\}$ denote the set of time steps. The value of the parameter T has to be large enough such that the planning horizon of $(T+1)\ell$ hours contains

[1] The `mipgap` is defined as the relative difference between the best lower bound LB and the best objective value z found. The definition used by CPLEX for version 12 is: `mipgap` $:= \frac{|z-\mathrm{LB}|}{10^{-10}+|z|} \cdot 100\%$ (for version 10.1.0, the denominator was defined as 1 + |LB|)

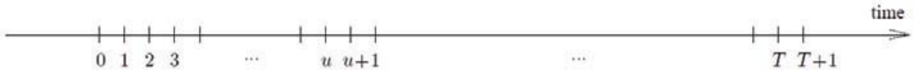

Figure 4: The planning horizon divided into $T+1$ time steps of length ℓ hours.

an optimal schedule. Later, when the models presented in [86, 87, 88] (Papers I–III) were developed, we redefined the end of the planning horizon, such that the horizon ends at time $T\ell$ hours. Here, in this work diary, we would like to present the time-indexed model for the machining model as originally defined in [80], and therefore we use the former definition here.

The variables needed to formulate the time-indexed model with nail variables for the machining problem are, besides C_j and T_j for completion times and tardiness, defined in Section 3.2,

$$x_{jku} = \begin{cases} 1, & \text{if the processing of operation 2 of job } j \text{ starts at} \\ & \text{resource } k \text{ at the beginning of time interval } u, \\ 0, & \text{otherwise,} \end{cases} \qquad j \in \mathcal{J},\ k \in \mathcal{K}^{\text{MPM}},\ u \in \mathcal{T}.$$

In the time-indexed model, all time-related parameters have to be expressed in integer multiples of time steps. In order to maintain the feasibility of every solution to the resulting model, this means that the real original values of the processing times, release dates, machine availability, and the time lag between jobs must be scaled and rounded up, while the due dates have to be scaled and truncated.

Since the machining problem includes the problem of finding an optimal sequence of jobs on each of the five multipurpose machines, the definitions of some parameters also need to be altered slightly. For example, the release date r_j^{MPM} for job j in the machining problem is composed by the actual release date r_j, the processing time p_{1j} of the first operation (mounting the part into a fixture), and the internal transportation time w, before it is scaled and rounded up, i.e.,

$$r_j^{\text{MPM}} := \left\lceil \frac{1}{\ell} \left(r_j + p_{1j} + w \right) \right\rceil, \qquad j \in \mathcal{J}. \tag{3a}$$

The definition of a job's completion time should reflect the completion time of the last operation of the job. Therefore, a new parameter representing the processing of the post-machining operations and the corresponding internal transports is needed in the formulation:

$$p_j^{\text{pm}} := \left\lceil \frac{1}{\ell} \left(\sum_{i=2}^{n_j} p_{ij} + (n_j - 2)w \right) \right\rceil, \qquad j \in \mathcal{J}. \tag{3b}$$

The time lag is redefined as the time required between the start of the machining operation of job j and the start of the machining operation of job q, according to

$$v_{jq}^{\mathrm{pm}} := \left\lceil \frac{1}{\ell}(p_{2j} + p_j^{\mathrm{pm}} + v_{jq} + p_{1q} + w) \right\rceil, \qquad (j,q) \in \mathcal{Q}. \tag{3c}$$

The remaining parameters are then redefined according to

$$\lambda_{jk} := \lambda_{2jk}; \qquad a_k := \left\lceil \frac{\tilde{a}_k}{\ell} \right\rceil, \qquad j \in \mathcal{J}, k \in \mathcal{K}^{\mathrm{MPM}}, \tag{3d}$$

$$p_j^{\mathrm{MPM}} := \left\lceil \frac{p_2 j}{\ell} \right\rceil; \quad d_j := \left\lfloor \frac{\tilde{d}_j}{\ell} \right\rfloor, \qquad j \in \mathcal{J}. \tag{3e}$$

The time-indexed model for the machining problem employing nail variables was developed using findings presented in [96]. It is formulated as that to

$$\text{minimize} \quad \sum_{j \in \mathcal{J}} (\alpha_j C_j + \beta_j T_j), \tag{4a}$$

$$\text{subject to} \quad \sum_{k \in \mathcal{K}^{\mathrm{MPM}}} \sum_{u \in \mathcal{T}} x_{jku} = 1, \qquad j \in \mathcal{J}, \tag{4b}$$

$$\sum_{u \in \mathcal{T}} x_{jku} \le \lambda_{jk}, \qquad j \in \mathcal{J},\ k \in \mathcal{K}^{\mathrm{MPM}}, \tag{4c}$$

$$\sum_{j \in \mathcal{J}} \sum_{\nu = (u - p_j^{\mathrm{MPM}} + 1)_+}^{u} x_{jk\nu} \le 1, \qquad k \in \mathcal{K}^{\mathrm{MPM}},\ u \in \mathcal{T}, \tag{4d}$$

$$\sum_{k \in \mathcal{K}^{\mathrm{MPM}}} \left(\sum_{\mu=0}^{u} x_{jk\mu} - \sum_{\nu=0}^{u+v_{jq}^{\mathrm{pm}}} x_{qk\nu} \right) \ge 0, \qquad u = 0, \ldots, T - v_{jq}^{\mathrm{pm}},\ (j,q) \in \mathcal{Q}, \tag{4e}$$

$$x_{jku} = 0, \qquad u = T - v_{jq}^{\mathrm{pm}}, \ldots, T,\ (j,q) \in \mathcal{Q},\ k \in \mathcal{K}^{\mathrm{MPM}}, \tag{4f}$$

$$\sum_{k \in \mathcal{K}^{\mathrm{MPM}}} \sum_{u \in \mathcal{T}} u x_{jku} + p_j^{\mathrm{pm}} = C_j, \qquad j \in \mathcal{J}, \tag{4g}$$

$$C_j - T_j \le d_j, \qquad j \in \mathcal{J}, \tag{4h}$$

$$T_j \ge 0, \qquad j \in \mathcal{J}, \tag{4i}$$

$$x_{jku} = 0, \qquad u = 0, \ldots, \max\{r_j^{\mathrm{MPM}}; a_k\},\ j \in \mathcal{J},\ k \in \mathcal{K}^{\mathrm{MPM}}, \tag{4j}$$

$$x_{jku} \in \{0,1\}, \quad j \in \mathcal{J},\ k \in \mathcal{K}^{\mathrm{MPM}},\ u \in \mathcal{T}, \tag{4k}$$

where $(s)_+ := \max\{s, 0\}$, $s \in \mathbb{R}$. The constraints (4b) ensure that every machining operation (i.e., every job in this model) is processed exactly once, and the constraints (4c) make sure that each machining operation is scheduled in an allowed resource. The constraints

(4d) regulate that at most one operation at a time is scheduled in each resource. The constraints (4e)–(4f) ensure that the machining operation of job j is scheduled to start at least the planned lead time of v_{jq}^{pm} time steps before the start of the machining operation of job q, whenever $(j, q) \in \mathcal{Q}$. The constraints (4g)–(4i) define the completion times, C_j and the tardiness, T_j (the function of these constraints rely on positive objective weights, i.e., $\alpha_j > 0$ and $\beta_j > 0$). The constraints (4j) make sure that no machining operation is scheduled on resource k before its release date or before the resource is available. During the development of the time-indexed model presented in [86] (Paper I), we discovered an error in the constraints (4j): the range for u should be $u = 0, \dots, \max\{r_j^{\text{MPM}} - 1; a_k - 1\}$, as an operation may start processing at time r_j^{MPM} or a_k if $a_k > r_j^{\text{MPM}}$. This was discovered as we had trouble feeding the model with a starting solution. Finally, the constraints (4k) impose binary restrictions on the variables.

3.4 On the choice of the time horizon and the length of the time steps

As both the number of variables and constraints in a time-indexed model depend on the total number of time steps, the choices of values for T and ℓ are of great importance. The best values of T and ℓ to use depend on the instance to be solved. The value of T should be as small as possible, but large enough such that the time interval defined by the set $\{0, \dots, T\}$ contains an optimal schedule. Since the makespan of the optimal schedule of a problem instance is not known beforehand, a suitable value of T is not easy to determine. In fact, even if this value is known, we could not use it without feeding the solver with a starting solution, since the time required for the optimization software to find the first feasible solution then risks being too long for practical purposes.

The length of the time step ℓ is equally important as the value of T. The solutions found by the time-indexed model are approximate as all data are either rounded up or truncated to the nearest integer multiple of ℓ, as in (3). The value of ℓ should be as small as possible—the best would of course be if no parameters were rounded—but large enough such that the computation time needed to solve the model is within a time frame that is acceptable for the practical purpose.

For the comparison of the three variants of the machining model presented in [80], a greedy heuristic was constructed in order to find a good feasible schedule as well as a suitable value of T. The machining model was then solved directly with one call of the optimization solver. In order to decrease the gap between the optimal objective value sought and the approximate objective value obtained by the time-indexed model,—the approximation level being due to the chosen value of ℓ—a *squeezing procedure* for postprocessing the solutions using original, non-approximated data was developed. This procedure was then used in [86] (Paper I) and [88] (Paper III). The squeezing procedure retains the ordering of the operations on each resource, but the starting time of each operation is recalculated such that it is scheduled as early as possible, however, without violating any constraints, such as precedence or release date constraints.

The effect of using the squeezing procedure is substantial, especially for large values of ℓ. In [80, Section 6], the squeezed schedules from the time-indexed model (4) for the machining problem with $\ell \in \{0.25\text{h}, 0.5\text{h}, 1\text{h}, 2\text{h}\}$ were compared for a set of test instances of various sizes. Only the results from the model with $\ell = 2\text{h}$ differed significantly from the others, for which the mean relative difference was less than 0.04% over the instances solved; see Figure 5.

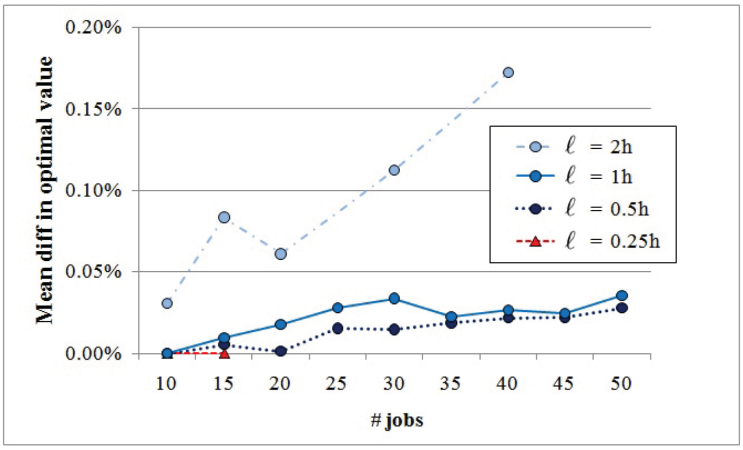

Figure 5: The mean differences between the relative objective values of the squeezed schedules obtained from the optimal solutions found using the time-indexed model (4) for $\ell \in \{0.25\text{h}, 0.5\text{h}, 1\text{h}, 2\text{h}\}$. The model (4) with $\ell = 0.25\text{h}$ is compared with the objective values obtained with the Manne model for the machining problem, while the model (4) with $\ell > 0.25\text{h}$ is compared with the same model with $\ell = 0.25\text{h}$.

During the work with [88] (Paper III) we calculated the relative difference between the objective value of the squeezed and the non-squeezed solutions for some instances. Since the model in Paper III includes the side constraints for the possibility to perform unmanned processing during night shifts, the gain of employing the squeezing procedure is even larger than for the case when these side constraints are excluded. For a set of twelve instances with $\ell = 4\text{h}$, the average relative difference between the objective value of the squeezed and the non-squeezed solution was 45%, while for a set of nine instances with $\ell = 0.5\text{h}$, the average relative difference was only 3.6%.

Since the results for the machining model with nail variables were very engouraging (see Figure 8 in Section 3.7) we moved on to once again tackle the whole problem of simultaneously scheduling the ten resources of the multitask cell. As it turned out, it was necessary to abandon our decomposition approach, since the definition of the jobs in the multitask cell changed such that the second operation was no longer always performed in one of the multipurpose machines, and some jobs included two or even three visits to these machines. Moreover, it turned out that it is very complicated to include the side constraints regarding a limited fixture availability (see [85]) in the decomposed model.

As our scope was extended to include more side constraints and all operations of the jobs in the mathematical model, the heuristic developed for determining a good feasible solution and the planning horizon needed to be altered. As we had noticed that the instances often could be solved in very short time for large values of ℓ, we instead developed an iterative procedure, in which the time-indexed model is repeatedly solved for diminishing values of ℓ, i.e., with an increasing accuracy. In each iteration the solution from the previous iteration is used as a starting solution and its makespan is used to determine the value of T for the next iteration. As the iterative procedure turned out to work well we abandoned the heuristic. In [88], in the first iteration, the value $T := \sum_{j \in \mathcal{J}} \sum_{i \in \mathcal{N}_j} p_{ij}$ is chosen along with a large value of ℓ. See [86, 87, 88] (Papers I–III) for further descriptions of the iterative procedure.

3.5 The inclusion of side constraints

Since side constraints usually complicate an optimization problem, we have included them one at a time, as we have become ready for new challenges. In the previous sections, we have discussed the side constraints defined by the machine relase dates [e.g., (4j)] and precedence constraints between jobs with a required time lag [e.g., (4e)–(4f)].

3.5.1 Limited fixture availability

Figure 6 shows a part mounted into a fixture at one of the three set-up stations of the multitask cell. Each part is mounted into a fixture through all the operations of a job, and since the number of each fixture type is limited, these side constraints are necessary for a schedule to be practically applicable in the multitask cell. When first including the side constraints defined by the limited fixture availability in the time-indexed model of the machining problem ([85]), the computation times became on average 23 times longer when the model was solved directly. For the full scheduling problem, and with three iterations of the iterative procedure, the difference in computation times decreased such that the times required to find an optimal solution became on average nine times longer when the limited fixture availability was considered.

3.5.2 Preventive maintenance activities

If, in addition to the processing jobs, also the preventive maintenance (PM) tasks are optimally scheduled, the utilization of the multitask cell will be increased. Therefore, we included the scheduling of PM activities simultaneously with the processing jobs in [81]. In [87] (Paper II) we suggest an alternative formulation of the PM side constraints, which would allow for solutions in which the PM activities are scheduled in-between the time windows currently prescribed in the multitask cell. This could yield more flexible schedules in favour of the processing jobs, but with a possibly larger number of scheduled PM occasions. The computation times became on average 1.8 times longer when the scheduling of PM activities was added to the time-indexed model including fixture availability constraints.

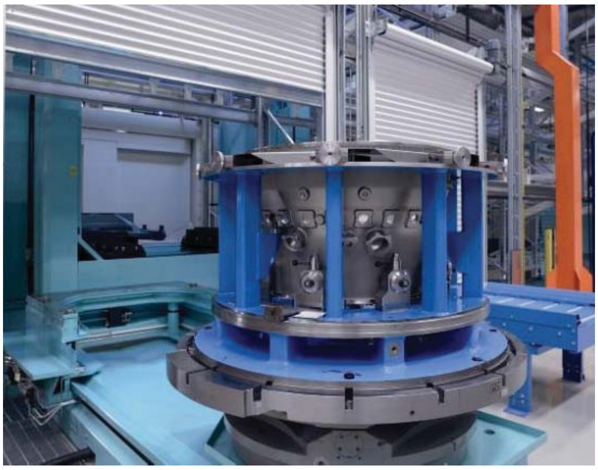

Figure 6: A part mounted into a fixture at a set-up station in the multitask cell.

3.5.3 Unmanned processing during night shifts

In our latest article, [88] (Paper III), side constraints regarding the possibility to perform unmanned processing during night shifts were included in the time-indexed scheduling model. These constraints turned out to be simple, comprising only intermediate deadlines and release dates, which are unique for each operation and depend on the amount of unmanned processing that is allowed at the start and end of the operation. We employed the iterative procedure to find a solution with a `mipgap` of at most 0.5% when $\ell = 1\text{h}$. When the night shift constraints were included in the time-indexed model (already including side constraints regarding fixture availability and PM activities), the computation times became on average 0.78 times those experienced without the night shift constraints. We conclude that it is beneficial for the computation times to include this kind of constraints.

3.6 Objective weights and a discussion on robustness

In our early computational tests, we employed the values $\alpha_j = \beta_j = 1,\ j \in \mathcal{J}$, for the objective weights. When studying some of the resulting schedules we discovered, however, some cases in which jobs with no tardiness, but short processing times, were scheduled in clusters before longer tardy jobs ([80]). Therefore, since minimizing the total tardiness is considered to be our main objective, in [85] we altered the weights such that $\alpha_j = 1$ and $\beta_j = 10,\ j \in \mathcal{J}$.

From time to time, however, some parts are delayed already at their arrival at the multitask cell. This means that the corresponding jobs possess negative due dates, and all of these jobs will possess a positive tardiness $T_j := (C_j - d_j)_+ = C_j + |d_j|$. Since there is no distinction—except for the constant term $|d_j|$—in the objective function between a late job and a *very* late job, we developed a definition of tardiness weights as a non-increasing function of the due date of the respective job. Such differentiated weights were first introduced in [81] and have been used in all testings from that point in time and onwards, specifically in Papers I–III. Letting $B > 0$, the tardiness weights are defined according to

$$\beta_j := B \left(1 - \frac{d_j}{|d_D|} \right)_+ , \tag{5}$$

where D denotes the job having the largest absolute due date and which is not an outlier; an outlier is here defined as ”a point which falls more than 1.5 times the interquartile range above the third quartile or below the first quartile” ([68]). The parameter $B = 10$ in Papers I–III, such that $\beta_j \in [0, 20]$, $j \in \mathcal{J}$ (except for the outliers, for which $\beta_j > 20$ may hold). By utilizing the definition (5), a job that is more delayed is assigned a higher objective weight than a job that is less delayed. This means that there is an increased probability that the jobs that are the most delayed are scheduled as early as possible.

In Section 2.3, we state that, in a dynamic environment, some jobs risk never being processed if the objective of minimizing the makespan is employed, since then nothing prevents a certain job from being scheduled late, in each of a set of subsequent schedules. This may occur if the rescheduling is performed before the last job in the previous schedule has started processing. This risk is considerably reduced if the objective function is changed to our proposed objective function including tardiness (4a) and the differentiated objective weights according to (5) are utilized. By employing our definition (5) of the objective weights, the scheduling becomes more robust than if equal weights are employed for the jobs. Our proposed objective function aims at reducing (and if possible eliminating) the tardiness, i.e., it strives to make the products leave the cell according to the overall production plan. The product flow is thus ”stabilized” through the use of the objective function (4a), which is beneficial for the production planning and control of the factory.

According to Pinedo ([61, Section 18.1]), the *schedule robustness* is not easy to measure or even to define. A robustness measure incorporating the deviation between the completion time of the rescheduled job and its original completion time is, however, defined in ibid. In [25], a robust schedule is obtained through the inclusion of a stability factor in the objective. In fact, the stability factor—which equals the starting time deviation plus a penalty function of the total deviation (see Section 2.4)—may be used as a measure of the schedule robustness. We have not yet measured the robustness of our schedules; this is an interesting area for future research.

The parameter a_k, which denotes the first point in time when resource k becomes available, can be utilized to increase the robustness of the schedules resulting from our scheduling procedure; see [87] (Paper II). This holds if no major unexpected event has occurred since the last scheduling occasion, such as for example a machine breakdown that has made the previous schedule infeasible. At the point in time of rescheduling, the value of a_k can be chosen such that the occupation of the machines for any jobs that are already scheduled (but not yet started) are also included. At these occasions, the planner of the multitask cell can thus control the start of the schedule, and therefore has a means to control the schedule robustness.

One of the reasons why a job will not start at the scheduled starting time is the variability of the processing times, which are typically ”skewed to the right”, since there is a physical lower limit on the length of a specific processing time, while no upper limit exists. In our time-indexed model, all processing times are rounded up, and hence the sequence of operations is near-optimal even for processing times that are slightly longer than the planned deterministic processing times, which we use in the squeezing procedure. Although not considering *any* realization of the uncertainty, as is done in the field of *robust optimization* (see [10] for a survey of this field), the schedules produced by our time-indexed model are in this sence robust.

3.7 Some additional computational results

In this section we start by a discussion on our two alternative ways of modelling allowed resources, then we describe our work with the generation of input data to our models. The section ends with a presentation of some computational results.

3.7.1 Alternative modelling of allowed resources

The full engineer’s model (2) as well as most of our mathematical models developed, use the parameter λ_{ijk} which equals 1 if operation i of job j is allowed to be processed in resource k, and 0 otherwise. We tested an alternative definition in [86] (Paper I), in which the models employ the set $\mathcal{M}_{ij} := \{k \in \mathcal{K} | \lambda_{ijk} = 1\}$. During the computational testing we noted that—before the preprocessing performed by the optimization solver—a model employing the parameters λ_{ijk} contains more constraints than a corresponding one employing the sets $\mathcal{M}_{ij}$, but after the preprocessing, the two models contain the same number of variables and constraints.

3.7.2 Generation of input data

All data used in all our computational tests, except for the benchmark instances used in [86] (Paper I), are generated from data collected at different times from the ERP system at GKN Aerospace. At the time of data extraction, the current operation of each job was also collected, such that the estimated arrival times for the jobs to the multitask cell could be calculated. In order to generate the data files, a database was constructed in Microsoft Office Access 2007 with code in Visual Basic for Application 2007. In the database, realistic release dates were calculated (see [80] for details), and due dates were transformed from the date and time format in the ERP system to the number of hours between the due date and the start of the schedule, i.e., the time of data extraction. The time lag v_{jq} was calculated as the sum of the planned lead times between all operations between jobs j and q plus the sum of their standard processing times.

During our computational testing, we sometimes experienced a test instance being ”not suited” for our scheduling models, in the sense that it included a lot of similar operations that, for example, can only be processed in one of the multipurpose machines, while only a few of the included operations that are allowed to be processed in the other machines. Scenario #11 in [88] (Paper III) is an example of such a case; see Figure 7. Each of the thirteen jobs that are scheduled in the multipurpose machine named 'MC2' in Figure 7 had a release date that was less than 8 hours, such that it should be considered when scheduling the coming shift. However, only the first three of these jobs need to be included in the schedule for the coming shift. The remaining ten jobs made this instance computationally demanding to solve, since the time horizon had to be large in order to include all thirteen jobs. Since an automated selection of the jobs to include in an instance is required, we need a procedure for excluding some of the "superfluous" jobs. One suggestion is to use the ratio $\mathtt{rat}^s := \frac{1}{|\mathcal{M}_s[} \sum_{(j,i)\in\mathcal{S}_s^{\mathrm{op}}} p_{ij}$, where the set $\mathcal{S}_s^{\mathrm{op}}$ contains all operations of type s, and the set $\mathcal{M}_s$ contains the resources that are allowed to process an operation of type s. If, in a problem instance, for some value of s, $\mathtt{rat}^s \gg \max_{r\neq s} \mathtt{rat}^r$, then the jobs possessing the largest due dates and including operations of type s, can be removed from the instance.

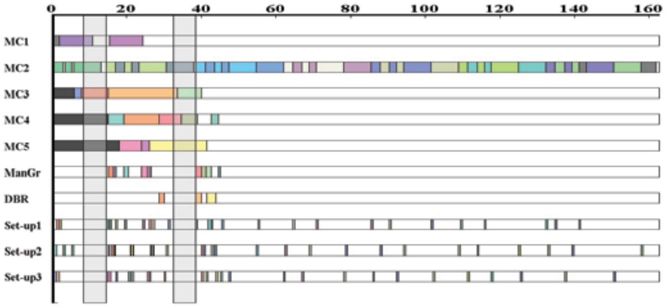

Figure 7: The best schedule found for the coming shift for scenario #11 in [88] (Paper III). The computation stopped after 2h (clocktime), with $\ell = 1\text{h}$. Not all of the thirteen jobs scheduled in machine 'MC2' are needed in the instance since the goal is to find a good schedule for the coming shift.

3.7.3 Comparisons between Manne and time-indexed models

In our first comparison (in [80]) between a Manne model and a time-indexed model with nail variables for the machining problem, the time-indexed model outperformed by far the Manne model; see Figure 8. The Manne model was only able to solve instances with 15 jobs or less within the time limit of 2 hours (clocktime).

When testing two other Manne models for the benchmark instances by Fattahi et al. [26] in order to compare these models with the iterative procedure including a time-indexed model ([86], Paper I), the result obtained was, however, different. When employing the makespan objective, the models performed almost equally well, while the time-indexed model yielded better results than the other two models when employing our tardiness ob-

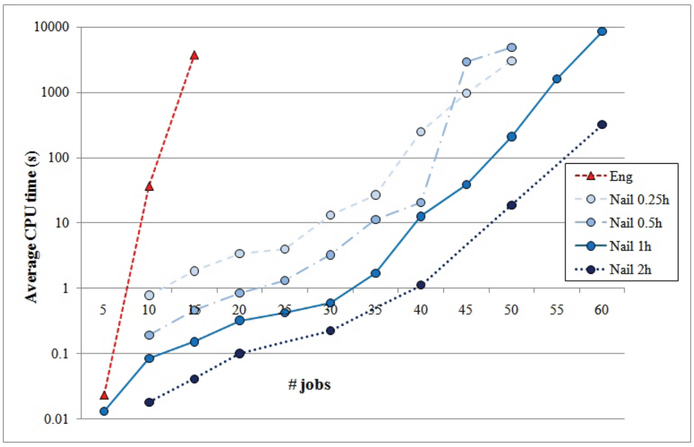

Figure 8: *Mean CPU times plotted for a Manne model (eng) and the time-indexed model* (4) *with nail variables for the machining problem for different values of time steps ℓ. The instances computed are represented by the markers in the graph. The mean values are calculated over six real data instances.*

jective. Below, some important differences between the two tests [of the machining problem in the multitask cell in [80] and for the Fattahi instances in [86] (Paper I)] are discussed:

- Only a few precedence constraints between the operations (i.e., jobs) are present in the machining problem, since it contains only one operation per job. In the Fattahi instances, most of the jobs comprise four operations to be scheduled in sequence.
- The time-indexed model for the machining problem was solved approximately, with $\ell = 1\text{h}$, and the resulting schedule was then squeezed. The highest possible level of accuracy was used for the Fattahi instances with (i.e., with $\ell = 1$), even though good results were achieved also for higher values of ℓ.
- The range of the processing times of $[40, 357]$ of the Fattahi test instances differ a lot in comparison with the range of $[0.5\text{h}, 23\text{h}]$ of the processing times in the multitask cell.
- The processing times of the Fattahi instances are resource-dependent: for a specific operation they differ more than 100% between the two resources. The processing times in the multitask cell are not resource-dependent.
- The symmetric constraints with the parameter M (so-called big M constraints) were employed in the Manne models used to solve the Fattahi instances, while constraints similar to (2d) and (2e) were employed to solve the machining problem (see the discussion in Section 3.2).

The perhaps most striking difference between the two tests, however, is the fact that the multitask cell instances are based on data collected from a real production cell, while the Fattahi instances are randomly generated.

4 Summary of the appended papers

The scopes, results and contributions of the four appended papers are summarized in this chapter. Paper I deals with the general FJSP without side constraints, and the computational tests are performed using the well-known benchmark test instances by Fattahi et al. [26], and Papers II–III deal with the problem of scheduling the multitask cell. Paper IV corrects a proof of the NP-completeness of flow shop scheduling with deteriorating jobs.

4.1 Paper I: A competitive iterative procedure using a time-indexed model for solving flexible job shop scheduling problems

In [86] (Paper I) we focus on the general FJSP with the objectives to minimize the makespan and our proposed tardiness function (4a), respectively.

4.1.1 Scope

We investigate the efficiency of our iterative solution procedure compared with two Manne models. The first Manne model, first presented by Özgüven et al. [58] is called the *benchmark model* in the article. The second model, called the *alternative model* in the article, is an enhanced version of the benchmark model, since some redundant variables are excluded. Our models are formulated and the iterative solution procedure is described. Computational results for the two objectives are presented. The benchmark test instances by Fattahi et al. [26] were chosen since they are commonly available and used in other comparisons; see e.g. [4, 7, 24, 58]. The benchmark model was chosen because it yielded the best results in the evaluation by Demir and İşleyen [24].

4.1.2 Results and contribution

We present a time-indexed formulation of the general FJSP employing nail variables. To the best of our knowledge, we are the first to develop this type of model for an FJSP. We show that there is a large difference in the performance of the scheduling models depending on which objective is considered. The makespan objective, although widely used in the research community, is not suitable for most practical applications. It is, however, important to evaluate scheduling models employing objectives well suited for real applications, such as our proposed objective (4a) including tardiness.

In order to evaluate the models considering the tardiness objective, we generated due dates for all the jobs in all the test instances. We intend to make these extensions of the Fattahi test instances available on the internet, such that they can serve as benchmark test instances employing objectives including tardiness, earliness, and/or the number of late jobs.

We show that the time-indexed model, combined with the iterative procedure, is able to find significantly better feasible solutions than those found by the two Manne models, for the largest instances considering both objectives studied. Further, our proposed scheduling method outperforms the two Manne models regarding the time to find the best feasible solution. For two of the largest instances only the alternative model was able to find and verify an optimal solution while minimizing the tardiness objective, although the time-indexed model was able to find optimal solutions to these instances however, not verify the optimality) within significantly shorter computing time.

4.2 Paper II: Scheduling optimization of a real flexible job shop including fixture availability and preventive maintenance

As being written prior to Paper I, Paper II ([87]) describes our first version of the iterative procedure, which is generic and can be used to solve any time-indexed model. The squeezing procedure (which is not generic) mentioned in Section 3.4 is only applied to the solution from the last iteration only.

4.2.1 Scope

We introduce the problem of scheduling the multitask cell including fixture availability and PM activities. The resulting schedules of our proposed scheduling method are aimed for the dynamic environment of the multitask cell, and the quality of our schedules is therefore discussed from this perspective.

4.2.2 Results and contribution

The time-indexed model with nail variables is presented as a basic model including precedence constraints between jobs. The three set-up stations are modelled as one resource wih a capacity of three in order to eliminate multiple, but equivalent, solutions due to symmetry. The side constraints modelling the limited fixture availability are included, as well as three alternative ways of scheduling the PM activities which are discussed and evaluated.

To our knowledge, we are the first to propose objective weights as functions of the jobs' due dates, such that a delayed job is assigned a higher weight than a job that is less delayed. This function works well in a dynamic environment, since a delayed job that has not been processed at the time of rescheduling will be even more delayed for the next schedule; therefore, it will be assigned a higher weight than in the previous schedule. We propose a hybrid event-driven rescheduling policy for the dynamic scheduling of the multitask cell, i.e., rescheduling is performed (i) periodically (before the start of every work shift), and (ii) whenever a major unexpected event, such as a machine breakdown, occurs.

Our computational results show that our iterative procedure is an efficient approach to find a suitable value of the time horizon as well as a good starting solution to the time-indexed model with the desired final length ℓ of the time step, such that the total computation times (including all iterations) are shorter than if the time-indexed model is solved directly.

Utilizing the method developed we are able to produce optimal, or near-optimal, schedules for real data instances within a time frame which is acceptable in practice (i.e., around 15 minutes).

4.3 Paper III: Scheduling optimization of a real flexible job shop including side constraints regarding maintenance, fixtures, and night shifts

In Paper III ([88]), we use the findings from Papers I–II to solve the problem of scheduling the multitask cell including side constraints regarding maintenance, fixtures, and night shifts.

4.3.1 Scope

The iterative procedure is further refined in Paper III. We test the procedure with, and without, squeezing of the schedules in every iteration, and also for different variants of the reduction between iterations of the length ℓ of the time steps. The inclusion of the possibility to schedule operations with unmanned processing during night shifts are evaluated. The test instances used are twelve real instances for the coming work shift collected in the multitask cell during 2012 and 2013. The solution procedure is terminated at the time limits of 15 minutes or 2 hours (clocktime), where the limit of 2 hours is employed in order to obtain good solutions to be used in the comparison between the different test settings. Therefore, the number of iterations performed by the iterative procedure differ both between the test settings employed and between the test instances.

In the evaluation of the machining problem of the multitask cell in [80], the results from the priority rules of EDD (earliest due date), SPT (shortest processing time), and FIFO (first–in, first–out) are compared with those from the time-indexed model using nail variables. The value of the total tardiness from the EDD, SPT, and FIFO rules, respectively, was on average 10%, 6%, and 22% higher than the total tardiness in the schedule found by the time-indexed model. However, since the FIFO rule is currently used as decision support in the detailed production planning of a majority of the workshops at GKN Aerospace, and since there are plans to use the built-in scheduling method of the control system in the multitask cell, which is based on the CR rule given in (1), we chose to use the two latter rules for the evaluation performed in Paper III.

4.3.2 Results and contribution

We are the first to propose a mathematical model for an FJSP which includes the opportunity to schedule unmanned processes during night shifts. Computational results show that the gain, in terms of objective value, of including the night shift constraints in the mathematical model is around 10%.

In a complex environment such as a flexible job shop with unmanned night shifts, it is not recommended to use priority dispatching rules: The objective values computed from the schedules constructed with both the FIFO and the CR rule were on average 1.4 times larger than those found after running the iterative procedure for 15 minutes.

The iterative procedure, implemented such that the computed schedules are squeezed in every iteration, performed equally well as when the computed schedule from the final iteration only is squeezed. The gain of applying the squeezing procedure is substantial when the squeezed schedules and the unsqueezed schedules are compared, and it is dependent on the value of ℓ (see Section 3.4). Since the relative difference between the results obtained after 15 minutes computing time differ only a little from those obtained after two hours (with eventually small values of ℓ), we conclude that the proposed iterative procedure is able to produce near-optimal schedules for real data instances for the coming shift within an acceptable time frame.

4.4 Paper IV: A note on the complexity of flow-shop scheduling with deteriorating jobs

In Paper IV ([83]), the NP-complexity of flow shop scheduling with deteriorating jobs is examined.

4.4.1 Scope

Paper IV is a note on the artcle ”Complexity analysis of job-shop scheduling with deteriorating jobs” (G. Mosheiov, Discrete Applied Mathematics, 117 (2002), pp. 195–209, [54]), which is quite well-cited and claims to provide a complete analysis of the complexity of flow shop, open shop, and job shop problems. A proportional deterioration rate is assumed, that is, the job processing times are increasing linear functions of the jobs' starting times. The objective is to minimize the makespan. Mosheiov [54] presents NP-hardness results for flow shops and open shops with three or more machines and for job shops with two or more machines. The proof of NP-hardness for the flow shop case is, however, not correct. Paper IV provides a correct proof.

4.4.2 Results and contribution

We provide a correct proof of the statement that the flow shop problem with deteriorating jobs is NP-complete even for three machines. In the proof, we transform the ”Equal Products Problem”, which is known to be NP-complete ([29]), into an instance of the flow shop problem.

During our work with this note, it came to our notice that a correct proof was published already in 1996 by Kononov, however, in Russian; see [44], but a summary of the proof of the NP-completeness result is found in the English language text book by Gawiejnowicz [31, Theorem 8.7] (incidentally, this proof summary includes a misprint: ”$1 \leq k \leq p-1$” should be replaced by ”$3 \leq k \leq p-1$”). Unfortunately the book also reprints, in the form of a summary, the erroneous proof of Mosheiov ([31, Theorem 8.6]); therefore, it does not correct the proof of Mosheiov in [54] for the flow shop problem with deteriorating jobs. A citation study was performed during the work with Paper IV in May 2010, primarily using the Web of Science databases ([95]). Mosheiov [54] had then been cited 31 times. Among them, six referred either to the polynomial algorithms or to one of the other theorems in the paper than its Theorem 2. Among the remaining 25 papers, 20 referred primarily to the result in [54, Theorem 2]. These papers had in turn generated about 150 ”descendants”, and their number is steadily growing.

The paper by Kononov [44] could not be found using Web of Science. Google Scholar ([33]) provided nine citations, among them four were written by S. Gawiejnowicz and co-authors, and two were written in Chinese. While some of them refer to the paper [54], none of them corrects the erroneous proof. Gawiejnowicz’s [31] book was not found using Web of Science. It had, according to Google Scholar, at the time been cited four times; none of them simultaneously cite Mosheiov [54].

We concluded that hitherto in the open literature, in no case had Mosheiov’s [54, Theorem 2] erroneous proof been corrected, and compared with the analysis made by Kononov [44]—and summarized in [31, Theorem 8.7]—it had found a much larger audience. Our correction in Paper IV was therefore both necessary and quite timely.

5 Conclusions

The work with this thesis has been a journey from the creation of the first engineer’s model, which was not able to solve instances of the size needed for the practical application—scheduling the jobs in the multitask production cell—, to the development of the iterative solution procedure with the time-indexed model with nail variables that is able to produce near-optimal schedules for the coming shift within a reasonable time frame.

We have developed a scheduling method that would substantially increase the output of the multitask cell as well as the machine utilization. Employing this method would lower the product costs since it is able to produce schedules that optimally utilize the unmanned night shifts for unmanned processing. Furthermore, the throughput times through the cell will decrease, which in turn will enable a reduction of the planned product lead times, with a greater flexibility in the production as an additional outcome.

The iterative solution procedure proposed is a generic procedure that can be used for the solution of any time-indexed model. We have shown that our iterative procedure with the time-indexed model with nail variables is highly competitive with other mathematical optimization models. We have also shown that the large difference in the performance of the scheduling models—which is dependent on the objective considered—imply that the evaluation of scheduling algorithms must be done with respect to an objective that is suitable for the real application for which they are intended. The minimization of the makespan is typically not such an objective, although it is the most widely used objective in research. We recommend the minimization of a weighted sum of the completion times plus a weighted total tardiness as objective, where the tardiness weights are non-increasing functions of the jobs’ due dates.

Since we schedule the preventive maintenance activities simultaneously with the processing jobs, we assure that the disturbance in the production due to maintenance is as small as possible. The managers of the multitask cell think positively of the results presented in this thesis, and their ambition is to implement our proposed scheduling method in the cell. A pilot study for a possible implementation will be launched after finishing the work with this thesis. The work with the implementation will probably set focus on some areas of future research that we are not aware of at the moment. In the next section we propose some areas of future research that we think may be of interest.

6 Future research

An interesting area of future research is to investigate the benefits of our scheduling model in a dynamic environment. One way to accomplish this is to simulate the production by the schedules produced by our iterative scheduling procedure, according to the proposed hybrid event-driven scheduling policy. This simulation should include, e.g., occasional machine breakdowns and a variability in the processing times. During this work, it would be very interesting to also find a means to measure—and to actually measure—the robustness of our schedules.

During the work with this thesis we have also come across some interesting inequalities that are valid for a hybrid flow shop in [8]. The transformation of these inequalities to our FJSP model and the implementation of a cutting plane algorithm, in order to obtain better lower bounds on the objective values are performed in an ongoing master's thesis project at the Department of Mathematical Sciences, by a student from Chalmers University of Technology.

Other areas for future research are the inclusion of the scheduling of personnel and of the limited availability of tools used in the machines in the multitask cell, in our scheduling models.

Glossary

CR	critical ratio priority dispatching rule
EDD	earliest due date priority dispatching rule
ERP	enterprise resource planning
FIFO	first–in, first–out priority dispatching rule
FJSP	flexible job shop scheduling problem
GKN	Guest, Keen & Nettlefolds, a global engineering group and owner of GKN Aerospace Engine Systems
JSP	job shop scheduling problem
MILP	mixed integer linear programming
NFFP	nationella flygforskningsprogrammet (Swedish National Aeronautics Research Programme)
NP	nondeterministic polynomial time (the time required to solve an NP-hard or NP-complete problem grows exponentially with its size)
PhD	Doctor of Philosophy (although this thesis deals with applied mathematics)
PM	preventive maintenance
SPT	shortest processing time priority dispatching rule

References

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

[2] N. Al-Hinai and T. Y. ElMekkawy, *An efficient hybridized genetic algorithm architecture for the flexible job shop scheduling problem*, Flexible Services and Manufacturing Journal, 23 (2011), pp. 64–85.

[3] S. Azem, R. Aggoune, and S. Dauzere-Peres, *Disjunctive and time-indexed formulations for non-preemptive job shop scheduling with resource availability constraints*, in 2007 IEEE International Conference on Industrial Engineering and Engineering Management, Dec. 2007, pp. 787–791.

[4] A. Bagheri, M. Zandieh, I. Mahdavi, and M. Yazdani, *An artificial immune algorithm for the flexible job-shop scheduling problem*, Future Generation Computer Systems, 26 (2010), pp. 533–541.

[5] P. Baptiste, M. Flamini, and F. Sourd, *Lagrangian bounds for just-in-time job-shop scheduling*, Computers & Operations Research, 35 (2008), pp. 906–915.

[6] A. Baykasoglu and L. Özbakir, *Analyzing the effect of dispatching rules in the scheduling performance through grammar based flexible scheduling system*, International Journal of Production Economics, 124 (2010), pp. 269–381.

[7] D. Behnke and M. J. Geiger, *Test instances for the flexible job shop scheduling problem with work centers*. `http://www.nbn\discretionary{-}{}{}resolving.de/urn:nbn:de:gbv:705-opus-29827` (Accessed on August 13, 2013), 2012.

[8] L. Berghman, *Machine scheduling models for warehousing docking operations*, PhD Thesis, Katholieke Universiteit Leuven, Belgium, 2012.

[9] L. Berghman, R. Leus, and F. C. R. Spieksma, *Optimal solutions for a dock assignment problem with trailer transportation*, Annals of Operations Research, (2011). doi 10.1007/s10479-011-0971-7.

[10] D. Bertsimas, D. Brown, and C. Caramanis, *Theory and applications of robust optimization*, SIAM Review, 53 (2011), pp. 464–501.

[11] L. Bigras, M. Gamache, and G. Savard, *Time-indexed formulations and the total weighted tardiness problem*, INFORMS Journal on Computing, 20 (2008), pp. 133–142.

[12] J. H. Blackstone, Jr., D. T. Phillips, and G. L. Hogg, *A state-of-the-art survey of dispatching rules for manufacturing job shop operations*, International Journal of Production Research, 20 (1982), pp. 27–45.

[13] J. BŁAŻEWICZ, W. DOMSCHKE, AND E. PESCH, *The job shop scheduling problem: Conventional and new solution techniques*, European Journal of Operational Research, 93 (1996), pp. 1–33.

[14] E. H. BOWMAN, *The schedule-sequencing problem*, Operations Research, 7 (1959), pp. 621–624.

[15] W. BOŻEJKO, M. UCHROŃSKI, AND M. WODECKI, *The new golf neighborhood for the flexible job shop problem*, Procedia Computer Science, 1 (2010), pp. 289–296.

[16] P. BRUCKER, *Scheduling Algorithms*, Springer-Verlag, Berlin Heidelberg, Germany, 5th ed., 2007.

[17] P. BRUCKER, B. JURISCH, AND A. KRÄMER, *Complexity of scheduling problems with multi-purpose machines*, Annals of Operations Research, 70 (1997), pp. 57–73.

[18] P. BRUCKER AND S. KNUST, *Complex Scheduling*, Springer-Verlag, Berlin, Germany, 2nd ed., 2012.

[19] K. BÜLBÜL AND P. KAMINSKY, *A linear programming-based method for job shop scheduling*, Journal of Scheduling, 16 (2013), pp. 161–183.

[20] G. CALLEJA AND R. PASTOR, *A dispatching algorithm for flexible job-shop scheduling with transfer batches: an industrial application*, Production Planning & Control, (2013), pp. 1–17.

[21] R. CHENG, M. GEN, AND Y. TSUJIMURA, *A tutorial survey of job-shop scheduling problems using genetic algorithms–I. representation*, Computers & Industrial Engineering, 30 (1996), pp. 983–997.

[22] T. CHIANG AND H. LIN, *A simple and effective evolutionary algorithm for multiobjective flexible job shop scheduling*, International Journal of Production Economics, 141 (2013), pp. 87–98.

[23] F. M. DEFERSHA AND M. CHEN, *A parallel genetic algorithm for a flexible job-shop scheduling problem with sequence dependent setups*, The International Journal of Advanced Manufacturing Technology, 49 (2010), pp. 263–279.

[24] Y. DEMIR AND S. K. İŞLEYEN, *Evaluation of mathematical models for flexible job-shop scheduling problems*, Applied Mathematical Modelling, 37 (2013), pp. 977–988.

[25] P. FATTAHI AND A. FALLAHI, *Dynamic scheduling in flexible job shop systems by considering simultaneously efficiency and stability*, CIRP Journal of Manufacturing Science and Technology, 2 (2010), pp. 114–123.

[26] P. Fattahi, M. Saidi Mehrabad, and F. Jolai, *Mathematical modeling and heuristic approaches to flexible job shop scheduling problems*, Journal of Intelligent Manufacturing, 18 (2007), pp. 331–342.

[27] R. Fourer, D. M. Gay, and B. W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*, Brooks/Cole Publishing Company/Cengage Learning, 2nd ed., 2002.

[28] J. Gao, M. Gen, and L. Sun, *Scheduling jobs and maintenances in flexible job shop with a hybrid genetic algorithm*, Journal of Intelligent Manufacturing, 17 (2006), pp. 493–507.

[29] M. R. Garey and D. S. Johnson, *Computers and intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co., New York, NY, USA, 1979.

[30] M. R. Garey, D. S. Johnson, and R. Sethi, *The complexity of flowshop and jobshop scheduling*, Mathematics of Operations Research, 1 (1976), pp. 117–129.

[31] S. Gawiejnowicz, *Time-Dependent Scheduling*, in Monographs in Theoretical Computer Science, Springer-Verlag, Berlin-Heidelberg, 2008.

[32] M. C. Gomes, A. P. Barbosa-Póvoa, and A. Q. Novais, *Optimal scheduling for flexible job shop operation*, International Journal of Production Research, 43 (2005), pp. 2323–2353.

[33] Google Scholar. Available at `http://scholar.google.se/`.

[34] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan, *Optimization and approximation in deterministic sequencing and scheduling: A survey*, Annals of Discrete Mathematics, 5 (1979), pp. 287–326.

[35] R. Haupt, *A survey of priority rule-based scheduling*, Operations-Research-Spektrum, 11 (1989), pp. 3–16.

[36] W. J. Hopp and M. L. Spearman, *Factory Physics*, McGraw-Hill/Irwin, New York, NY, USA, 3d ed., 2008.

[37] IBM Corp., *IBM ILOG CPLEX V12.1 User's Manual for CPLEX*, 2009.

[38] A. S. Jain and S. Meeran, *Deterministic job-shop scheduling: Past, present and future*, European Journal of Operational Research, 113 (1999), pp. 390–434.

[39] T. Jansson, *Resource utilization in a multitask cell*, Master's Thesis, Department of Mathematical Sciences, Chalmers University of Technology, Göteborg, Sweden, 2006.

[40] A. Jones, L. C. Rabelo, and A. T. Sharawi, *Survey of job shop scheduling techniques*. Available at `http://onlinelibrary.wiley.com`, 1999.

[41] H. Karimi, S. H. A. Rahmati, and M. Zandieh, *An efficient knowledge-based algorithm for the flexible job shop scheduling problem*, Knowledge-Based Systems, 36 (2012), pp. 236–244.

[42] S. Kedad-Sidhoum, Y. Rios-Solis, and F. Sourd, *Lower bounds for the earliness–tardiness scheduling problem on single and parallel machines*, European Journal of Operational Research, 189 (2008), pp. 1305–1316.

[43] K. Kim and P. Egbelu, *Scheduling in a production environment with multiple process plans per job*, International Journal of Production Research, 37 (1999), pp. 2725–2753.

[44] A. Kononov, *Combinatorial complexity of scheduling jobs with simple linear deterioration*, Discrete Analysis and Operations Research, 3 (1996), pp. 15–32. In Russian.

[45] Y. H. Lee, C. S. Jeong, and C. Moon, *Advanced planning and scheduling with outsourcing in manufacturing supply chain*, Computers & Industrial Engineering, 43 (2002), pp. 351–374.

[46] J. K. Lenstra, A. H. G. Rinnooy Kan, and P. Brucker, *Complexity of machine scheduling problems*, in Studies in Integer Programming, P. Hammer, E. Johnson, B. Korte, and G. Nemhauser, eds., vol. 1 of Annals of Discrete Mathematics, Elsevier, 1977, pp. 343–362.

[47] C. Low, Y. Yip, and T. Wu, *Modelling and heuristics of FMS scheduling with multiple objectives*, Computers & Operations Research, 33 (2006), pp. 674–694.

[48] G. B. Mainieri and D. P. Ronconi, *New heuristics for total tardiness minimization in a flexible flowshop*, Optimization Letters, 7 (2013), pp. 665–684.

[49] A. S. Manne, *On the job-shop scheduling problem*, Operations Research, 8 (1960), pp. 219–223.

[50] Y. Mati, S. Dauzère-Pérès, and C. Lahlou, *A general approach for optimizing regular criteria in the job-shop scheduling problem*, European Journal of Operational Research, 212 (2011), pp. 33–42.

[51] Y. Mati and X. Xie, *Multiresource shop scheduling with resource flexibility and blocking*, IEEE Transactions on Automation Science and Engineering, 8 (2011), pp. 175–189.

[52] L. Mönch and R. Driessel, *A distributed shifting bottleneck heuristic for complex job shops*, Computers & Industrial Engineering, 49 (2005), pp. 363–380.

[53] E. Moradi, S. M. T. Fatemi Ghomi, and M. Zandieh, *Bi-objective optimization research on integrated fixed time interval preventive maintenance and production for scheduling flexible job-shop problem*, Expert Systems with Applications, 38 (2011), pp. 7169–7178.

[54] G. Mosheiov, *Complexity analysis of job-shop scheduling with deteriorating jobs*, Discrete Applied Mathematics, 117 (2002), pp. 195–209.

[55] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*, John Wiley & Sons, Inc., New York, NY, USA, 1988.

[56] T. Nishi, Y. Hiranaka, and M. Inuiguchi, *Lagrangian relaxation with cut generation for hybrid flowshop scheduling problems to minimize the total weighted tardiness*, Computers & Operations Research, 37 (2010), pp. 189–198.

[57] D. Ouelhadj and S. Petrovic, *A survey of dynamic scheduling in manufacturing systems*, Journal of Scheduling, 12 (2009), pp. 417–431.

[58] C. Özgüven, L. Özbakir, and Y. Yavuz, *Mathematical models for job-shop scheduling problems with routing and process plan flexibility*, Applied Mathematical Modelling, 34 (2010), pp. 1539–1548.

[59] C.-H. Pan, *A study of integer programming formulations for scheduling problems*, International Journal of Systems Science, 28 (1997), pp. 33–41.

[60] S. Pettersson, *The interaction between optimization model and production at Volvo Aero*, Master's Thesis, Department of Technology Management and Economics, Chalmers University of Technology, Göteborg, Sweden, 2010.

[61] M. L. Pinedo, *Scheduling: Theory, Algorithms, and Systems*, Springer, New York, NY, USA, 4th ed., 2010.

[62] J. Poppenborg, S. Knust, and J. Hertzberg, *Online scheduling of flexible job-shops with blocking and transportation*, European Journal of Industrial Engineering, 6 (2012), pp. 497–518.

[63] A. A. B. Pritsker, L. J. Watters, and P. M. Wolfe, *Multiproject scheduling with limited resources: A zero-one programming approach*, Management Science, 16 (1969), pp. 93–108.

[64] S. Rahimifard and S. T. Newman, *Simultaneous scheduling of workpieces, fixtures and cutting tools within flexible machining cells*, International Journal of Production Research, 35 (1997), pp. 2379–2396.

[65] S. H. A. Rahmati and M. Zandieh, *A new biogeography-based optimization (BBO) algorithm for the flexible job shop scheduling problem*, The International Journal of Advanced Manufacturing Technology, 58 (2012), pp. 1115–1129.

[66] M. Rajkumar, P. Asokan, and V. Vamsikrishna, *A GRASP algorithm for flexible job-shop scheduling with maintenance constraints*, International Journal of Production Research, 48 (2010), pp. 6821–6836.

[67] R. Rangsaritratsamee, W. G. Ferrell Jr., and M. B. Kurz, *Dynamic rescheduling that simultaneously considers efficiency and stability*, Computers & Industrial Engineering, 46 (2004), pp. 1–15.

[68] J. Renze, *Outlier. From MathWorld–A Wolfram Web Resource, created by Weisstein, E. W.* `http://mathworld.wolfram.com/Outlier.html` (Accessed on August 2013), 2013.

[69] J. Roslöf, I. Harjunkoski, T. Westerlund, and J. Isaksson, *Solving a large-scale industrial scheduling problem using MILP combined with a heuristic procedure*, European Journal of Operational Research, 138 (2002), pp. 29–42.

[70] R. Rudek, *A note on proving the strong NP-hardness of some scheduling problems with start time dependent job processing times*, Optimization Letters, 6 (2012), pp. 1021–1025.

[71] R. Sadykov and L. A. Wolsey, *Integer programming and constraint programming in solving a multimachine assignment scheduling problem with deadlines and release dates*, INFORMS Journal on Computing, 18 (2006), pp. 209–217.

[72] J. P. Sousa and L. A. Wolsey, *A time indexed formulation of non-preemptive single machine scheduling problems*, Mathematical Programming, 54 (1992), pp. 353–367.

[73] P. P. M. Stoop and V. C. S. Wiers, *The complexity of scheduling in practice*, International Journal of Operations & Production Management, 16 (1996), pp. 37–53.

[74] A. Syberfeldt, *A multi-objective evolutionary approach to simulation-based optimisation of real-world problems*, PhD thesis, DeMontfort University, Leicester, UK, 2009.

[75] A. Syberfeldt, I. Karlsson, A. Ng, J. Svantesson, and T. Almgren, *A web-based platform for the simulation-optimization of industrial problems*, Computers & Industrial Engineering, 64 (2013), pp. 987–998.

[76] A. Syberfeldt, A. Ng, R. I. John, and P. Moore, *Multi-objective evolutionary simulation-optimisation of a real-world manufacturing problem*, Robotics and Computer-Integrated Manufacturing, 25 (2009), pp. 926–931.

[77] J. Tang, G. Zhang, B. Lin, and B. Zhang, *A hybrid algorithm for flexible job-shop scheduling problem*, Procedia Engineering, 15 (2011), pp. 3678–3683.

[78] J. C. TAY AND N. B. HO, *Evolving dispatching rules using genetic programming for solving multi-objective flexible job-shop problems*, Computers & Industrial Engineering, 54 (2008), pp. 453–473.

[79] C. S. THOMALLA, *Job shop scheduling with alternative process plans*, International Journal of Production Economics, 74 (2001), pp. 125–134.

[80] K. THÖRNBLAD, *On the Optimization of Schedules of a Multitask Production Cell*, Licentiate Thesis, Chalmers University of Technology and University of Gothenburg, Göteborg, Sweden, Sep. 2011.

[81] K. THÖRNBLAD, T. ALMGREN, M. PATRIKSSON, AND A.-B. STRÖMBERG, *Mathematical optimization of a flexible job shop problem including preventive maintenance and availability of fixtures*, in Proceedings of the 4th World P&OM Conference/19th International Annual EurOMA Conference, July 1–5, 2012, Amsterdam, The Netherlands, 2012, pp. 1–10.

[82] K. THÖRNBLAD AND L. KJELLSDOTTER IVERT, *A comparison of schedules resulting from priority rules and mathematical optimization for a real production cell*, in Proceedings of PLANs forsknings- och tillämpningskonferens, August 18–19, 2010, Skövde, Sweden, 2010, pp. 99–110.

[83] K. THÖRNBLAD AND M. PATRIKSSON, *A note on the complexity of flow-shop scheduling with deteriorating jobs*, Discrete Applied Mathematics, 159 (2011), pp. 251–253. Paper IV in this thesis.

[84] K. THÖRNBLAD, A.-B. STRÖMBERG, T. ALMGREN, AND M. PATRIKSSON, *Optimization of schedules for a multitask production cell*, in Proceedings of the 22$^{\text{nd}}$ NOFOMA conference, June 10–11, 2010, Kolding, Denmark, 2010, pp. 1–13.

[85] K. THÖRNBLAD, A.-B. STRÖMBERG, M. PATRIKSSON, AND T. ALMGREN, *A time-indexed formulation of a flexible job shop problem with a limited number of fixtures*, in Proceedings of the 13th International Conference on Project Management and Scheduling, April 1–4, 2012, Leuven, Belgium, 2012, pp. 335–338.

[86] ——, *A competitive iterative procedure using a time-indexed model for solving flexible job shop scheduling problems*. Paper I in this thesis. Available at `http://www.optimization-online.org/DB_HTML/2013/08/3991.html`, August 2013.

[87] ——, *Scheduling optimization of a real flexible job shop including fixture availability and preventive maintenance*, Revised for publication in European Journal of Industrial Engineering, (2013). Paper II in this thesis.

[88] ——, *Scheduling optimization of a real flexible job shop including side constraints regarding maintenance, fixtures, and night shifts*. Paper III in this thesis. Available at `http://www.optimization-online.org/DB_HTML/2013/08/3995.html`, August 2013.

[89] J. M. van den Akker, H. Hoogeveen, and S. Velde, *Applying column generation to machine scheduling*, in Column Generation, G. Desaulniers, J. Desrosiers, and M. M. Solomon, eds., Springer US, 2005, pp. 303–330.

[90] J. M. van den Akker, C. A. J. Hurkens, and M. W. P. Savelsberg, *Time-indexed formulations for machine scheduling problems: Column generation*, INFORMS Journal on Computing, 12 (2000), pp. 111–124.

[91] J. Van den Bergh, J. Beliën, P. De Bruecker, E. Demeulemeester, and L. De Boeck, *Personnel scheduling: A literature review*, European Journal of Operational Research, 226 (2013), pp. 367–385.

[92] H. M. Wagner, *An integer linear-programming model for machine scheduling*, Naval Research Logistics Quarterly, 6 (1959), pp. 131–140.

[93] L. Wang, G. Zhou, Y. Xu, S. Wang, and M. Liu, *An effective artificial bee colony algorithm for the flexible job-shop scheduling problem*, The International Journal of Advanced Manufacturing Technology, 60 (2012), pp. 303–315.

[94] S. Wang and J. Yu, *An effective heuristic for flexible job-shop scheduling problem with maintenance activities*, Computers & Industrial Engineering, 59 (2010), pp. 436–447.

[95] Web of Science databases. Available at `http://thomsonreuters.com/web-of-science/`.

[96] L. A. Wolsey, *MIP modelling of changeovers in production planning and scheduling problems*, European Journal of Operational Research, 99 (1997), pp. 154–165.

[97] Y. Yuan, H. Xu, and J. Yang, *A hybrid harmony search algorithm for the flexible job shop scheduling problem*, Applied Soft Computing, 13 (2013), pp. 3259–3272.

[98] G. Zhang, L. Gao, and Y. Shi, *An effective genetic algorithm for the flexible job-shop scheduling problem*, Expert Systems with Applications, 38 (2011), pp. 3563–3573.

[99] X. Zhang, *Scheduling with Time Lags*, PhD Thesis, Erasmus Research Institute of Management, Erasmus University Rotterdam, The Netherlands, 2010.

[100] Z. Zhu and R. B. Heady, *Minimizing the sum of earliness/tardiness in multi-machine scheduling: a mixed integer programming approach*, Computers & Industrial Engineering, 38 (2000), pp. 297–305.