New solution approaches for scheduling problems in production and logistics

At the faculty of
Rechts- und Wirtschaftswissenschaften
of the Technischen Universität Darmstadt

Submitted in fulfilment of the requirements for the degree of Doctor rerum politicarum
(Dr. rer. pol.)

Doctoral Thesis
By Hamid Abedinnia

First assessor: Prof. Dr. Christoph Glock
Second assessor: Jun.-Prof. Dr. Simon Emde

Darmstadt 2018 (year of the viva voce)

Abedinnia, Hamid: New solution approaches for scheduling problems in production and logistics
Darmstadt, Technische Universität Darmstadt,
Year thesis published in TUprints 2019
Day of the viva vice: 09.11.2018

Ehrenwörtliche Erklärung

Ich erkläre hiermit ehrenwörtlich, dass ich die vorliegende Dissertation selbstständig angefertigt habe. Sämtliche aus fremden Quellen direkt oder indirekt übernommene Gedanken sind als solche kenntlich gemacht.

Die Dissertation wurde bisher keiner anderen Prüfungsbehörde vorgelegt.

Darmstadt, den 23. May 2018

Outline

Inhalt

Table of Contents

List of Abbreviations

ABS	Absolute value
ACO	Ant colony optimization
AGB	The heuristic developed by Abedinnia, Glock and Brill
AGV	Automated guided vehicles
AI	Artificial intelligence
AM	Adaptive memory
AMP	Adaptive memory programming
ANN	Artificial neural network
ARPD	Average relative percentage deviation
AS/RS	Automated storage and retrieval system
AVE	Average dependency
ave	Average
CA	Content analysis
cm	Centimetre
CON	Constant
cont.	Continued
CPU	Central processing unit
e.g.	Exempli gratia (for example)
EA	Evolutionary algorithm
Eq.	Equation
EVMSP	Electric vehicle milk-run scheduling problem
FCFS	First come first served
FF	The heuristic developed by Fernandez-Viagas and Framinan
FIFO	First in first out
Fig	Figure
FL	The heuristic developed by Framinan and Laha
FMS	Flexible manufacturing systems
FSP	Flowshop scheduling problem
FT	Flowtime
GA	Genetic algorithm
GB	Gigabyte
GDP	Gross domestic product
GHz	Gigahertz
GLS	Guided local search
GSP	Generalized set partitioning
h.	Heuristic
HFS	Hybrid flowshop
HFSP	Hybrid flowshop scheduling problem
i.e.	Id est (in other words)
ILP	Integer Linear programming
ILS	Iterated local search

kg	Kilogram
LB	Lower bound
LP	Linear programming
LS	The heuristic developed by Laha and Sarin
max	Maximum
MILP	Mixed ingeger linear programming
min	Minimum
MIP	Mixed ingeger programming
MJTWPT	Machine & Job-based Total Weighted Processing Time
MSPP	Machine scheduling problems in production
MTWPT	Machine-based Total Weighted Processing Time
N	Newton
n.R.	No rotation
NEH	The heuristic developed by Nawas, Enscor and Ham
No.	Number
NP	Non-deterministic polynomial-time
OEM	Original Equipment Manufacturer
OH	Opening heuristic
opt	Optimum
PC	Personal computer
PFSP	Permutation flowshop scheduling problem
PSO	Particle swarm optimization
RAM	Random Access Memory
RCPSP	Resource-constrained project scheduling problem
rnd	Uniformly distributed random integer
RPD	Relative percentage deviation
SA	Simulated annealing
sec.	Second
SI	Shortest imminent
SIG	Slope Indices of Gupta
SIP	Slope Indices of Palmer
SIR	Slope Indices of Rajendran
SJR	SCImago Journal Rank
SNIP	Source Normalized Impact per Paper
SPT	Shortest processing time
SS	Scatter search
TFT	Total flowtime
TS	Tabu search
TSP	Travelling salesman problem
TWFT	Total weighted flow-time
TWK	Total work-content
ULD	Unit load devices

ULDSP	The problem of scheduling the build-up of unit load devices at an air cargo terminal under space and personnel constraints
US	United states of America
VNS	Variable neighborhood search
vs.	Versus

List of Figures

Paper 1

Paper 2

Paper 3

Paper 4

Paper 5

Paper 6

List of Tables

Introduction

Paper 1

Paper 2

Paper 3

Paper 4

Paper 5

Paper 6

Abstract

The current cumulative PhD thesis consists of six papers published in/submitted to scientific journals. The focus of the thesis is to develop new solution approaches for scheduling problems encountering in manufacturing as well as in logistics. The thesis is divided into two parts: “machine scheduling in production” and “scheduling problems in logistics” each of them consisting three papers.

To have most comprehensive overview of the topic of machine scheduling, the first part of the thesis starts with two systematic review papers, which were conducted on tertiary level (i.e., reviewing literature reviews). Both of these papers analyze a sample of around 130 literature reviews on machine scheduling problems. The first paper use a subjective quantitative approach to evaluate the sample, while the second papers uses content analysis which is an objective quantitative approach to extract meaningful information from massive data. Based on the analysis, main attributes of scheduling problems in production are identified and are classified into several categories. Although the focus of both these papers are set to review scheduling problems in manufacturing, the results are not restricted to machine scheduling problem and the results can be extended to the second part of the thesis. General drawbacks of literature reviews are identified and several suggestions for future researches are also provided in both papers.

The third paper in the first part of the thesis presents the results of 105 new heuristic algorithms developed to minimize total flow time of a set of jobs in a flowshop manufacturing environment. The computational experiments confirm that the best heuristic proposed in this paper improves the average error of best existing algorithm by around 25 percent.

The first paper in second part is focused on minimizing number of electric tow-trains responsible to deliver spare parts from warehouse to the production lines. Together with minimizing number of these electric vehicles the paper is also focused to maximize the work load balance among the drivers of the vehicles. For this problem, after analyzing the complexity of the problem, an opening heuristic, a mixed integer linear programing (MILP) model and a taboo-search neighborhood search approach are proposed. Several managerial insights, such as the effect of battery capacity on the number of required vehicles, are also discussed.

The second paper of the second part addresses the problem of preparing unit loaded devices (ULDs) at air cargos to be loaded latter on in planes. The objective of this problem is to minimize number of workers required in a way that all existing flight departure times are met and number of available places for building ULDs is not violated. For this problem, first, a MILP model is proposed and then it is boosted with a couple of heuristics which enabled the model to find near optimum solutions in a matter of 10 seconds. The paper also investigates the inherent tradeoff between labor and space utilization as well as the uncertainty about the volume of cargo to be processed.

The last paper of the second part proposes an integrated model to improve both ergonomic and economic performance of manual order picking process by rotating pallets in the warehouse. For the problem under consideration in this paper, we first present and MILP model and then propose a neighborhood search based on simulated annealing. The results of numerical experiment indicate that selectively rotating pallets may reduce both order picking time as well as the load on order picker, which leads to a quicker and less risky order picking process.

Zusammenfassung

Die vorliegende kumulative Doktorarbeit besteht aus sechs Artikeln, die in wissenschaftlichen Zeitschriften veröffentlicht wurden. Der Schwerpunkt dieser Doktorarbeit liegt in der Entwicklung neuer Lösungsansätze für die Planungsprobleme, die in der Fertigung und in der Logistik auftreten. Die Arbeit gliedert sich in zwei Teile: „Maschinenplanung in der Produktion“ und „Planungsprobleme in der Logistikplanung“, die jeweils aus drei Artikeln bestehen.

Um einen möglichst umfassenden Überblick über das Thema Maschinenplanung zu erhalten, beginnt der erste Teil der Arbeit mit zwei systematischen Literaturüberblickarbeiten, die im Tertiärstudie durchgeführt wurden (d.h. Überblick von Literaturüberblicke). In beiden diese arbeiten wird eine Stichprobe von rund 130 Literaturüberblicken zu Problemen in Bereich der Maschinenplanung analysiert. Der erste Artikel verwendet einen subjektiven quantitativen Ansatz zur Bewertung der Stichprobe, während der zweite Artikel die sogenannte Inhaltsanalyse (Content Analyse, CA) verwendet. Dies ist ein objektiver quantitativer Ansatz, um aussagekräftige Informationen aus massiven Daten zu extrahieren. Basierend auf der Analyse werden die Attribute von Planungsproblemen in der Produktion identifiziert und in verschiedene Kategorien klassifiziert. Obwohl liegt der Fokus dieser beiden Arbeiten auf der Überprüfung von Planungsproblemen in der Fertigung, die Ergebnisse beschränken sich jedoch nicht auf das Maschinenplanungsproblem, und die Ergebnisse können auf den zweiten Teil der Doktorarbeit ausgedehnt werden. Allgemeine Nachteile als auch Hindernisse von Literaturüberblickarbeiten werden identifiziert, und in beiden Artikeln werden auch einige Vorschläge für zukünftige Forschungen gegeben.

Der dritte Artikel im ersten Teil der Arbeit stellt die Ergebnisse von 105 neuen heuristischen Algorithmen vor, die entwickelt wurden, um die Gesamtauslaufzeit einer Reihe von Jobs in einer Fertigungsumgebung eines Flowshops zu minimieren. Die rechnerischen Experimente bestätigen, dass die in diesem Artikel vorgeschlagene beste Heuristik den durchschnittlichen Fehler des besten vorhandenen Algorithmus um etwa 25 Prozent verbessert.

Der erste Artikel im zweiten Teil konzentriert sich auf die Minimierung der Anzahl elektrischer Schleppzüge, die für die Lieferung von Ersatzteilen vom Lager an die Produktionslinien verantwortlich sind. Zusammen mit der Minimierung der Anzahl dieser Elektrofahrzeuge ist das Papier auch darauf ausgerichtet, die Arbeitslast unter den Fahrern der Fahrzeuge abzugleichen. Für dieses Problem werden nach Analyse der Komplexität des Problems eine Öffnungsheuristik, ein MILP-Modell (Mixed Integer Linear Programing) und ein Nachbarschaftssuchansatz mit hilfe die so genante Tabu-Search meta-heuristik-vorgeschlagen. Darüber hinaus werden einige Erkenntnisse aus dem Management diskutiert, beispielsweise der Einfluss der Batteriekapazität auf die Anzahl der benötigten Fahrzeuge.

Der zweite Artikel des zweiten Teils befasst sich mit dem Problem der Vorbereitung von Unit-Loaded Devices (ULDs) an Luftfrachtgütern, die in Ebenen geladen werden sollen. Das Ziel dieses Problems besteht darin, die Anzahl der erforderlichen Arbeitskräfte so zu minimieren, dass alle bestehenden Abflugzeiten eingehalten werden und die Anzahl der verfügbaren Plätze für die Ladung von ULDs nicht gestört wird. Für dieses Problem wird zunächst ein MILP-Modell vorgeschlagen und anschließend mit einigen Heuristiken verstärkt, wodurch das Modell innerhalb von 10 Sekunden nahezu optimale Lösungen finden konnte. Der Artikel untersucht auch den inhärenten Kompromiss zwischen Arbeits- und Raumnutzung sowie die Ungewissheit über das zu verarbeitende Frachtvolumen.

Der letzte Artikel des zweiten Teils schlägt ein integriertes Modell vor, um die Ergonomie und die Wirtschaftlichkeit des manuellen Kommissionierprozesses durch rotierende Paletten im Lager zu verbessern. Für das in dieser Arbeit behandelte Problem stellen wir zuerst ein MILP-Modell vor und schlagen dann eine Nachbarschaftssuche vor, die auf so genannte Simuliertem Anealing meta-heuristik basiert ist. Die Ergebnisse des numerischen Experiments zeigen, dass selektiv rotierende Paletten sowohl die Kommissionierzeit als auch die Belastung des Kommissionierers reduzieren können, was zu einem schnelleren und weniger riskanten Kommissionierprozess führt.

Introduction

Scheduling is a decision-making process to determine the allocation of restricted resources to accomplish a set of tasks, and its goal is to optimize given (and sometime conflicting) objectives subject to the considered constraints (Pinedo, 2008). This definition of scheduling makes it relevant to almost all planning activities, from planning personal daily activities to sophisticated production planning in the manufacturing of semiconductor circuits, for example. Depending on the area where the schedule needs to be prepared, resources, tasks, objectives and constraints can be of different forms. Table 1 illustrates some examples for these different forms in four different industrial disciplines in which scheduling is relevant: I) machine scheduling in production (which is covered in the first part of this thesis); II) scheduling problems in logistics (which is the subject of the second part of this thesis); III) project scheduling; and IV) scheduling in software development.

This broad applicability of scheduling made it popular among practitioners and simultaneously attractive for academics. The scientific literature on scheduling is extensive and, as Gorman (2016) stated, scheduling is the second most frequently studied optimization problem in the Operations Research and Management Science literature.

This cumulative dissertation consists of six papers either published in or submitted to different scientific journals (see Table 2 for more information). All six papers are concerned with scheduling problems. Due to differing foci of the papers, this dissertation is divided into two parts. Part A includes Papers 1 to 3 and contributes to a research stream that investigates scheduling problems in production, which are also known as machine scheduling problems. Part B features Papers 4 to 6 and investigates scheduling problems encountered in planning intra-plant logistics activities. Aside from the differing foci, the six papers also vary in the methodologies employed. The first two papers present systematic literature reviews on the state-of-research of machine scheduling problems in production. These literature reviews are conducted on a tertiary level, which means they analyze a literature sample consisting exclusively of literature review papers. As the topic of machine scheduling is very general and not restricted to a specific industry, and as there are some similarities between scheduling problems in production and in logistics, these two literature reviews also cover some studies that focus on the application of scheduling in logistics (e.g., the scheduling of automated guided vehicles (AGVs) is discussed in both papers). The remaining four papers develop mathematical models for supporting production/logistics planning decisions. Papers 3 develops a set of constructive heuristic algorithms to minimize total throughput time in a basic production system. Papers 4 to 6 first present mixed-integer programing (MIP) models for the problems under consideration and then propose different heuristic and meta-heuristic algorithms, which are all able to obtain near-optimal solutions for their respective problems with reasonable runtime. The following provides an introduction to the research areas that the papers contribute to and explains the research gaps they are looking to fill.

A regular literature review (which is frequently also referred to as a secondary study) aims to analyze primary works published in a specific research area (Garousi & Mäntylä, 2016; Hochrein et al., 2015).

Area of application	Resource(s)	Task(s)	Objective(s)	Constraint(s)	Sample references
Machine scheduling in production	machines, manpower, electricity	parts/orders to be manufactured	minimizing throughput time, maximizing machine utilization	machine breakdown, controllable processing times	Sanlaville & Schmidt (1998) Panwalkar et al. (2013) Hall & Sriskandarajah (1996)
Scheduling problem in logistics	transportation vehicles, manpower	products to be packed/transported	minimizing number of vehicles in use, maximizing	just-in-time deliveries, ergonomic constraints	Behnamian & F. Ghomi (2014) Chen & Lee (2008)
Project scheduling	financial resources, manpower, machines	Project parts to be done	minimizing delay, maximizing	Precedence constraints, unexpected hazards	Herroelen et al. (1999) Brucker et al. (1999)
Scheduling in software development	processors, network links or expansion cards, computing units	computation elements, software segments, threads	maximizing load balancing, minimizing response time	must-meet deadlines, dynamic arrival times	Yuan & Nahrstedt (2003) Singh et al. (2010)

Table 1: Application of scheduling problems in different industries

However, the massive body of literature in some research streams, such as machine scheduling, prohibits a regular secondary review that covers the entire available literature. Tertiary studies (i.e., reviews of literature reviews) may support structuring and synthesizing a particular research area in such cases (Kache & Seuring, 2014; Seuring & Gold, 2012). According to Paper 1, the contributions of a tertiary study is manifold: (I) giving an aggregated overview of a research domain; (II) analyzing research trends in the domain of interest; (III) evaluating the methodological rigor of literature reviews in the domain; and (IV) identifying research gaps (on the secondary and, if possible, also on the primary level). Papers 1 and 2 present the results of tertiary studies conducted on the domain of machine scheduling problems in production (MSPP). The comprehensive samples of both papers, containing around 130 literature reviews, have been generated in a systematic way to make the results reproducible and reliable.

As describing the review methodology employed is necessary for making the generation of the sample reproducible and more reliable, Paper 1 first describes the review methodology used by outlining both the literature search strategy as well as inclusion/exclusion criteria employed. It then proposes a conceptual framework that considers the main attributes of MSPP in 7 categories and 75 sub-categories. After a descriptive analysis of the sampled papers that gives insight into publication patterns for MSPP, a quantitative analysis of the sampled review papers is carried out based on the proposed framework. A synthesis of research findings describes the state-of-knowledge and unveils general deficiencies of literature reviews on MSPP. In addition, the paper provides a comprehensive overview of MSPP, which supports researchers in positioning their own work in the literature and in finding potential innovative research areas. Based on a discussion of the statistical findings, opportunities for future research on MSPP are proposed.

Paper 2 presents the results of a content analysis (CA) on a comprehensive sample generated in a similar way as in Paper 1. Paper 2 starts with a short description of the CA, which is an objective and quantitative approach to extract worthwhile information from massive data (Neuendorf, 2002). The objective of a CA is to make replicable and valid inferences from texts or other meaningful matters to the context of their use (Krippendorff, 2012). Among four different types of CAs that have been discussed in the literature (i.e., descriptive, inferential, psychometric, and predictive CA), Paper 2 employs a descriptive approach, whose conclusions are limited to the content under study. As recording units, the paper selects words (e.g., “makespan”), abbreviations (e.g., “HFS”, which stands for “hybrid flow shop”) and symbols (e.g., “Cmax”, which stands for maximum completion time). The appearance frequency of the recording units in the sample are set as the objects of the analysis. At the end of the search phase of the CA, the recording units that are used more frequently in the sample are rated as more important. To ensure that the process of categorizing recording units is exclusive and exhaustive, both inductive and deductive categorization approaches are employed (more information on categorization approaches in the CA can be found in Neuendorf (2002) and Krippendorff (2012)), which results in an identification of a total of 179 main attributes of machine scheduling problems in production that can be allocated to 7 groups and 48 subgroups. The reliability of the results of the content analysis is then examined in a sensitivity analysis. Finally, a close analysis of the results unveils several research gaps in the literature and enables us to propose promising avenues for future research.

Paper#	Focus of the paper	Authors	Title	Journal	Status
1	Scheduling in Production (Part A)	H. Abedinnia C. H. Glock E. H. Grosse M. Schneider	Machine scheduling problems in production: A tertiary study	Computers & Industrial Engineering	Published
2		H. Abedinnia C. H. Glock M. Schneider	Machine scheduling in production: A content analysis	Applied Mathematical Modelling	Published
3		H. Abedinnia C. H. Glock A. Brill	New simple constructive heuristic algorithms for minimizing total flow-time in the permutation flow shop scheduling problem	Computers & Operations Research	Published
4	Scheduling in Logistics (Part B)	S. Emde H. Abedinnia C. H. Glock	Scheduling electric vehicles making milk-runs for just-in-time Delivery	IIE Transactions	Accepted
5		S. Emde H. Abedinnia A. Lange C. H. Glock	Scheduling personnel for the build-up of unit load devices at an air cargo terminal with limited space	OR-Spectrum	Under review
6		C. H. Glock E. H. Grosse H. Abedinnia S. Emde	An integrated model to improve ergonomic and economic performance in order picking by rotating pallets	European Journal of Operational Research	Under review

Table 2: Overview of the papers included in this cumulative dissertation

Paper 3 develops a set of new simple constructive heuristic algorithms to minimize total flow-time, i.e., the summation of completion times of all jobs, for an n-jobs x m-machines permutation flowshop scheduling problem. A flowshop production system is commonly defined as a production system in which a set of n jobs undergoes a series of operations in the same order (Pinedo, 2008). Flowshop scheduling problems are in most cases proven to belong to the class of NP-hard problems. This is even the case for permutation flowshop scheduling problems, i.e. for flowshop scheduling problems with the same job order on all machines. The heuristic algorithms proposed in this paper are based on the popular simple heuristic presented by Nawaz et al. (1983), which is known as NEH in the literature. The NEH heuristic consists of two phases, namely (I) the sorting (prioritizing) phase and (II) the insertion phase. In the sorting phase, the jobs are sorted in descending order of their total processing time. This sorted list is then used in the insertion phase to determine the sequence in which jobs are added to an existing partial sequence. Based on a considered decision criterion, in each iteration, the best partial sequence will be selected for the next iteration. In Paper 3, we first propose a modification for the insertion phase of NEH (which is actually a smart neighborhood search) and then integrate new indicator variables for weighting jobs into this algorithm. We also propose new decision criteria to select the best partial sequence in each iteration of our algorithm. A comprehensive numerical experiment reveals that our modifications and extensions improve the effectiveness of the best existing simple heuristic without affecting its computational efficiency by 24%.

In Paper 4, we modeled the problem of assigning a set of timetabled milk-run trips to a fleet of electric vehicles such that battery capacities are not exceeded, the fleet size is minimal, and fairness, which is measured by calculating the difference of workload on the most and least busy driver, is maximal. Battery-operated electric vehicles are frequently used in in-plant logistics systems to feed parts from a central depot to work cells on the shop floor. These vehicles, often so-called tow trains, make many milk-run trips during a typical day, with the delivery timetable depending on the production schedule. To operate such a milk-run delivery system efficiently, not only do the timetabled trips need to be assigned to vehicles, it is also important to take the limited battery capacity into consideration. Moreover, since most tow trains in use today are still operated by human drivers, fairness aspects with respect to the division of the workload also need to be considered. In this context, we tackle the following problem that we encountered at a large manufacturer of engines for trucks and buses in Germany. Given a fixed schedule of milk-runs (round trips) to be performed during a planning horizon and a fleet of homogeneous electric vehicles stationed at a depot, which vehicle should set out on which milk-run and when should recharging breaks be scheduled, such that all runs can be completed with the minimum number of vehicles and ensuring all vehicles are approximately equally busy? The computational complexity of this problem, as well as the complexity of some important sub-problems (such as the case in which battery restriction is relaxed), are investigated in Paper 4. A constructive heuristic algorithm has been proposed, which can solve the problem in pseudo-polynomial time. Although this heuristic is quite fast (it solves large instances in split seconds), the solution quality, especially with regard to fairness, is suboptimal. To overcome this issue, Paper 4 also suggests a solution approach based on taboo search, which is capable of solving realistic instances to near-optimality in less than two minutes. Comprehensive computational experiments conducted in this paper enabled us, first, to examine the effectiveness as well as the efficiency of the solution approaches proposed, and then, to offer some insight into how battery technology influences vehicle utilization.

Paper 5 investigates a problem we encountered at a terminal of a major German airfreight carrier and addresses the preparation of unit load devices (ULD), which later have to be loaded onto aircrafts. This process is difficult to plan for many airlines, which face the challenge of assigning a limited number of workers to a limited number of workspaces available for preparing the ULD, while respecting the requirements imposed by an existing flight schedule. While preparing ULD, the objectives are to comply with the flight schedule, not to exceed the available space at the terminal, and to minimize the maximum workforce employed over time. To support airlines in realizing efficient ULD preparation processes, this paper proposes a mixed-integer programming model as well as a generalized set partitioning reformulation of this problem. Based on the latter formulation, we develop different heuristic strategies, some of which are shown to solve this NP-hard problem to near-optimality (using the right parameter settings for the proposed heuristic, our heuristic performs very well, delivering optimality gaps well below 1% on average) in a matter of merely 10 seconds. (Near-)optimal schedules as obtained by our heuristics are significantly better at avoiding large peak workloads than the simple rule of thumb we encountered in practice. On average, the required peak workforce could be more than halved. We also investigate the inherent tradeoff between labor and space utilization as well as the effect of uncertainty about the volume of the cargo to be shipped.

Paper 6 studies manual order picking activities in a U-shaped warehouse where items are stored on pallets in two rows one above the other, inspired by a situation observed in practice. Picking products directly from pallets renders order picking a high-risk environment for developing musculoskeletal disorders due to the required handling of heavy loads and continuous bending, stretching and lifting. The challenge that arises for warehouse managers in this case is to organize the order picking process as efficiently as possible, simultaneously keeping in mind health and safety issues. The first aspect can be addressed by planning short order picking routes and implementing storage assignment methods that allocate frequently required items close to the depot of the warehouse. Worker well-being, in turn, can be improved by searching for opportunities to reduce the load on the warehouse worker. Our observations revealed that, in many cases, picking an item from the back side of a pallet led to excessive load on the order picker compared to picking the same item from the front side of the pallet. Therefore, Paper 6 investigated the case where the company has the opportunity to rotate pallets once their front part has been depleted, which helps to reduce the extent of bending and stretching required on the part of the order picker, and therefore the load on the worker's spine. In this way, a biomechanical model was developed to measure the peak L4/L5 spinal compression that acts on the order picker during the picking of items as an ergonomic objective. In addition, as an economic objective, the total order picking time was considered. A mathematical model was also proposed for sequencing orders, routing the order picker through the warehouse, and scheduling pallet rotation tours. The developed model allows studying the impact of rotating pallets on two different measures: order picking time and peak spinal load on the order picker. The results of a numerical experiment indicate that selectively rotating pallets may both reduce order picking time as well as the load on the order picker, leading to a quicker and less risky order picking process. The model proposed in this paper supports the decision of which pallet to rotate (and which not to rotate) against the company's cost objectives and its strive for worker well-being. Several managerial insights, limitations of our study, as well as several suggestions for further research are also discussed.

Overall, this thesis adds to the literature two systematic literature reviews on the state-of-research of machine scheduling problem in production (Paper 1 and Paper 2), one decision support model to sequence jobs in a classic production system (Papers 3), and three effective algorithms to support decision making in scheduling problems encountered in planning intra-plant logistics activities (Paper 4 to 6). Despite the scientific character of the papers, the literature synthesis and the comprehensive computational experiments conducted clearly underscore the practical applicability of the reviewed and developed decision support models and hence their relevance for practitioners. Detailed managerial insights together with limitations of the chosen research approaches and future research opportunitiesm can be found in the final section of each paper.

References

Behnamian, J., F. Ghomi, S.M.T., 2014. A survey of multi-factory scheduling. Journal of Intelligent Manufacruring. 27, 1–19.

Brucker, P., Drexl, A., Möhring, R., Neumann, K., Pesch, E., 1999. Resource-constrained project scheduling: Notation, classification, models, and methods. European Journal of Operational Research. 112, 3–41.

Brucker, P., Sotskov, Y.N., Werner, F., 2007. Complexity of shop-scheduling problems with fixed number of jobs: a survey. Mathematical Methods of Operations Research. 65, 461–481.

Chen, B., Lee, C.Y., 2008. Logistics scheduling with batching and transportation. European Journal of Operational Research. 189, 871–876.

Garousi, V., Mäntylä, M. V., 2016. A systematic literature review of literature reviews in software testing. Information and Software Technology. 80, 195–216.

Gorman, M.F., 2016. A "Metasurvey" analysis in Operations Research and Management Science: A survey of literature reviews. Surveys in Operations Research and Management Science. 21, 18–28.

Hall, N.G., Sriskandarajah, C., 1996. A survey of machine scheduling problems with blocking and no-wait in process. Operations Research. 44, 510–525.

Herroelen, W., Demeulemeester, E., De Reyck, B., 1999. A classification scheme for project scheduling, in: International Series in Operations Research & Management Science, Vol 14. Springer, Boston, MA, Boston.

Hochrein, S., Glock, C.H., Bogaschewsky, R., Heider, M., 2015. Literature reviews in supply chain management: a tertiary study. Management Review Quarterly. 65, 239–280.

Kache, F., Seuring, S., 2014. Linking collaboration and integration to risk and performance in supply chains via a review of literature reviews. Supply Chain Management: An International Journal. 19, 664–682.

Krippendorff, K., 2012. Content Analysis: An Introduction to Its Methodology, third ed., SAGE Publications, Inc.

Lenstra, J.K., Kan, A.H.G.R., Brucker, P., 1977. Complexity of machine scheduling problems. Annals of Discrete Mathematics. 1, 343–362.

Levner, E., Kats, V., Alcaide López de Pablo, D., Cheng, T.C.E., 2010. Complexity of cyclic scheduling problems: A state-of-the-art survey. Computers & Industrial Engineering. 59, 352–361.

Nawaz, M., Enscore, Jr.E.E., Ham, I., 1983. A heuristic algorithm for the m-Machine, n-Job flow-shop sequencing problem. Omega. 11, 91–95.

Neuendorf, K.A., 2002. The Content Analysis Guidebook., SAGE Publications, Inc.

Panwalkar, S.S., Smith, M.L., Koulamas, C., 2013. Review of the ordered and proportionate flow shop scheduling research. Naval Research Logistics. 60, 46–55.

Pinedo, M.L., 2008. Scheduling Theory, Algorithms, and Systems, Third. ed. Springer.

Sanlaville, E., Schmidt, G., 1998. Machine scheduling with availability constraints. Acta Informatica. 35, 795–811.

Seuring, S., Gold, S., 2012. Conducting content-analysis based literature reviews in supply chain management. Supply Chain Management: An International Journal. 17, 544–555.

Singh, A., Goyal, P., Batra, S., 2010. An optimized round robin scheduling algorithm for CPU scheduling. International Journal on Computer Science and Engineering. 2, 2383–2385.

Yuan, W., Nahrstedt, K., 2003. Energy-efficient soft real-time CPU scheduling for mobile multimedia systems. ACM SIGOPS Operating Systems Review. 37, 149–163.

Part A Machine scheduling in production

Paper 1 Machine scheduling problems in production: A tertiary study

Authors: Hamid Abedinnia, Christoph H. Glock, Michael Schneider, Eric H. Grosse

Type of publication: Journal article

Publication details: *Computers & Industrial Engineering*, Volume 111, September 2017, Pages 403-416

Abstract

This paper presents the results of a comprehensive and systematic review of 129 literature reviews on machine scheduling problems in production (MSPP). The paper first proposes a conceptual framework that considers the main attributes of MSPP in 7 categories and 75 sub-categories. After a descriptive analysis of the sampled papers that give insights into publication patterns for MSPP, a quantitative analysis of the sampled review papers is carried out based on the proposed framework. A synthesis of research findings describes the state of knowledge and unveils general deficiencies of literature reviews on MSPP. In addition, the paper provides a comprehensive overview of MSPP, which supports researchers in positioning their own work in the literature and in finding potential innovative research areas. Based on a discussion upon the statistical findings, opportunities for future research on MSPP are proposed.

Keywords: *systematic literature review, review of reviews, scheduling in production, machine scheduling*

1 Introduction

1.1. Machine scheduling problems in production

Scheduling can generally be defined as the process of assigning restricted resources to a set of tasks that need to be accomplished. Scheduling is a relevant problem in many different areas, e.g. in project management (Leyman & Vanhoucke, 2015; Nkasu & Leung, 1997), software optimization (Li, Singhoff, Rubini, & Bourdellès, 2016), and personnel management (Cochran, Chu, & Chu, 1997; Prot, Lapègue, & Bellenguez-Morineau, 2015). Among the most prominent and important research fields in scheduling are production systems (Pinedo, 2008), which is also the focus of the paper at hand. In many production systems, jobs (that represent tasks) need to be processed on machines (that represent resources). The aim of machine scheduling problems in production (MSPP) is to find a sequence of jobs to be processed on machines in a way that optimizes a set of objective(s) without violating any of the constraints (Graves, 1981; Sen & Gupta, 1984). Even in modern production systems, such as in semiconductor manufacturing plants, scheduling techniques play an important role in reducing idle times, speeding up the production process and reducing cost by improving operational processes (Mönch, Fowler, Dauzère-Pé rès, Mason, & Rose, 2011).

As MSPP need to be solved in almost any production system to plan operational activities, related solution approaches are broadly applicable in practice (Tuncel & Bayhan, 2007). Solving MSPP is challenging in most cases, however, as modifying one simple assumption often leads to a new problem that requires new solution approaches. This fact renders MSPP not only a challenging problem for practitioners, but also a popular research topic for academia. It is thus not surprising that MSPP belong to the most frequently studied optimization problems in management and engineering. Simple database searches may illustrate the scope of this research stream: The keyword combination ‘‘scheduling” and ‘‘production”, for example, leads to about 1,750,000 hits in Google Scholar and 48,000 hits in Business Source Premier[1], which gives an impression of the high number of publications in this area.

The continuous high publication output on MSPP makes it necessary to regularly synthesize and consolidate research topics and findings to give researchers and practitioners an overview of the existing state-of-knowledge and to identify research gaps that could be addressed in future research efforts. This general understanding already inspired many researchers to review specific sub-domains of this research field (see, for example, the metasurvey of Gorman (2016) on literature reviews in operations research and management science). Specific MSPP literature reviews help researchers to gain insights into the topic covered by the review, but they may also contribute to a loss of overview of the research domain itself (here: MSPP), whose state-ofknowledge may be scattered over a large number of specific review papers. As the high number of published works on MSPP prohibitsa single review that covers the entire (primary) literature, one established way to synthesize research in this area is to conduct a tertiary study on a comprehensive sample of review papers (secondary works) on MSPP. The next section gives an overview of tertiary studies and their contribution to the literature and outlines the contribution of the present paper to the literature on MSPP. Section 1.3 then outlines the organization of this paper.

[1] Numbers effective March 2017

1.2. Tertiary study

Research streams that enjoyed a high publication output in the past often suffer from the fact that the high amount of published research makes it very difficult (if not impossible) to maintain an overview of the entire domain. For the same reason, reviewing the entire domain in a single literature review is often prohibitive. Tertiary studies (i.e., reviews of literature reviews) may support structuring and synthesizing a research area in this case, as their object of analysis are (fewer) literature reviews instead of a prohibitively large number of primary research papers. The primary objective of tertiary studies is to investigate core themes that have been studied in a particular research area by reviewing and analyzing secondary works (i.e., literature reviews). The aims of tertiary studies are to (I) give an aggregated overview of a research domain, to (II) analyze research trends in the domain of interest, to (III) evaluate the methodological rigor of literature reviews in the domain, and to (IV) identify research gaps (on the secondary and, if possible, also on the primary level). Tertiary studies usually apply a systematic literature review to a sample of literature reviews, and they are an established research methodology in many different areas including operations and production management (e.g., Bushuev, Guiffrida, Jaber, & Khan, 2015; Glock, Grosse, & Ries, 2014), supply chain management (e.g., Hochrein & Glock, 2012; Hochrein, Glock, Bogaschewsky, & Heider, 2015; Kache & Seuring, 2014; Seuring & Gold, 2012), or software engineering (e.g., Garousi & Mäntylä, 2016). Tertiary studies provide a compact and comprehensive overview of the state-of-knowledge in a specific research area, and unveil general deficiencies of published literature reviews on the subject under consideration. Tertiary studies are thus valuable sources for finding potential areas for future research.

In the area of MSPP, a prohibitively large number of primary works and a high as well as an increasing number of secondary research motivated the tertiary study at hand. The tertiary study enables us to analyze the entire domain of MSPP, which would not be possible in a regular literature review that analyzes primary works (Garousi & Mäntylä, 2016; Hochrein et al., 2015). The contribution of an easy-to-understand but comprehensive overview of the vast research field of MSPP provided by our tertiary study is manifold. First, our paper gives a broad overview of the research field of MSPP and synthesizes findings that were obtained in literature surveys covered in our sample. As a review of the MSPP on the primary level is not possible due to the massive number of papers that have been published on this topic, only a tertiary study is able to review this research stream in such a broad manner. To the best of our knowledge, our paper is the first and only work that applies a systematic tertiary analysis to MSPP, so it is the only paper that gives such a broad overview of research on MSPP. Secondly, we develop a content-related and technical classification framework for MSPP based on an in-depth analysis of the sampled review papers. This framework for classifying MSPP (Section 3) can be seen as a synthesis of the different classification schemes for MSPP applied or derived in the sampled literature reviews. Thirdly, the content-related analysis (Section 4.3.2) illustrates major topics and applications that were discussed in the sampled review papers, which helps readers in gaining insights into the topics that were emphasized by prior research (and, in return, into topics that did not receive much attention so far). This, in turn, assists readers in identifying possible avenues for future research or positioning their own work in the existing literature. For readers interested in conducting a secondary study on their own, this tertiary study helps to identify areas where a new or an initial literature review is required. The latter aspect is supported in detail in our content discussion that identifies topics where secondary studies are required in the future. Finally, this tertiary study could also serve as a guideline for the application of systematic review techniques in the area of MSPP, which is an aspect that is of increasing relevance in the scientific literature.

1.3. Organization of the paper

To accomplish the objectives outlined in Section 1.2, this paper, first, generates a sample of literature reviews on MSPP in a systematic search of the literature (Section 2). Subsequently, the paper proposes a comprehensive conceptual framework that reflects the main characteristics of MSPP in seven categories and 75 subcategories (Section 3). The framework is used to evaluate literature reviews in this area. The methodology of the tertiary study is explained in detail in Section 4.1. After a descriptive analysis of the sample that gives insights into publication patterns (Section 4.2), a quantitative and content-related analysis of the sampled review papers is carried out (Section 4.3). This step contains an evaluation of the review methodology as well as a content examination of the sampled review papers based on the proposed framework to identify the most popular streams of research on MSPP. Finally, the study identifies methodological drawbacks of existing literature reviews on MSPP and highlights areas where future research might be promising (Section 5). Section 6 concludes the paper. Fig. 1 illustrates the steps of the tertiary study.

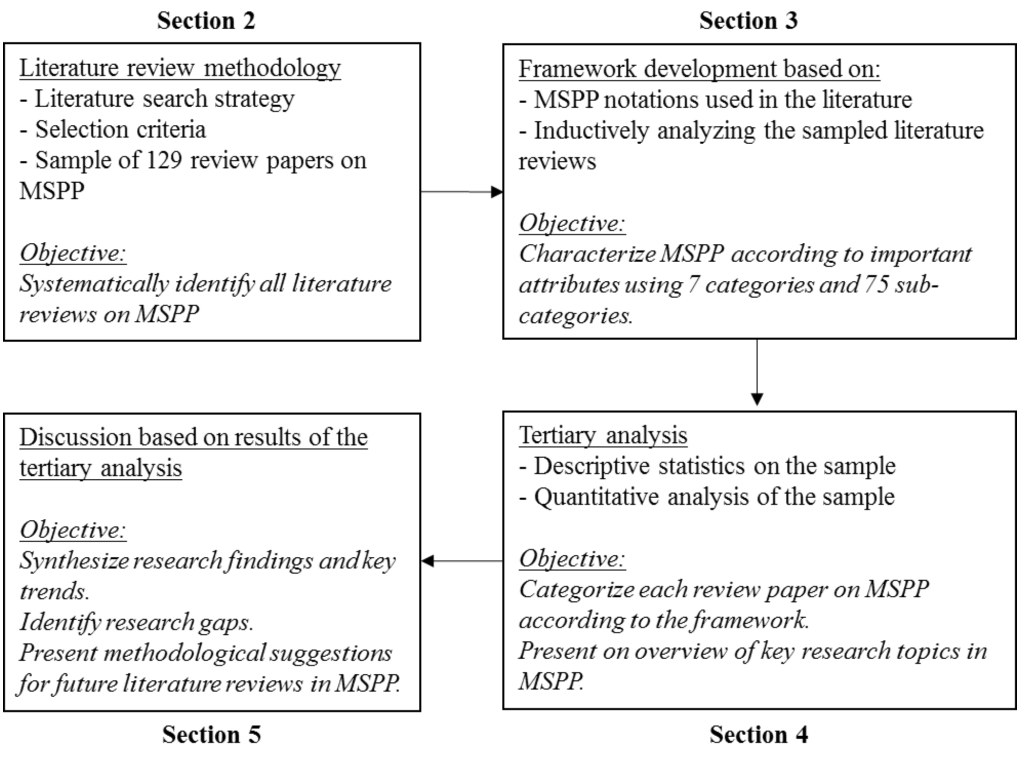


Figure 1 Methodology of the tertiary study

2 Review methodology

2.1. Literature search strategy

Tertiary studies require a rigorously developed literature sample to ensure that readers are able to reproduce sample generation and evaluation. As a result, tertiary studies require a systematic, well-structured and documented search of the literature (Hochrein et al., 2015). In the following, we describe the search strategy that was used to identify literature reviews on MSPP in detail. The literature search was conducted in November 2016. In a first step, keywords were defined that were later used to identify relevant works in the literature. The final keyword list was generated using three groups of keywords, where group A contained keywords related to scheduling (‘‘scheduling” and ‘‘sequencing”), group B keywords that limited the search to MSPP (‘‘machine”, ‘‘shop”, ‘‘manufacturing”, ‘‘production”, ‘‘process”, ‘‘flow shop”, ‘‘job shop”, and ‘‘open shop”), and group C keywords related to literature reviews (‘‘survey”, ‘‘review”, ‘‘overview”, ‘‘taxonomy”, and ‘‘trends”). The keyword ‘‘shop” was included in group B to ensure that different spellings of shop-related production systems are covered (e.g., ‘‘flow shop” is often also referred to as ‘‘flowshop” or ‘‘flow-shop”). The final keyword list was generated by combining all keywords from the three initial lists. The database Scopus was then searched for works that contain keywords from the final keyword list either in their title, abstract or list of keywords. In this step, 7253 papers were identified. The database search was complemented by a forward and backward snowball search, where the references of papers contained in the sample were checked, and where works that cited papers contained in the sample were evaluated for possible relevance (Hochrein & Glock, 2012). In this step of the literature search, 28 additional articles were added to the sample.

2.2. Selection criteria

After an initial sample had been generated based on the database and snowball searches, all pre-selected works were independently checked for relevance by three researchers. Works were included in the final sample based on the following requirements:

- The language was limited to English.
- Only works that appeared in peer-reviewed, high-quality journals were considered, and book chapters and conference papers were excluded from the analysis. To assess the quality and reputation of a journal, the SNIP and the SJR metrics were used, and only works published in journals with SNIP≥0.87 and SJR≥0.51 were considered relevant (see also González-Pereira, Guerrero-Bote, & Moya-Anegón, 2010; Moed, 2010). Two journals, for which the SNIP and SJR metrics were not available, were manually added to the list of relevant journals (i.e. *Annals of Discrete Mathematics and Production and Inventory Management*).
- Only works that review the literature on MSPP (secondary studies) were considered. Works that compare methods for a specific scheduling problem or that propose models or solution procedures (even if they contain a section on reviewing the literature) were excluded from the sample.
- To keep the tertiary study focused on MSPP, we excluded works on project scheduling, personnel scheduling, timetabling, scheduling in agriculture, and computer scheduling.

Applying the defined inclusion and exclusion criteria to the initial sample based on an analysis of title and abstract reduced the size of the sample from 7281 to 230 articles. In a final step, all papers that remained in the sample were completely read to assess their relevance in light of the defined criteria. This step led to a final sample of 129 relevant review papers.

3 MSPP: a framework

This section proposes a conceptual framework to categorize the literature on MSPP. The framework can be seen as an extension of the 3-field notation of Graham, Lawler, Lenstra, and Kan (1979), which characterizes scheduling problems according to machine environment, constraints and objectives. To capture all characteristics of MSPP, an inductive approach was used where all sampled papers were carefully read to identify important characteristics of MSPP. Based on this analysis, a comprehensive framework containing 75 different attributes of MSPP (referred to as subcategories hereafter) divided into seven categories was developed. The proposed framework facilitates the analysis of literature reviews on MSPP and allows a structured and rigorous examination of this research field. In the following, we briefly introduce each category of the framework (see also Table 1 in Section 4.3.2, which contains a list of all categories and sub-categories).

Type of problem

To classify scheduling problems, Nagar, Haddock, and Heragu (1995) proposed the *type of problem* as a categorization criterion that summarizes structural properties of the problem at hand. We use the category *type of problem* to classify problems as stochastic or deterministic, static or dynamic, and offline or online. Note that prior research has often used the terms static/offline and dynamic/online interchangeably; however, as some authors see a difference in this terminology (e.g., Lee, Leung, & Pinedo, 2010), our framework considers them separately.

Theory of complexity

The category *theory of complexity* is used to evaluate if reviewed papers explicitly report complexity results for the considered problem or not.

Practical application of scheduling

Many researchers pointed out a gap between the theory of scheduling and its application in practice (e.g., MacCarthy & Liu, 1993). As Sabuncuoglu (1998) noted, the ‘‘well established and rich body of scheduling theory has scarcely been used in practice”. Thus, our framework evaluates if the sampled papers focus on exploring and bridging the gap between scheduling in theory and in practice. In addition, we also examine the application of MSPP in modern manufacturing systems, such as flexible manufacturing systems (FMS), robotic and semiconductor manufacturing, and wafer fabrication.

Solution approaches

The category *solution approaches* summarizes the methods used to solve specific MSPP, and it is a classification criterion that has frequently been used in the past (e.g., Reza Hejazi & Saghafian, 2005). In the following, we distinguish between *exact*, *approximation* (i.e., worst-case ratios), *heuristic* and *metaheuristic* algorithms as well as algorithms based on *artificial intelligence* (AI) and *simulation*. We note that some researchers categorize AI-algorithms (such as expert systems or artificial neural networks) as metaheuristics (e.g., Reza Hejazi & Saghafian, 2005), while others distinguish between these two (e.g., Ouelhadj & Petrovic, 2009). To be compatible with both interpretations, our framework considers them separately.

Constraints

The *constraints* considered in MSPP influence the applicability and the complexity of the corresponding models, and they have frequently been used as classification criteria in surveys on MSPP (e.g., Dileepan & Sen, 1988). This category considers all constraints used for MSPP, such as setup (or change-over), machine breakdown, pre-emption and precedence constraints.

Objectives (performance criteria)

The objective of MSPP models is usually to find a sequence of jobs to be processed on machines that optimizes either a single or a set of performance criteria. In addition, the objective of some problems is to determine due dates for jobs, which are commonly referred to as *due date assignment* problems (see e.g., Gordon, Proth, & Chu, 2002). Performance criteria, which measure the quality of the solutions, can be *cost-based* (e.g., total setup cost, total inventory cost, total resource cost), *penalty-based* (e.g., total tardiness/earliness, number of tardy jobs, maximum lateness), or based on *throughput time* (Kiran & Smith, 1984). We split up the performance criterion *throughput time* into *makespan* (defined as completion time of the last job) and *flow time* (defined as the difference between completion time and the release time of the jobs).

Machine configuration (flow pattern)

The *machine configuration* (or *flow pattern*) describes how jobs are routed through the production system as well as the configuration of the machines on the shop floor (Graham et al., 1979). This category evaluates if a single- or a multi-stage production system is considered and how many machines are available on each stage (*single machine* vs. *parallel machine* problems). For multi-stage problems, the *machine configuration* distinguishes between several options for routing jobs through the system, namely (a) *flow shops* (routing is predetermined and identical for all jobs), (b) *job shops* (predetermined, but individual routings for each job), and (c) *open shops* (no limitation on routing; a job can be processed in any sequence). For more details on different machine configurations for MSPP, see for example Pinedo (2008). Early research on scheduling problems often used on a discrete-time formulation that divides the time horizon into a number of intervals of equal duration (Li & Ierapetritou, 2008). The category *machine configuration* therefore also captures if the continuous- or discrete-time formulation is used for the problem (*process scheduling*). It further determines if inventory-related costs are considered in the problem (such problems are commonly referred to as *lot-streaming* problems), and if the problem under consideration contains a *cyclic* production system or not.

4 Tertiary analysis

4.1. Methodology

The methodology of our tertiary analysis consists of three steps. First, descriptive statistics of the sample including article count per year, per journal and per author are presented (Section 4.2). Secondly, the sample is evaluated with respect to type of literature review and the methodology employed (Section 4.3.1). Thirdly, a quantitative and content-related analysis is conducted to classify each sampled literature review according to the developed framework (Section 4.3.2).

The literature generally differentiates between three types of literature reviews: (a) narrative reviews, (b) systematic reviews, and (c) meta-analyses (Cooper, 2010; Hochrein & Glock, 2012; Hochrein et al., 2015). Narrative reviews usually do not describe how the sample was developed and/or do not document the literature search process in a systematic way. The lack of transparency in the literature search and selection phase makes it difficult or even impossible to reproduce the findings of narrative literature reviews, and it also makes it difficult to assess if the results of the review hold for the research field per se, or only for a selection of papers reviewed from this field. Systematic reviews, in contrast, employ a reliable and reproducible methodology to generate the literature sample. Meta analyses extract data from a literature sample that was systematically developed and analyze the sample using statistical techniques. In Section 4.3.1, we analyze our sample in light of the type of literature review and examine if the keywords used in retrieving the sample, sample size (i.e., number of papers reviewed), and coverage (i.e., the time period considered) are reported in the sampled papers. This approach enables us to evaluate the methodological quality of the literature reviews, as clearly documenting the literature search is vital to ensuring transparency, reproducibility and rigor.

As was stated before, the sample analyzed in this tertiary study consists of 129 literature reviews. Given that each of the sampled literature reviews on its part analyzes samples of primary studies that were in several cases quite comprehensive, discussing all 129 literature reviews in the work at hand is prohibitive. Instead, we decided to analyze the sample on a meta level using a quantitative analysis that provides compact results and that enables us to keep the length of the paper at hand within reasonable limits (results are presented in Section 4.3.2). A second challenge we faced when classifying the sampled papers was that many papers covered more than a single sub-category, with the extent of coverage varying from sub-category to sub-category. To correctly reflect the content of the sampled papers in the assignment of the papers to our framework, we considered four 'degrees of coverage' for each sub-category in classifying papers, namely a) *clearly focused* (3), b) *relatively focused* (2), c) *covered without special focus* (1), and d) *not mentioned* (0). We used the scores given in brackets to code the coverage. All sampled review papers were categorized by each author of this paper individually to ensure intercoder reliability. Only a few papers were categorized differently in the first step. These papers were discussed in detail within the research team to arrive at a consensus. The four degrees of coverage can be explained as follows:

- Papers are classified as *clearly focused* on a specific sub-category if the sub-category is mentioned in the title of the paper, or if it becomes apparent from the title, abstract, list of keywords or introduction that the review in question is dedicated to the respective sub-category.
- A paper was classified *as relatively focused* on a specific sub-category if special attention is given to the respective sub-category in the entire paper, but if no clear focus on the sub-category could be identified. This includes reviews that use the sub-category in question to classify the primary works discussed in the review, or reviews that are clearly focused on another sub-category, but that also devote a single or a few section(s) to the sub-category in question.
- If a specific sub-category is only mentioned in the review without discussing primary works on this sub-category in a separate section, the paper is labelled as *covered without special focus*.
- The sub-categories that are not covered in the paper are labelled as *not mentioned*.

Our quantification procedure is illustrated for the review of Baker and Scudder (1990) in the following: First, it becomes apparent from the title that the main focus of the paper is on reviewing penalty-based objectives. Thus, we assign the score 3 to the subcategory ‘‘*penalty-based objectives*” from the category ‘‘*objective*”. Baker and Scudder (1990) also reviews both dynamic and static studies. As a consequence, we assign the score 3 to the subcategories ‘‘*dynamic*” and ‘‘*static*” from the category ‘‘*type of problem*”. The authors further divided the sampled papers into ‘‘*single machine*” and ‘‘*parallel machine*” problems, which concurs with our definition of *relatively focused* reviews. As a result, the subcategories ‘‘*single machine*” and ‘‘*parallel machines*” of the category ‘‘*machine configuration*” are weighted with 2. Although the main focus of the exemplary review is on studies with penalty-based objectives, some multi-objective studies, which consider objectives based on flow time, are included as well. Thus, for the category ‘‘*objective*”, we assign the score 1 to the sub-categories ‘‘*multi-objective*” and ‘‘*flow time*”. Other sub-categories that are not considered in Baker and Scudder (1990), such as *setup* or *buffer space* constraints, are labelled as *not mentioned* and their score in the framework is 0.

An overview of all categorized reviews in our sample according to this methodology is given in Tables 2a–2e in Section 4.3.2.

4.2. Descriptive results

This section presents a descriptive analysis of the sample with respect to the number of sampled reviews per year, per journal and per author (see for a similar analysis Gorman, 2016).

4.2.1. Article count by year

The 129 articles contained in the final sample were published between 1959 and November 2016 in 42 different scientific journals (a comprehensive list of papers included in the sample can be found in Appendix A, where each paper is labelled with a number to facilitate referencing). The number of sampled papers published over time is presented in Fig. 2. As can be seen, reviewing the literature on MSPP has attracted a continuously increasing attention over the last decades, where 53 of the sampled surveys (i.e., around 41% of the whole sample) were published during the last ten years. These results clearly underline the ongoing significance of MSPP in academic research.

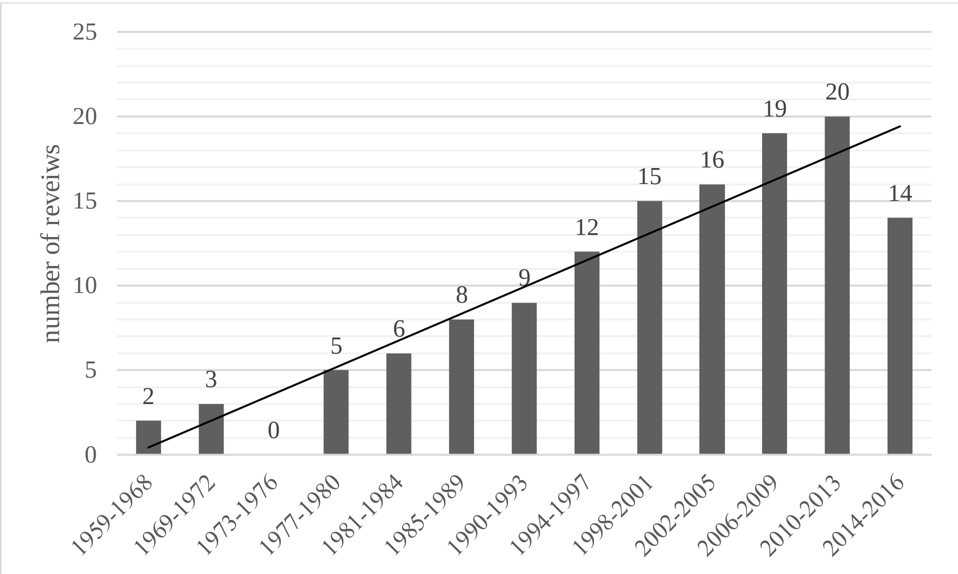


Figure 2 Number of surveys published on MSPP over time

4.2.2. Article count by journal

Fig. 3 summarizes the distribution of survey papers across the ten scientific journals that published the highest number of literature reviews on MSPP. These 10 journals published more than 66% (i.e., 86 surveys) of the sampled papers. The *European Journal of Operational Research* (23), the *International Journal of Production Research* (10), *Operations Research* (9), and *Omega* (9) were the four most popular outlets for literature reviews on MSPP, and together they published around 40% of the sampled papers.

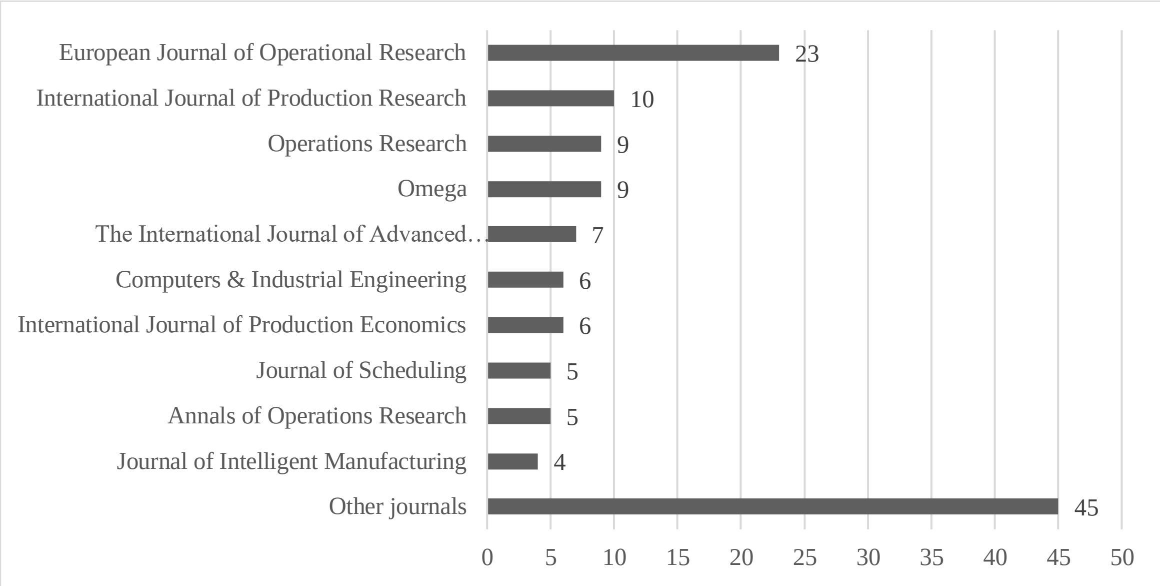


Figure 3 Number of published surveys on MSPP per journal

4.2.3. Article count by author

The analysis of the sample shows that 226 different authors contributed to the publication of the 129 sampled review papers. Fig. 4 ranks all authors who published three or more literature reviews on MSPP according to the number of reviews they have authored.

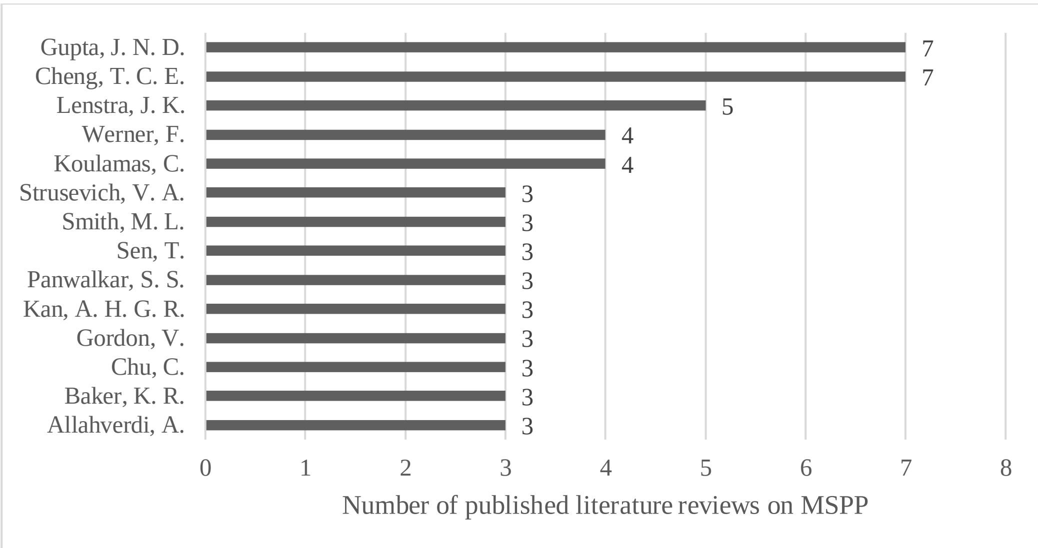


Figure 4 Authors that published at least three of the sampled literature reviews on MSPP

4.3. Data evaluation

This section presents a quantitative analysis of the sampled papers. We first examine the type of literature review by evaluating their research methodology and then evaluate the content of the review papers quantitatively according to the developed framework in Section 3.

4.3.1. Type of literature review

Our analysis shows that the majority of the sampled literature reviews on MSPP are narrative (non-systematic) reviews according to the definition provided in Section 4.1. The first systematic literature review on MSPP appeared as late as 2013. Only four literature reviews contained in the sample can be classified as systematic ([113], [122], [127] and [129], see Appendix A), and only a single meta-analysis exists [44]. Only 25 reviews in our sample reported the sample size (which is 105.24 primary works on average in these reviews). The coverage, i.e. the timeframe of the primary studies included in the review paper was also reported in only 27 of the sampled papers. Only three reviews ([122], [127] and [129]) reported the keywords used in retrieving their sample and, finally, only 12 papers analyzed it statistically. Further implications of the results regarding the type of literature review are discussed in Section 5.

4.3.2. Quantitative analysis

The total score obtained for each category after assigning weights to the sub-categories according to the methodology outlined in Section 4.1 and the relative coverage of each category in the sample (i.e., the percentage of sampled reviews that has at least weight 1 for each category) are presented in Table 1.

Due to the high number of sub-categories (in total 75, see Table 1), a comprehensive definition and discussion of each single sub-category is not within the scope of this paper. The reader is instead referred to the reviews tagged as *clearly focused* and *relatively focused* in Tables 2a–2e that contain comprehensive discussions of the respective sub-categories. Tables 2a–2e also summarize the total score obtained and reports the overall rank of each sub-category among all 75 sub-categories. As can be seen, the ten sub-categories that obtained the highest scores belong to the categories ‘‘*flow pattern*” (i.e., single machine, flow shop, parallel machines, and job shop), ‘‘*objective*” (i.e., penalty, makespan and flow time), ‘‘*solution approaches*” (i.e., heuristic algorithms & dispatching rules, and exact algorithms) and ‘‘*type of problem*” (i.e., deterministic).

Category	Sub-categories	Total number of sub-categories	Total score	Coverage in the sample
1) Type of problem	dynamic, static, deterministic, stochastic, online, offline	6	321	65.11%
2) Theory of complexity	theory of complexity	1	89	50.38%
3) Practical application of scheduling	lack of application in scheduling theory, industrial applications of scheduling theory, modern production systems (FMS, semiconductor manufacturing, wafer fabrication, robotic manufacturing)	6	159	62.01%
4) Solution approaches	exact algorithms, heuristic algorithms & dispatching rules, metaheuristic algorithms, artificial intelligence, simulation, approximation algorithms, specific solution approach (hyper heuristics, robust scheduling, petri nets, decomposition-based algorithms, artificial neural networks, TSP-based approaches, mixed-integer programming models, tabu search, Johnson's rule, expert systems)	16	566	84.49%
5) Constraints	dual resource operations, ordered operations, 1-job-m-machine problems, human factors, time-dependent processing time, inserted idle-time, interval scheduling problems, fixed number of jobs with various number of operations, learning effects, processing set restrictions, machine eligibility constraints, distributed scheduling, equal processing times, positional effects, remanufacturing scheduling, group scheduling, batch machines, scheduling with job rejection, transportation lags, rescheduling, machine breakdown, common due-dates, precedence constrains, preemptive scheduling, buffer space constraints, no-wait operations, setup	27	574	92.24%
6) Objectives	cost, penalty, objectives based on throughput time (makespan, flow time), multi-objective problems, due-date assignment, minimum number of machines	7	536	91.47%
7) Flow pattern	single machine, parallel machines, flow shop, job shop, open shop, mixed shop, hybrid flow shop, hybrid job shop, dedicated machines, cyclic production, process scheduling, lot streaming	12	526	94.57%
	total	75	2771	

Table 1: Framework with categories, sub-categories, total score obtained and coverage

Category	Sub-category		Cleary focuced literature reviews	Relatively focuced literature reviews	Total score	Percentage of the respective category's total score %	Rank in the respective sub-category	Overall rank among all 75 sub-categories
1) Type of problem	deterministic		[2], [5], [10], [25], [29], [33], [35], [37], [39], [42], [48], [49], [52], [56], [57], [59], [61], [62], [88], [99], [104], [114], [115]	[1], [17], [68], [76], [80], [94], [107], [117], [123]	113	35.20%	1	6
	static		[16], [17], [19], [25], [29], [55], [56], [61], [62], [65], [75]	[4], [20], [76], [107], [117], [123], [129]	71	22.12%	2	16
	stochastic		[27], [54], [71], [93], [116], [125], [126]	[17], [39], [68], [94], [117], [123]	57	17.76%	3	18
	dynamic		[14], [27], [32], [67], [95], [126]	[4], [20], [22], [78], [117], [123], [129]	53	16.51%	4	20
	online		[64], [97]	[80], [86], [92]	18	5.61%	5	38
	offline			[92]	9	2.80%	6	46
2) Theory of complexity			[6], [10], [29], [43], [60], [85], [88], [98]	[39], [40], [41], [59], [67], [99], [103], [112]	89	100.00%	1	13
3) Practical application of scheduling	industrial applications of scheduling theory		[11], [42], [78], [81]	[9], [18], [27], [39], [41], [44], [63], [73], [80], [82], [106], [111]	95	59.75%	1	11
	lack of application in scheduling theory		[29], [31], [45]	[5], [44]	17	10.69%	2	39
	modern production systems	FMS	[28], [51], [66]	[31]	16	10.06%	4	41
		robotic manufacturing	[42], [47], [57]	[98]	17	10.69%	2	39
		semiconductor manufacturing	[78], [81], [104]		11	6.92%	5	43
		wafer fabrication	[105]		3	1.89%	6	56

Table 2 a: Focused and relatively focused reviews for each sub-category including their total score and overall rank

Category	Sub-category		Cleary focuced literature reviews	Relatively focuced literature reviews	Total score	Percentage of the respective category's total score %	Rank in the respective sub-category	Overall rank among all 75 sub-categories
4) Solution approaches	heuristic algorithms & dispatching rules		[7], [12], [22], [35], [79], [105], [109], [128]	[3], [4], [5], [8], [31], [76], [78], [87], [89], [95], [96], [100], [101], [112], [116], [120], [123], [127]	144	27.38%	1	1
	exact algorithms		[46], [72], [75]	[3], [5], [20], [31], [35], [36], [63], [65], [74], [76], [78], [87], [89], [96], [100], [101], [112], [116], [120], [123], [127]	123	23.38%	2	3
	metaheuristic algorithms		[30], [34], [40], [42], [73], [116], [119]	[31], [35], [76], [78], [89], [95], [101], [112], [120], [127]	91	17.30%	3	12
	approximation algorithms		[23], [79]	[10], [26], [92], [96], [97], [112]	50	9.51%	4	22
	simulation		[14], [27], [66], [123]	[4], [20]	42	7.98%	5	25
	artificial intelligence		[122]	[45], [66], [78], [95]	37	7.03%	6	29
	specific solution approach	robust scheduling		[71], [93], [95]	7	1.33%	7	47
		decomposition-based algorithms		[78], [87]	6	1.14%	8	48
		artificial neural networks	[47], [84]		6	1.14%	8	48
		petri nets	[89]		3	0.57%	10	56
		expert systems	[18]		3	0.57%	10	56
		TSP-based approaches	[77]		3	0.57%	10	56
		mixed integer programming models	[72]		3	0.57%	10	56
		taboo search	[30]		3	0.57%	10	56
		hyper heuristics	[126]		3	0.57%	10	56
		Johnson’s rule		[29]	2	0.38%	16	72

Table 2 b: Focused and relatively focused reviews for each sub-category including their total score and overall rank

Category	Sub-category	Cleary focuced literature reviews	Relatively focuced literature reviews	Total score	Percentage of the respective category's total score %	Rank in the respective sub-category	Overall rank among all 75 sub-categories
5) Constraints	setup	[43], [50], [54], [55], [56], [83], [90], [109], [121]	[17], [46], [68], [117], [129]	86	15.19%	1	15
	dual resource operations	[24], [38], [88], [112]	[27], [33], [74], [80], [82], [86], [108]	65	11.48%	2	17
	precedence constrains		[17], [82], [108], [117]	54	9.54%	3	19
	machine breakdown	[48], [59], [99]	[42], [64], [71], [108], [110]	50	8.83%	4	22
	buffer space	[39]	[42], [57], [70], [77]	44	7.77%	5	24
	preemptive scheduling		[10], [48], [85], [92], [99], [103]	42	7.42%	6	25
	batch machines	[37], [68], [81], [82]	[50], [72], [87], [88], [90], [105], [117]	40	7.07%	7	27
	rescheduling problems	[64], [71], [95]	[68], [82], [93]	35	6.18%	8	31
	no-wait operations	[39], [128]	[10], [77], [90]	33	5.83%	9	32
	transportation lags	[123]		26	4.59%	10	34
	common due-dates	[70]	[110]	23	4.06%	11	35
	ordered operations	[114]		10	1.77%	12	44
	learning effects	[91]	[108]	10	1.77%	12	44
	interval scheduling problems	[80], [86]		6	1.06%	14	48
	distributed scheduling	[102], [118]		6	1.06%	14	48
	positional effects	[110], [113]		6	1.06%	14	48
	time-dependent processing time	[67]	[49]	5	0.88%	17	53
	fixed number of jobs with various number of operations	[85]		3	0.53%	18	56
	processing set restrictions	[92]		3	0.53%	18	56
	machine eligibility constraints	[97]		3	0.53%	18	56
	equal processing times	[103],		3	0.53%	18	56
	remanufacturing scheduling	[113]		3	0.53%	18	56
	scheduling with job rejection	[115]		3	0.53%	18	56
	inserted idle-time	[58]		3	0.53%	18	56
	group scheduling	[129]		3	0.53%	18	56
	1-job-m-machine problems		[42]	2	0.35%	26	72
	human factors		[45]	2	0.35%	26	72

Table 2 c: Focused and relatively focused reviews for each sub-category including their total score and overall rank

Category	Sub-category		Cleary focuced literature reviews	Relatively focuced literature reviews	Total score	Percentage of the respective category's total score %	Rank in the respective sub-category	Overall rank among all 75 sub-categories
6) Objective	penalty		[16], [25], [35], [61], [62], [65], [70], [96], [106], [117], [124]	[13], [17], [26], [27], [36], [37], [43], [48], [49], [74], [88], [110], [115]	134	25.00%	1	2
	objectives based on throughput time	makespan	[52], [60], [69], [75], [76], [79], [97]	[17], [27], [39], [43], [49], [67], [87], [110], [115]	119	22.20%	2	4
		flow time	[3], [75]	[26], [27], [36], [37], [39], [43], [48], [49], [110], [115]	106	19.78%	3	8
	multi objective problems		[19], [21], [28], [36], [74], [88], [94], [107], [115], [119], [120]	[17], [23], [64], [66]	89	16.60%	4	13
	cost		[54], [63], [88]	[27], [43], [51], [55]	52	9.70%	5	21
	due-date assignment		[20], [61], [62], [108]	[12], [13], [88], [110]	31	5.78%	6	33
	minimum number of machines		[80]	[86]	5	0.93%	7	53

Table 2 d: Focused and relatively focused reviews for each sub-category including their total score and overall rank

Category	Sub-category	Cleary focuced literature reviews	Relatively focuced literature reviews	Total score	Percentage of the respective category's total score %	Rank in the respective sub-category	Overall rank among all 75 sub-categories
7) Flow pattern	single machine	[10], [17], [19], [21], [35], [37], [49], [96], [117]	[1], [4], [11], [12], [16], [25], [30], [36], [43], [48], [50], [55], [58], [59], [61], [62], [65], [67], [74], [83], [88], [90], [94], [99], [106], [110], [115], [118], [121], [125]	117	20.38%	1	5
	flow shop	[10], [29], [44], [56], [57], [69], [76], [77], [79], [107], [114], [120], [128], [129]	[11], [35], [39], [48], [50], [55], [59], [73], [74], [83], [88], [90], [94], [98], [99], [111], [118], [121], [124], [125]	113	19.69%	2	6
	parallel machines	[10], [26], [35], [92], [97], [103], [112]	[11], [16], [25], [33], [43], [48], [50], [55], [59], [62], [70], [74], [83], [88], [90], [94], [99], [106], [110], [111], [115], [118], [121], [124], [125]	104	18.12%	3	9
	job shop	[10], [22], [27], [38], [40], [52], [64], [78], [116], [122], [123]	[9], [11], [35], [50], [55], [73], [83], [85], [88], [90], [94], [99], [111], [118], [121], [124], [125]	98	17.07%	4	10
	hybrid flow shop	[53], [75], [76], [87], [100], [101]	[39], [85], [111], [121], [129]	40	6.97%	5	27
	open shop	[10]	[74], [85], [88], [90], [99], [111], [118], [121], [124], [125]	37	6.45%	6	29
	process scheduling	[46], [63], [68], [72], [82], [93]	[41]	21	3.66%	7	36
	lot streaming	[8], [54], [111]	[2], [83], [84], [113]	21	3.66%	7	36
	cyclic production	[57], [98]		13	2.26%	9	42
	mixed shop	[60]	[85]	5	0.87%	10	53
	hybrid job shop	[127]		3	0.52%	11	56
	dedicated machines		[106]	2	0.35%	12	72

Table 2 e: Focused and relatively focused reviews for each sub-category including their total score and overall rank

5 Discussion

One important finding of this tertiary analysis concerns the research methodology employed in the sampled literature reviews. As our evaluation of the sample shows that most of the sampled literature reviews are narrative ones and only four of them are classified as systematic, which corresponds to only 3% of the sampled papers (see Section 2 for the drawbacks of narrative literature reviews). This clearly points to a lack of scientific rigor in reviewing the literature on MSPP, as systematic literature reviews are today state-of-the-art in other research disciplines (e.g., da Silva et al., 2011; Hochrein and Glock, 2012). The results further call for future secondary research on MSPP that employs established literature review methodologies that are structured and evidence-based using a robust auditable and repeatable scientific procedure (Denyer and Tranfield, 2009). Thus, future reviews on MSPP should clearly document and report selection criteria as well as sample generation and sample characteristics to generate results the reader can reproduce and validate. Thus, it can be concluded that a clear weakness of most literature reviews on MSPP is that they do not contain either information about the methodology of generating the literature sample, or present any statistical report (such as the sample size, the coverage of the sample, or most contributed outlets).

With respect to the results of our framework analysis, we note that the category "*type of problem*" was neglected in many sampled papers (see, e.g., [74], [77] and [79]). Although the related sub-categories describe important attributes of the considered scenario, such as parameter uncertainty or a dynamic behaviour of the system that strongly influence both the applicability of a model and computational complexity, it is surprising that around 35% of the sampled review papers did not report information about the type of MSPP.

In the category "*constraints*", our analysis reveals that *setup* has attracted more attention in the MSPP literature reviews than any other constraint. As a basic assumption in the literature on MSPP, setups are either integrated in the processing time of jobs, or they are neglected. Setups in MSPP can be classified as "family and non-family" or as "sequence-dependent and sequence-independent" setups. Our tertiary analysis shows that "sequence-dependent setups" (total score: 60) were investigated in greater detail than family-based setups (total score: 30). Apart from setups, the constraints *buffer space* (also denoted as intermediate storage in process operations) and *dual resource operations* (also denoted as constrained resources or controllable processing times) were investigated in many secondary studies. In contrast, other constraints such as *human factors*, *remanufacturing* and *manufacturing with rejection*, which have attracted much attention in other areas of management science and operation management, are clearly underrepresented in secondary studies on MSPP. This implies the necessity of further primary and secondary research considering these constraints in MSPP.

With respect to the category "*objectives*", the sub-category *penalty-based objectives* has attracted the most attention by literature reviews on MSPP in the past. *Makespan* and *flow time*, both belonging to the *throughput time-based objectives*, are the next most discussed sub-categories in this category. As penalty-based objectives are usually set according to conformance to prescribed deadlines of the customers (MacCarthy and Liu, 1993), and as objectives based on throughput time aim on optimizing the facility's utilization ratio by increasing production speed (Hall and Sriskandarajah, 1996), the popularity of penalty-based objectives compared to throughput timed-based objectives in secondary studies on MSPP might reflect that researchers on MSPP realized the importance of customer-centric production systems for the success of companies in today's markets with intense competition.

It is also worth highlighting that some literature reviews on MSPP are *clearly focused* on specific solution approaches. Looking at metaheuristic solution approaches, this tertiary analysis identified only one dedicated review on tabu search [30]. For all other metaheuristics, there is no *clearly focused* literature review available that is devoted only to a single metaheuristic. Future secondary research might focus on reviewing the performance of other metaheuristics (in particular genetic algorithms, simulated annealing, or particle swarm optimization, which are especially popular in this field) to MSPP. This also includes pointing out different research opportunities for applying metaheuristics in MSPP in future primary works.

Finally, to identify the development of sub-categories over time and highlight recent popular sub-categories in MSPP, a sensitivity evaluation was conducted by limiting the sample to literature reviews published between 2007 and 2016. The total score for each sub-category in the reduced sample (2007-2016) was then compared to the original scores presented in Table 2. The sub-categories belonging to both groups, together with their respective categories, are shown in Table 3. Part A reports sub-categories that were not mentioned in literature reviews in the last 10 years, whereas part B highlights sub-categories that were first mentioned in literature reviews since year 2007. The categories in part A may imply a need for an update of secondary research considering the respective sub-categories, while the sub-categories in part B may point towards emerging research topics.

A. sub-categories that were not discussed in literature reviews in the last 10 years

Category	Sub-category
practical application of scheduling	lack of application in scheduling theory
constraints	1-job-m-machine problems
	human factors
	time-dependent processing time
	inserted idle-time
	remanufacturing scheduling
solution approaches	expert systems
	TSP-based approaches
	mixed integer programming models
	taboo search
	Johnson's rule

B. Sub-categories that were first discussed in literature reviews within the last 10 years

Category	Sub-category
practical application of scheduling	wafer fabrication
constraints	fixed number of jobs with various number of operations
	learning effects
	processing set restrictions
	machine eligibility constraints
	distributed scheduling
	equal processing times
	scheduling with job rejection
	group scheduling
flow pattern	hybrid job shop
	dedicated machines
solution approaches	petri nets
	hyper heuristics

Table 3: Results of the sensitivity evaluation

6 Conclusion

This paper reported the results of a tertiary study on machine scheduling problems in production (MSPP). The paper employed a state-of-the-art methodology for systematically reviewing and evaluating literature reviews that appeared in this area. Applying a rigorous methodology for searching relevant review papers, a literature sample containing 129 literature reviews was generated. A comprehensive conceptual framework that includes all dimensions and characteristics of MSPP was then developed. The framework was used to categorize each literature review included in the sample according to its focus.

This work contributes to the literature in various ways. To the best of our knowledge, no comprehensive review of literature reviews on MSPP exists to date. The paper therefore extends the existing literature on MSPP by giving a broad overview of the research field and synthesizing findings of reviews in this area, which is valuable for researchers in getting a broad overview of MSPP in general as well as in finding specific key areas of MSPP. In addition, the developed framework classifies the respective reviews and synthesizes their findings, which allows readers to identify major achievements and research streams in MSPP as well as research gaps. This paper further supports researchers in positioning their own work in the literature and in finding starting points for future research directions, and it also encourages future secondary research by showing in which areas updates on existing literature reviews or new literature reviews might be promising. With the use of the summarizing tables, it is easy to identify relevant literature reviews for a specific MSPP of interest. Finally, this paper also made suggestions for improving the methodological quality of future literature reviews on MSPP and called for more systematic, robust auditable and repeatable research in this area. In addition, this tertiary study highlighted literature reviews focusing on the practical application of MSPP in industry (see category 3 in Table 2a), which can be of decision support for managers.

This work has limitations. First, to ensure methodical rigor and scientific quality, we limited our sample to review papers that were published in high-quality peer-reviewed journals excluding conference papers and book chapters, which might have biased the sample by neglecting possible literature reviews on MSPP that were not published in peer-reviewed journals, but that could also contain interesting insights. Secondly, due to the large number of sampled reviews (129), it was not possible to discuss the findings of each literature review in detail, which made it necessary to synthesize the sample on an aggregated level, which ruled out an in-depth analyses of all characteristics of MSPP. Thirdly, the results of this tertiary study depend on the defined keywords and the database Scopus. Future works could thus use alternative search engines or keywords to evaluate the validity of our results. Fourthly, we excluded other scheduling topics that are not included in MSPP, which might be interesting to examine in future research.

References

Baker, K.R., Scudder, G.D., 1990. Sequencing with Earliness and Tardiness Penalties: A Review. Operations Research. 38, 22–36.

Bushuev, M.A., Guiffrida, A., Jaber, M.Y., Khan, M., 2015. A review of inventory lot sizing review papers. Management Research Review, 38, 283-298.

Cochran, J.K., Chu, D.E., Chu, M.D., 1997. Optimal staffing for cyclically scheduled processes. International Journal of Production Research. 35, 3393–3404.

Cooper, H.M., 2010. Research synthesis and meta-analysis – a step-by-step approach, 4th ed., Los Angeles et al. 2010.
Da Silva, F.Q., Santos, A.L., Soares, S., França, A.C.C., Monteiro, C.V., Maciel, F.F., 2011. Six years of systematic literature reviews in software engineering: An updated tertiary study. Information and Software Technology, 53, 899-913.
Denyer, D., Tranfield, D., 2009. Producing a systematic review (Chapter 39), in Bu-chanan, D. and Bryman, A. (Eds), The Sage Handbook of Organizational Research Meth-ods, Sage Publications Ltd, London, pp. 671-89.
Dileepan, P., Sen, T., 1988. Bicriterion static scheduling research for a single machine. Omega 16, 53–59.
Garousi, V., Mäntylä, M. V., 2016. A systematic literature review of literature reviews in software testing. Information and Software Technology. 80, 195–216.
Glock, C.H., Grosse, E.H., Ries, J.M., 2014. The lot sizing problem: A tertiary study. International Journal of Production Economics. 155, 39–51.
González-Pereira, B., Guerrero-Bote, V.P., Moya-Anegón, F., 2010. A new approach to the metric of journals scientific prestige: The SJR indicator. Journal of Informetrics. 4, 379–391.
Gordon, V.S., Proth, J.-M., Chu, C., 2002. Due date assignment and scheduling: Slk, TWK and other due date assignment models. Production Planning & Control. 13, 117–132.
Gorman, M.F., 2016. A “Metasurvey” analysis in Operations Research and Management Science: A survey of literature reviews. Surveys in Operations Research and Management Science. 21, 18–28.
Graham, R.L., Lawler, E.L., Lenstra, J.K., Kan, A.H.G.R., 1979. Optimization and approximation in deterministic sequencing and scheduling: A survey. Annals of Discrete Mathematics. 5, 287–326.
Graves, S.C., 1981. A review of production scheduling. Operations Research. 29, 646–675.
Hall, N.G., Sriskandarajah, C., 1996. A survey of machine scheduling problems with blocking and no-wait in process. Operations Research. 44, 510–525.
Hochrein, S., Glock, C., 2012. Systematic literature reviews in purchasing and supply management research: a tertiary study. International Journal of Integrated Supply Management. 7, 215–245.
Hochrein, S., Glock, C.H., Bogaschewsky, R., Heider, M., 2015. Literature reviews in supply chain management: a tertiary study. Management Review Quarterly. 65, 239–280.
Kache, F., Seuring, S., 2014. Linking collaboration and integration to risk and performance in supply chains via a review of literature reviews. Supply Chain Management: An International Journal. 19, 664–682.
Kiran, A.S., Smith, M.L., 1984. Simulation studies in job shop scheduling: a survey. Computers & Industrial Engineering. 8, 87–93.
Lee, K., Leung, J.Y.-T., Pinedo, M.L., 2010. Makespan minimization in online scheduling with machine eligibility. 4OR: A Quarterly Journal of Operations Research. 8, 331–364.
Leyman, P., Vanhoucke, M., 2015. A new scheduling technique for the resource–constrained project scheduling problem with discounted cash flows. International Journal of Production Research. 53, 2771–2786.
Li, S., Singhoff, F., Rubini, S., Bourdellès, M., 2016. Scheduling analysis of tasks constrained by TDMA: Application to software radio protocols. Journal of Systems Architecture. 2016, 1–18.
Li, Z., Ierapetritou, M., 2008. Process scheduling under uncertainty: Review and challenges. Computers & Chemical Engineering. 32, 715–727.
MacCarthy, B.L., Liu, J., 1993. Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling. International Journal of Production Research. 31, 59–79.
Moed, H.F., 2010. Measuring contextual citation impact of scientific journals. Journal of Informetrics. 4, 265–277.
Mönch, L., Fowler, J.W., Dauzère-Pérès, S., Mason, S.J., Rose, O., 2011. A survey of problems, solution techniques, and future challenges in scheduling semiconductor manufacturing operations. Journal of Scheduling. 14, 583–599.
Nagar, A., Haddock, J., Heragu, S., 1995. Multiple and bicriteria scheduling: A literature survey. European Journal of Operational Research. 81, 88–104.
Nkasu, M.M., Leung, K.H., 1997. A resources scheduling decision support system for concurrent project management. International Journal of Production Research. 35, 3107–3132.
Ouelhadj, D., Petrovic, S., 2009. A survey of dynamic scheduling in manufacturing systems. Journal of Scheduling. 12, 417–431.
Pinedo, M.L., 2008. Scheduling Theory, Algorithms, and Systems, Third. ed. Springer.
Prot, D., Lapègue, T., Bellenguez-Morineau, O., 2015. A two-phase method for the shift design and personnel task scheduling problem with equity objective. International Journal of Production Research. 7543, 1–13.
Reza Hejazi, S., Saghafian, S., 2005. Flowshop-scheduling problems with makespan criterion: a review. International Journal of Production Research. 43, 2895–2929.
Sabuncuoglu, I., 1998. Scheduling with neural networks: A review of the literature and new research directtions. Production Planning & Control. 9, 2–12.
Sen, T., Gupta, S.K., 1984. A state-of-art survey of static scheduling research involving due dates. Omega 12, 63–76.
Seuring, S., Gold, S., 2012. Conducting content-analysis based literature reviews in supply chain management. Supply Chain Management: An International Journal. 17, 544–555.

Tuncel, G., Bayhan, G.M., 2007. Applications of Petri nets in production scheduling: A review. The International Journal of Advanced Manufacturing Technology. 34, 762–773.

Appendix. List of papers included in the sample

To ease referencing to the sampled review papers, we first sorted them chronologically according to the year of publication, and then assigned the numbers 1 to 129.

[1]. Sisson, R.L., 1959. Methods of sequencing in job shops-A review. Operations Research. 7, 10–29.
[2]. Elmaghraby, S.E., 1968. The machine sequencing problem review and extensions. Naval Research Logistics. 15, 205–232.
[3]. Bakshi, M.S., Arora, S.R., 1969. The Sequencing Problem. Management Science. 16, 247–263.
[4]. Day, J.E., Hottenstein, M.P., 1970. Review of sequencing research. Naval Research Logistics Quarterly. 17, 11–39.
[5]. Gupta, J.N.D., 1971. M-stage scheduling problem—a critical appraisal. International Journal of Production Research. 9, 267–281.
[6]. Lenstra, J.K., Kan, A.H.G.R., Brucker, P., 1977. Complexity of machine scheduling problems. Annals of Discrete Mathematics. 1, 343–362.
[7]. Panwalkar, S.S., Iskander, W., 1977. A survey of scheduling rules. Operations Research. 25, 45–61.
[8]. Elmaghraby, S.E., 1978. The economic lot scheduling problem (ELSP): review and extensions. Management Science. 24, 587–598.
[9]. Godin, V.B., 1978. Interactive scheduling: historical survey and state of the art. AIIE Transactions. 10, 331–337.
[10]. Graham, R.L., Lawler, E.L., Lenstra, J.K., Kan, A.H.G.R., 1979. Optimization and approximation in deterministic sequencing and scheduling: A survey. Annals of Discrete Mathematics. 5, 287–326.
[11]. Graves, S.C., 1981. A review of production scheduling. Operations Research. 29, 646–675.
[12]. Blackstone, J.H., Phillips, D.T., Hogg, G.L., 1982. A state-of-the-art survey of dispatching rules for manufacturing job shop operations. International Journal of Production Research. 20, 27–45.
[13]. Baker, K.R., 1984. Sequencing rules and due-date assignments in a job shop. Management Science. 30, 1093–1104.
[14]. Kiran, A.S., Smith, M.L., 1984. Simulation studies in job shop scheduling: a survey. Computers & Industrial Engineering. 8, 87–93.
[15]. Lenstra, J.K., Kan, A.H.G.R., 1984. New directions in scheduling theory. Operations Research Letters. 2, 255–259.
[16]. Sen, T., Gupta, S.K., 1984. A state-of-art survey of static scheduling research involving due dates. Omega 12, 63–76.
[17]. Gupta, S.K., Kyparisis, J., 1987. Single Machine Scheduling Research. Omega 15, 207–227.
[18]. Kanet, J.J., Adelsberger H, 1987. Expert systems in production scheduling. European Journal of Operational Research. 29, 51–59.
[19]. Dileepan, P., Sen, T., 1988. Bicriterion static scheduling research for a single machine. Omega 16, 53–59.
[20]. Cheng, T.C.E., Gupta, M.C., 1989. Survey of scheduling research involving due date determination decisions. European Journal of Operational Research. 38, 156–166.
[21]. Fry, T.D., Armstrong, R.D., Lewis, H., 1989. A framework for single machine multiple objective sequencing research. Omega 17, 595–607.
[22]. Haupt, R., 1989. A survey of priority rule-based scheduling. OR Spektrum 11, 3–16.
[23]. Kovalev, M.Y., Shafranskij, Y.M., Strusevich, V.A., Tanaev, V.S., Tuzikov, A.., 1989. Approximation scheduling algorithms: a survey. Optimization 20, 859–878.
[24]. Treleven, M., 1989. A review of the dual resource constrained system research. IIE Transactions. 21, 279–287.
[25]. Baker, K.R., Scudder, G.D., 1990. Sequencing with earliness and tardiness penalties: A review. Operations Research. 38, 22–36.
[26]. Cheng, T.C.E., Sin, C.C.S., 1990. A state-of-the-art review of parallel-machine scheduling research. European Journal of Operational Research. 47, 271–292.
[27]. Ramasesh, R., 1990. Dynamic job shop scheduling: A survey of simulation research. Omega 18, 43–57.
[28]. Gupta, Y.P., Evans, G.W., Gupta, M.C., 1991. A review of multi-criterion approaches to FMS scheduling problems. International Journal of Production Economics. 22, 13–31.
[29]. Dudek, R. a, Panwalkar, S.S., Smith, M.L., 1992. The lessons of flowshop scheduling research. Operations Research. 40, 7–13.

[30]. Barnes, J.W., Laguna, M., 1993. A tabu search experience in production scheduling. Annals of Operations Research. 41, 141–156.
[31]. MacCarthy, B.L., Liu, J., 1993. Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling. International Journal of Production Research. 31, 59–79.
[32]. Suresh, V., Chaudhuri, D., 1993. Dynamic scheduling—a survey of research. International Journal of Production Economics. 32, 53–63.
[33]. Tzafestas, S., Triantafyllakis, A., 1993. Deterministic scheduling in computing and manufacturing systems: a survey of models and algorithms. Mathematics and Computers in Simulation. 35, 397–434.
[34]. Aytug, H., Bhattacharyya, S., Koehler, G.J., Snowdon, J.L., 1994. A review of machine learning in scheduling. IEEE Transactions on Engineering Management. 41, 165–171.
[35]. Koulamas, C., 1994. The total tardiness problem: review and extensions. Operations Research. 42, 1025–1042.
[36]. Nagar, A., Haddock, J., Heragu, S., 1995. Multiple and bicriteria scheduling: A literature survey. European Journal of Operational Research. 81, 88–104.
[37]. Webster, S., Baker, K.R., 1995. Scheduling groups of jobs on a single machine. Operations Research. 43, 692–703.
[38]. Gargeya, V.B., Deane, R.H., 1996. Scheduling research in multiple resource constrained job shops: a review and critique. International Journal of Production Research. 34, 2077–2097.
[39]. Hall, N. G., Sriskandarajah, C., 1996. A survey of machine scheduling problems with blocking and no-wait in process. Operations Research. 44, 510–525.
[40]. Vaessens, R.J.M., Aarts, E.H.L., Lenstra, J.K., 1996. Job shop scheduling by local search. INFORMS Journal on Computing. 8, 302–317.
[41]. Appiequist, G., Samikog, O., Pekny, J., Reklaifis, G., 1997. Issues in the use, design and evolution of process scheduling and planning systems. ISA Transactions. 36, 81–121.
[42]. Lee, C., Lei, L., Pinedo, M., 1997. Current trends in deterministic scheduling. Annals of Operations Research. 70, 1–41.
[43]. Liaee, M.M., Emmons, H., 1997. Scheduling families of jobs with setup times. International Journal of Production Economics. 51, 165–176.
[44]. Reisman, A., Kumar, A., Motwani, J., 1997. Flowshop scheduling/sequencing research : A statistical review of the literature, 1952 – 1994. IEEE Transactions on Engineering Management. 44, 316–329.
[45]. Wiers, V.C.S., 1997. A review of the applicability of OR and AI scheduling techniques in practice. Omega 25, 145–153.
[46]. Pinto, J.M., Grossmann, I.E., 1998. Assignment and sequencing models for the scheduling of process systems. Annals of Operations Research. 81, 433–466.
[47]. Sabuncuoglu, I., 1998. Scheduling with neural networks: A review of the literature and new research directions. Production Planning & Control. 9, 2–12.
[48]. Sanlaville, E., Schmidt, G., 1998. Machine scheduling with availability constraints. Acta Informatica. 35, 795–811.
[49]. Alidaee, B., Womer, N.K., 1999. Scheduling with time dependent processing times: Review and extensions. Journal of the Operational Research Society. 50, 711–720.
[50]. Allahverdi, A., Gupta, J.N.., Aldowaisan, T., 1999. A review of scheduling research involving setup considerations. Omega 27, 219–239.
[51]. Balogun, O.O., Popplewell, K., 1999. Towards the integration of flexible manufacturing system scheduling. International Journal of Production Research. 37, 3399–3428.
[52]. Jain, A.S., Meeran, S., 1999. Deterministic job-shop scheduling: Past, present and future. European Journal of Operational Research. 113, 390–434.
[53]. Linn, R., Zhang, W., 1999. Hybrid flow shop scheduling: a survey. Computers & Industrial Engineering. 37, 57–61.
[54]. Sox, C.R., Jackson, P.L., Bowman, A., Muckstadt, J. a., 1999. A review of the stochastic lot scheduling problem. International Journal of Production Economics. 62, 181–200.
[55]. Yang, W.-H., 1999. Survey of scheduling research involving setup times. International Journal of Systems Science. 30, 143–155.
[56]. Cheng, T.C.E., Gupta, J.N.D., Wang, G.Q., 2000. A review of flowshop scheduling research with setup times. Production and Operations Management. 9, 262–282.
[57]. Crama, Y., Kats, V., Klundert, J. Van De, Levner, E., 2000. Cyclic scheduling in robotic flowshops. Annals of Operations Research. 96, 97–124.
[58]. Kanet, J.J., Sridharan, V., 2000. Scheduling with inserted idle time: problem taxonomy and literature review. Operations Research. 48, 99–110.

[59]. Schmidt, G., 2000. Scheduling with limited machine availability. European Journal of Operational Research. 121, 1–15.

[60]. Shakhlevich, N. V., Sotskov, Y.N., Werner, F., 2000. Complexity of mixed shop scheduling problems: A survey. European Journal of Operational Research. 120, 343–351.

[61]. Gordon, V.S., Proth, J.-M., Chu, C., 2002. Due date assignment and scheduling: Slk, TWK and other due date assignment models. Production Planning & Control. 13, 117–132.

[62]. Gordon, V., Proth, J.-M., Chu, C., 2002. A survey of the state-of-the-art of common due date assignment and scheduling research. European Journal of Operational Research. 139, 1–25.

[63]. Kallrath, J., 2002. Planning and scheduling in the process industry. OR Spectrum. 24, 219–250.

[64]. Raheja, a. S., Subramaniam, V., 2002. Reactive recovery of job shop schedules - A review. The International Journal of Advanced Manufacturing Technology. 19, 756–763.

[65]. Sen, T., Sulek, J.M., Dileepan, P., 2003. Static scheduling research to minimize weighted and unweighted tardiness: A state-of-the-art survey. International Journal of Production Economics. 83, 1–12.

[66]. Chan, F.T.S., Chan, H.K., 2004. A comprehensive survey and future trend of simulation study on FMS scheduling. Journal of Intelligent Manufacturing. 15, 87–102.

[67]. Cheng, T.C.., Ding, Q., Lin, B.M.., 2004. A concise survey of scheduling with time-dependent processing times. European Journal of Operational Research. 152, 1–13.

[68]. Floudas, C. a., Lin, X., 2004. Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. Computers & Chemical Engineering. 28, 2109–2129.

[69]. Framinan, J.M., Gupta, J.N.D., Leisten, R., 2004. A review and classification of heuristics for permutation flowstop scheduling with makespan objective. Journal of the Operational Research Society. 55, 1243–1255.

[70]. Lauff, V., Werner, F., 2004. Scheduling with common due date, earliness and tardiness penalties for multi-machine problems: A survey. Mathematical and Computer Modelling. 40, 637–655.

[71]. Aytug, H., Lawley, M. a., McKay, K., Mohan, S., Uzsoy, R., 2005. Executing production schedules in the face of uncertainties: A review and some future directions. European Journal of Operational Research. 161, 86–110.

[72]. Floudas, C. a., Lin, X., 2005. Mixed integer linear programming in process scheduling: modeling, algorithms, and applications. Annals of Operations Research. 139, 131–162.

[73]. Hart, E., Ross, P., Corne, D., 2005. Evolutionary scheduling: A review. Genetic Programming and Evolvable Machines. 6, 191–220.

[74]. Hoogeveen, H., 2005. Multicriteria scheduling. European Journal of Operational Research. 167, 592–623.

[75]. Kis, T., Pesch, E., 2005. A review of exact solution methods for the non-preemptive multiprocessor flowshop problem. European Journal of Operational Research. 164, 592–608.

[76]. Reza Hejazi, S., Saghafian, S., 2005. Flowshop-scheduling problems with makespan criterion: a review. International Journal of Production Research. 43, 2895–2929.

[77]. Bagchi, T.P., Gupta, J.N.D., Sriskandarajah, C., 2006. A review of TSP based approaches for flowshop scheduling. European Journal of Operational Research. 169, 816–854.

[78]. Gupta, A.K., Sivakumar, A.I., 2006. Job shop scheduling techniques in semiconductor manufacturing. The International Journal of Advanced Manufacturing Technology. 27, 1163–1169.

[79]. Gupta, J.N.D., Koulamas, C., Kyparisis, G.J., 2006. Performance guarantees for flowshop heuristics to minimize makespan. European Journal of Operational Research. 169, 865–872.

[80]. Mathirajan, M., Sivakumar, a. I., 2006. A literature review, classification and simple meta-analysis on scheduling of batch processors in semiconductor. The International Journal of Advanced Manufacturing Technology. 29, 990–1001.

[81]. Méndez, C. a., Cerdá, J., Grossmann, I.E., Harjunkoski, I., Fahl, M., 2006. State-of-the-art review of optimization methods for short-term scheduling of batch processes. Computers & Chemical Engineering. 30, 913–946.

[82]. Zhu, X., Wilhelm, W.E., 2006. Scheduling and lot sizing with sequence-dependent setup: A literature review. IIE Transactions. 38, 987–1007.

[83]. Akyol, D.E., Bayhan, G.M., 2007. A review on evolution of production scheduling with neural networks. Computers & Industrial Engineering. 53, 95–122.

[84]. Brucker, P., Sotskov, Y.N., Werner, F., 2007. Complexity of shop-scheduling problems with fixed number of jobs: a survey. Mathematical Methods of Operations Research. 65, 461–481.

[85]. Kolen, A.W.J., Lenstra, J.K., Papadimitriou, C.H., Spieksma, F.C.R., 2007. Interval scheduling: A survey. Naval Research Logistics. 54, 530–343.

[86]. Kovalyov, M.Y., Ng, C.T., Cheng, T.C.E., 2007. Fixed interval scheduling: Models, applications, computational complexity and algorithms. European Journal of Operational Research. 178, 331–342.

[87]. Quadt, D., Kuhn, H., 2007. A taxonomy of flexible flow line scheduling procedures. European Journal of Operational Research. 178, 686–698.

[88]. Shabtay, D., Steiner, G., 2007. A survey of scheduling with controllable processing times. Discrete Applied Mathematics. 155, 1643–1666.

[89]. Tuncel, G., Bayhan, G.M., 2007. Applications of Petri nets in production scheduling: A review. The International Journal of Advanced Manufacturing Technology. 34, 762–773.

[90]. Allahverdi, A., Ng, C.T., Cheng, T.C.E., Kovalyov, M.Y., 2008. A survey of scheduling problems with setup times or costs. European Journal of Operational Research. 187, 985–1032.

[91]. Biskup, D., 2008. A state-of-the-art review on scheduling with learning effects. European Journal of Operational Research. 188, 315–329.

[92]. Leung, J.Y.-T., Li, C.-L., 2008. Scheduling with processing set restrictions: A survey. International Journal of Production Economics. 116, 251–262.

[93]. Li, Z., Ierapetritou, M., 2008. Process scheduling under uncertainty: Review and challenges. Computers & Chemical Engineering. 32, 715–727.

[94]. Lei, D., 2009. Multi-objective production scheduling: a survey. The International Journal of Advanced Manufacturing Technology. 43, 926–938.

[95]. Ouelhadj, D., Petrovic, S., 2009. A survey of dynamic scheduling in manufacturing systems. Journal of Scheduling. 12, 417–431.

[96]. Koulamas, C., 2010. The single-machine total tardiness scheduling problem: Review and extensions. European Journal of Operational Research. 202, 1–7.

[97]. Lee, K., Leung, J.Y.-T., Pinedo, M.L., 2010. Makespan minimization in online scheduling with machine eligibility. 4OR: A Quarterly Journal of Operations Research. 8, 331–364.

[98]. Levner, E., Kats, V., Alcaide López de Pablo, D., Cheng, T.C.E., 2010. Complexity of cyclic scheduling problems: A state-of-the-art survey. Computers & Industrial Engineering. 59, 352–361.

[99]. Ma, Y., Chu, C., Zuo, C., 2010. A survey of scheduling with deterministic machine availability constraints. Computers & Industrial Engineering. 58, 199–211.

[100]. Ribas, I., Leisten, R., Framiñan, J.M., 2010. Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective. Computers & Operations Research. 37, 1439–1454.

[101]. Ruiz, R., Vázquez-Rodríguez, J.A., 2010. The hybrid flow shop scheduling problem. European Journal of Operational Research. 205, 1–18.

[102]. Toptal, A., Sabuncuoglu, I., 2010. Distributed scheduling: a review of concepts and applications. International Journal of Production Research. 48, 5235–5262.

[103]. Kravchenko, S. a., Werner, F., 2011. Parallel machine problems with equal processing times: a survey. Journal of Scheduling. 14, 435–444.

[104]. Mönch, L., Fowler, J.W., Dauzère-Pérès, S., Mason, S.J., Rose, O., 2011. A survey of problems, solution techniques, and future challenges in scheduling semiconductor manufacturing operations. Journal of Scheduling. 14, 583–599.

[105]. Sarin, S.C., Varadarajan, A., Wang, L., 2011. A survey of dispatching rules for operational control in wafer fabrication. Production Planning & Control. 22, 4–24.

[106]. Sterna, M., 2011. A survey of scheduling problems with late work criteria. Omega 39, 120–129.

[107]. Sun, Y., Zhang, C., Gao, L., Wang, X., 2011. Multi-objective optimization algorithms for flow shop scheduling problem: a review and prospects. The International Journal of Advanced Manufacturing Technology. 55, 723–739.

[108]. Gordon, V., Strusevich, V., Dolgui, A., 2012. Scheduling with due date assignment under special conditions on job processing. Journal of Scheduling. 15, 447–456.

[109]. Pickardt, C.W., Branke, J., 2012. Setup-oriented dispatching rules – a survey. International Journal of Production Research. 50, 5823–5842.

[110]. Rustogi, K., Strusevich, V.A., 2012. Simple matching vs linear assignment in scheduling models with positional effects: A critical review. European Journal of Operational Research. 222, 393–407.

[111]. Cheng, M., Mukherjee, N.J., Sarin, S.C., 2013. A review of lot streaming. Int. J. Prod. Res. 51, 1–24.

[112]. Edis, E.B., Oguz, C., Ozkarahan, I., 2013. Parallel machine scheduling with additional resources: Notation, classification, models and solution methods. European Journal of Operational Research. 230, 449–463.

[113]. Morgan, S.D., Gagnon, R.J., 2013. A systematic literature review of remanufacturing scheduling. International Journal of Production Research. 51, 4853–4879.

[114]. Panwalkar, S.S., Smith, M.L., Koulamas, C., 2013. Review of the ordered and proportionate flow shop scheduling research. Naval Research Logistics. 60, 46–55.

[115]. Shabtay, D., Gaspar, N., Kaspi, M., 2013. A survey on offline scheduling with rejection. Journal of Scheduling. 16, 3–28.

[116]. Abdullah, S., Abdolrazzagh-Nezhad, M., 2014. Fuzzy job-shop scheduling problems: A review. Information Sciences. (Ny). 278, 380–407.
[117]. Adamu, M.O., Adewumi, A.O., 2014. A survey of single machine scheduling to minimize weighted number of tardy jobs. Journal of Industrial and Management Optimization. 10, 219–241.
[118]. Behnamian, J., Fatemi Ghomi, S.M.T., 2014. A survey of multi-factory scheduling. Journal of Intelligent Manufacturing. 27, 1–19.
[119]. Gen, M., Lin, L., 2014. Multiobjective evolutionary algorithm for manufacturing scheduling problems: state-of-the-art survey. Journal of Intelligent Manufacturing. 25, 849–866.
[120]. Mutlu, M., Yagmahan, B., 2014. Multi-objective permutation flow shop scheduling problem: Literature review, classification and current trends. Omega 45, 119–135.
[121]. Allahverdi, A., 2015. The third comprehensive survey on scheduling problems with setup times/costs. European Journal of Operational Research. 246, 345–378.
[122]. Çaliş, B., Bulkan, S., 2015. A research survey: review of AI solution strategies of job shop scheduling problem. Journal of Intelligent Manufacturing. 26, 961–973.
[123]. Xie, C., Allen, T.T., 2015. Simulation and experimental design methods for job shop scheduling with material handling: a survey. The International Journal of Advanced Manufacturing Technology. 80, 233–243.
[124]. Adamu, M.O., Adewumi, A.O., 2016. Minimizing the weighted number of tardy jobs on multiple machines: A review. Journal of Industrial and Management Optimization. 12, 1465–1493.
[125]. Behnamian, J., 2016. Survey on fuzzy shop scheduling. Fuzzy Optimization and Decision Making. 15, 331–366.
[126]. Branke, J., Nguyen, S., Pickardt, C.W., Zhang, M., 2016. Automated design of production scheduling heuristics: a Review. IEEE Transactions on Evolutionary Computation. 20, 110–124.
[127]. Chaudhry, I.A., Khan, A.A., 2016. A research survey: Review of flexible job shop scheduling techniques. International Transactions in Operational Research. 23, 551–591.
[128]. Nagano, M.S., Miyata, H.H., 2016. Review and classification of constructive heuristics mechanisms for no-wait flow shop problem. International Journal of Advanced Manufacturing Technology. 86, 1–14.
[129]. Neufeld, J.S., Gupta, J.N.D., Buscher, U., 2016. A comprehensive review of flowshop group scheduling literature. Computers and Operations Research. 70, 56–74.

Paper 2 Machine scheduling in production: A content analysis

Authors: Hamid Abedinnia, Christoph H. Glock, Michael Schneider

Type of publication: Journal article

Publication details: *Applied Mathematical Modelling*, Volume 50, 2017, Pages 279-299

Abstract

This paper presents the results of a content analysis on a comprehensive and systematically generated sample of 132 literature surveys on machine scheduling problems in production (MSPP). The paper identifies the main attributes of MSPP by analyzing these surveys and proposes a classification scheme for MSPP consisting of seven main groups with several subgroups. The reliability of the results of the content analysis is examined in a sensitivity analysis. A close analysis of the results unveils several research gaps in the literature and enables us to propose promising avenues for future research.

Keywords: *Machine scheduling, Production scheduling, Content analysis, Tertiary study, Systematic literature review*

1 Introduction

Scheduling problems arise whenever restricted resources have to be assigned to task elements for accomplishing these tasks over time. Scheduling is relevant in different disciplines such as project management [1], aerospace industry [2], computer science [3,4], and personnel management [5]. The focus of the paper at hand is on the scheduling of production activities, where a set of jobs needs to be processed on a set of machines (we refer to this type of problems as “machine scheduling problems in production” (MSPP) in the following). The aim of MSPP is to plan the work flow through the ma- chines in a way that optimizes a set of specific objectives.

A massive body of research on MSPP exists because of its applicability in different production industries (see, e.g., [6,7]). The keyword combination “scheduling” and “production” leads to 1,770,000 hits in Google Scholar and 425,013 hits in Scopus, which roughly illustrates the scope of this research stream[1]. The substantial amount of research on this subject is also witnessed by a high number of survey articles on MSPP. These survey papers aim at structuring the literature by identifying main research streams, synthesizing research findings and highlighting interesting future research opportunities. The literature on MSPP distinguishes between different problem variants, and for many problem variants, one or even several surveys have been published in the past, e.g., on static and dynamic problems [8–10], on deterministic and stochastic problems [11,12], on single-objective and multi-objective problems [13,14], or on specific flow patterns of jobs between machines [15–17].

As the high number of published works on MSPP prohibits a review that covers the entire literature, we apply a content analysis (CA)—a methodology that helps to evaluate large amounts of data in a structured and systematic way—on a comprehensive sample of survey papers on MSPP. The intention of analyzing literature reviews (i.e., so-called secondary studies) in a CA is that a tertiary study enables us to analyze the entire domain of MSPP, which would not be possible in a regular secondary literature review that analyzes primary works [18,19]. The comprehensive overview provided by our study is useful for researchers who wish to inform themselves about the state-of-research of the MSPP discipline, who wish to position their own work in the existing literature, or who intend to conduct a literature review on an MSPP problem on their own. In addition, the classification scheme derived in the survey at hand can be used to guide future research in this area. Therefore, the objective of the CA at hand is to provide an overview of the overall development and the state-of-the-art of MSPP research and thus to support researchers and practitioners in accessing this research field.

More precisely, we:

- provide an overview of the literature on MSPP by analyzing secondary-level works (i.e., survey papers).
- investigate publication patterns of surveys on MSPP.
- highlight which dimensions of the problem attracted the most attention from researchers on the secondary level, and which areas have not been studied to an adequate degree and might thus offer potential for further research. As dimensions of the problem, we consider (i) the main

[1] Numbers effective March 2017.

research topics studied in secondary works, (ii) the most important assumptions made, (iii) the most frequently investigated constraints, (iv) the solution approaches used, and (v) the objectives employed.
- develop a structured framework for classifying studies on MSPP. This framework supports researchers in positioning their works in the literature and in identifying promising areas for future research.

A related work is Reisman et al. [20], which is the only paper we are aware of that applied a CA in the context of MSPP. In contrast to our paper, Reisman et al. [20] restricted themselves to flow shop scheduling problems, which allowed them to focus on primary studies. We adopt a more comprehensive perspective, and our object of analysis are MSPP in general. Therefore, we have to focus on secondary works.

The rest of the paper is organized as follows. Section 2 introduces the CA employed in the paper. First, we explain our approach to retrieve a reproducible sample of survey papers on MSPP and present descriptive results of the literature search. Secondly, we discuss the employed recording and coding units and show how these units can be categorized. Section 2 concludes with presenting the results obtained during the CA. Section 3 reports the results of a sensitivity analysis con- ducted to validate the reliability of the CA results, and it provides some suggestions for improving the reliability of the CA. Section 4 concludes the paper.

2 Methodology and results of the study

A CA is a systematic, objective and quantitative approach to extract worthwhile information from massive data [21]. The objective of a CA is to make replicable and valid inferences from texts or other meaningful matters to the context of their use [22]. The history of CA was reviewed by Krippendorff [22], who saw World War II as a driver for the popularity of CA, where it was used for extracting information from propaganda. The method was originally used in communication science, journalism and sociology before it was adopted by other disciplines, e.g. physics [23], education [24], or production planning and logistics [25,26].

Detailed guidelines for conducting a CA can be found in Weber [27], Neuendorf [21], Krippendorff [22], and Babbie [28]. Neuendorf [21] distinguished between four different types of CAs, namely descriptive, inferential, psychometric, and predictive CAs. The approach selected for the paper at hand is a descriptive CA, whose conclusions are limited to the content under study. Although the generic stages of the approach are largely fixed (see [25]), the detailed steps of each stage need to be tailored towards the nature of the research project. In the following, we explain the steps of the CA applied in this paper, which were designed according to the methodology described in Neuendorf [21] and Krippendorff [22].

The CA starts with formulating research questions (Section 1) and then gathers the relevant material (sample) the CA is applied to (Section 2.1). In Steps 3 and 4, recording and coding units are developed for the sample (Section 2.2), and then the recorded material is categorized (Section 2.3). Step 5 summarizes the results (Section 2.4) for the final step: the interpretation of results (Section 3).

2.1. Sampling

After the formulation of a research question, sampling is the second key step of a CA. Krippendorff [22] stated that the "universe of available texts is too large to be examined as a whole, so content analysts need to limit their research to a manageable body of text". As explained in the introduction, the number of primary works on MSPP is enormous, which makes it impossible to analyze the entire literature on MSPP in a single paper. Secondary research works are representative of this immense body of primary literature, but their number is much smaller. To gain a broad picture of the literature on MSPP, the paper at hand therefore focuses on the tertiary level and analyses literature surveys in a CA. In tertiary studies, a particular field of research is analyzed by studying secondary research works to identify research streams and research patterns. Such studies are valuable for both primary and secondary researchers, and they are becoming more and more popular in operations management and operations research [29,30] because:

- secondary researchers can identify research streams in the literature that require an initial or updated survey article.
- primary researchers can better position their own work in the literature, and they can find proper starting points for conducting research in a new direction.

To generate the sample for our CA, a systematic search of the literature was conducted to identify all literature surveys on MSPP that have been published in high-quality international journals. One advantage of such a systematic literature search is that it ensures a transparent and reproducible generation of the sample (see [31] for a detailed description of the advantages of a systematic literature search). In the following, Section 2.1.1 describes the methodology of the literature search, and Section 2.1.2 provides a descriptive analysis of the final sample.

2.1.1. Methodology of the literature search

To search the literature, we follow the methodology advocated by Tranfield et al. [31], Cooper [32], Glock and Hochrein [33], and Hochrein and Glock [34] to identify relevant papers in a multi-step search approach, which can be summarized as follows:

1. **Initial search:** In the first step, keywords were defined that were then used to search the database Scopus (http://www.scopus.com) for works that contain a word from the set of keywords either in their title, abstract, or list of keywords. To generate the final set of keywords, three initial groups of keywords were defined. Group A aimed at retrieving scheduling-related works and contained "scheduling" and "sequencing". Group B intended to limit the search to scheduling problems in manufacturing and contained "machine", "shop", "manufacturing", "production", "process", "flow shop", "job shop", and "open shop". The keyword "shop" was included to ensure that different spellings of shop-related production systems are covered (e.g., "flow shops" are often also referred to as "flowshops" or "flow-shops"). The keywords in Group C were defined to ensure that only survey papers are included in the final sample, and the group contained "survey", "review", "overview", "taxonomy", and "trends". The final keyword list was generated by combining all keywords from the three groups. Thus, the search string used

in the database search is [("Scheduling" ∨ "Sequencing") ∧ ("Machine" ∨ "Shop" ∨ "Manufacturing" ∨ "Production" ∨ "Process" ∨ "Flow shop" ∨ "Job shop" ∨ "Open shop") ∧ ("Survey" ∨ "Review" ∨ "Overview" ∨ "Taxonomy" ∨ "Trends")] in (Title ∨ Abstract ∨ Keywords). In Step 1 of our literature search, 7643 papers were identified.

2. **First refinement:** First, papers found in Step 1 that do not have an English title and/or abstract were removed. For the remaining papers, the titles and abstracts were read, and irrelevant works were excluded. The size of the literature sample was reduced to 523 papers in this step.
3. **Second refinement:** The following exclusion criteria were applied to the literature sample obtained in Step 2:
   a) The language of the full papers (not only the titles and abstracts) was restricted to English.
   b) Only papers published in peer-reviewed journals were considered relevant, and so-called grey literature (e.g., book chapters or conference papers) was excluded.
   c) Only works with a focus on reviewing the literature on MSPP were considered relevant. Works that contain a literature review (e.g., to clarify a research question) but whose focus is on something else (e.g., developing a model or algorithm) were excluded from the sample. This also includes papers that compare a set of methods for a particular scheduling problem.
   d) To keep the size of the sample manageable and to make sure that only works that appeared in high-quality journals were considered relevant, the sample was limited to papers published in journals with SNIP ≥ 0.87 and SJR ≥ 0.51. The SNIP and SJR are metrics to evaluate the reputation of journals and are described in detail in [35] and [36]. In addition, we manually added two journals to the list of relevant journals because we believe them to be relevant journals in the field of operations research and management science: *Annals of Discrete Mathematics* and *Production and Inventory Management*. For these two journals, the SNIP and SJR metrics were not available.

   The size of the sample was reduced to 210 articles in this step.
4. **Snowball search:** In Step 4, a backward and a forward snowball search were carried out by checking the references of the papers contained in the current sample and by evaluating articles citing the sampled papers for relevance. In this way, 28 additional articles were added to the sample.
5. **Final assessment:** In this step, all papers contained in the current sample were completely read to assess their relevance in light of the exclusion criteria defined above. Although scheduling problems in all fields are similar to some extent, we excluded works on project scheduling, personnel scheduling, timetabling, scheduling in agriculture, and computer scheduling. The sample was reduced to a size of 132 papers at the end of this step.

2.1.2. Descriptive analysis of the sample

The articles contained in the final sample were published between 1959 and 2016 in 41 different scientific journals. Figure 1 shows the distribution of the publication year of the sampled papers. As can be seen, reviewing the literature on MSPP attracted attention of researchers with an increasing trend over the years. This is in line with the findings of several secondary works on the subject of operations research and management science (e.g., [37,38]), which observed that the number of primary works published in the field has increased over time. 48 of the surveys (i.e., around 36% of the whole sample) were published during the last 10 years, which underlines the ongoing significance of the topic. Figure 2 summarizes the distribution of survey papers across scientific journals. The *European Journal of Operational Research, Omega,* the *International Journal of Production Research,* and *Operations Research* were the four most popular outlets for literature surveys on MSPP, and together they published more than 40% of the sampled papers.

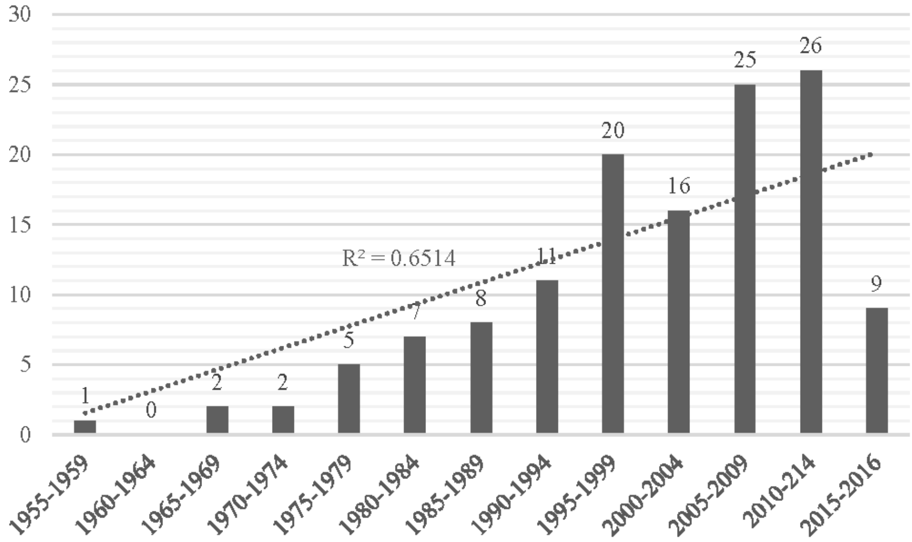


Figure 5: Number of published surveys per year.

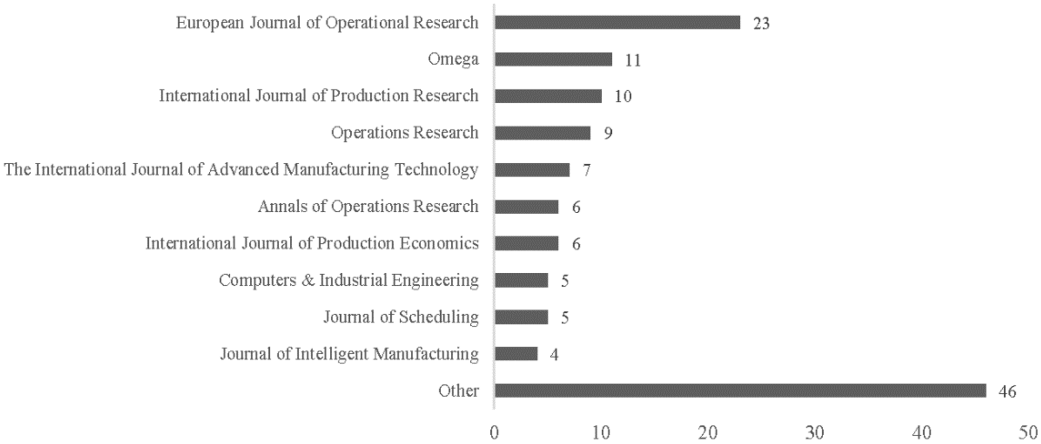


Figure 6: Number of published surveys per journal.

2.2. Recording and coding units

After the literature sample has been generated, it needs to be coded. Coding is defined as the process where raw data is transformed into standardized form suitable for machine processing and analysis. The literature differentiates between coding of *manifest content* (*objective*) and *latent content* (*subjective*) [28]. Manifest coding relies on the objective appearance of keywords in the sample, and it can thus easily be performed using computer software. Latent coding, in turn, is based on the researchers' interpretation and is therefore difficult to automate.

To ensure that the results of this analysis can be reproduced, we applied a manifest coding procedure to our sample. As recording units, we selected words (e.g., "makespan"), abbreviations (e.g., "HFS", which stands for "hybrid flow shop") and symbols (e.g., "Cmax", which stands for maximum completion time) for our analysis. According to Cullinane and Toy [25], this set of recoding units is the most widely applied one for analysing manifest content (Other types of recording units, such as characters and paragraphs, are described in Frankfort-Nachmias et al. [39]). The appearance frequency of the recording units in the sample are the objects of our analysis. At the end of the search phase of the CA, the recording units that are used more frequently in the sample are rated as more important.

2.3. Categorization

In a CA, the analytic categories used for classifying the coded material are derived either deductively or inductively [40]. In the *deductive approach*, categories (in the case of this paper: keywords, and the groups and subgroups that these keywords are assigned to) are defined before conducting the CA. The sample is then analyzed to see how pervasive and thus appropriate the categories are and to gain insights into the relative importance of each category. In the *inductive approach*, in contrast, the categories are derived directly from the analysis of the sample.

In this paper, the categories were identified through both inductive and deductive approaches to ensure that the categories are reliable, exclusive and exhaustive as proposed by Cullinane and Toy [25]. In the following subsections, the categories obtained are described in detail.

2.3.1. Categories obtained through the deductive approach

Each paper contained in the sample was carefully examined to identify the focus and methodology of the paper. In this way, 118 keywords were derived as representatives of the main attributes of MSPP. The occurrences of the keywords in the sample were then determined using the software MAXQDA 11 (a software for analyzing structured and unstructured data such as books, audio recordings and videos). Subsequently, the keywords were arranged into seven groups with the following labels:

- **Type of problem**: comprises keywords describing fundamental characteristics of a particular MSPP, such as whether the problem is deterministic or stochastic, static or dynamic, offline or online, or periodic or non-periodic.
- **Flow pattern**: contains keywords that describe the way jobs are routed through the production system.

- **Objectives**: contains keywords that refer to the objective(s) of MSPP. A particular MSPP is concerned with the optimization of either one or a set of objectives, and, generally, the objectives are either cost-based, penalty-based, or based on throughput time.
- **Job and machine characteristics**: consists of keywords dealing with job- or machine-related assumptions. Examples include "pre-emption" or "machine breakdowns".
- **Solution approaches**: summarizes keywords on the solution method used to address the MSPP. Exact algorithms, simple heuristics, and metaheuristic approaches are examples of frequently-used solution methods.
- **Scheduling in practice**: keywords summarized in this group are related to the application of scheduling models to real-life problems.
- **Pure theory**: consists of keywords exclusively related to theoretical/mathematical aspects of MSPP (e.g., keywords related to the computational complexity of a problem). The name of this group is taken from Reisman et al. [20].

To facilitate interpreting the results obtained by the CA, the groups of keywords discussed above are further divided into a total of 34 subgroups with the intention of identifying additional concepts within the groups. Subgroups are defined in such a way that keywords contained in a subgroup refer to a similar concept. For instance, the keywords "makespan", "flow time", and "completion-time" belonging to the category "objectives" were categorized into a subgroup "throughput-time- based objectives". Where necessary, the subgroups defined in this step were refined during the inductive approach.

2.3.2. Categories obtained through the inductive approach

The *inductive* approach was carried out by counting all words, abbreviations, and symbols (which will henceforth be referred to as "words" for the sake of simplicity) in the sampled papers using the software MAXQDA 11. All function words, i.e., conjunctions, prepositions, pronouns, and single letters were excluded using a filter option provided by the software.

About 83,000 different words were found by the software during the screening of the sampled papers. All identified words were then examined and re-examined in several sessions by the authors of this paper to filter out words related to MSPP. These words were then added to the list of keywords (and thus to the groups and subgroups) defined in the deductive step. Some of the newly found keywords made it necessary to extend or refine the subgroups formulated in Section 2.3.1 (e.g., keywords related to learning effects were not matched perfectly with any of the subgroups obtained by the deductive approach and were therefore summarized in a new subgroup in the inductive step). New subgroups that were defined in this step were created using the same method as in the deductive step. In addition, a number of keywords and subgroups defined in the deductive approach were removed because the inductive approach did not confirm their frequent usage in the sampled papers (For example, "random search" as a solution approach was not discussed in the sampled texts at all, and therefore the corresponding keywords were removed). During the inductive approach, the number of keywords strongly increased to 610 and the number of subgroups to 48.

It is worth mentioning here that examining each word identified in the inductive approach individually enabled us to evaluate which "general" words have a specific meaning in scheduling. For instance, the word "identical", which was identified during the screening of the sampled papers, at first glance appears to be a general word without a specific meaning in scheduling, so one might tend not to consider it further in defining subgroups. Yet, this word has a specific meaning in the scheduling context because it is used to describe the machine-related assumptions of specific parallel machines problems. Therefore, we decided to replace the word “identical” by more scheduling-related words, such as “identical-machine”, “identical-processor” and “identical-server” (we also added other spelling options for these new keywords, e.g. for “identical-machine” we also checked “identical machine”, “identical_machine”, and also the plural of the keywords) to ensure that only words used in a scheduling context are classified as keywords.

During the screening of the sampled papers, certain keywords had been identified in different spellings. Therefore, we summarized words with an identical meaning (but different spellings) in a single keyword and thus reduced the total number of keywords to 179.

2.4. Results of the CA

During the deductive and inductive stages of the CA, a total number of 179 keywords related to MSPP were found that were allocated to 7 groups and 48 subgroups. The comprehensive list of keywords and subgroups is presented in the Appendix. The subgroups are not mutually exclusive but instead reflect the general structure of MSPP. For example, a three- machine problem can be a job shop or a flow shop problem. The identified keywords appeared a total of 78,701 times in the sampled papers. Fig. 3 summarizes the share of each group of keywords in the sampled papers in the total keyword count.

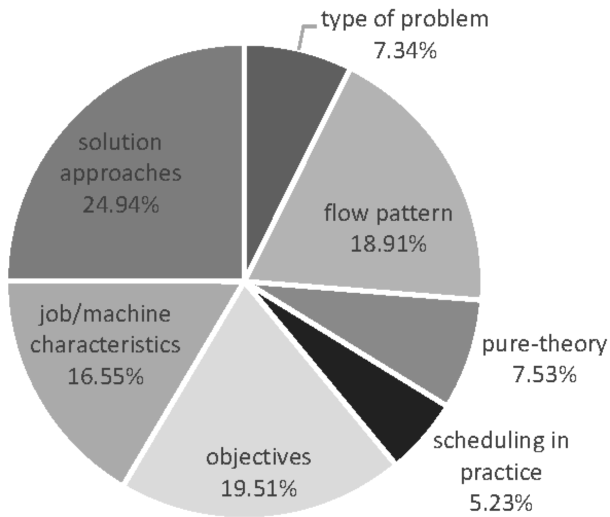


Figure 7: Shares of the keyword groups in the total keyword count of the sampled papers.

Fig. 3 illustrates that most keyword hits were obtained for the groups “solution approaches“, “flow pattern”, “objectives”, and “job/machine characteristics”, which together accounted for around 80% of the total keyword hits. We can infer from these numbers that prior secondary publications on MSPP had a strong focus on these four groups, as compared, for example, to the application of scheduling in practice, which accounted only for 5.23% of the total keyword hits. Given the large number of surveys contained in our sample, the share each keyword has in the total number of keyword hits may indicate the relative importance researchers attributed to the different dimensions of MSPP. The results for each keyword group will be presented in detail in the following subsections.

Table 1 presents another look at the sample by giving an overview of the six most-discussed subgroups. As can be seen, these six subgroups generated 46.27% of the total hits in the sample. The remaining 53.73% of the keyword hits are distributed over the other 43 subgroups. This could imply that the subgroups displayed in Table 1 were seen as particularly important by researchers in the past.

Subgroup	Contained in group	% of total hits in sample
Simple heuristics	Solution approaches	10.06
Penalty-based objectives	Objectives	8.35
Metaheuristics	Solution approaches	8.31
Theory of complexity	Pure theory	7.57
throughput-time-based objectives	Objectives	6.27
Setup time	Jobs/machine characteristics	5.70
Total		46.27

Table 1: Most discussed subgroups in the sampled papers and the groups they belong to.

2.4.1. Type of problem

The group “type of problem” consists of 11 keywords divided into 7 subgroups, and it is one of the groups that received the least attention in the sample with only 7.34% of the total keyword hits. Because this group contains keywords that describe fundamental assumptions of MSPP, one could have expected that it would have received more attention in the sampled papers. However, analyzing the sampled papers in more detail revealed that in most cases, such assumptions were only briefly explained or even only listed in early sections of the papers without a detailed discussion. One possible reason is that the assumptions determining the type of the problem have been discussed comprehensively in early papers, such that later works saw no need to discuss them in detail again.

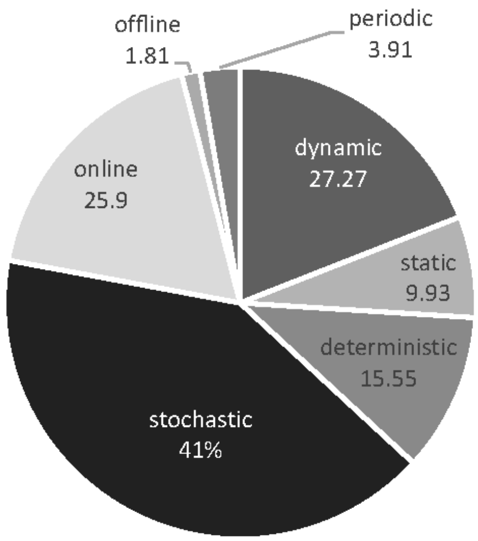


Figure 8: Subgroups of the group "Type of problem" and their shares in the keyword count.

Fig. 4 gives an overview of the subgroups of "type of problem" that emerged from the sample. As can be seen, our results indicate that dynamic and stochastic scheduling problems received more attention than static and deterministic problems in the sampled survey papers. Thus, our results do not confirm the general opinion expressed in the literature that the majority of scheduling models are deterministic and static (see, e.g., [9,41]). There are some possible reasons for this result. One could be that deterministic and static MSPP models can easily be described with the help of standard assumptions and terms, such that researchers would only use keywords on the "type of problem" in case standard assumptions are violated (i.e., when the parameters are stochastic and/or dynamic). The second reason could be that dynamic and stochastic scheduling problems form a larger part of the literature on MSPP than assumed by many researchers, which has led to a significant number of surveys on this topic.

Offline and periodic scheduling problems did not receive much attention in survey papers on MSPP, which could point to research opportunities in this area.

2.4.2. Flow pattern

The group "flow pattern" contains 38 keywords that were divided into 11 subgroups. The flow pattern describes the way jobs are routed through the production system [42], and it can regulate

- the type of jobs and machines in the production system. It determines, for example, if the problem deals with continuous jobs (known as *process scheduling*, which is common in chemical and metal industries), whether or not machines are able to process more than one job at a time (known as *batch machine scheduling*, which is common in metal casting and machinery industries), or if the products are being manufactured and delivered in lots (known as *lot scheduling*).
- the configuration of machines in production systems, e.g., whether the production system is composed of a *single machine* or whether there are multiple machines in use. In the case of multiple machines, the flow pattern determines if these machines perform the same task (such

systems are referred to as *parallel machines* production systems) or not. When the machines process different tasks, there are several possibilities to route jobs through the system, such as routing them according to one of the following flow patterns: *flow shops* are production systems in which the routing is predetermined and identical for all jobs; *job shops* are systems with predetermined but individual routings for each job; and *open shops* set no limitation on the routing such that the operations of a job can be processed in any sequence (for more details on different machine configurations for MSPP, see Pinedo [43]).

Fig. 5 gives an overview of the subgroups of the group “flow pattern”. As can be seen, five flow patterns received a particularly high attention in the sampled texts, namely batch machine, flow shop, job shop, single machine, and parallel machines problems. These five subgroups accounted for around 75% of the total keyword hits obtained for this category.

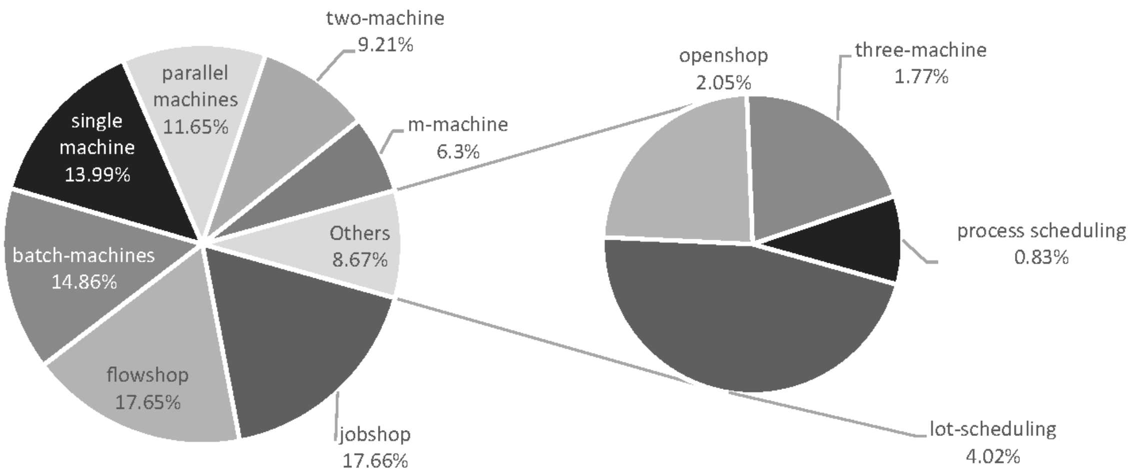


Figure 9: Subgroups of the group “flow pattern” and their shares in the keyword count.

The “single machine” problem is the most basic MSPP, and the literature describes it as a building block for modelling and solving more general and complicated problems. Thus, one could have expected that this problem received the most attention in MSPP surveys. Surprisingly, in the sampled texts, it received fewer keyword hits than three other flow patterns, namely batch machines, flow shop and job shop. One possible explanation for this result is that the single machine MSPP is well-researched and nowadays primarily used to develop more general problems, such that researchers tend not to dedicate entire surveys or long sections of surveys to discussing the state-of-the-art of this problem. Another explanation could be that in early works, no strict standard terminology for flow patterns of MSPP had been developed yet. For example, the word “job shop” was in use to address different types of flow patterns for many years before the terminology for MSPP became more specific: Sisson [44] used the term “job shop” to refer to a parallel machines problem, and some models discussed in Baker [45] as “job shop” problems are single machine problems.

Most flow shop problems consist of multiple stages with each stage containing only a single machine for processing jobs. To increase the flexibility of the production system or to balance the capacity of the stages, hybrid flow shops (HFS) consider multiple machines at several stages [46]. In the sample, only 9.17% of the keyword hits belonging to the flow shop subgroup refer to HFS. This could indicate that HFS might be an interesting candidate for future literature reviews.

The subgroup “parallel machines” consists of four different clusters of keywords: general terms related to parallel ma- chines problems, identical machines, uniform machines, and unrelated machines. The last three clusters refer to parallel machines problems with specific machine characteristics (for a detailed description of different parallel machines scheduling problems, see Cheng and Sin [47]). As Fig. 6 shows, the identical machines problem, which is the most basic type of a parallel machines scheduling problem, is responsible for 12.69% of the hits and thus has been discussed more frequently than uniform machines and unrelated machines problems. One possible reason for the popularity of identical machines scheduling problems in the sampled papers is that some algorithms that were developed for identical machines problems can be extended to or at least can give some insights into developing effective algorithms for more complicated scenarios considering uniform/unrelated parallel machines.

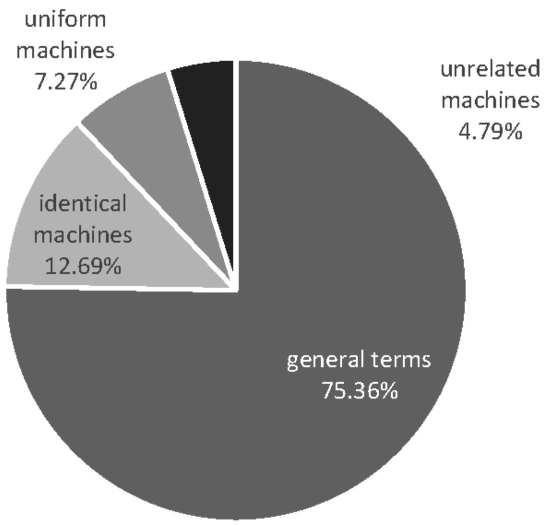


Figure 10: Different clusters of keywords for the subgroup “parallel machines” and their shares in the keyword count.

2.4.3. Objectives

Another important dimension of MSPP are the objectives considered in the model formulation. For this group, we found 20 keywords in the sample and assigned them to five subgroups (see Fig. 7). In general, the objectives of MSPP are either cost-based, penalty-based, based on throughput-time, or a combination of—usually conflicting—objectives leading to a multi- objective problem.

More than 40% of the hits of objective-related keywords were obtained for penalty-based objectives such as minimizing tardiness, lateness, or earliness of jobs. As Table 1 shows, penalty-based objective is the second most discussed subgroup in the sample, which highlights its popularity in secondary researches. One interesting observation is that among all keywords related to penalty-based objectives, “earliness” received the least attention in the sampled surveys. One possible reason for this may be that the earliness of jobs is not a critical objective in many practical problem settings, which could have led to infrequent discussions in primary studies and, in turn, to less attention in secondary works.

Objectives based on the throughput-time of jobs can be further divided into “flow time” and “makespan”. Flow time is defined as the time each job needs to be completed, and in MSPP in most cases the total flow time (i.e., the sum of the flow times of all jobs) is considered as an objective. Makespan, in turn, is the completion time of the last job. In case all jobs are ready to be processed at time zero, the makespan is identical to the maximum flow time. As Fig. 7 indicates, makespan has been discussed much more often than flow time in the sampled papers.

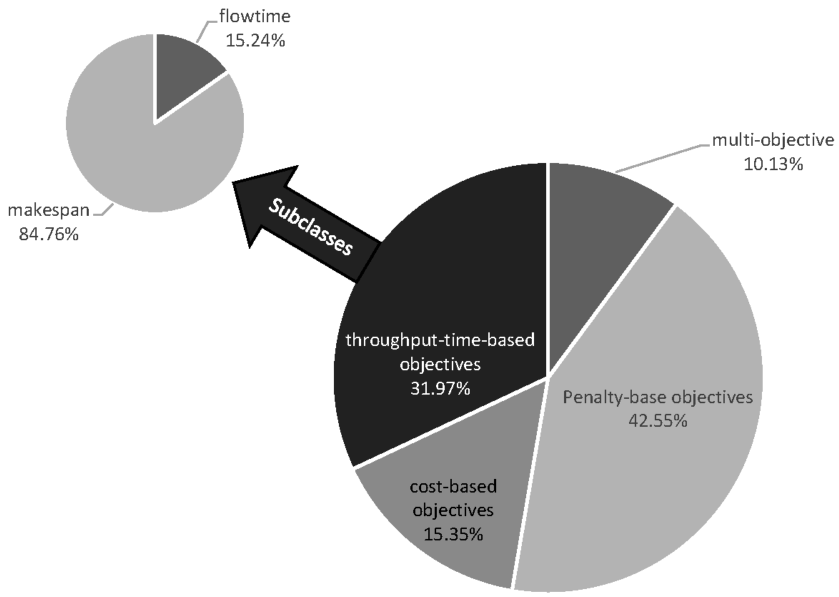


Figure 11: Subgroups of the group “objectives” and their shares in the keyword count.

In most real-life problems, the decision maker follows more than a single objective. Thus, it is surprising that only 10.13% of the keywords found in the group “objectives” refer to multi-objective problems.

Summarizing, the keywords pertaining to earliness and flow time do not appear as frequently as other keywords in their corresponding subgroups (i.e., “throughput time-based” and “penalty-based objective”, respectively) in the sample. A similar observation was made regarding multi-objective problems, which is a subgroup that has received less attention than other subgroups in the group “objectives”. These findings could imply three potential research gaps either on the primary or on the secondary level.

2.4.4. Job and machine characteristics

This group consists of 37 keywords distributed over 14 subgroups. 43% of the keyword hits of this group were obtained for the seven subgroups that belong to job-related characteristics, while the remaining 57% of the keyword hits were obtained for the other seven subgroups that comprise machine-related characteristics. The job/machine-related characteristics determine the complexity and often influence the practical applicability of the considered problem variant. For example, considering limited buffer space, which is a realistic assumption in most cases, can increase the complexity of a problem dramatically and at the same time make it more applicable to real-world problems. Fig. 8 presents the results for this group divided into job- and machine-related characteristics.

Setup time is defined as the time required to prepare machines for processing jobs, e.g., the time needed for loading/unloading machines. Although we assigned setup time to the subgroup "machine-related characteristics", it is worth noting that setup time may also be job-related. Our findings show that setup time-related keywords are the most frequently discussed machine-related characteristic, and they are the sixth most discussed subgroup in the sample (see Table 1). It is also worth noting that the keyword "setup" is the most frequently used keyword in the entire sample, and this keyword alone (out of the total 78,701 keyword hits recorded) generated 5.67% of the total keyword count.

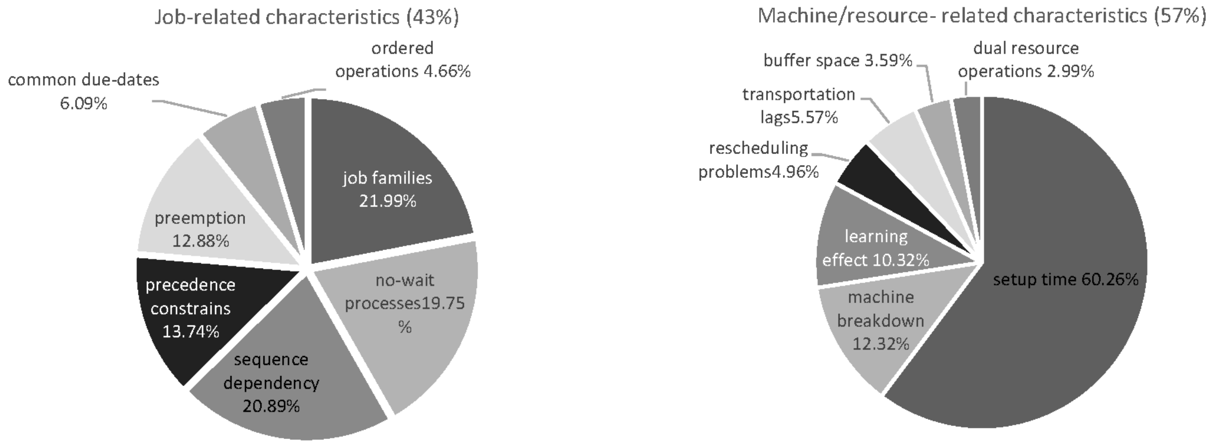


Figure 12: Subgroups of the group "Job and machine characteristics" and their shares in the keyword count.

Another interesting point is that some of the subgroups of both job- and machine-related assumptions can be interdependent. For example, considering "learning effects" may change the rate of "machine breakdowns" or the length of "setup times", and considering "job families" can create "sequence dependencies" as well as "precedence constraints". Fig. 9 schematically presents some more examples for interdependencies that may arise between job- and machine-related assumptions. One opportunity for future research might be to conduct a survey on MSPP studies to explore the effect of different job-/machine-related assumptions on each other by investigating the interdependencies mentioned above. Another possibility might be to conduct surveys devoted to the job- and machine-related characteristics that were discussed only infrequently in our sample (for example, transportation lags or dual-resource operations).

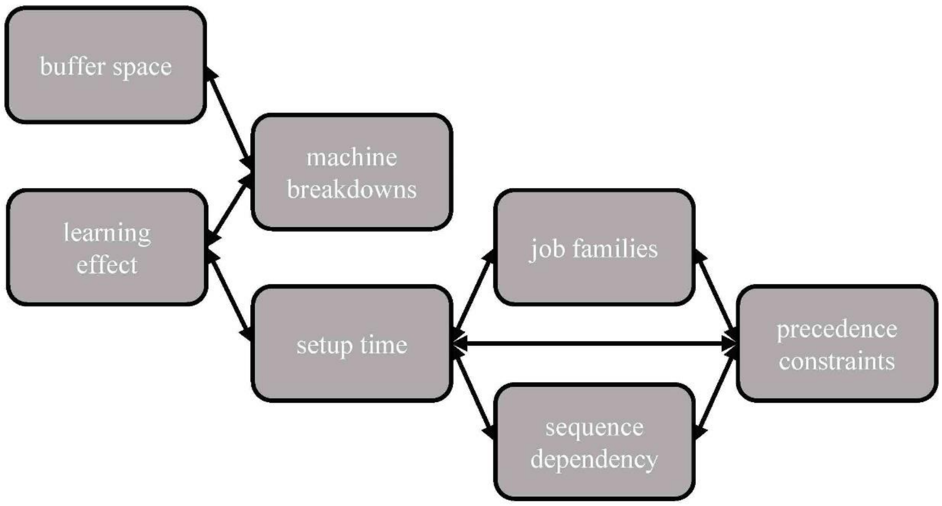


Figure 13: Examples of interdependencies that may arise between job- and machine-related assumptions.

2.4.5. Solution approaches

Another important dimension of research on MSPP is the approaches used for solving the problem. The keyword group "solution approaches", which contains 50 keywords building seven subgroups, is not only the category with the highest number of keywords, but it is also the group that received the most keyword hits in the sample (see Fig. 3). As Table 1 shows, "solution approaches" is one of the groups of keywords that contains two of the top six most discussed subgroups. Fig. 10 illustrates that simple heuristic algorithms, metaheuristics and exact algorithms triggered around 85% of the keyword hits of the group "solution approaches".

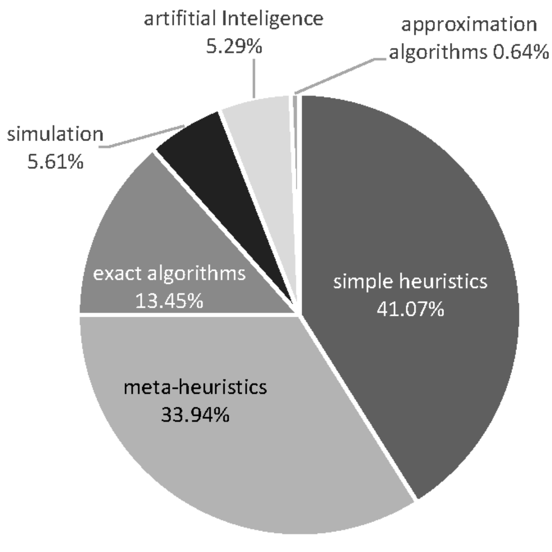


Figure 14: Subgroups of the group "solution approaches" and their shares in the keyword count.

For the subgroup “exact algorithms”, we distinguished between five clusters of keywords. As Fig. 11 illustrates, keywords related to branch-and-bound methods build the most frequently used cluster in the sample.

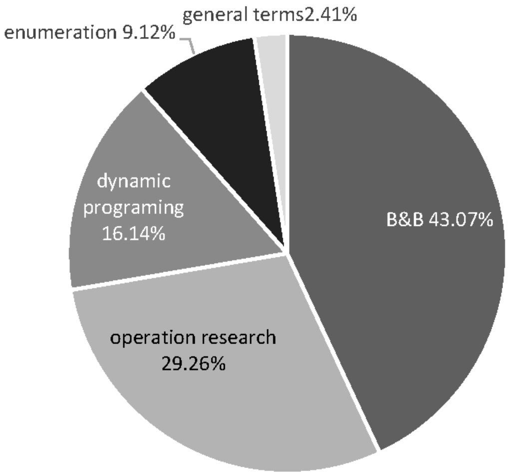


Figure 15: Distribution of keyword counts of the subgroup “exact algorithms”.

Simple heuristics are the most frequently used type of solution approach for scheduling jobs in MSPP. They are simple to implement, easy to understand and require a low computational effort [48]. According to the CA conducted in this paper, such algorithms have not only been discussed more than any other solution approach in the sampled review papers, but they also represent the most discussed subgroup as illustrated in Table 1. The subgroup “simple heuristics” is divided into “general terms” and “specific simple heuristics”. The first category includes general keywords related to simple heuristics that do not refer to a particular heuristic (such as “heuristic” or “priority rules”). The latter category includes specific keywords that refer to a particular heuristic (such as “Johnson’s rule” or “FIFO”). Around 76% of the total keyword hits belong to “general terms”, while the rest are related to “specific simple heuristics”. 13 different specific simple heuristics were identified in the sampled text, among them “shortest processing time” (SPT), “first come first served” (FCFS), “shortest imminent” (SI), “total work-content” (TWK), and “constant” (CON). As Fig. 12 demonstrates, the SPT and “Johnson’s rule” were discussed more frequently than any other simple heuristic in the sampled review papers (for a detailed explanation of the heuristics displayed in Fig. 12, see Blackstone et al. [49] and Haupt [50]).

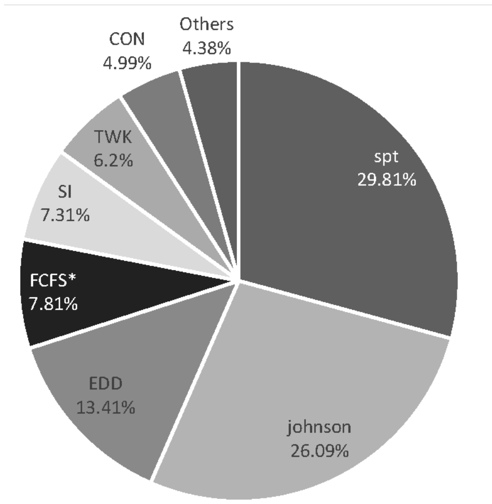


Figure 16: Share of specific simple heuristic algorithms in the keyword count of the subgroup "simple heuristics".

Similarly, to the subgroup "simple heuristics", the subgroup "metaheuristics" can also be divided further into "general terms" and "specific metaheuristics". "General terms" includes 33% of the keywords hits obtained for the subgroup "meta- heuristics", and the rest of the keywords belong to "specific metaheuristics". The keyword hits for specific metaheuristics could be assigned to ten different metaheuristic algorithms. According to Fig. 13, genetic algorithm (GA) is the most dis- cussed metaheuristic in surveys on MSPP with more than 50% of the total keyword count in this subgroup. Tabu search (TS), simulated annealing (SA), particle swarm optimization (PSO), ant colony optimization (ACO), and variable neighborhood search (VNS) are the next most discussed metaheuristics in the sampled texts with a total of about 40% of the keyword hits. The rest of the keyword hits (with a share of less than 6%) were obtained for "memetic", "scatter search" (SS), "GRASP", "guided local search" (GLS), "iterated local search" (ILS), and "adaptive memory programming" (AMP).

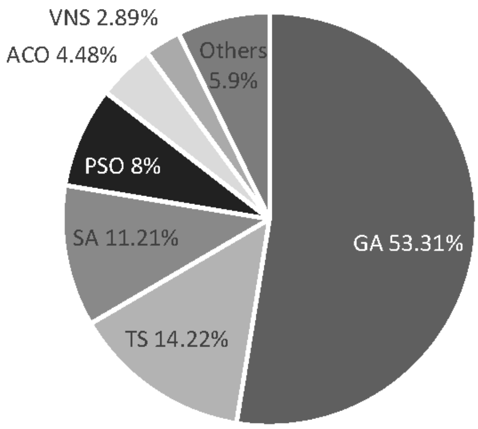


Figure 17: Share of specific metaheuristics in the keyword count of the subgroup "metaheuristics".

2.4.6. Scheduling in practice

16 keywords referring to the application of scheduling in practice were assigned to this group, and as Fig. 3 shows, this is the least-discussed group in the sample. The keywords for "scheduling in practice" were divided into two subgroups. The first subgroup contains general application-related terms (e.g., "real-life", "real-world" and "industry"), and they accounted for around 25% of the total number of keyword hits obtained for this group. The second subgroup was termed "industrial application of scheduling" and contained keywords such as "time-tabling", "chemical", "textile", or "automobile industry". The other 75% of hits were obtained for the second subgroup (see Fig. 14 for an overview).

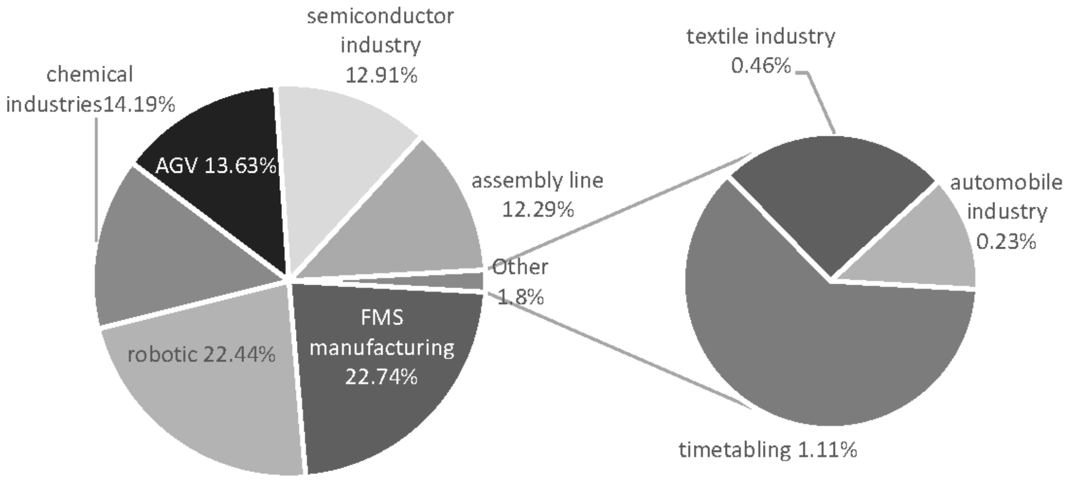


Figure 18: Distribution of the keyword count of the subgroup "scheduling in practice".

One important finding is that the application of MSPP in a real industrial context has not received much attention in the sampled papers as compared to other aspects of MSPP, which confirms the results of several review papers on MSPP that hypothesized that more application-oriented research is necessary (see, for example, McCarthy and Liu [51]). These results could imply that reviewing the primary literature with a specific focus on the application of scheduling models to real-life problems might be promising and that more application-oriented research is possibly needed also on the primary level.

Another interesting finding is that the scheduling of robotic manufacturing received more attention than any other sub-group. It is worth noting that problems with automated guided vehicles (AGVs) are often also treated as robotic manufacturing problems; however, we separated them from robotic manufacturing to highlight their role in the literature on MSPP.

2.4.7. Pure theory

This group includes only seven keywords (which makes it the smallest group with respect to the number of keywords) building two subgroups, namely "general theory" with keywords such as "theory" and "theorem", and "complexity theory" (e.g., "complexity", "NP-hard", and "NP-complete"). 21% of the keywords counted in the group "pure theory" belong to the subgroup "general theory", while the rest belongs to the subgroup “complexity theory”. As Table 1 shows, an interesting finding is that the keywords related to “complexity theory” with 7.57% of the total keyword hits build the fourth most discussed subgroup in the sampled papers. This reflects the relative popularity of the theory of complexity in survey papers on MSPP as compared to the industrial applications of scheduling models, for example.

3 Discussion

This section presents the results of a sensitivity analysis conducted to examine the reliability of the CA results. It also discusses some limitations observed in the sampled papers together with some general limitations of CAs.

3.1. Sensitivity analysis

Content analyses have often been criticized with respect to the reliability of results (see, for example, Neuendorf [21]). To validate the results obtained in this paper and to gain insights into the extent to which individual papers may bias the results obtained, a sensitivity analysis is conducted based on the method proposed by Grosse et al. [26]. The idea of the sensitivity analysis is to identify papers that use specific terms very frequently, which may lead to an overrepresentation and incorrect interpretation of individual recording units. To assess the validity of results, Grosse et al. [26] selected a set of papers with a specific focus on recording units that had frequently been found in their sample, removed these papers from the sample and conducted the CA again to isolate the influence of these papers on the results of the CA. In contrast to Grosse et al. [26], in this study, a specific set of papers was selected for every single subgroup. Because the resulting sets consisted of papers with a specific focus on the respective subgroup, these papers could have introduced biases into our earlier analyses. For example, our sample contains a single survey with a focus on ordered operations in scheduling studies. This survey produced so many keyword hits related to ordered operations that it may have introduced biases into the results of the CA.

To identify whether a survey is dedicated to a specific subgroup, the title of each of the sampled papers was checked for subgroup keywords. For example, the paper entitled “Parallel machine scheduling with additional resources: Notation, classification, models and solution methods” was allocated to the “parallel machines” and “dual resource operations” sub-groups because the title contains keywords from both subgroups. Table 2 presents subgroups with more than five dedicated papers in the sample (in the tables that follow, # indicates the number of dedicated papers). Four of the subgroups in Table 2 belong to the group “flow pattern”, which is the group that has the highest number of dedicated papers (41 out of 132 papers in the sample).

Group	Subgroup	#	Group	Subgroup	#
Flow pattern	Flowshop	17	job & machine characteristics	Setup time	8
Flow pattern	Jobshop	13	Flow pattern	Single machine	6
Objectives	Penalty-base objectives	12	Pure theory	Pure theory	5
Objectives	Multi-objective	11	nature of problem	Deterministic	5
Scheduling in practice	Scheduling in practice	9	nature of problem	Stochastic	5
Solution approaches	Simple heuristics	9	flow pattern	lot-scheduling	5

Table 2: Subgroups with more than five dedicated papers.

In the next step, to evaluate how the results of the CA were influenced by the dedicated papers, we counted the keyword frequency of each subgroup in the corresponding dedicated papers. Table 3 illustrates the ten subgroups for which the strongest dependency on their dedicated papers could be observed.

According to Table 4, the subgroup "flowshop" with around 64–70% has the highest dependency on its dedicated papers. The percentage for the subgroup "ordered operation" is approximately 40.84%. However, as shown in Table 3, there are 17 dedicated papers for the former subgroup and only one paper for the latter. As a result, the numbers in Table 4 might be misleading. To remedy this drawback, the average dependency of subgroups on their dedicated papers was calculated by dividing the percentage dependency on dedicated papers of each subgroup by the number of dedicated papers of this subgroup. Table 4 summarizes the ten subgroups with the highest average dependency (AVE).

Subgroup	#	% dependency on dedicated papers	Subgroup	#	% dependency on dedicated papers
Flowshop	17	64.07	Stochastic	5	53.69
Multi-objective	11	62.89	Dual resource operations	4	46.61
Lot-scheduling	5	60.60	ordered operations	1	40.84
Artificial Intelligence	3	56.51	learning effect	2	36.91
Setup time	8	54.43	process scheduling	4	36.59

Table 3: Ten subgroups with the highest percentage dependency on their dedicated papers.

The average dependency measure facilitates the interpretation of the results of the CA. For example, the subgroups “artificial intelligence” and “dynamic” are responsible for 1.29% and 1.4% of the total keyword hits obtained in the sample. Yet, while the average dependency of the first subgroup on its dedicated papers is 18.84%, the AVE for the second subgroup is only 4.75% This might be interpreted as follows: Although both subgroups occurred with a similar frequency in the sample, the “dynamic” subgroup occurred more evenly distributed across the sample as compared to the subgroup “artificial intelligence”. The list of all subgroups with the number of their dedicated papers as well as the percentage of their dependency on dedicated papers and the average dependency can be found in the Appendix.

Subgroup	#	AVE	Subgroup	#	AVE
Ordered operations	1	40.84	Common due-dates	2	13.74
Online	1	22.18	Approximation algorithms	2	12.70
Artificial Intelligence	3	18.84	Lot-scheduling	5	12.12
Learning effect	2	18.46	Dual resource operations	4	11.65
Job families	1	15.29	No-wait processes	2	11.26

Table 4: Ten subgroups with the highest average percentage dependency on their dedicated papers.

3.2. Limitations of the sample

3.2.1. Lack of application in scheduling studies

Reisman et al. [20] revealed an interesting fact about 50 years of literature on flow shop scheduling: “Over its lifetime, which is now in its fifth decade, the entire literature of FSS (flow shop scheduling/sequencing) has recorded only five articles (3% of the total) that were judged to be true applications .”Although the scope of the paper at hand is much broader than that of Reisman et al. [20] , the results of this CA also strongly indicate that the literature suffers from a lack of application: only 5.23% of the sampled papers investigated the application of scheduling in reality. Based on the results of our CA, the following research opportunities may contribute to increasing the applicability of MSPP studies:

- develop methods for multi-objective problems to take account of the fact that decisions in practice are often based on multiple objectives (only 1.98% of the total hits in our sample are related to multi-objective problems).
- focus on more realistic constraints for MSPP. Transportation lags and dual/multi-resource operations are two examples of such constraints (they accounted for only 0.52% and 0.28% of the total hits in our sample, respectively).
- develop models that tackle specific industry-related problems. For example, we found that MSPP have been studied in the automobile and textile industries, but that the application of MSPP to these industries accounted for only 0.03% of total hits in the sample.

- study the use of MSPP models in practice and report how the use of such models influences the production processes.

3.2.2. Missing metaheuristics

As was mentioned in Section 2.4.5, some well-known metaheuristics for addressing combinatorial optimization problems (such as scatter search, guided local search, iterated local search, and adaptive memory programming) collectively appeared only 147 times (i.e., 0.18% of total hits) in the sampled papers. Furthermore, there are some other well-known metaheuristics (such as large neighborhood search) that did not appear in the sample at all. This could be interpreted in two ways: either these algorithms are not well suited for solving MSPP, or they are underrepresented in solving it, which could indicate that further research in this area is promising.

3.3. Limitations of the CA

Conducting a CA on a sample consisting of secondary studies enables researchers to obtain a general overview of the subject under consideration. The results are also helpful for developing a framework to classify the literature. This framework, in turn, can help researchers to position their own works in the literature. However, one drawback is that the results obtained cannot necessarily be extended to the primary level. To illustrate this aspect, consider an example from the results of the paper at hand. As discussed in Section 2.4.4, transportation lags have not been discussed in the sampled papers very frequently. However, despite the correlations that usually exist between research at the primary and the secondary level, we cannot reliably infer from these results that transportation lags have not been investigated on the primary level. The only valid conclusion is that this particular assumption has been reviewed less intensively than other machine-related characteristics in the literature. Another disadvantage of conducting a CA on the secondary level is that surveys are not published on a regular basis and with delays regarding the initial publication of the primary works. Therefore, it is difficult to draw conclusions concerning the development of different subjects over time from the results.

Using a “manifest coding” approach ensures that the results of a CA are reproducible. However, in order to make the results of a CA that uses “manifest coding” reliable, a standard terminology for the entire sample is required. In the case of our sample, a standard terminology was not available for all subgroups. For example, as explained in Section 2.4.2, in some of the sampled papers, the term “job shop” has been used for describing a “single machine” flow pattern. Such violations bias the results, especially if they occur frequently. Using “latent coding” is one effective way to overcome this drawback and to make the results of a CA more reliable for samples in which the standard terminology is often violated. In our CA, a low level of violations was observed, which is why the manifest coding scheme was used.

Another way to improve the reliability of the results could be to assign different weights to different subgroups found in the sampled papers. In this case, keywords that appear in the titles of the sampled papers could receive higher weights than the keywords that appeared in the main text of the papers, for example to correctly reflect the fact that a key- word mentioned in the title may reflect the overall importance of the keyword somewhat stronger than a keyword that appears in the regular text. The challenge here clearly would be to assign reasonable weights to the different places of occurrence.

4 Summary and future research

The paper at hand analyzed secondary works on MSPP by conducting a CA on a comprehensive and systematically generated sample consisting of 132 surveys. The manifest coding technique was used to identify the keywords. To ensure that the process of categorizing recording units is exclusive and exhaustive, both inductive and deductive categorization approaches were employed. As a result, 179 keywords were identified in the sample, which were allocated to 7 groups and 48 subgroups. These groups and subgroups establish a framework that helps to classify the literature on MSPP. The results of the CA revealed that the most discussed attributes of MSPP are the solution approaches, objectives, and flow patterns. The results also indicate a lack of practical applications in the sampled papers. To verify the reliability of the results, a sensitivity analysis was conducted to identify surveys that may have introduced biases into our results. The limitations of the CA conducted in this paper together with some suggestions to remedy these drawbacks were also discussed.

In the discussion of the results of our CA, several opportunities for future research were identified. Another interesting opportunity might be to apply the CA procedure proposed in the paper at hand to a sample of primary studies on MSPP published in flagship journals over a certain timespan. Such a study could compare the results obtained for primary works with those obtained for secondary works in the paper at hand. It could also be used to check whether the framework to classify MSPP studies developed in this paper is comprehensive. Another opportunity for future research could be to conduct a tertiary study on the sample studied in the paper at hand. This might help to gain further insights into the state-of-the-art of MSPP.

Acknowledgment

The authors are grateful to the associate editor and the anonymous reviewers for their constructive comments on an earlier version of this paper that helped to improve its quality significantly. The first and second authors further wish to thank the Carlo and Karin Giersch Stiftung for funding their research.

References

[1]. F. Fu, Integrated scheduling and batch ordering for construction project, Appl. Math. Model 38 (2014) 784–797.
[2]. J. Fliege, K. Kaparis, B. Khosravi, Operations research in the space industry, Eur. J. Oper. Res. 217 (2) (2012) 233–240.
[3]. A. Mishra, A .K. Tripathi, A Monte Carlo algorithm for real time task scheduling on multi-core processors with software controlled dynamic voltage scaling, Appl. Math. Model 38 (2014) 1929–1947.
[4]. A. Mishra, A .K. Tripathi, Energy efficient voltage scheduling for multi-core processors with software controlled dynamic voltage scaling, Appl. Math. Model 38 (2014) 3456–3466.
[5]. C. Özgüven, B. Sungur, Integer programming models for hierarchical workforce scheduling problems including excess off-days and idle labour times, Appl. Math. Model 37 (22) (2013) 9117–9131.
[6]. H. Aytug, M.a. Lawley, K. McKay, S. Mohan, R. Uzsoy, Executing production schedules in the face of uncertainties: a review and some future directions, Eur. J. Oper. Res. 161 (1) (2005) 86–110.
[7]. M. Ebrahimi, S.M.T. Fatemi Ghomi, B. Karimi, Hybrid flow shop scheduling with sequence dependent family setup time and uncertain due dates, Appl. Math. Model 38 (2014) 2490–2504.
[8]. P. Dileepan, T. Sen, Bicriterion static scheduling research for a single machine, Omega 16 (1) (1988) 53–59.
[9]. D. Ouelhadj, S. Petrovic, A survey of dynamic scheduling in manufacturing systems, J. Sched. 12 (4) (2009) 417–431.
[10]. V. Suresh, D. Chaudhuri, Dynamic scheduling-a survey of research, Int. J. Prod. Econ. 32 (1) (1993) 53–63.

[11]. A.S. Jain, S. Meeran, Deterministic job-shop scheduling: past, present and future, Eur. J. Oper. Res. 113 (2) (1999) 390–434.
[12]. M.L. Pinedo, Optimal policies in stochastic shop scheduling, Ann. Oper. Res. 1 (3) (1984) 305–329.
[13]. J.M. Framinan, J.N.D. Gupta, R. Leisten, A review and classification of heuristics for permutation flow-stop scheduling with makespan objective, J. Oper. Res. Soc. 55 (12) (2004) 1243–1255.
[14]. H. Hoogeveen, Multicriteria scheduling, Eur. J. Oper. Res. 167 (3) (2005) 592–623.
[15]. J. Behnamian, S.M.T. Fatemi Ghomi, Hybrid flowshop scheduling with machine and resource-dependent processing times, Appl. Math. Model 35 (2011) 1107–1123.
[16]. T. Kis, E. Pesch, A review of exact solution methods for the non-preemptive multiprocessor flowshop problem, Eur. J. Oper. Res. 164 (3 SPEC. ISS.) (2005) 592–608.
[17]. N.V. Shakhlevich, Y.N. Sotskov, F. Werner, Complexity of mixed shop scheduling problems: A survey, Eur. J. Oper. Res. 120 (2) (20 0 0) 343–351.
[18]. V. Garousi, M.V. Mäntylä, A systematic literature review of literature reviews in software testing, Inf. Softw. Technol 80 (2016) 195–216.
[19]. S. Hochrein, C.H. Glock, R. Bogaschewsky, M. Heider, Literature reviews in supply chain management: a tertiary study, Manage. Rev. Q 65 (2015) 239–280.
[20]. A. Reisman, A. Kumar, J. Motwani, Flowshop scheduling/sequencing research: a statistical review of the literature, 1952 –1994, IEEE Trans. Eng. Manage. 44 (3) (1997) 316–329.
[21]. K.A. Neuendorf, The Content Analysis Guidebook, SAGE Publications, Inc, 2002.
[22]. K. Krippendorff, Content Analysis: An Introduction to Its Methodology, third ed., SAGE Publications, Inc, 2012.
[23]. H. Uzunboylu, G. Asiksoy, Research in physics education: a study of content analysis, Proc. Soc. Behav. Sci. 136 (9) (2014) 425–437.
[24]. T. Erdogan, Research trends in dissertations on PBL: a content analysis study, Proc.—Soc. Behav. Sci. 197 (2015) 308–315.
[25]. K. Cullinane, N. Toy, Identifying influential attributes in freight route/mode choice decisions: a content analysis, Transp. Res. Part E: Logist. Transp. Rev. 36 (20 0 0) 41–53.
[26]. E.H. Grosse, C.H. Glock, W.P. Neumann, Human factors in order picking system design: a content analysis, Int. J. Prod. Res. (2016), doi: 10.1080/ 00207543.2016.1186296.
[27]. R.P. Weber, Basic Content Analysis, second ed., SAGE Publications, Inc, 1990.
[28]. E.R. Babbie, The Practice of Social Research, thirteenth International ed., Cengage Learning, Wadsworth, 2013.
[29]. C.H. Glock, E.H. Grosse, J.M. Ries, The lot sizing problem: a tertiary study, Int. J. Prod. Econ. 155 (2014) 39–51.
[30]. M.F.A. Gorman, “Metasurvey”analysis in operations research and management science: a survey of literature reviews, Surv. Oper. Res. Manage. Sci. 21 (1) (2016) 18–28.
[31]. D. Tranfield, D. Denyer, P. Smart, Towards a methodology for developing evidence-informed management knowledge by means of systematic review, Br. J. Manage. 14 (3) (2003) 207–222.
[32]. H.M. Cooper, Research Synthesis and Meta-Analysis: A Step-By-Step Approach (Applied Social Research Methods), fourth ed., Sage Publications, Inc, 2010.
[33]. CH. Glock, S. Hochrein, Purchasing organization and design: a literature review, Bus. Res. 4 (2011) 149–191.
[34]. S. Hochrein, C. Glock, Systematic literature reviews in purchasing and supply management research: a tertiary study, Int. J. Integr. Supply Manage. 7 (4) (2012) 215–245.
[35]. B. González-Pereira, V.P. Guerrero-Bote, F. Moya-Anegón, A new approach to the metric of journals scientific prestige: the SJR indicator, J. Infometrics 4 (2010) 379–391.
[36]. H.F. Moed, Measuring contextual citation impact of scientific journals, J. Informetrics 4 (3) (2010) 265–277.
[37]. K.A.H. Kobbacy, S. Vadera, A survey of AI in operations management from 2005 to 2009, J. Manuf. Technol. Manage. 22 (6) (2011) 706–733.
[38]. P. Kouvelis, C. Chambers, D.Z. Yu, Manufacturing operations manuscripts published in the first 52 issues of POM: Review, trends, and opportunities, Prod. Oper. Manage. 14 (4) (2005) 450–467.
[39]. C. Frankfort-Nachmias, D. Nachmias, J. Dewaard, Research Methods in the Social Sciences, eighth ed., Worth Pub., 2014.
[40]. S. Seuring, S. Gold, Conducting content-analysis based literature reviews in supply chain management, Supply Chain Manage. Int. J. 17 (5) (2012) 544–555.
[41]. C.R. Sox, P.L. Jackson, A. Bowman, J.a. Muckstadt, A review of the stochastic lot scheduling problem, Int. J. Prod. Econ. 62 (3) (1999) 181–200.
[42]. R. Ramasesh, Dynamic job shop scheduling: a survey of simulation research, Omega 18 (1) (1990) 43–57.
[43]. M.L. Pinedo, Scheduling Theory, Algorithms, and Systems, third. ed., Springer, 2008.
[44]. R.L. Sisson, Methods of sequencing in job shops—a review, Oper. Res. 7 (1) (1959) 10–29.
[45]. K.R. Baker, Sequencing rules and due-date assignments in a job shop, Manage. Sci. 30 (9) (1984) 1093–1104.

[46]. C. Oguz, M.F. Ercan, A genetic algorithm for hybrid flow-shop scheduling with multiprocessor tasks, J. Sched. 8 (4) (2005) 323–351.
[47]. T.C.E. Cheng, C.C.S. Sin, A state-of-the-art review of parallel-machine scheduling research, Eur. J. Oper. Res. 47 (1990) 271–292.
[48]. CD. Geiger, R. Uzsoy, H. Aytug, Rapid modeling and discovery of priority dispatching rules: an autonomous learning approach, J. Sched. 9 (1) (2006) 7–34.
[49]. J.H. Blackstone , D.T. Phillips , G.L. Hogg , A state-of-the-art survey of dispatching rules for manufacturing job shop operations, Int. J. Prod. Res. 20 (1) (1982) 27–45.
[50]. R. Haupt, A survey of priority rule-based scheduling, OR Spectrum 11 (1) (1989) 3–16.
[51]. B.L. MacCarthy, J. Liu, Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling, Int. J. Prod. Res. 31 (1) (1993) 59–79.

Further reading

The following list of references contains papers that are included in the sample of the content analysis, but that were not explicitly cited in the paper.

Abdullah, S., Abdolrazzagh-Nezhad, M. (2014). Fuzzy job-shop scheduling problems: A review. Information Sciences. 278, 380–407.

Adamu, M.O., Adewumi, A.O. (2014). A survey of single machine scheduling to minimize weighted number of tardy jobs. Journal of Industrial and Management Optimization. 10, 219–241.

Adamu, M.O., Adewumi, A.O. (2016). Minimizing the weighted number of tardy jobs on multiple machines: A review. Journal of Industrial and Management Optimization. 12, 1465–1493.

Akyol, D. E., & Bayhan, G. M. (2007). A review on evolution of production scheduling with neural networks. Computers & Industrial Engineering, 53(1), 95–122.

Alidaee, B., & Womer, N. K. (1999). Scheduling with time dependent processing times: Review and extensions. Journal of the Operational Research Society, 50(7), 711–720.

Allahverdi, A., Gupta, J. N. D., & Aldowaisan, T. (1999). A review of scheduling research involving setup considerations. Omega, 27(2), 219–239.

Allahverdi, A., Ng, C. T., Cheng, T. C. E., & Kovalyov, M. Y. (2008). A survey of scheduling problems with setup times or costs. European Journal of Operational Research, 187(3), 985–1032.

Allahverdi, A., (2015). The third comprehensive survey on scheduling problems with setup times/costs. European Journal of Operational Research. 246, 345–378.

Appiequist, G., Samikog, O., Pekny, J., & Reklaifis, G. (1997). Issues in the use, design and evolution of process scheduling and planning systems. ISA Transactions, 36(2), 81–121.

Aytug, H., Bhattacharyya, S., Koehler, G. J., & Snowdon, J. L. (1994). A review of machine learning in scheduling. IEEE Transactions on Engineering Management, 41(2), 165–171.

Bagchi, T. P., Gupta, J. N. D., & Sriskandarajah, C. (2006). A review of TSP based approaches for flowshop scheduling. European Journal of Operational Research, 169(3), 816–854.

Baker, K. R., & Scudder, G. D. (1990). Sequencing with Earliness and Tardiness Penalties: A Review. Operations Research, 38(1), 22-36.

Bakshi, M. S., & Arora, S. R. (1969). The Sequencing Problem. Management Science, 16(4), 247–263.

Balogun, O. O., & Popplewell, K. (1999). Towards the integration of flexible manufacturing system scheduling. International

Journal of Production Research, 37(15), 3399–3428.

Barnes, J. W., & Laguna, M. (1993). A tabu search experience in production scheduling. Annals of Operations Research, 41(3), 141–156.

Behnamian, J. (2016). Survey on fuzzy shop scheduling. Fuzzy Optimization and Decision Making. 15, 331–366.

Behnamian, J., Fatemi Ghomi, S.M.T. (2014). A survey of multi-factory scheduling. Journal of Intelligent Manufacturing. 27, 1–19.

Biskup, D. (2008). A state-of-the-art review on scheduling with learning effects. European Journal of Operational Research, 188(2), 315–329.

Branke, J., Nguyen, S., Pickardt, C.W., Zhang, M. (2016). Automated design of production scheduling heuristics: a Review. IEEE Transactions on Evolutionary Computation. 20, 110–124.

Brucker, P., Sotskov, Y. N., & Werner, F. (2007). Complexity of shop-scheduling problems with fixed number of jobs: a survey. Mathematical Methods of Operations Research, 65(3), 461–481.

Çaliş, B., Bulkan, S. (2015). A research survey: review of AI solution strategies of job shop scheduling problem. Journal of Intelligent Manufacturing. 26, 961–973.

Chan, F. T. S., & Chan, H. K. (2004). A comprehensive survey and future trend of simulation study on FMS scheduling. Journal of Intelligent Manufacturing, 15, 87–102.

Chaudhry, I.A., Khan, A.A. (2016). A research survey: Review of flexible job shop scheduling techniques. International Transactions in Operational Research. 23, 551–591.

Cheng, M., Mukherjee, N. J., & Sarin, S. C. (2013). A review of lot streaming. International Journal of Production Research, 51(23–24), 1–24.

Cheng, T. C. E., Gupta, J. N. D., & Wang, G. Q. (2000). A review of flowshop scheduling research with setup times. Production and Operations Management, 9(3), 262–282.

Cheng, T. C. E., & Gupta, M. C. (1989). Survey of scheduling research involving due date determination decisions. European Journal of Operational Research, 38(2), 156–166.

Cheng, T. C. E., Ding, Q., & Lin, B. M. T. (2004). A concise survey of scheduling with time-dependent processing times. European Journal of Operational Research, 152(1), 1–13.

Crama, Y., Kats, V., Klundert, J. V. D, & Levner, E. (2000). Cyclic scheduling in robotic flowshops. Annals of Operations Research, 96, 97–124.

Day, J. E., & Hottenstein, M. P. (1970). Review of sequencing research. Naval Research Logistics Quarterly, 17(1), 11–39.

Dudek, R. a, Panwalkar, S. S., & Smith, M. L. (1992). The lessons of flowshop scheduling research. Operations Research, 40(1), 7–13.

Edis, E. B., Oguz, C., & Ozkarahan, I. (2013). Parallel machine scheduling with additional resources: notation, classification, models and solution methods. European Journal of Operational Research, 230(3), 449–463.

Elmaghraby, S. E. (1978). The economic lot scheduling problem (ELSP): review and extensions. Management Science, 24(6), 587–598.

Elmaghraby, S. E. (1968). The machine sequencing problem review and extensions. Naval Research Logistics, 15(2), 205–

232.

Floudas, C. A., & Lin, X. (2005). Mixed integer linear programming in process scheduling: modeling, algorithms, and applications. Annals of Operations Research, 139(1), 131–162.

Floudas, C. A., & Lin, X. (2004). Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. Computers & Chemical Engineering, 28(11), 2109–2129.

Fry, T. D., Armstrong, R. D., & Lewis, H. (1989). A framework for single machine multiple objective sequencing research. Omega, 17(6), 595–607.

Gargeya, V. B., & Deane, R. H. (1996). Scheduling research in multiple resource constrained job shops: a review and critique. International Journal of Production Research, 34(8), 2077–2097.

Gen, M., & Lin, L. (2014). Multiobjective evolutionary algorithm for manufacturing scheduling problems: state-of-the-art survey. Journal of Intelligent Manufacturing, 25(5), 849–866.

Godin, V. B. (1978). Interactive scheduling: historical survey and state of the art. AIIE Transactions, 10(3), 331–337.

Gordon, V. S., Proth, J. M., & Chu, C. (2002). Due date assignment and scheduling: Slk, TWK and other due date assignment models. Production Planning & Control, 13(2), 117–132.

Gordon, V., Proth, J. M., & Chu, C. (2002). A survey of the state-of-the-art of common due date assignment and scheduling research. European Journal of Operational Research, 139(1), 1–25.

Gordon, V., Strusevich, V., & Dolgui, A. (2012). Scheduling with due date assignment under special conditions on job processing. Journal of Scheduling, 15(4), 447–456.

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

Graves, S. C. (1981). A review of production scheduling. Operations Research, 29(4), 646–675.

Gupta, A. K., & Sivakumar, A. I. (2006). Job shop scheduling techniques in semiconductor manufacturing. The International Journal of Advanced Manufacturing Technology, 27(11–12), 1163–1169.

Gupta, J. N. D. (1971). M-stage scheduling problem—a critical appraisal. International Journal of Production Research, 9(2), 267–281.

Gupta, J. N. D., Koulamas, C., & Kyparisis, G. J. (2006). Performance guarantees for flowshop heuristics to minimize makespan. European Journal of Operational Research, 169(3), 865–872.

Gupta, S. K., & Kyparisis, J. (1987). Single machine scheduling research. Omega, 15(3), 207–227.

Gupta, Y. P., Evans, G. W., & Gupta, M. C. (1991). A review of multi-criterion approaches to FMS scheduling problems. International Journal of Production Economics, 22(1), 13–31.

Hall, N. G., & Sriskandarajah, C. (1996). A survey of machine scheduling problems with blocking and no-wait in process. Operations Research, 44(3), 510–525.

Hart, E., Ross, P., & Corne, D. (2005). Evolutionary scheduling: a review. Genetic Programming and Evolvable Machines, 6(2), 191–220.

Kallrath, J. (2002). Planning and scheduling in the process industry. OR Spectrum, 24, 219–250.

Kanet, J. J., & Adelsberger H. (1987). Expert systems in production scheduling. European Journal of Operational Research, 29, 51–59.

Kanet, J. J., & Sridharan, V. (2000). Scheduling with inserted idle time: problem taxonomy and literature review. Operations

Research, 48(1), 99–110.

Kiran, A. S., & Smith, M. L. (1984). Simulation studies in job shop scheduling: a survey. Computers & Industrial Engineering, 8(2), 87–93.

Kolen, A. W. J., Lenstra, J. K., Papadimitriou, C. H., & Spieksma, F. C. R. (2006). Interval scheduling: A survey. Naval Research Logistics, 54(5), 530–343.

Koulamas, C. (2010). The single-machine total tardiness scheduling problem: Review and extensions. European Journal of Operational Research, 202(1), 1–7.

Koulamas, C. (1994). The total tardiness problem: review and extensions. Operations Research, 42(6), 1025–1042.

Kovalev, M. Y., Shafranskij, Y. M., Strusevich, V. A., Tanaev, V. S., & Tuzikov, A. V. (1989). Approximation scheduling algorithms: a survey. Optimization, 20(6), 859–878.

Kovalyov, M. Y., Ng, C. T., & Cheng, T. C. E. (2007). Fixed interval scheduling: Models, applications, computational complexity and algorithms. European Journal of Operational Research, 178(2), 331–342.

Kravchenko, S. a., & Werner, F. (2011). Parallel machine problems with equal processing times: a survey. Journal of Scheduling, 14(5), 435–444.

Lauff, V., & Werner, F. (2004). Scheduling with common due date, earliness and tardiness penalties for multimachine problems: A survey. Mathematical and Computer Modelling, 40(5–6), 637–655.

Lee, C., Lei, L., & Pinedo, M. (1997). Current trends in deterministic scheduling. Annals of Operations Research, 70(0), 1–41.

Lee, K., Leung, J. Y. T., & Pinedo, M. L. (2010). Makespan minimization in online scheduling with machine eligibility. 4OR: A Quarterly Journal of Operations Research, 8(4), 331–364.

Lei, D. (2009). Multi-objective production scheduling: a survey. The International Journal of Advanced Manufacturing Technology, 43(9–10), 926–938.

Lenstra, J. K., Kan, A. H. G. R., & Brucker, P. (1977). Complexity of machine scheduling problems. Annals of Discrete Mathematics, 1(8), 343–362.

Lenstra, J. K., & Kan, A. H. G. R. (1984). New directions in scheduling theory. Operations Research Letters, 2(6), 255-259.

Leung, J. Y. T., & Li, C. L. (2008). Scheduling with processing set restrictions: A survey. International Journal of Production Economics, 116(2), 251–262.

Levner, E., Kats, V., Alcaide López de Pablo, D., & Cheng, T. C. E. (2010). Complexity of cyclic scheduling problems: A state-of-the-art survey. Computers & Industrial Engineering, 59(2), 352–361.

Li, Z., & Ierapetritou, M. (2008). Process scheduling under uncertainty: Review and challenges. Computers & Chemical Engineering, 32(4–5), 715–727.

Liaee, M. M., & Emmons, H. (1997). Scheduling families of jobs with setup times. International Journal of Production Economics, 51, 165–176.

Linn, R., & Zhang, W. (1999). Hybrid flow shop scheduling: a survey. Computers & Industrial Engineering, 37(1–2), 57–61.

Ma, Y., Chu, C., & Zuo, C. (2010). A survey of scheduling with deterministic machine availability constraints. Computers & Industrial Engineering, 58(2), 199–211.

Mathirajan, M., & Sivakumar, A. I. (2006). A literature review, classification and simple meta-analysis on scheduling of batch processors in semiconductor. The International Journal of Advanced Manufacturing Technology, 29(9–10), 990–1001.

Méndez, C. a., Cerdá, J., Grossmann, I. E., Harjunkoski, I., & Fahl, M. (2006). State-of-the-art review of optimization methods for short-term scheduling of batch processes. Computers & Chemical Engineering, 30(6–7), 913–946.

Mönch, L., Fowler, J. W., Dauzère-Pérès, S., Mason, S. J., & Rose, O. (2011). A survey of problems, solution techniques, and future challenges in scheduling semiconductor manufacturing operations. Journal of Scheduling, 14(6), 583–599.

Morgan, S. D., & Gagnon, R. J. (2013). A systematic literature review of remanufacturing scheduling. International Journal of Production Research, 51(16), 4853–4879.

Mutlu, M., Yagmahan, B. (2014). Multi-objective permutation flow shop scheduling problem: Literature review, classification and current trends. Omega 45, 119–135.

Nagano, M.S., Miyata, H.H. (2016). Review and classification of constructive heuristics mechanisms for no-wait flow shop problem. International Journal of Advanced Manufacturing Technology. 86, 1–14.

Nagar, A., Haddock, J., & Heragu, S. (1995). Multiple and bicriteria scheduling: A literature survey. European Journal of Operational Research, 81(1), 88–104.

Neufeld, J.S., Gupta, J.N.D., Buscher, U. (2016). A comprehensive review of flowshop group scheduling literature. Computers and Operations Research. 70, 56–74.

Panwalkar, S. S., & Iskander, W. (1977). A survey of scheduling rules. Operations Research, 25(1), 45–61.

Panwalkar, S. S., Smith, M. L., & Koulamas, C. (2013). Review of the ordered and proportionate flow shop scheduling research. Naval Research Logistics, 60(1), 46–55.

Pickardt, C. W., & Branke, J. (2012). Setup-oriented dispatching rules – a survey. International Journal of Production Research, 50(20), 5823–5842.

Pinto, J. M., & Grossmann, I. E. (1998). Assignment and sequencing models for the scheduling of process systems. Annals of Operations Research, 81(0), 433–466.

Quadt, D., & Kuhn, H. (2007). A taxonomy of flexible flow line scheduling procedures. European Journal of Operational Research, 178(3), 686–698.

Raheja, A. S., & Subramaniam, V. (2002). Reactive recovery of job shop schedules - a review. The International Journal of Advanced Manufacturing Technology, 19(10), 756–763.

Ribas, I., Leisten, R., & Framinan, J. M. (2010). Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective. Computers & Operations Research, 37(8), 1439–1454.

Ruiz, R., & Vázquez-Rodríguez, J. A. (2010). The hybrid flow shop scheduling problem. European Journal of Operational Research, 205(1), 1–18.

Rustogi, K., & Strusevich, V. A. (2012). Simple matching vs linear assignment in scheduling models with positional effects: a critical review. European Journal of Operational Research, 222(3), 393–407.

Sanlaville, E., & Schmidt, G. (1998). Machine scheduling with availability constraints. Acta Informatica, 35(9), 795–811.

Sarin, S. C., Varadarajan, A., & Wang, L. (2011). A survey of dispatching rules for operational control in wafer fabrication. Production Planning & Control, 22(1), 4–24.

Schmidt, G. (2000). Scheduling with limited machine availability. European Journal of Operational Research, 121, 1–15.

Sen, T., & Gupta, S. K. (1984). A state-of-art survey of static scheduling research involving due dates. Omega, 12(1), 63–76.

Sen, T., Sulek, J. M., & Dileepan, P. (2003). Static scheduling research to minimize weighted and unweighted tardiness: a state-of-the-art survey. International Journal of Production Economics, 83(1), 1–12.

Shabtay, D., Gaspar, N., & Kaspi, M. (2013). A survey on offline scheduling with rejection. Journal of Scheduling, 16(1), 3–28.

Shabtay, D., & Steiner, G. (2007). A survey of scheduling with controllable processing times. Discrete Applied Mathematics, 155(13), 1643–1666.

Sterna, M. (2011). A survey of scheduling problems with late work criteria. Omega, 39(2), 120–129.

Sun, Y., Zhang, C., Gao, L., & Wang, X. (2011). Multi-objective optimization algorithms for flow shop scheduling problem: a review and prospects. The International Journal of Advanced Manufacturing Technology, 55, 723–739.

Toptal, A., & Sabuncuoglu, I. (2010). Distributed scheduling: a review of concepts and applications. International Journal of Production Research, 48(18), 5235–5262.

Treleven, M. (1989). A review of the dual resource constrained system research. IIE Transactions, 21(3), 279–287.

Tuncel, G., & Bayhan, G. M. (2007). Applications of Petri nets in production scheduling: a review. The International Journal of Advanced Manufacturing Technology, 34(7–8), 762–773.

Tzafestas, S., & Triantafyllakis, A. (1993). Deterministic scheduling in computing and manufacturing systems: a survey of models and algorithms. Mathematics and Computers in Simulation, 35(5), 397–434.

Vaessens, R. J. M., Aarts, E. H. L., & Lenstra, J. K. (1996). Job shop scheduling by local search. INFORMS Journal on Computing, 8(3), 302–317.

Webster, S., & Baker, K. R. (1995). Scheduling groups of jobs on a single machine. Operations Research, 43(4), 692–703.

Wiers, V. C. S. (1997). A review of the applicability of OR and AI scheduling techniques in practice. Omega, 25(2), 145–153.

Xie, C., Allen, T.T. (2015). Simulation and experimental design methods for job shop scheduling with material handling: a survey. The International Journal of Advanced Manufacturing Technology. 80, 233–243.

Yang, W. H. (1999). Survey of scheduling research involving setup times. International Journal of Systems Science, 30(2), 143–155.

Zhu, X., & Wilhelm, W. E. (2006). Scheduling and lot sizing with sequence-dependent setup: a literature review. IIE Transactions, 38(11), 987–1007.

Appendix

The following table gives an overview of the groups and subgroups obtained in the work at hand. The table shows the recording units (keywords) belonging to each subgroup, together with their detailed results and the results of the sensitivity analysis.

Group	Subgroup	Subclass	Recording unit	Number of hits in the sample	Percent-age of hits in sample	No. of dedicated papers	Percentage of depend-ency on dedicated papers	Average percentage of depend-ency on dedicated papers
Type of the problem	Dynamic		Dynamic (Dynamic programming is excluded)	1101	1.41	3	14.26	4.75
	Static		Static	401	0.51	3	7.48	2.49
	Stochastic		Stochastic, probabilistic, Fuzzy	2371	3.03	5	53.69	10.74
	Deterministic		Deterministic	628	0.80	5	22.13	4.43
	Online		Online, real-time	1046	1.34	1	22.18	22.18
	Offline		Offline	73	0.09	1	8.22	8.22
	Periodic		Periodic	158	0.20	0	---	---
Flow pattern	Batch machines		Batch machines	2211	2.83	2	18.23	9.11
	Flow shop	General flow shop	Flow shop (excluding hybrid flow shop), FSP, PFSP	2627	3.36	17	64.07	3.77
		Hybrid flow shop	Hybrid flow shop, HFS, HFSP					
	Job shop		Job shop	2628	3.36	13	34.21	2.63
	Single machine		Single machine, single server, single processor, single stage, one machine, one processor	2082	2.66	6	13.93	2.32
	Parallel machines	General terms	Parallel machine, parallel server, parallel pro-cessor	1734	2.22	3	13.38	4.46
		Identical machines	Identical machine, identical processor					
		Uniform machines	Uniform machines, uniform processor					
		Irrelevant ma-chines	Irrelevant machines, irrelevant processor					
	Two machines		Two machine, two processor	1371	1.75	0	---	---
	m machines		m machine, multi-machine, multi-processor	938	1.20	4	6.72	1.68
	Lot scheduling		Lot scheduling, lot-sizing, lot-streaming	599	0.77	5	60.60	12.12
	Three machines		Three machine, three processor	264	0.34	0	---	---

	Open shop		Open shop	305	0.39	0	---	---
	Process scheduling		Process scheduling	123	0.16	4	36.59	9.15
Objectives	Penalty-based objectives		Penalty, delay, due-date (excluding common due-date), earliness, lateness, tardiness, earliness-tardiness	6532	8.35	12	25.05	2.09
	throughput-time-based objectives	Flow time	Flow time, FT, TFT, total completion time	4908	6.27	4	11.59	2.90
		Makespan	Makespan, Cmax, Max completion time, maximum completion time					
	Cost-based objectives		Cost, cost-based	2357	3.01	1	3.65	3.65
	Multi-objective problems		Multi-objective, bi-criteria, multi-criteria, pareto	1555	1.99	11	62.89	5.72
Job and machine characteristics	Job families		Family, family-based	1236	1.58	1	15.29	15.29
	No-wait processes		no-wait, deterioration	1110	1.42	2	22.52	11.26
	Sequence dependency		sequence dependent	1174	1.50	1	7.50	7.50
	Precedence constraints		Precedence	772	0.99	0	---	---
	Preemption		Preemption, preemptive	724	0.93	0	---	---
	Common due-dates		Common due-date	342	0.44	2	27.49	13.74
	Ordered operations		Ordered, semi-ordered	262	0.33	1	40.84	40.84
	Setup time		Setup, changeover, loading	4461	5.70	8	54.43	6.80
	Machine breakdown		Breakdown, stability, machine availability, maintenance, disturbance, repair, machine reliability	912	1.17	3	4.50	1.50
	Learning effect		Learning, forgetting	764	0.98	2	36.91	18.46
	Rescheduling problems		rescheduling, resequencing	367	0.47	0	---	---
	Transportation lags		Transportation, delivery, shipment	412	0.53	0	---	---

	Buffer space		Buffer	266	0.34	0	---	---
	Dual resource operations		Dual resource, dual-constrained, multi-resource, resource-constrained, resource-dependent, two-resource, controllable processing time	221	0.28	4	46.61	11.65
Solution approaches	Simple heuristics	General terms	Priority rule, dispatching rule, dispatch rule	7869	10.06	9	17.58	1.95
		Specific heuristics	SPT, Johnson, EDD, SI, TWK, LR, NOP, RAN, CON, SLACK, FIFO, SR, FCFS					
	Metaheuristics	General terms	Local search, neighbourhood, population, metaheuristic, EA, evolutionary	6502	8.31	4	14.40	3.60
		Specific metaheuristics	GA, genetic, chromosome, tabu-search, tabu, taboo, annealing, ant, PSO, swarm, GRASP, guided local search, GLS, iterated local search, ILS, memetic, adaptive memory, AM, SS, scatter search, variable neighbourhood search, VNS					
	Exact algorithms	General terms	Exact algorithm	2577	3.29	2	4.81	2.41
		Branch & bound	Branch and bound, branching, lomnicki					
		Linear programming	Linear programming, LP, ILP, integer programming, mixed-integer					
		Dynamic programming	Dynamic programming					
		Enumeration	Enumeration, enumerate					
	Simulation		Simulation	1075	1.37	4	14.23	3.56
	Artificial intelligence		Artificial intelligence, ANN, ANNs, AN, neural	1014	1.30	3	56.51	18.84
	Approximation algorithms		Worst-case analysis, performance guarantee, performance ratio	122	0.16	2	25.41	12.70
Scheduling in practice	General terms		Practice, real life, real world, realistic, industry	4116	5.26	9	32.90	3.66
	Specific industrial applications		AGV, AGNs, assembly, automated, automobile, chemical, FMS, flexible manufacturing, robotic, robots, semiconductor, textile, timetabling, wafer					
Pure theory	General terms		Theory, Theorem, theoretical	5925	7.57	5	8.64	1.73
	Complexity theory		Complexity, NP-hard, NP-complete, polynomial					

Paper 3 New simple constructive heuristic algorithms for minimizing total flow-time in the permutation flowshop scheduling problem

Authors: Hamid Abedinnia, Christoph H. Glock, Andreas Brill

Type of publication: Journal article

Publication details: *Computers & Operations Research,* Volume 74, 2016, Pages 165-174

Abstract

This paper develops a set of new simple constructive heuristic algorithms to minimize total flow-time for an *n*-jobs *x m*-machines permutation flowshop scheduling problem. We first propose a new iterative algorithm based on the best existing simple heuristic algorithm, and then integrate new indicator variables for weighting jobs into this algorithm. We also propose new decision criteria to select the best partial sequence in each iteration of our algorithm. A comprehensive numerical experiment reveals that our modifications and extensions improve the effectiveness of the best existing simple heuristic without affecting its computational efficiency.

Keywords: *Flowshop scheduling problem, Permutation flowshop, Total flow-time, Heuristics*

1 Introduction

A flowshop production system is commonly defined as a production system in which a set of n jobs undergoes a series of operations in the same order [23]. Determining optimal job sequences for flowshop scheduling problems can be based on various objectives; minimizing makespan and minimizing total flow-time have, however, most often been considered as objectives for flowshop scheduling problems in the past. The first objective refers to the minimization of the last job's completion time, while the second one aims on minimizing the total in-process time, which reduces work-in-progress inventory [6]. For makespan minimization, problems with more than two machines have been shown to be strongly NP hard [23]; this is even the case for Permutation Flowshop Scheduling Problems, i.e. for flowshop scheduling problems with the same job order on all machines. Garey et al. [11] showed that the problem of minimizing total flow-time with more than one machine belongs to the category of NP complete problems. Accordingly, for large-size problems, heuristic procedures have to be used to find solutions in reasonable computational time. A comprehensive review of research on flowshop scheduling that appeared during the last 50 years is the one of Gupta and Stafford [14]. A review of scheduling problems that aim on minimizing makespan can be found in Ruiz and Maroto [28] and Gupta et al. [13]. The Permutation Flowshop Scheduling Problem with the objective of flow-time minimization was reviewed by Pan and Ruiz [22] and Framinan et al. [8], where the latter also reviewed works that consider makespan minimization. Mutlu and Yagmahan [18] recently reviewed multi-objective flowshop scheduling problems.

Framinan et al. [10] provided a framework to categorize heuristic algorithms for the Permutation Flowshop Scheduling Problem (PFSP) according to their structure. This framework distinguished between the phases of (a) index development, (b) solution construction, and (c) solution improvement. Framinan et al. [9] categorized existing heuristics, which can address one or more of these phases, into two classes: simple and composite heuristics. An algorithm was categorized as a simple heuristic if it does not include another heuristic. Composite heuristics are heuristics that contain at least one simple heuristic for conducting one or more of the three above-mentioned phases. Pan and Ruiz [22] showed that composite heuristics outperform simple heuristics in minimizing flow-time. Yet, as simple heuristics are the basic building blocks of composite heuristics, improving their performance is still of interest for the research community, as this improvement can boost the performance of composite heuristics as well. The aim of this paper is to propose a set of new simple heuristics to improve the performance of the best existing simple heuristic algorithm for minimizing total flow-time in the PFSP.

A popular simple heuristic for minimizing makespan in the general PFSP was presented by Nawaz et al. [19] (we refer to this heuristic as NEH in the following), which outperformed other algorithms developed earlier, such as the heuristics of Palmer [20], Gupta [12], or Campbell et al. [2]. Despite its good performance for makespan-related PFSPs, another advantage of NEH is that it leads to good solutions for other objectives as well, such as minimizing total flow-time (as was shown, for example, by Allahverdi and Aldowaisan [1]). The NEH heuristic consists of two phases, namely (I) the sorting (prioritizing) phase and (II) the insertion phase. In the sorting phase, jobs are sorted in descending order of their total processing time. This sorted list is used in the insertion phase to determine the sequence in which jobs are added to an existing partial sequence. For an n-job PFSP, the insertion phase consists of n iterations. In step k $(1 \leq k \leq n)$ of the insertion phase, the k^{th} job on the sorted list is successively assigned to the k possible slots in the current partial sequence that was obtained in the previous iteration, which consists of $k-1$ jobs. The partial sequence that leads to the best value for the objective function (minimum partial makespan) is used as the current k-jobs partial sequence for the next iteration.

Since 1983, many researchers have tried to improve NEH for different objective functions by modifying either its sorting or its insertion phase. One example is the work of Framinan et al. [7], which tried to improve the performance of NEH for three objectives (i.e. makespan, idle time and total flow-time minimization) by applying 177 new ordering policies to the sorting phase of NEH. These policies are combinations of different indicator values and sorting criteria. Most extensions of NEH are more effective than the original version (i.e., they lead to better solutions), but they are usually less efficient (i.e., they are usually more complex and require more computational time than the original NEH).

The relatively high efficiency of NEH is primarily due to the idea of keeping an established partial sequence of a set of jobs unchanged from one iteration until the algorithm terminates. This idea, however, also restricts the effectiveness of NEH, as it does not search for potentially better local solutions once a partial sequence has been established. One option to improve the insertion phase of NEH is to optimize partial sequences by testing alternative positions for jobs at the end of each iteration, i.e. to evaluate the neighborhood of each partial sequence. A similar idea was presented by Rajendran [24], who optimized partial sequences by exchanging adjacent jobs pairwise with the objective to minimize total flow-time. Framinan and Leisten [6] combined this idea with NEH and performed pairwise exchanges at the end of each iteration to improve partial sequences. The authors showed that their algorithm (to which we refer as FL hereafter) outperformed other constructive algorithms for the total flow-time criterion. Framinan et al. [9] evaluated different heuristic algorithms for the PFSP and concluded that the FL heuristic led to better solutions for the total flow-time criterion. Laha and Sarin [17] extended FL by allowing all jobs assigned to a partial sequence to change their respective position by checking all other $k-1$ slots at the end of each iteration. They showed that their algorithm (to which we refer as LS in the following) leads to a better performance, in terms of the quality of the solutions, and only a small loss in efficiency as compared to the FL heuristic. Pan and Ruiz [22] reviewed the most promising constructive heuristics and indicated that LS is the best existing simple heuristic to minimize total flow-time in general PFSPs in terms of the quality of the results. Since LS is computationally complex, the authors developed some new composite heuristics that outperform LS and at the same time consume about one order of magnitude less CPU time. Recently, Fernandez-Viagas and Framinan [5] proposed a set of new constructive heuristics (we refer to them as FF heuristics in the following) and compared them with some of the heuristics considered in Pan and Ruiz [22]. Although some of their composite algorithms showed a better performance than the ones proposed in Pan and Ruiz [22], their proposed simple heuristics (all pure FF heuristics, i.e. FF(1)-FF(n)) are outperformed by LS.

As mentioned above, having promising composite heuristics does not render efforts to improve simple heuristics worthless. Better simple heuristics may open the gate for the development of even better composite heuristics. Based on LS, this paper proposes several new simple heuristics for the PFSP. Numerical experiments illustrate that our modifications lead to a significant improvement in terms of the quality of the solutions without affecting the computational efficiency as compared to the best existing simple heuristic.

The remaining sections of this paper are organized as follows. Section 2 outlines the heuristic of Laha and Sarin [17] and possible modifications for its extensions. Section 3 describes the proposed heuristics in detail. A comprehensive comparison of the proposed heuristics and LS, together with a detailed evaluation of the effect of proposed modifications on NEH and LS, are given in Section 4, and Section 5 concludes the paper.

2 The heuristic of Laha and Sarin and its modifications

Both the FL and LS heuristics optimize partial sequences at the end of each iteration of NEH's insertion phase. This paper focuses on improving the LS heuristic, which outperforms all other existing simple heuristics for optimizing total flow-time in a permutation flowshop manufacturing system [22]. The pseudocode of LS is the following:

Step 1: $P_i = \sum_{j=1}^{m} p_{ij}$, $i = 1,2,\dots,n$, where P_i is the indicator value of job i and p_{ij} is the processing time of job i on machine j.

Step 2: Sort the jobs in an **ascending** order of their indicator values.

Step 3: Select jobs $k = 1$ and $k = 2$ and keep the partial sequence (i.e. $1 - 2$ or $2 - 1$) that results in a shorter total flow-time as the current partial sequence.

Step 4: For $k = 3,\dots,n$, repeat the following:

4.1. Insert the k^{th} job in all k possible slots in the partial sequence obtained in the last iteration, which consists of $k - 1$ jobs.
4.2. Select the best k-job partial sequence that results in the shortest total flow-time as the current partial sequence.
4.3. For $i = 1,\dots,k - 1$, remove job i from the current partial sequence and insert it into the $k - 1$ positions of the remaining partial sequence. Calculate the corresponding total flow-time for all new combinations.
4.4. If the best of the new $k(k - 1)$ k-job partial sequences generated in Step 4.3 is better than the current partial sequence, replace it by the best partial sequence obtained in Step 4.3. Set $k = k + 1$.

It is worth noting that the first three steps of LS are almost identical to those of the original NEH heuristic. The only difference is that in LS, unlike in the original NEH, the jobs are sorted in ascending order of their weights (Step 2). Framinan et al. [8] showed that minimizing total flow-time in a PFSP by using a modified version of the NEH heuristic, with jobs sorted in an *ascending* order of their total processing times, performs better than the original NEH. It is also worth noting that partial sequences are optimized starting with Step 4 of LS.

As LS is based on NEH, we have the same options for improving LS as for extending the NEH heuristic. Framinan et al. [7] named six attributes of the NEH heuristic that offer rooms for extensions:

(1) Consider different objective functions, such as makespan, total flow-time or idle-time minimization.
(2) Employ different indicator values in Step 1 (e.g. as by Framinan et al. [7]).
(3) Choose a different sorting criterion in Step 2 (e.g. as by Framinan et al. [7]).
(4) Restrict the insertion of job k in Step 4 to a subset of the k possible positions (e.g. as by Rajendran [24]).

(5) Insert multiple unscheduled jobs into a subsequence in each iteration (e.g. as by Woo and Yim [32]).
(6) Keep more than a single sequence within the iterations of Step 4 (e.g. as by Jena et al. [33]).

In addition to the suggestions of Framinan et al. [7], we propose the following additional options for extending NEH:

(7) Optimize partial sequences, e.g. by performing a local search in the neighborhood of the partial sequences obtained at the end of each iteration of NEH, as in FL or LS.
(8) Apply different tie-breaking rules in Step 2, where two or more jobs may have the same weight, and in Steps 4.2 and 4.4, where two or more inserting positions may result in the same value for the objective function. [4], for example, reviewed existing tie-breaking rules and proposed a new one.
(9) Chose a different decision criterion for the selection of the best k-job partial sequence than in the objective function of the problem.

The heuristics proposed in the following are based on extension options 2, 7 and 9.

3 Proposed heuristics

3.1. AGB (focuses on option 7 to improve LS):

The sorted list (obtained in Step 2) is used both in the FL and LS heuristics only to determine the order in which jobs are inserted into the partial sequence. The heuristic proposed here uses the priority orders more effectively in the insertion phase. AGB uses the first three steps of LS and then continues as follows:

Step 4: for $k = 3, \dots, n$ repeat the following:

4.1. Insert the k^{th} job in all k possible slots in the **current** partial sequence obtained in the last iteration, which consists of $k-1$ jobs.
4.2. Select the best k-job partial sequence that results in the lowest total flow-time as the current partial sequence.
4.3. $i = 1$
4.4. Remove job i (i.e., the i^{th} job in the sorted list obtained in Step 2) from the current partial sequence and insert it into the $(k-1)$ positions of the remaining partial sequence. Calculate the corresponding total flow time for all new combinations.
4.5. If the best of the new $(k-1)$ k-job partial sequences generated in Step 4.4 is better than the current partial sequence, replace the current partial sequence by the best partial sequence obtained in Step 4.4.
4.6. If $i < k$, then $i = i + 1$ and go to Step 4.4. Otherwise $k = k + 1$.

One important difference between AGB and LS is Step 4.4, where we initiate our local search with the most important job and then proceed to the less important ones. Another difference is Step 4.5, in which we replace the current sequence immediately if this leads to an improvement, whereas LS replaces its current partial sequence only after having checked all exchanges in an iteration. The idea behind our modifications is that the change in the position of more important jobs has a higher impact on the final solution than a change in the position of less important jobs. It is easy to show that the complexity of LS is $O(k((k-1)^2+k)km)$ and the complexity of AGB is $O(k(k^2+k)km)$. It is clear that both algorithms have the same complexity, namely $O(n^4m)$, which is identical to the complexity of FL.

3.2. AGB/α/β (focuses on option 2 to improve AGB)

Following the notation of Framinan et al. [7], Heuristic/α/β describes an algorithm where α denotes the indicator value and β the sorting criterion employed. AGB/α/β is a set of new heuristics to enhance the performance of the AGB heuristic developed in Section 3.1 by employing different criteria for sorting jobs (option 2). Preliminary computational experiments revealed that all algorithms proposed here show a better performance with an ascending rather than a descending order of jobs. Since Rajendran [24], Framinan et al. [8] and Laha and Sarin [17] used an ascending order of the indicator values in their algorithms for PFSP with total flow-time objective, we restrict our analysis to the same sorting rule. Thus, for all analyzed algorithms, β = A. Accordingly, the heuristics proposed in this section are the same as AGB with respect to their sorting criterion, but differ from AGB by using new indicator values, instead of the sum of processing times, for indexing jobs in Step 1 of AGB. We examine six new indicators to prioritize the jobs. Although three of these indicators have been studied before (see Palmer's, Gupta's, and Rajendran's indices), they have not yet been tested as weighting rules in the sorting phase of NEH. The following indicator variables are considered:

1. P_i: Total processing times, i.e. $P_i = \sum_{j=1}^{m} p_{ij}\ \ i = 1,2,\dots,n$. This is the original indexing policy of the NEH and LS heuristics, and it is used as a benchmark here.
2. SIP_i: Slope Indices of Palmer [20]: The total weighted processing time follows the expression $SIP_i = \sum_{j=1}^{m}(2j-m-1)\times p_{ij}$ for jobs $i = 1,2,\dots,n$. We use the Palmer indices as weights for the jobs; thus, unlike in the original heuristic of Palmer, the indices here do not indicate the final sequence of jobs.
3. $ABS(SIP_i)$: Absolute Slope Indices of Palmer, i.e., $ABS(SIP_i) = |SIP_i|$, where SIP_i is Palmer's index of job i. It is clear that Palmer's indices can be positive or negative. According to Palmer's heuristic, to reduce the chance of machines/jobs idle times, jobs are sequenced in such a way that the difference between the slopes of successive jobs is minimized. As was already mentioned, Palmers' slopes indicate the weight (the importance) of the jobs in our algorithm. Our idea behind using the absolute value of Palmer's slopes is that the absolute slopes might reflect the importance of jobs better, i.e. a job with a Palmer index of -6 is a more important job than a job with a Palmer index of 4, and it thus needs to be sequenced earlier.
4. SIG_i: Slope Indices of Gupta [12]: Job weights follow the expression $SIG_i = \frac{e_i}{\min_{1\le j\le m-1}\{p_{ij}+p_{i,j+1}\}}$ for $i = 1,2,\dots,n$, where $e_i = \begin{cases} 1 & if\ p_{i1} < p_{im} \\ -1 & otherwise \end{cases}$. As with SIP_i, Gupta's indices are only used to determine the weights of jobs in Step 1 of the algorithm, and not to determine the sequence of jobs. Gupta's indices assign a higher weight to jobs that are more likely to cause bottlenecks. For this reason, we examined the effect of this indexing method on AGB.
5. SIR_i: Slope Indices of Rajendran [24]: Rajendran [24] proposed an alternative way to assign a higher weight to machines that are more likely to cause bottlenecks in production. The impact of this sorting method on AGB is investigated as well in this paper. Weighted processing times are calculated as $SIR_i = \sum_{j=1}^{m}(m-j+1)\times P_{ij}$ for $i = 1,2,\dots,n$. Again, we use the indices only to determine the weights of the jobs.

6. $MTWPT_i$ (Machine-based Total Weighted Processing Time): For this indicator variable, we first assign weights to each machine, and then use these weights in calculating the total weighted processing time of each job. The total processing time on each machine (utilization of machines $u_j, j = 1, 2, \dots, m$) is used as machine weight and follows the expression $u_j = \sum_{i=1}^{n} p_{ij}, j = 1,2, \dots, m$. The idea behind this weighing policy is that the processing times on more important machines should play a more prominent role in weighting jobs. Indicator values then follow the expression $MTWPT_i = \sum_{j=1}^{m} u_j \times p_{ij}$ for $i = 1,2, \dots, n$.
7. $MJTWPT_i$ (Machine & Job-based Total Weighted Processing Time): This indicator assigns weights both to jobs and machines, and uses these weights in the calculation of the total weighted processing time of each job. The weight of machines and jobs are the same as their weights in indicators 6 and 1, respectively. The indicator values follow the expression $MJTWPT_i = \sum_{j=1}^{m} \left(u_j \times P_i \times P_{ij}\right)$.

3.3. AGB/α/β/γ (focuses on option 9 to improve AGB)

AGB/α/β/γ is a set of new heuristics that try to improve AGB/α/β by employing new decision criteria (option 9). These heuristics are the same as AGB/α/β, with the exception that they use new decision criteria for selecting the best current sequence in Steps 3, 4.2 and 4.5 of AGB. Two new decision criteria are suggested here, which both improve the effectiveness of the algorithm. The parameter γ determines the selected decision criterion. We consider the following decision criteria:

a) TFT: Total flow-time, i.e. $TFT = \sum_{\rho=1}^{k} FT_\rho$, where FT_ρ is the flow-time of the job at position ρ in the current partial sequence. It is clear that AGB/P_i/A/TFT is the same as AGB.
b) $TWFT$: Total weighted flow-time, i.e. $TWFT = \sum_{\rho=1}^{k} \rho \times FT_\rho$. The idea behind considering TWFT as a decision criterion is that jobs positioned at the end of the sequence have a higher contribution to the total flow-time than jobs scheduled early in the sequence. It is important to note that TWFT proposed here is different from the decision criterion used by Rajendran and Ziegler [25]. In our heuristic, the weights of the flow-times are only determined by the position of the job in the current partial sequence, while in the heuristic of Rajendran and Ziegler [25], it is determined by considering holding costs in addition, which are part of their problem formulation.
c) $TWFT_k$: Uses TFT as the decision criterion in inserting the first $(k-1)$ jobs into the partial sequence, and TWFT in inserting the next jobs. It is clear that $TWFT_0 = TWFT$ and $TWFT_{n+1} = TFT$. In the computational experiment, the performance for $k = 8, 16, 24$ is analyzed.

Combining all improvement options presented in Sections 3.1 to 3.3 leads to 35 new simple constructive heuristics, which all have the same complexity as the original LS heuristic. We also integrate the improvement options presented in Sections 3.2 and 3.3 into the LS and NEH heuristics to examine the effect of these sorting policies on the performance of the heuristics. This paper thus investigates the performance of 103 new simple constructive heuristic algorithms.

4 Computational experiment

This section examines the performance of the heuristics proposed in Section 3. As mentioned before, Framinan et al. [8] showed that using a modified version of the NEH algorithm with jobs sorted in ascending order leads to better results. Therefore, we compare the results of the heuristics proposed in this paper with the modified version of NEH (named hereafter NEH/P_i/A). We also compared the performance of the newly developed heuristics with the performance of the LS heuristic. In addition, we evaluated the effect of employing the improvement options presented in Sections 3.2 and 3.3 in the LS and NEH heuristics.

The performance of any heuristic algorithm should be evaluated in two respects, namely (I) its effectiveness, i.e. the quality and goodness of the solution it obtains, and (II) its efficiency, i.e. the complexity or computational time of the heuristic, or, in other words, the CPU-time of the heuristic. To have a standard data set for comparing the performance of different heuristic algorithms for the PFSP, Taillard [30] suggested a collection of random problems of different sizes. Using standard data set makes it easier to evaluate the effectiveness of different heuristics, especially since the size of the problems contained in the set are, according to the author, representative for real industrial problems. Taillard's data set contains 120 random problems with different parameters. The number of jobs (the size of the problems) is either 20, 50,100, 200 or 500, while the number of machines ranges from 5 to 20. Taillard's data sets have frequently been used in almost all PFSP papers to compare heuristics [21], for example by Reza Hejazi and Saghafian [27], Reisman et al. [26], Ruiz and Maroto [28], Pan and Ruiz [22], and Fernandez-Viagas and Framinan [5].

It is common practice in the literature to measure an algorithms' effectiveness by the average relative percentage deviation (ARPD) of the obtained results. The relative percentage deviation for a problem instance ρ is calculated as $RPD_\rho = \left(\frac{Heuristic_\rho - Benchmark_\rho}{Benchmark_\rho} * 100\right)$, where $Heuristic_\rho$ denotes the value of the objective function obtained by a heuristic, and $Benchmark_\rho$ is a benchmark value for the objective function of the same problem instance.

In a study of the literature, we found that ARPDs reported for identical heuristics applied on the same data set vary, e.g. in Pan and Ruiz [22] vs. Laha and Sarin [17] and Dong et al. [3] vs. Semančo and Modrák [29]. Concluding from our literature study, these variations have been caused by the following differences:

1. *Employing different benchmarks:* The selection of benchmarks in the calculation of the ARPD can be made in several ways. As in Pan and Ruiz [21] and Jarboui et al. [15], taking the best-known objective values from the literature is one alternative (in the following denoted as $ARPD_L$). $ARPD_L$ denotes the average relative percentage deviation for a set of large-size problems, where we usually do not have access to the optimum solution. $ARPD_L$ compares the solutions for each problem with the best known value of this problem from the literature, and uses the expression $ARPD_L = \frac{1}{\Omega}\sum_{\rho=1}^{\Omega}\left(\frac{Heuristic_\rho - BK_\rho}{BK_\rho} * 100\right)$, where BK_ρ denotes the best known value of the considered objective of problem instance ρ, and Ω stands for the total number of problem instances. Another alternative (in the following denoted as $ARPD_P$) is that the authors pick the best solution obtained during their own investigations as the benchmark. Pan and Ruiz [22] and Fernandez-Viagas and Framinan [5] used this option in their computational experiment. $ARPD_P$ is also commonly used in the literature and calculated as $ARPD_P = \frac{1}{\Omega}\sum_{\rho=1}^{\Omega}\left(\frac{Heuristic_\rho - BOC_\rho}{BOC_\rho} * 100\right)$, where BOC_ρ stands for the best objective value obtained by

the considered heuristics for problem instance ρ. The last option, which is not as frequently used as the first two alternatives, is to employ available lower/upper bounds as in Dong et al. [3].
Although capturing general information by comparing the reported $ARPD_P$s of different papers is possible, this measure also has some drawbacks. Problems arise, for example, when comparing the relative performance of different heuristics reported in different papers. For example, both Pan and Ruiz [22] and Fernandez-Viagas and Framinan [5] considered some common heuristics in their computational analyses (i.e. Raj, LR(1), LR-NEH(5,10), ICi(i =1,2,3) and PR(1,10,15)). The reported $ARPD_P$-values for all common heuristics are higher in the paper of Pan and Ruiz [22]. This reveals that the best obtained solutions by the heuristics considered in Pan and Ruiz [22] are better than the best obtained by the heuristics of the other paper. This general information is the only thing that we can conclude by comparing these two papers.
For the above-mentioned reasons, we think that using $ARPD_L$ can ease the tractability and the comparison of the results of different papers and facilitate comparing the obtained solutions with the best-known of the literature. It is worth to note, that $ARPD_L$ is applicable for those optimization problems in which using a common data-set is commonplace and the best-known solutions are being updated time to time. As this is the case for the PFSP with total flow-time objective, we suggest $ARPD_L$ as the comparison measure for the literature.

2. *Different tie-breaking rules:* Ties that may occur in the sorting phase (Step 2) or in the insertion phase (Steps 3, 4.2 and 4.4) can be handled differently. The tie-breaking mechanism applied in Step 2 depends on the employed sorting algorithm, in particular on the algorithm's stability properties in the coding of the heuristic. Vasiljevic and Danilovic [31] stated that imprecise definition of the employed tie-breaking rules may lead to different implications of the same algorithms. Fernandez-Viagas and Framinan [4] also underlined the significant influence of tie-breaking rules in the insertion phase on the performance of NEH and its modifications.

In our computational experiment, NEH//A, LS and the heuristics proposed in this paper were coded in Java and run on a 2.88 GHz Intel Core processor with 8.00 GB RAM. A stable sorting algorithm was applied in Step 2. In the insertion phase, the first-obtained best k-job partial sequence was taken as the current partial sequence. This is not identical, but similar to one of the tie-breaking rules proposed by Kalczynski and Kamburowski [16]. Following the performance evaluation of Laha and Sarin [17], we carried out experiments on different types of problems. We considered small-size problems with a number of jobs equal to 6, 7 and 8 and a number of machines equal to 5, 10, 15 and 20. 100 instances were generated for each combination of jobs and machines. The processing times were generated randomly using a *U(1,99)* distribution. To evaluate the performance of the algorithms for large-size problems, we considered the benchmark problems of Taillard [30]. To measure the performance of the algorithms, the ARPD and the percentage of optimal solutions for small-size problems were reported. The optimum total flow-time was obtained by a full enumeration of all possible job sequences. The average relative percentage deviation for a set of small-size problems is hence calculated as $APRD = \frac{1}{100} * \sum_{\rho=1}^{100} \left(\frac{Heuristic_\rho - Optimum_\rho}{Optimum_\rho} * 100 \right)$, where $Optimum_\rho$ denotes the optimum solution for the problem instance ρ. Table 1 compares, LS and AGB for small-size problems, and Table 2 reports the performance on large-size problems. The average computation times (in seconds) for solving the related problem instances are also reported in these tables. In addition we report $ARPD_L$ and $ARPD_p$ to compare our best-obtained solutions with the ones of Pan and Ruiz [22] and those obtained with other heuristics.

4.1. Comparing AGB and LS for the general case

This section compares the performance of AGB and LS. As mentioned before, both algorithms have the same complexity. Tables 1 and 2 show that using AGB leads to better results than using the LS heuristic; Table 1 summarizes the results for small-size problems and Table 2 the performance for large size-problems. Considering that CPU times for small-size problems are negligible, we refrained from reporting the same. In our experiments, AGB showed a better performance than LS in terms of the quality of the solutions, with an average increase of 2.2% in CPU time for large-size problems. This means that without losing a significant amount of efficiency, AGB outperforms LS. The average value of all performance figures is also presented in Tables 1 and 2.

According to Table 1, for small-size problems, AGB leads to the optimal solution in 78% of the cases. LS shows a similar performance in this respect by finding the optimal solution in 76.2% of the cases. The average error of AGB for small-size problems is 0.16%, which is an improvement of almost 15% as compared to LS, whose average error is 0.19%. Table 2 shows that the proposed algorithm outperforms LS for large-size problems as well. The average error ($ARPD_L$) of AGB is 2.219% for large-size problems, whereas LS led to an average error of 2.396%. Thus, our heuristic led to an average of 7.38% improvement as compared to LS with almost identical CPU time.

n	m	No. of Problems	LS		AGB	
			ARPD	% Opt.	ARPD	% Opt.
5	5	100	0.096	88.0	0.086	90.0
	10	100	0.039	96.0	0.039	96.0
	15	100	0.053	90.0	0.051	91.0
	20	100	0.074	90.0	0.072	92.0
		Ave	**0.066**	**91.0**	**0.062**	**92.3**
6	5	100	0.316	76.0	0.225	79.0
	10	100	0.134	83.0	0.093	85.0
	15	100	0.127	81.0	0.117	85.0
	20	100	0.087	86.0	0.079	87.0
		Ave	**0.166**	**81.5**	**0.129**	**84.0**
7	5	100	0.235	73.0	0.223	73.0
	10	100	0.329	64.0	0.291	65.0
	15	100	0.168	76.0	0.132	75.0
	20	100	0.169	70.0	0.162	72.0
		Ave	**0.225**	**70.8**	**0.202**	**71.3**
8	5	100	0.381	62.0	0.342	63.0
	10	100	0.326	60.0	0.265	65.0
	15	100	0.292	54.0	0.213	62.0
	20	100	0.201	70.0	0.187	68.0

	Ave	**0.300**	**61.5**	**0.252**	**64.5**
	Total Ave	**0.189**	**76.2**	**0.161**	**78.0**

Table 1: Comparing AGB and LS for the general case: Summary of results for small-size problems

4.2. Evaluating the effect of the improvement options for field α on the NEH//A, LS and AGB heuristics

This section analyzes the effect of the indicator variables proposed in Section 3.2 on the performance of the NEH//A, LS and AGB heuristics. Table 3 shows the ARPD and the percentage of optimal solutions obtained for the 105 considered algorithms for small-size problems. Similarly than in the previous chapter, we restrain from reporting CPU times for small-size problems due to their negligibility. As the objective here is to evaluate the effect of different sorting mechanisms, we focus only on the results of algorithms with TFT as decision criterion.

We conclude from Table 3 that the original NEH//A sorting criterion, P_i, is in most cases amongst the best choices for small size problems. For NEH//A, the best-performing indicator values are SIP_i and $ABS(SIP_i)$. The best ARPD has the value 0.758 and was achieved with the $ABS(SIP_i)$ indicator variable and shows an improvement of 14.5% as compared to the original NEH. For LS, the best results were obtained with the $MTWPT_i$ method and led to an ARPD-value of 0.175, which is a 7.4% improvement as compared to the original LS. The best-performing indicator variables for AGB are P_i and $MTWPT_i$. The best-performing algorithm with respect to ARPD employs $MTWPT_i$ as indicator variable and results in an ARPD-value of 0.151. This is a 6.2% improvement as compared to the basic version of AGB with P_i as indicator.

n	m	No. of Problems	LS			AGB		
			$ARPD_L$	$ARPD_P$	CPU time [s]	$ARPD_L$	$ARPD_P$	CPU time [s]
20	5	10	1.927	1.357	0.014	1.969	1.486	0.014
	10	10	1.384	0.941	0.012	1.259	1.02	0.011
	20	10	1.429	1.231	0.013	1.273	1.12	0.014
		Ave	**1.580**	**1.177**	**0.014**	**1.500**	**1.209**	**0.013**
50	5	10	2.238	1.031	0.325	1.923	0.737	0.327
	10	10	3.334	1.666	0.363	3.318	1.65	0.388
	20	10	2.821	1.256	0.430	2.679	1.225	0.417
		Ave	**2.798**	**1.318**	**0.373**	**2.64**	**1.204**	**0.377**
100	5	10	2.356	1.117	5.013	2.293	1.09	4.975
	10	10	3.273	1.480	5.138	3.07	1.291	5.147
	20	10	3.730	1.634	6.072	3.388	1.326	5.941
		Ave	**3.12**	**1.411**	**5.408**	**2.917**	**1.236**	**5.354**
200	10	10	2.356	1.236	86.927	2.053	0.954	85.026
	20	10	2.805	1.413	91.709	2.314	0.931	91.427

		Ave	**2.761**	**1.354**	**61.348**	**2.184**	**0.943**	**88.227**
500	20	10	1.099	0.706	3484.262	1.096	0.703	3474.173
		Total Ave	**2.396**	**1.256**	**306.690**	**2.219**	**1.127**	**305.655**

Table 2: Comparing AGB and LS for the general case: Summary of results for large-size problems.

Table 4 shows the $ARPD_L$- and $ARPD_P$-values as well as the CPU times for the 105 different combinations of the α- and γ-parameters used in NEH//A, LS and AGB for large-size problems. The first conclusion we draw from Table 4 is that employing different indicator values or decision criteria does not affect the complexity of each individual heuristic; it can, however, dramatically influence the quality of the results. By focusing on the results of the algorithms with the TFT decision criterion in Table 4, we conclude for NEH//A that P_i, $ABS(SIP_i)$, $MTWPT_i$ and $MJTWPT_i$ perform better than the other methods introduced above. The best $ARPD_L$ is of value 4.238 and was achieved with the $ABS(SIP_i)$ indicator for NEH//A. This is an improvement of 15.5% as compared to the original P_i indicator. For LS, P_i, $ABS(SIP_i)$, SIR_i and $MTWPT_i$ showed the best performance, and the best $ARPD_L$ is of value 2.251 obtained by the $MTWPT_i$ indicator, which is a 6% improvement as compared to the original LS. The most promising indicators for AGB are P_i, SIR_i, and $MTWPT_i$. The best performing algorithm with respect to $ARPD_L$ employs SIR_i as indicator variable and leads to a 2.098% average error, which is a 5.5% improvement as compared to original AGB.

		P_i	SIP_i	$ABS(SIP_i)$	SIG_i	SIR_i	$MTWPT_i$	$MJTWPT_i$
NEH//A								
TFT	ARPD	0.887	0.764	0.758	0.797	0.896	0.879	0.885
	%Opt	39.0	42.9	42.3	40.8	38.8	39.8	39.2
$TWFT$	ARPD	1.154	1.070	1.119	1.243	2.490	1.155	1.172
	%Opt	31.4	31.7	31.8	28.7	17.2	32.2	31.1
$TWFT_8$	ARPD	0.887	0.764	0.758	0.797	1.559	0.879	0.885
	%Opt	39.0	42.9	42.3	40.8	25.8	39.8	39.2
$TWFT_{16}$	ARPD	0.887	0.764	0.758	0.797	1.559	0.879	0.885
	%Opt	39.0	42.9	42.3	40.8	25.8	39.8	39.2
$TWFT_{24}$	ARPD	0.887	0.764	0.758	0.797	1.559	0.879	0.885
	%Opt	39.0	42.9	42.3	40.8	25.8	39.8	39.2
LS								
TFT	ARPD	0.887	0.764	0.758	0.797	0.896	0.879	0.885
	%Opt	39.0	42.9	42.3	40.8	38.8	39.8	39.2
$TWFT$	ARPD	1.154	1.070	1.119	1.243	2.490	1.155	1.172
	%Opt	31.4	31.7	31.8	28.7	17.2	32.2	31.1
$TWFT_8$	ARPD	0.887	0.764	0.758	0.797	1.559	0.879	0.885
	%Opt	39.0	42.9	42.3	40.8	25.8	39.8	39.2
$TWFT_{16}$	ARPD	0.887	0.764	0.758	0.797	1.559	0.879	0.885
	%Opt	39.0	42.9	42.3	40.8	25.8	39.8	39.2
	ARPD	0.887	0.764	0.758	0.797	1.559	0.879	0.885

$TWFT_{24}$	%Opt	39.0	42.9	42.3	40.8	25.8	39.8	39.2
AGB								
TFT	ARPD	0.887	0.764	0.758	0.797	0.896	0.879	0.885
	%Opt	39.0	42.9	42.3	40.8	38.8	39.8	39.2
$TWFT$	ARPD	1.154	1.070	1.119	1.243	2.490	1.155	1.172
	%Opt	31.4	31.7	31.8	28.7	17.2	32.2	31.1
$TWFT_8$	ARPD	0.887	0.764	0.758	0.797	1.559	0.879	0.885
	%Opt	39.0	42.9	42.3	40.8	25.8	39.8	39.2
$TWFT_{16}$	ARPD	0.887	0.764	0.758	0.797	1.559	0.879	0.885
	%Opt	39.0	42.9	42.3	40.8	25.8	39.8	39.2
$TWFT_{24}$	ARPD	0.887	0.764	0.758	0.797	1.559	0.879	0.885
	%Opt	39.0	42.9	42.3	40.8	25.8	39.8	39.2

Table 3: Summary of results for different α- and γ-values for small-size problems.

It can further be seen in Table 3 and Table 4 that there is no universally best-performing indicator variable; thus, for each heuristic, the individually best indicator variable needs to be chosen. Our results indicate that in most cases, using Gupta's indices - SIG_i - as indicators results in worse solutions with an average loss of 39% and 14% in the quality of solutions for large- and small-size problems, respectively, as compared to the original indicator P_i. On the other hand, applying $MTWPT_i$ or $ABS(SIP_i)$ indicators leads to better solutions in most cases. The average errors of all heuristics using $MTWPT_i$ indicators were improved by 4.1% and 4.8% for large- and small-size problems, respectively. This improvement was 5.5% and 8.1% for $ABS(SIP_i)$. SIP_i performed well on LS- and AGB-based algorithms, while it did not perform well on NEH-based heuristics.

4.3. Evaluating the effect of the improvement options for the field γ on the NEH//A, LS and AGB heuristics

This section analyzes the effect of the new decision criteria γ proposed in Section 3.3 on the performance of the NEH//A, LS and AGB heuristics. We conclude from Table 3 that for NEH//A, LS and AGB, the original TFT decision criterion leads to the best results for small-size problems. This is not surprising, as the effect of using TWFT is expected to increase as the number of jobs in the sequence gets larger. When the number of jobs in the sequence is small (i.e., smaller than 10), the effect of assigning higher weights to the jobs at the end of the sequence does not make a significant difference. As the maximum number of jobs in our small-size problems is 8, it is evident that the results for TFT and all varieties of $TWFT_k$ are identical.

This section evaluates the effect of the different decision criteria (i.e., the different options for the γ parameter), which is why we focus only on the results of algorithms with P_i as indicator variable. It can be seen in Table 4 that the effect of applying different γ-parameters on different heuristics depends on the employed sorting indicator, the α-parameter, and the utilized heuristic. Although almost none of the γ-parameters performs well when they are applied on heuristics using SIP_i, SIG_i, or $MJTWPT_i$ as their α-parameter, they all outperform their respective original heuristics when the α-parameter of the heuristics is P_i, SIR_i or $MTWPT_i$. For the NEH//A heuristic, the best $ARPD_L$ is of value 4.423 and was achieved with the $TWFT$ method. This is an improvement of 11.6% as compared to the original NEH//A. The best $ARPD_L$ for LS is of value 1.962 and was obtained by the $TWFT_{24}$ approach, which shows an 18.11% improvement as compared to the original LS. The best-performing algorithm with respect to

AGB employs $TWFT_{16}$ as decision criterion and has a 17.7% improvement in $ARPD_L$ as compared to the basic AGB, which leads to a value of 1.826.

According to Tables 3 and 4, combining the improvement options for the α-indicator and the γ decision criteria leads to a higher performance improvement than using isolated improvement options. For example, combining all improvement options can enhance the $ARPD_L$ of NEH//A to 4.09, which is a 18.4% improvement. The best LS-based heuristic is $LS/MTWPT_i/A/TWFT_{24}$ with an average error of 1.941, which is a 19% improvement as compared to LS.

The results presented in this section and Section 4.2 indicate that $AGB/MTWPT_i/A/TWFT_8$ is the best-performing algorithm for both small- and large-size problems. In order to analyze the performance of $AGB/MTWPT_i/A/TWFT_8$ in detail, we compare this heuristic to LS analogue to the analysis in Section 4.1. Tables 5 and 6 illustrate that using $AGB/MTWPT_i/A/TWFT_8$ leads to better results than LS. While CPU times have not been reported in Table 5 due to their negligibility for small-size problems, it can be seen from Table 6 that $AGB/MTWPT_i/ASC/TWFT_8$ consumes on average 4.04% less CPU time than LS. According to Table 5, $AGB/MTWPT_i/A/TWFT_8$ leads to the optimal solution in 79.25% of the cases for small-size problems, which is an improvement of more than 3%. The average error of $AGB/MTWPT_i/A/TWFT_8$ for small-size problems is 0.151%, which is an improvement of more than 20% as compared to LS. Table 6 shows that the proposed algorithm outperforms LS for all categories of large-size problems as well. The $ARPD_L$-value for $AGB/MTWPT_i/A/TWFT_8$ in this case is 1.813%, which is a 24.33% improvement as compared to LS. This algorithm shows a 42% improvement as compared to LS on the $ARPD_P$ metrics.

		P_i	SIP_i	$ABS (SIP_i)$	SIG_i	SIR_i	$MTWPT_i$	$MJTWPT_i$	P_i	SIP_i	$ABS (SIP_i)$	SIG_i	SIR_i	$MTWPT_i$	$MJTWPT_i$
	NEH//A								**LS**						
TFT	$ARPD_L$	5.013	5.521	4.283	5.944	5.390	4.865	5.013	2.396	3.249	2.334	3.627	2.330	2.251	2.765
	$ARPD_P$	3.890	4.393	3.168	4.811	4.262	3.743	3.907	1.256	2.100	1.195	2.473	1.191	1.113	1.622
	CPU time [s]	7.784	7.690	7.820	7.749	7.707	7.785	7.653	306.690	319.704	305.170	320.408	323.295	300.178	296.015
$TWFT$	$ARPD_L$	4.423	7.499	4.090	7.560	4.824	4.500	4.423	1.962	4.115	2.098	4.783	2.032	2.012	2.775
	$ARPD_P$	3.309	6.349	2.979	6.409	3.70	3.384	3.836	0.828	2.956	0.962	3.616	0.896	0.877	1.632
	CPU time [s]	1.000	1.006	0.996	1.006	1.001	0.991	0.987	299.230	308.742	297.168	306.365	306.422	298.981	295.371
$TWFT_8$	$ARPD_L$	4.537	7.221	4.249	7.379	5.069	4.456	4.537	2.017	4.017	2.171	4.811	2.057	1.955	2.776
	$ARPD_P$	3.421	6.073	3.136	6.230	3.946	3.340	3.958	0.882	2.858	1.035	3.644	0.921	0.820	1.633
	CPU time [s]	0.990	1.000	0.994	1.001	0.902	0.995	0.983	298.948	307.392	296.891	307.713	307.840	305.823	302.725
$TWFT_{16}$	$ARPD_L$	4.720	6.862	4.357	7.203	5.172	4.677	4.720	2.097	3.965	2.171	4.567	2.174	1.974	2.868
	$ARPD_P$	3.603	5.718	3.243	6.054	4.049	3.559	4.197	0.962	2.807	1.035	3.402	1.037	0.839	1.724
	CPU time [s]	0.993	1.000	0.994	0.997	0.899	0.989	0.988	306.721	307.201	309.331	307.188	310.067	302.651	294.648
$TWFT_{24}$	$ARPD_L$	4.880	6.493	4.334	6.975	5.295	4.658	4.880	2.018	3.7727	2.162	4.389	2.107	1.941	2.800
	$ARPD_P$	3.760	5.352	3.220	5.829	4.169	3.539	4.120	0.883	2.616	1.026	3.225	0.971	0.807	1.657
	CPU time [s]	0.991	1.009	0.998	1.005	1.007	0.991	0.988	302.811	317.690	305.643	317.651	319.841	304.754	302.154
	AGB														
TFT	$ARPD_L$	2.219	2.872	2.232	3.300	2.098	2.141	2.666							
	$ARPD_P$	1.127	1.774	1.141	2.197	1.008	1.050	1.570							
	CPU time [s]	305.655	320.171	306.764	321.969	319.646	302.774	307.892							
$TWFT$	$ARPD_L$	1.917	3.563	2.069	4.262	1.912	1.831	2.708							
	$ARPD_P$	0.829	2.457	0.981	3.148	0.825	0.744	1.612							
	CPU time [s]	311.826	309.214	310.495	307.300	307.311	308.244	306.895							
$TWFT_8$	$ARPD_L$	1.890	3.539	2.059	4.238	1.868	1.813	2.764							
	$ARPD_P$	0.802	2.433	0.970	3.125	0.781	0.727	1.668							
	CPU time [s]	308.450	307.670	299.829	307.763	307.972	294.448	291.386							
$TWFT_{16}$	$ARPD_L$	1.826	3.415	2.012	4.130	1.884	1.836	2.726							
	$ARPD_P$	0.740	2.311	0.923	3.017	0.797	0.750	1.631							
	CPU time [s]	292.754	307.794	291.924	307.770	310.866	292.918	292.065							
$TWFT_{24}$	$ARPD_L$	1.908	3.357	1.973	3.967	1.941	1.830	2.685							
	$ARPD_P$	0.821	2.252	0.885	2.855	0.852	0.744	1.590							
	CPU time [s]	293.394	320.925	291.949	317.383	320.386	299.351	300.854							

Table 4: Summary of results for different α- and γ-values for large-size problems.

n	m	No. of Problems	LS		$AGB/MTWPT_i/A/TWFT_8$	
			ARPD	% Opt.	ARPD	% Opt.
5	5	100	0.096	88.0	0.087	91.0
	10	100	0.039	96.0	0.039	96.0
	15	100	0.053	90.0	0.052	89.0
	20	100	0.074	90.0	0.056	94.0
		Ave	**0.066**	**91.0**	**0.059**	**92.5**
6	5	100	0.316	76.0	0.206	83.0
	10	100	0.134	83.0	0.110	85.0
	15	100	0.127	81.0	0.117	85.0
	20	100	0.087	86.0	0.071	88.0
		Ave	**0.166**	**81.5**	**0.126**	**85.25**
7	5	100	0.235	73.0	0.170	76.0
	10	100	0.329	64.0	0.233	68.0
	15	100	0.168	76.0	0.101	78.0
	20	100	0.169	70.0	0.132	77.0
		Ave	**0.225**	**70.8**	**0.159**	**74.75**
8	5	100	0.381	62.0	0.352	65.0
	10	100	0.326	60.0	0.253	64.0
	15	100	0.292	54.0	0.243	60.0
	20	100	0.201	70.0	0.190	69.0
		Ave	**0.300**	**61.5**	**0.259**	**64.5**
		Total Ave	**0.189**	**76.2**	**0.151**	**79.25**

Table 5: Comparing LS and $AGB/MTWPT_i/A/TWFT_8$: Summary of results for small-size problems.

n	m	No. of Problems	LS			$AGB/MTWPT_i/A/TWFT_8$		
			$ARPD_L$	$ARPD_P$	CPU time [s]	$ARPD_L$	$ARPD_P$	CPU time [s]
20	5	10	1.927	1.357	0.014	1.604	1.123	0.010
	10	10	1.384	0.941	0.012	1.322	1.083	0.012
	20	10	1.429	1.231	0.013	0.979	0.826	0.013
		Ave	**1.580**	**1.177**	**0.014**	**1.302**	**1.011**	**0.012**
50	5	10	2.238	1.031	0.325	1.729	0.546	0.316
	10	10	3.334	1.666	0.363	2.513	0.859	0.362
	20	10	2.821	1.256	0.430	2.276	0.829	0.433
		Ave	**2.798**	**1.318**	**0.373**	**2.173**	**0.745**	**0.37**
100	5	10	2.356	1.117	5.013	1.721	0.526	4.887
	10	10	3.273	1.480	5.138	2.214	0.450	5.101
	20	10	3.730	1.634	6.072	3.018	0.963	5.898
		Ave	**3.12**	**1.411**	**5.408**	**2.318**	**0.646**	**5.295**
200	10	10	2.356	1.236	86.927	1.698	0.603	85.021
	20	10	2.805	1.413	91.709	1.813	0.437	91.587
		Ave	**2.761**	**1.354**	**61.348**	**1.756**	**0.521**	**58.869**
500	20	10	1.099	0.706	3484.262	0.870	0.478	3339.737
		Total Ave	**2.396**	**1.256**	**306.690**	**1.813**	**0.7273**	**252.384**

Table 6: Comparing LS and $AGB/MTWPT_i/A/TWFT_8$: Summary of results for small-size problems.

4.4. Statistical analysis

This section compares the results of AGB and the best-performing heuristic identified in Section 4.3, $AGB/MTWPT_i/A/TWFT_8$, to those obtained by the LS heuristics by calculating t-statistics. The purpose of this statistical analysis is to ensure that the proposed heuristics outperform LS in the general case, and not only for the generated random small-size problems and Taillard's benchmark set. For each group of problems, characterized by the number of jobs, n, and the number of machines, m, the mean and the standard deviation were calculated. The difference in total flow-times for each problem was obtained by subtracting the result obtained by the proposed algorithms from the total flow-time obtained by LS. For testing the null hypothesis $H_o\colon\ \mu = 0$, we computed the t-statistic as follows: $t = \sqrt{N}\frac{\bar{X}-\mu_0}{S}$ with the sample size N, the sample mean $\bar{X}$, the standard deviation S, $\mu = 0$ and $N-1$ degrees of freedom. If the null hypothesis holds, the difference between the two methods is statistically insignificant. We obtained the critical value, c, from the relation: $Probability(t > c) = \alpha = 5\%$. Using the standard tables of the t-distribution, we obtain $c = 1.66$ for 99 degrees of freedom (for small-size problems) and $c = 1.833$ for 9 degrees of freedom (for large-size problems). Table 7 reports the results for the test of statistical significance. As can be seen, AGB performs statistically better than LS in 8 out of 28 cases, and $AGB/MTWPT_i/A/TWFT_8$ performs better in 22 out of 28 cases. It is therefore evident that the latter heuristic outperforms LS.

n	m	No. of Problems	LS vs. AGB			LS vs. $AGB/MTWPT_i/A/TWFT_8$		
			Total flow time difference		t	Total flow time difference		t
			Mean	St. Dev.		Mean	St. Dev.	
5	5	100	0.18	1.445	1.246	1.06	5.312	1.996
	10	100	0.00	0.000	0.000	0.00	0.00	0.000
	15	100	0.09	1.001	0.899	1.24	5.726	2.166
	20	100	0.17	11.362	0.150	0.50	2.707	1.847
6	5	100	2.11	9.041	2.334	2.61	10.727	2.433
	10	100	1.58	10.462	1.510	1.67	9.010	1.853
	15	100	0.46	5.326	0.864	1.19	6.792	1.752
	20	100	0.59	5.400	1.093	2.58	12.727	2.027
7	5	100	0.46	7.231	0.636	1.82	8.767	2.076
	10	100	1.78	8.964	1.986	4.44	22.210	1.999
	15	100	2.37	18.726	1.266	4.32	22.260	1.941
	20	100	0.56	9.883	0.567	3.18	14.278	2.227
8	5	100	1.19	9.228	1.290	2.11	9.041	2.334
	10	100	3.54	15.082	2.347	4.33	23.417	1.849
	15	100	6.32	24.925	2.536	4.01	18.092	2.216
	20	100	1.47	17.309	0.849	2.77	16.527	1.676
20	5	10	-7.50	77.386	-0.306	45.90	173.308	0.838
	10	10	31.8	157.205	0.640	17.40	132.571	0.415
	20	10	50.70	104.875	1.529	147.60	262.315	1.779
50	5	10	212.00	266.751	2.513	341.30	638.151	1.691
	10	10	1.30	951.114	0.004	688.80	829.202	2.627
	20	10	165.10	702.177	0.744	660.00	864.324	2.415
100	5	10	134.00	1143.042	0.371	1510.10	1983.277	2.408
	10	10	587.50	1453.332	1.278	3078.20	1362.411	7.145
	20	10	1258.80	1730.372	2.300	2646.50	2630.821	3.181
200	10	10	3130.00	4665.167	2.122	6771.30	6693.018	3.199
	20	10	6060.70	8177.792	2.344	12283.80	5478.073	7.091
500	20	10	298.10	18732.177	0.050	15424.00	34203.407	1.426

Table 7: Results of the statistical test.

5 Summary and conclusion

The best known simple constructive heuristics algorithm for optimizing the permutation flowshop scheduling problem with total flow-time as objective, developed by Laha and Sarin (2009), was modified in this paper by employing different improvement options, which led to 103 new heuristics. A comprehensive computational experiment was conducted to examine the effects of each modification in detail. One of our metrics for comparing the performance of algorithms showed that our modifications result in a 24% improvement in $ARPD_L$ as compared to the LS heuristic. Another metric, the $ARPD_P$, showed a 42% improvement. Our statistical analyses demonstrate that the modification proposed and evaluated in this paper lead to a superior simple constructive heuristic, which outperforms the LS heuristic in the general case. None of the improvements discussed in this paper affected the time-efficiency of the LS heuristic.

Besides modifying the LS algorithm to improve its performance, we also developed new ideas for weighting jobs to be scheduled and indexing them. Our numerical studies indicated that using alternative sorting methods (i.e. the indicator variables) for weighting jobs can improve/worsen the performance of the algorithm. Another contribution of the paper is the utilization of newly-defined decision criteria for selecting the best partial sequence in each iteration of the NEH heuristic. Our numerical studies showed that using decision criteria different from the objective of the problem can lead to better results. Besides these contributions, this paper addressed the problem of imprecise reporting practices in the computational experiments of the published work and proposed the indicator $ARPD_L$ for evaluating the effectiveness of heuristics.

Future research could aim on developing more effective sorting approaches and decision criteria, and examine the effects of different indicator variables on the performance of the decision criteria. The application of the proposed heuristics in developing new composite heuristics could also be interesting. A modified version of the AGB and the ideas proposed here to improve it could also be employed for other objectives, such as minimizing makespan.

References

Allahverdi, A., & Aldowaisan, T. (2002). New heuristics to minimize total completion time in m-machine flowshops. International Journal of Production Economics, 77(1), 71–83.

Campbell, H. G., Dudek, R. A., & Smith, M. L. (1970). A HEURISTIC ALGORITHM FOR THE n JOB, m MACHINE SEQUENCING PROBLEM. Management Science, 16(10), B630–B637.

Dong, X., Huang, H. & Chen, P. (2008). An improved NEH-based heuristic for the permutation flowshop problem. Computers & Operations Research, 35, 3962–3968.

Fernandez-Viagas, V. & Framinan, J. M. (2014). On insertion tie-breaking rules in heuristics for the permutation flowshop scheduling problem. Computers & Operations Research, 45, 60-67.

Fernandez-Viagas, V. & Framinan, J.M. (2015): A new set of high-performing heuristics to minimize flowtime in permutation flowshops. Computers & Operations Research, 53, 68-80.

Framinan, J. M., & Leisten, R. (2003). An efficient constructive heuristic for flowtime minimisation in permutation flow shops. Omega, 31(4), 311–317.

Framinan, J. M., Leisten, R., & Rajendran, C. (2003). Different initial sequences for the heuristic of Nawaz, Enscore and Ham to minimize makespan, idletime or flowtime in the static permutation flowshop sequencing problem. International Journal of Production Research, 41(1), 121–148.

Framinan, J. M., Leisten, R., & Ruiz-Usano, R. (2002). Efficient heuristics for flowshop sequencing with the objectives of makespan and flowtime minimisation. European Journal of Operational Research, 141, 559–569.

Framinan, J. M., Leisten, R., & Ruiz-Usano, R. (2005). Comparison of heuristics for flowtime minimisation in permutation flowshops. Computers & Operations Research, 32(5), 1237–1254.

Framinan, J. M., Gupta, J. N. D. & Leisten, R. (2004). A review and classification of heuristics for permutation flow-shop scheduling with makespan objective. Journal of the Operational Research Society, 55(12), 1243-1255.

Garey, M. R., Johnson, D. S., & Sethi, R. (1976). The Complexity of Flowshop and Jobshop Scheduling. Mathematics of Operations Research, 1(2), 117–129.

Gupta, J. N. (1971). A functional heuristic algorithm for the flowshop scheduling problem. Operational Research Quarterly, 22(1), 39–47.

Gupta, J. N. D., Koulamas, C., & Kyparisis, G. J. (2006). Performance guarantees for flowshop heuristics to minimize makespan. European Journal of Operational Research, 169(3), 865–872.

Gupta, J. N. D., & Stafford, E. F. (2006). Flowshop scheduling research after five decades. European Journal of Operational Research, 169(3), 699–711.

Jarboui, B., Eddaly, M. & Siarry, P. (2009): An estimation of distribution algorithm for minimizing the total flowtime in permutation flowshop scheduling problems. Computers & Operations Research, 36, 2638–2646.

Kalczynski, P. J., & Kamburowski, J. (2009). An empirical analysis of the optimality rate of flow shop heuristics. European Journal of Operational Research, 198(1), 93–101.

Laha, D., & Sarin, S. (2009). A heuristic to minimize total flow time in permutation flow shop. Omega, 37(3), 734–739.

Mutlu, M., & Yagmahan, B. (2014). Multi-objective permutation flow shop scheduling problem: Literature review, classification and current trends. Omega, 45, 119–135.

Nawaz, M., Enscore Jr., E. E., & Ham, I. (1983). A Heuristic Algorithm for the m-Machine, n-Job Flow-shop Sequencing Problem. Omega, 11(1), 91–95.

Palmer, D. (1965). Sequencing jobs through a multi-stage process in the minimum total time -A quick method of obtaining a near optimum. Operational Research Quarterly, 16(1), 101–107.

Pan, Q., & Ruiz, R. (2012). Local search methods for the flowshop scheduling problem with flowtime minimization. European Journal of Operational Research, 222(1), 31-43.

Pan, Q., & Ruiz, R. (2013). A comprehensive review and evaluation of permutation flowshop heuristics to minimize flowtime. Computer and Operations Research, 40, 117–128.

Pinedo, M. L. (2008). Scheduling Theory, Algorithms, and Systems (Third., p. 664). Springer.

Rajendran, C. (1993). Heuristic algorithm for scheduling in a flowshop to minimize total flowtime. International Journal of Production Economics, 29(1), 65–73.

Rajendran, C. & Ziegler, H. (1997). An efficient heuristic for scheduling in a flowshop to minimize total weighted flowtime of jobs. European Journal of Operational Research, 103, 129-138.

Reisman, A., Kumar, A., & Motwani, J. (1997). Flowshop scheduling/sequencing research: a statistical review of the literature, 1952-1994. IEEE Transactions on Engineering Management, 44(3), 316–329.

Reza Hejazi, S., & Saghafian, S. (2005). Flowshop-scheduling problems with makespan criterion: a review. International Journal of Production Research, 43(14), 2895–2929.

Ruiz, R., & Maroto, C. (2005). A comprehensive review and evaluation of permutation flowshop heuristics. European Journal of Operational Research, 165(2), 479–494.

Semančo, P. & Modrák, V. (2012). A Comparison of Constructive Heuristics with the Objective of Minimizing Makespan in the Flow-Shop Scheduling Problem. Acta Polytechnica Hungarica, 9(5), 177-190.

Taillard, E. (1993). Benchmarks for basic scheduling problems. European Journal of Operational Research, 64(2), 278–285.

Vasiljevic, D. & Danilovic, M. (2015): Handling ties in heuristics for the permutation flow shop scheduling problem. Journal of Manufacturing Systems, 35, 1-9.

Part B Scheduling problems in logistics

Paper 4 Scheduling electric vehicles making milk-runs for just-in-time delivery

Authors: Simon Emde, Hamid Abedinnia, Christoph Glock

Type of publication: Journal article

Publication details: *IISE Transactions*: Accepted for publication

Abstract

Battery-operated electric vehicles are frequently used in in-plant logistics systems to feed parts from a central depot to workcells on the shopfloor. These vehicles, often so-called tow trains, make many milk-run trips during a typical day, with the delivery time-table depending on the production schedule. To operate such a milk-run delivery system efficiently, not only do the timetabled trips need to be assigned to vehicles, it is also important to take the limited battery capacity into consideration. Moreover, since most tow trains in use today are still operated by human drivers, fairness aspects with respect to the division of the workload also need to be considered. In this context, we tackle the following problem we encountered at a large manufacturer of engines for trucks and busses in Germany. Given a fixed schedule of milk-runs (round trips) to be performed during a planning horizon and a fleet of homogeneous electric vehicles stationed at a depot, which vehicle should set out on which milk-run and when should recharging breaks be scheduled, such that all runs can be completed with the minimum number of vehicles and all vehicles are about equally busy? We investigate the computational complexity of this problem and develop suitable heuristics, which are shown to solve instances of realistic size to near-optimality in a matter of a few minutes. We also offer some insight into how battery technology influences vehicle utilization.

Keywords: *Production logistics, Electric vehicles, Vehicle scheduling, Tow trains, Fairness*

1 Introduction

Tow train delivery systems have become increasingly popular in a number of manufacturing industries to ensure a steady and reliable in-plant part supply (e.g., Faccio et al., 2013). In this context, tow trains, i.e., small electric tow trucks connected to a handful of waggons, are used to make periodic deliveries of parts and subassemblies from a depot (often a so-called supermarket, Battini et al. (2013)) to nearby workcells, often in the form of milk-runs (round trips). Tow trains, or more generally small electric delivery vehicles, have become particularly popular in final assembly of automobiles, where they are used on a large scale (e.g., Emde et al., 2012, Emde and Boysen, 2012). Further practical examples are discussed, e.g., by Vaidyanathan et al. (1999), who report on such vehicles being used in a factory producing exhaust systems, and Akillioğlu et al. (2006), who describe a case from a company making diesel injectors.

The paper at hand, however, is specifically motivated by a case we observed at a large manufacturer of engines for trucks and busses in Germany. The OEM stores engine parts in a warehouse with different workplaces, where each workplace in the warehouse is assigned to a single workcell at the assembly lines in the production facility of the company. Upon arrival of an order at the production facility, the production department issues orders to the warehouse for retrieving the parts required for assembling the respective engine. The warehouse workers then collect the required parts in the warehouse and place them in a pre-specified sequence in a stillage at their respective workplaces. Electric tow trains are used for transporting the filled stillages from the warehouse to the workplaces of the production department. Due to limited storage space at the assembly line, it is not possible to store parts for more than two engines at the workplaces of this facility. As a consequence, parts are transported from the warehouse to the production department on an order-by-order basis. At the end of each tour, the tow trains return to the depot, where they are either recharged or sent on the next tour. As different workplaces may need to be visited both in the warehouse and in the production facility depending on the engine to be produced, the OEM faces a set of different tours that differ in their length and the weight of items that need to be transported. The scenario observed at the OEM is illustrated schematically in Figure 1.

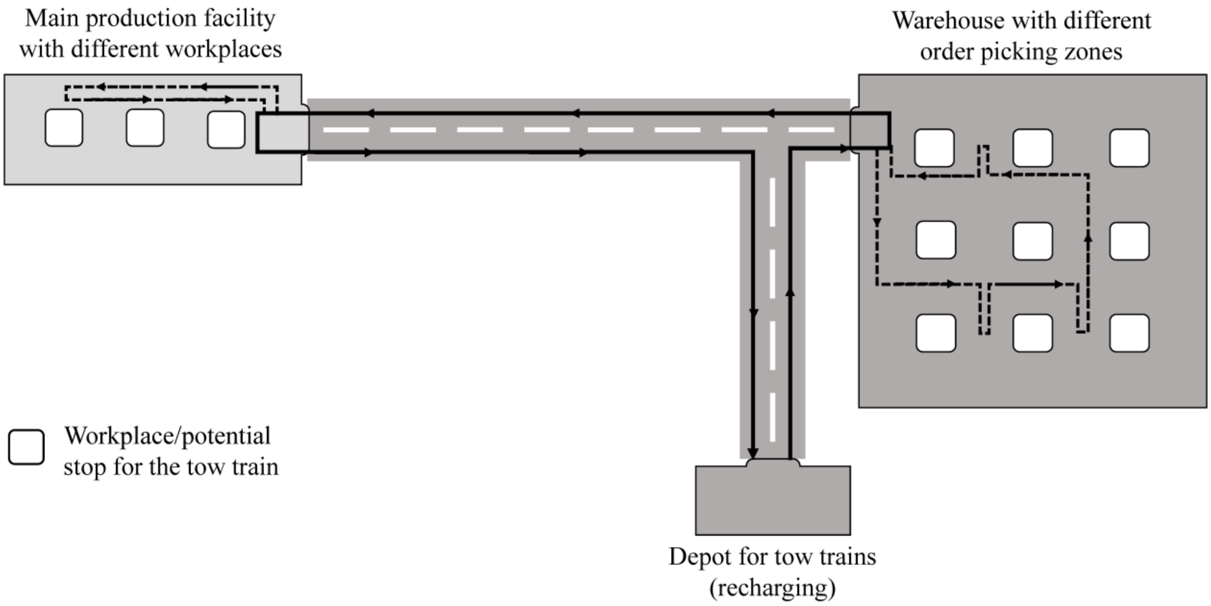


Figure 1: Part feeding with electric vehicles at our OEM partner.

While electric vehicles have proven to be an efficient means of in-plant part feeding, where engines burning fossil fuels are not an attractive (and in many scenarios not even legal) option due to exhaust fumes, they do have the drawback of limited battery capacity. Battery powered tuggers have to return to the depot periodically for recharging, which complicates the process of planning and executing timely deliveries. Apart from performing scheduled trips, i.e., carrying parts to points of consumption on the shopfloor and returning to the depot, recharging intervals also need to be carefully planned lest the vehicle fleet grow excessively large.

A second aspect that should be taken into consideration in this context is the notion of fairness. Tow trains are occasionally fully automated guided vehicles, but in most cases they still require a human operator (Emde and Gendreau, 2015). Merely keeping the vehicle fleet small may lead to an uneven workload, where some tuggers and their operators have a significantly higher workload than their colleagues. This situation is often perceived as unfair, and can be remedied by load balancing (e.g., Lee and Ueng, 1999). Apart from this driver-centric perspective, load balancing among electric vehicles can also be desirable to ensure longevity of the batteries. The expected lifetime of lead acid batteries, for example, drops if they are used while the state of charge is low (e.g., Sauer and Wenzl, 2008, Dufo-López et al., 2014). Balancing the load among vehicles makes it easier to afford every battery sfficient time to recharge.

In this context, we consider the following electric vehicle scheduling problem. Given are a fixed schedule of milk-runs (round trips) to be performed during the planning horizon and a fleet of homogeneous electric vehicles (typically tow trains) stationed at a depot. Each milk-run must be performed by exactly one vehicle. The vehicles are electrically powered, and each milk-run requires a certain, known amount of charge from the battery. Therefore, a vehicle can only be assigned to a run if it has sufficient charge left. The battery can be recharged at the depot, which will take the more time the more depleted the battery already is. The question asked is: Which vehicle should set out on which milk-run, and when should recharging breaks be scheduled, such that all runs can be completed with the minimum number of vehicles and all vehicles are about equally busy?

The contribution of this paper is threefold. First, we present and model the new problem of planning milk-run deliveries with electric vehicles. Second, we develop a powerful heuristic solution method based on tabu search for this problem, which is shown to solve instances of realistic size to (near-) optimality in a matter of a few minutes. Finally, in a comprehensive computational study, we explore the inherent tradeoff between battery capacity and fleet size, deriving some managerial insight into the ideal composition of the vehicle fleet.

The remainder of this paper is structured as follows. In Section 2, we will review the pertinent literature. In Section 3, we will formally define the problem and investigate its computational complexity in Section 4. In Section 5, we propose a MIP model as well as several heuristic algorithms, which are tested in a computational study in Section 6, where we will also analyze the connection between battery technology and vehicle utilization. We also investigate how balancing the load among vehicles may improve fairness among drivers. Finally, Section 7 concludes the paper.

2 Literature review

The problem of scheduling electric vehicles obviously bears some resemblance to classic vehicle scheduling problems. The problem is mostly considered in the context of bus scheduling and defined as assigning a set of timetabled trips to busses such that the total number of vehicles or the total deadheading time (or a combination thereof) is minimal. Since the seminal paper by Saha (1970), this topic has received substantial attention from the scientific community, with many realistic aspects of bus scheduling integrated into the base model (such as multiple depots and heterogeneous vehicles). Surveys are provided by Bodin and Golden (1981), Bunte and Kliewer (2009).

Since these models almost exclusively focus on public transportation scheduling, they are only of limited use for scheduling electrically powered tow trains in a just-in-time production setting. For one, when scheduling busses, the so-called deadheading time that occurs when the bus moves from the end point of one trip to the start point of the next, plays a major role, making models and solution methods more complicated. Deadheading, however, is a negligible issue in production logistics, where all vehicles start from and return to the depot after each trip anyways. Moreover, there are very few vehicle scheduling models that specifically deal with electric vehicles. While some publications consider time constraints (e.g., Freling and Paixao, 1995, Haghani and Banihashemi, 2002), forcing the busses to return to the depot before they run out of fuel, this is a far cry from actually scheduling recharging intervals, the length of which depends on the current state of the battery level.

To the best of the authors' knowledge, there are only three papers dealing with single-depot electric vehicle scheduling. Li (2014) consider both conventional as well as electrically powered transit busses; for the latter, they assume that the vehicles are recharged via battery swapping or fast charging in constant time (i.e., irrespective of the current charge of the battery). Similarly, Reuer et al. (2015) also consider the case of electric busses which can be charged in constant time. Finally, Chao and Xiaohong (2013) present a case study from Chinese metropolitan areas, where several different aspects of operating a fleet of electric busses is taken into account, like, e.g., the adequate power supply at the recharging stations. None of these papers are suitable for in-plant logistics, however, as the main issue in the scheduling of public busses in light of the limited battery capacity is to ensure that they return to the depot (or a battery swap station) on time before the battery runs dry. This is not an issue for tow trains because they return to the depot after each trip anyways. Unlike busses, on the other hand, tow trains are usually not recharged by battery swapping but by connecting them to a recharging station, making the problem more complicated as recharging times cannot be assumed to be constant.

The problem of assigning trips (jobs) that have to be executed at given fixed times to vehicles (machines) also bears some similarity to interval scheduling. In interval scheduling, not only the processing times of the jobs but also their start times are given. Recent surveys were written by Kolen et al. (2007) and Kovalyov et al. (2007). Interval scheduling, however, is not concerned with scheduling recharging events.

In the machine scheduling context, jobs that take longer the later they are executed (like recharging a battery) are somewhat reminiscent of so-called deteriorating jobs; a survey of related papers is provided by Cheng et al. (2004). However, most models relating to deteriorating jobs proposed so far assume that the deterioration is some function of the scheduled starting time of a job. In our case, on the other hand, the recharging time depends on the current battery level, not necessarily the time. While there have been a few publications in recent years dealing with position-dependent deteriorating jobs (e.g., Yang et al., 2013, Yin et al., 2015), none of these are from the field of vehicle or interval scheduling.

Finally, there are a few papers specifically dealing with the scheduling of tow trains in assembly plants (Emde and Boysen, 2012, Emde and Gendreau, 2015). However, these papers only deal with the super-ordinate problem of drawing up a timetable for the vehicles given a production sequence. They do not explicitly assign vehicles to trips or take into account battery recharging.

3 Problem description

The electric vehicle milk-run scheduling problem (EVMSP) consists of assigning timetabled trips to a fleet of vehicles such that the energy consumption of each trip does not exceed the remaining charge. The energy consumption of a trip may depend on such things as its length, the weight of the cargo to be carried, the terrain (e.g., there might be steep ramps to be traversed on some trips) etc. Each trip starts and ends at the depot. In addition to the given fixed trips, recharging intervals can also be scheduled for each vehicle at the depot, where the amount of charge regained depends on the length of time spent at the recharger at the depot. Vehicles as well as operators can be a significant cost driver for assembly plants (e.g., Golz et al., 2012), therefore trips and recharge events should be assigned such that the total number of used vehicles is minimal. As a secondary objective, plant managers are often concerned about fairness issues: operators perceive a schedule that assigns many lengthy tasks to one vehicle while another one is almost completely idle as unfair. Moreover, such schedules may also be undesirable due to the strain they put on batteries. Unbalanced schedules should therefore be avoided. We investigate this fairness aspect further in our computational study (Section 6).

To model EVMSP concisely, we make the following assumptions.

- EVMSP is an operational problem, with a planning horizon of one day or one shift. Therefore, we assume that all parameters, especially the timetables of the trips as well as their length and energy consumption, are static and deterministic. Excepting unforeseen disturbances, which are hard to model at any rate, this is certainly a realistic assumption in many assembly plants, where the production sequence is fixed several days in advance and the exact demand as well as routes and timetables are determined with some lead time (Emde and Boysen, 2012, Golz et al., 2012).
- The battery recharge rate is linear. This is a slight simplification because in reality, charging times somewhat increase for the last 10-20% of capacity. It is common practice in the literature, however, to abstract from this (e.g., Schneider et al., 2014).
- All trips start and end at the depot, which is also the location of the recharging station, that is, there is no deadheading. At the OEM we visited, electric tow trains are used to ferry parts from a storage area to the assembly line, always departing from and returning to the depot after each trip.
- All vehicles are identical.
- The recharging station has sufficient capacity to service all calling vehicles at all times.
- At the beginning of the planning horizon, all vehicles have fully charged batteries.
- All parameters are integer. This is not a very strong assumption as any real-valued EVMSP instances can be converted to integer with arbitrary precision.

3.1. Formal description

Let $J = \{1, \dots, n\}$ be a set of n trips (transport tasks). With each trip $j \in J$ is associated a given start time s_j (the time when the tow train departs from the depot), a given end time e_j (the time when the tow train returns to the depot), where $e_j > s_j$, and an amount c_j of energy that is consumed when the trip is executed. Without loss of generality, we assume that the trips are sorted in ascending order of their start time s_j, i.e., $j < j'$ if $s_j < s_{j'}$. Furthermore, there is a variable number of vehicles m, each of which has the same limited battery capacity C. The battery of a vehicle can be recharged with a rate of r; i.e., it takes C/r time units to fully recharge an empty battery.

To define a schedule in a concise manner, we make the following observation.

Observation 3.1. *For each vehicle, the set of trips to perform implies its recharging intervals.*

This is because we can assume that the vehicle will recharge in-between any two successive trips because each trip starts and ends at the depot in any case. The vehicle is hence either on a trip, or recharging. Of course, it may not actually be necessary for a vehicle to recharge its battery in any given interval between trips; however, it would be theoretically possible for it to do so.

Consequently, we can define a schedule $S = \{\pi_1, \dots, \pi_m\}$ as a set of m sequences. Each sequence π_i of length $|\pi_i|$ defines the order in which the trips assigned to vehicle *i* are processed. Let $\pi_i(k) \in J$, $\forall k = 1, \dots, |\pi_i|$, be the *k*-th trip executed by vehicle *i*. We say that a schedule *S* is feasible if it satisfies the following conditions.

1. Each trip is executed exactly once, i.e., $\bigcup_{i=1}^{m} \bigcup_{k=1}^{|\pi_i|} \{\pi_i(k)\} = J$ and $\bigcup_{k=1}^{|\pi_i|} \{\pi_i(k)\} \cap \bigcup_{k=1}^{|\pi_{i'}|} \{\pi_{i'}(k)\} = \emptyset, \forall i, i' \in \{1, \dots, m\}, i \neq i'$.
2. No vehicle executes two distinct trips at the same time, $s_{\pi_i(k)} \geq e_{\pi_i(k-1)}, \forall i = 1, \dots, m; k = 2, \dots, |\pi_i|$.
3. The remaining charge of the vehicle executing a trip is sufficient to complete that trip. Let $\bar{C}_i(k)$ be the remaining charge of the battery of vehicle *i* after its *k*-th trip has been executed, which is calculated recursively as
$$\bar{C}_i(k) = \begin{cases} \min\{C; \bar{C}_i(k-1) + r.\left(s_{\pi_i(k)} - e_{\pi_i(k-1)}\right)\} - c_{\pi_i(k)} \text{ if } \mathrm{k} \geq 2 \\ C - c_{\pi_i(1)} \text{if } \mathrm{k} = 2 \end{cases}. \tag{1}$$
Then it must hold that $\bar{C}_i(k) \geq 0, \forall i = 1, \dots, m; k = 1, \dots, |\pi_i|$.

Note that Eq. (1) follows from Observation 3.1. At any time, the remaining charge is given recursively by the charge left after the last trip, plus the recharged energy in-between the two consecutive trips, minus the energy consumed on the current trip.

Among all feasible schedules *S*, we seek one that minimizes the number of vehicles. As a secondary objective, the schedule with the minimum number of vehicles should also be fair, i.e., the difference between the busiest vehicle and the idlest vehicle (as determined by the time they spend on trips) should be minimal. Consequently, our objective function is

$$f(S) = \gamma.m + \max_{i=1,\dots,m}\left\{\sum_{j=1}^{|\pi_i|}(e_{\pi_i(j)} - s_{\pi_i(j)})\right\} - \min_{i=1,\dots,m}\left\{\sum_{j=1}^{|\pi_i|}(e_{\pi_i(j)} - s_{\pi_i(j)})\right\}. \quad (2)$$

To obtain a lexicographic ordering of objectives, weighting factor γ needs to be set to a sufficiently great value, e.g., $\gamma = \sum_{j\in J}(e_j - s_j)$, the total processing time of all jobs.

Example: Consider an example problem with 4 trips. The start and end times as well as the energy consumption of the trips is given in Table 1. Let the battery capacity be $C = 10$ and the recharge rate $r = 1$. Then a feasible and optimal (for a sufficiently great weight γ) solution is $S = \{\langle 1,3]; \langle 2,4]\}$, indicating that one vehicle performs trips 1 and 3 (total time spent on trips: 3), and another vehicle takes care of trips 2 and 4 (total time spent on trips: 5), yielding a minimum number of vehicles of $m = 2$ and a difference between the busiest and the least busy vehicles of $5 - 3 = 2$. Note that although trips 2 and 4 consume a total of $c_2 + c_4 = 12$ units of energy, which is more than the battery capacity of $C = 10$, the solution is nonetheless feasible because the vehicle can recover two units of charge in-between the trips.

j	1	2	3	4
s_j	0	1	4	6
e_j	2	4	5	8
c_j	4	7	6	5

Table 1: An example problem.

4 Computational complexity

Disregarding the fairness objective and limited battery capacity, EVMSP can be solved in polynomial time (Saha, 1970) by constructing an acyclic network where each node represents one trip $j \in J$ and an arc between two nodes (j, j') is inserted if trips j and j' can be performed by the same vehicle in succession, i.e., if $e_j \leq s_{j'}$. The problem then reduces to finding the minimum number of paths through the network, such that each node lies on exactly one path. Each path can be interpreted as a vehicle schedule. The problem can be solved efficiently using a maximum flow or minimum cost flow algorithm (Bodin and Golden, 1981, Bertossi et al., 1987). Hence, we get the following proposition.

Proposition 4.1. *EVMSP with infinite battery capacity* ($C = \infty$) *and without fairness objective, denoted as EVMSP-∞, is in P.*

Of course, it is unrealistic to assume that vehicles never need to be recharged. However, if battery capacities are very generous and rarely constitute a bottleneck, this may be a viable simplification. Moreover, an optimal solution to EVMSP-∞ constitutes a lower bound on the number of vehicles, *m*.

Note that calculating a lower bound does not necessarily require solving a maximum flow problem. It is also possible to get this value by checking for each point in time *t* which jobs are active at that time. The maximum number of jobs active at any time constitutes a lower bound on *m*, where only the times *t* where a job starts or ends need to be considered. This can be done in $O(n^2)$ time. Specifically,

$$m^{LB} = \max_{t \in \{s_1,\dots,s_n,e_1,\dots,e_n\}} \{ |\{j | s_j \leq t < e_j ; j \in J\}| \}. \quad (3)$$

However, even if the battery capacity is relaxed, finding an optimal solution without ignoring fairness is strongly NP-hard as we will show in the following.

Proposition 4.2. *EVMSP with infinite battery capacity (C = 1) is NP-hard in the strong sense.*

Proof. To show that EVMSP-1 with fairness objective is NP-hard, we will present a reduction from 3-PARTITION, which is well known to be strongly NP-hard (Garey and Johnson, 1979).

3-PARTITION: Given $3q$ positive integers a_p, $p = 1, \dots, 3q$, and a positive integer B, where $\sum_{p=1}^{3q} a_p = qB$ and $B/4 < a_p < B/2$, $\forall p = 1, \dots, 3q$, does a partition of the set $\{1,2, \dots, 3q\}$ into q sets $A_1, A_2, \dots, A_q$ exist such that $\sum_{p \in A_i} a_p = B$, $\forall i = 1, \dots, q$?

Consider the following transformation from a 3-PARTITION instance to an EVMSP instance. For each of the $3q$ integers in 3-PARTITION, we introduce one trip j in EVMSP. The trips take place sequentially starting from time 2, such that no two trips overlap, i.e., $s_j = 2 + \sum_{p=1}^{j-1} a_p$, $e_j = 2 + \sum_{p=1}^{j} a_p$, $\forall j = 1, \dots, 3q$. Moreover, we introduce q trips $3q + 1, 3q + 2, \dots, 4q$, each with the same start and end time $s_j = 1$ and $e_j = 2, j = 3q + 1, \dots, 4q$. Let $\gamma = \sum_{j=1}^{4q} (e_j - s_j)$. The question asked is: Is there an EVMSP schedule S which uses exactly q vehicles and is perfectly fair, i.e., with $f(S) \leq q\gamma$?

A solution to 3-PARTITION is also a solution to the corresponding EVMSP instance: All jobs in set $A_i \cup \{3q + i\}$ are executed by one distinct vehicle ($\forall i = 1, \dots, q$). Since the only overlapping jobs $3q + i$ are assigned to diffrent vehicles, the schedule is feasible. Seeing that each vehicle has the exact same total load of $\sum_{p \in A_i} a_p + 1 = \sum_{p \in A_i} (e_p - s_p) + 1 = B + 1$, the solution must be perfectly fair (i.e., all q vehicles are exactly equally busy), leading to an objective value of $f(S) = q\gamma$.

The transformation also works in the opposite direction: A solution to EVMSP with $f(S) \leq q\gamma$ must also be a solution to the corresponding 3-PARTITION instance. Since minimizing the vehicle count is the primary objective, m must be exactly q in the optimal solution because only jobs $3q + i, i = 1, \dots, q$, cannot be executed by the same vehicle. The other jobs will be divided among the vehicles as equally as possible to maximize fairness, which is only possible if each vehicle performs exactly 3 trips corresponding to the 3-PARTITION integers totalling B. The proposition follows. □

Proposition 4.2 obviously implies that the general EVMSP (without relaxed battery capacity constraint but with fairness objective) is also NP-hard in the strong sense. Note that if the battery capacity is limited but the fairness objective is neglected, the problem is still NP-hard in the strong sense, as per the following proposition.

Proposition 4.3. *EVMSP without fairness objective but with given finite battery capacity is NP-hard in the strong sense.*

Proof. We prove this proposition by reduction from BIN PACKING, which is well known to be strongly NP-hard (Garey and Johnson, 1979).

BIN PACKING: Given q positive integers $a_p, p = 1, \ldots, q$, and two positive integers ϵ and B, does a partition of the set $\{1,2,\ldots,q\}$ into ϵ sets $A_1, A_2, \ldots, A_\epsilon$ exist such that $\sum_{p \in A_i} a_p \leq B$, $\forall i = 1, \ldots, \epsilon$?

Consider the following transformation from a BIN PACKING instance to an EVMSP instance. For each of the q integers in BIN PACKING, we introduce one trip j in EVMSP. The trips can have arbitrary start and end times so long as no two trips overlap, i.e., $e_j \leq s_{j'} \vee e_{j'} \leq s_j$ must hold $\forall j, j' \in J, j \neq j'$. The energy consumption of each trip is set to correspond to the integers from BIN PACKING, i.e., $c_j = a_j$, $\forall j = 1, \ldots, q$. The battery capacity of the vehicles is $C = B$, and the recharge rate is very slow, i.e., $r < \left(\max_{j=1,\ldots,q} \{e_j\}\right)^{-1}$. The question asked is: Is there an EVMSP schedule S which uses no more than ϵ vehicles?

Given that the processing intervals of no two trips overlap, trips can be arbitrarily assigned to any vehicle. Since it is impossible to recover any significant charge during the planning horizon due to the slow recharge rate r, the sum of the energy consumption of all trips assigned to one vehicle must not be greater than $C = B$, however. Seeing that the energy consumption of each trip corresponds to the integers in BIN PACKING, the equivalence of an EVMSP solution with no more than ϵ vehicles and a BIN PACKING solution with no more than ϵ bins is hence apparent.

There is, however, one special case that is solvable in pseudo-polynomial time. We will make use of this later to generate solutions heuristically.

Proposition 4.4. *Given an instance of EVMSP, the problem of assigning as many trips as possible to a single vehicle, i.e., maximizing $|\pi_i|$ for one single vehicle i, denoted as EVMSP-1, can be solved in pseudo-polynomial time.*

Proof. Consider an acyclic digraph consisting of nodes (j, b_j), where $j \in J$ and $b_j \in \{0,1,\ldots,C\}$, indicating that a charge of b_j units remains after trip j has been executed. Moreover, consider an arc from node (j, b_j) to $(j', b_{j'})$, indicating that the vehicle executes trip j' immediately after trip j, if

- the times do not overlap, i.e., $e_j \leq s_{j'}$ (note that this implies that $j' > j$ because we assume that the trips are sorted according to s_j),
- the remaining charge after executing trip j' is $b_{j'} = min\{C; b_j + r.(s_{j'} - e_j)\} - c_{j'}$, and
- the charge is actually sufficient to complete trip j', i.e., $b_j + r.\left(s_{j'} - e_j\right) - c_{j'} \geq 0$.

Finally, let $(0, C)$ be a dummy source node, connected to nodes $(j, C - c_j)$, $\forall j \in J$, and $(n+1, 0)$ be a dummy sink node, receiving an inbound arc from every other node that would otherwise have an outdegree of 0. Assigning the maximum number of trips to the vehicle is then equivalent to finding a longest path (that is, a path with the maximum number of edges) from source to sink in this graph. Note that nodes with an indegree of 0 need not be considered and can be deleted from the graph (excepting the source node) as they can obviously not lie on any longest path.

Concerning the asymptotic runtime, the graph consists of no more than $O(nC)$ vertices. Even if each vertex were connected to every other, the total number of edges would be no more than $O(n^2C^2)$. Finding the longest path in an acyclic directed graph can be done in $O(V+E)$ time, where V is the number of vertices and E the number of edges (Lawler, 1976). Hence EVMSP-1 can be solved in $O(n^2C^2)$ time, which completes the proof. □

Example (cont.): Consider the example from Section 3.1. The corresponding EVMSP-1 graph is depicted in Figure 2; one longest path is bold-faced, indicating that trips 1 and 4 get assigned to one vehicle.

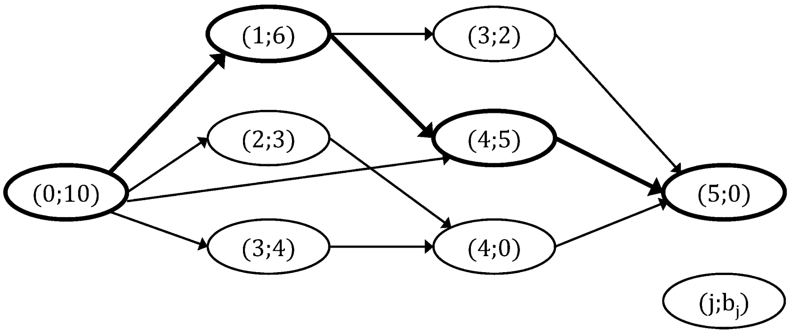


Figure 2: Example graph for EVMSP-1.

5 Algorithms

Given the strongly NP-hard nature of EVMSP, it is unlikely that the problem can be solved to optimality by an exact algorithm for realistic instances in acceptable time (see Section 6 for our computational results). Still, for smaller instances and to have a benchmark, we propose a MIP model that enables the use of a default solver. For larger instances, we propose an opening heuristic as well as a neighborhood search-based metaheuristic, tabu search.

5.1. MIP model

Using the notation summarized in Table 2, we propose the following mixed-integer program.

	sets
J	set of trips (indices $j, j' \in J = \{1, \dots, n\}$)
V	set of vehicles (index $i \in V = \{1, \dots, m\}$))
E_j	set of potential successors of trip j, i.e., $E_j = \{j' \mid j' \in J; s_{j'} \geq e_j\}, \forall j \in J$
	parameters
γ	weighting factor of the number of vehicles in the objective function
M	big integer
C	battery capacity of the vehicles
r	battery recharge rate (units of charge per time unit)

s_j	start time of trip *j*
e_j	end time of trip *j*
c	energy consumption of trip *j*
	decision variables
x_{ij}	binary variable: 1, if trip *j* is assigned to vehicle *i*; 0, otherwise
y_i	binary variable: 1, if vehicle *i* is used; 0, otherwise
b_j	continuous variable: battery charge remaining after trip *j* has been completed
α	continuous variable: total trip time of the busiest vehicle
β	continuous variable: total trip time of the least busy vehicle

Table 2: Notation for the MIP model

$$Minimize\ F(x, b, y, \alpha, \beta) = \gamma . \sum_{i \in V} y_i + \alpha - \beta \tag{4}$$

subject to

$$\sum_{i \in V} x_{ij} = 1 \qquad \forall j \in J \tag{5}$$

$$x_{ij} < y_i \qquad \forall i \in V; j \in J \tag{6}$$

$$\sum_{j \in J} x_{ij} . (e_j - s_j) \leq \alpha \qquad \forall i \in V \tag{7}$$

$$\sum_{j \in J} x_{ij} . (e_j - s_j) + (1 - y_i). M \geq \beta \geq 0 \qquad \forall i \in V \tag{8}$$

$$b_{j'} - b_j + c_{j'} - 2C . (2 - x_{ij} - x_{ij'}) \leq r . (s_{j'} - e_j) \qquad \forall i \in V; j \in J; j' \in E_j \tag{9}$$

$$x_{ij} + x_{ij'} \leq 1 \qquad \forall i \in V; j, j' \in J; j' \notin E_j; j < j' \tag{10}$$

$$0 \leq b_j \leq C - c_j \qquad \forall j \in J \tag{11}$$

$$x_{ij} \in \{0,1\} \qquad \forall i \in V; j \in J \tag{12}$$

$$y_i \in \{0,1\} \qquad \forall i \in V \tag{13}$$

Objective function (4) minimizes the weighted sum of the number of vehicles and the difference between the workload of the busiest (α) and the least busy (β) vehicle. Constraints (5) ensure that each trip is executed exactly once. Inequalities (6) force y_i to 1 if vehicle *i* performs any job. Constraints (7) and (8) set α and β, respectively. Constraints (9) enforce that the battery can only be recharged in-between trips: the charge $b_{j'}$ after finishing trip j' cannot be greater than the charge b_j after completing the previous trip *j*, minus the energy $c_{j'}$ consumed on trip j', plus the energy recharged in-between trips, $r. (s_{j'} - e_j)$. Inequalities (10) make it impossible for two overlapping trips to be assigned to the same vehicle. Finally, (11) make sure that no vehicle runs out of power, and (12) and (13) are the binary constraints.

Note that since the optimal number of vehicles *m* is not known in advance, this value has to be initialized to an upper bound, e.g., trivially, $m := n$ or, alternatively, a value obtained from one of the heuristics described in the following. The big integer M need not be greater than $\sum_{j\in J}(e_j - s_j)$ as β can never be greater than this value at any rate.

5.2. Opening heuristic

To construct a first feasible solution, we propose to use the procedure from Proposition 4.4 to assign trips to one vehicle after another. It is outlined in Algorithm 1. In each iteration, as many of the trips in set $\bar{J}$, which initially equals the set of all trips J, as possible are assigned to vehicle i by solving EVMSP-1. The trips thus assigned are then removed from $\bar{J}$, and the next vehicle is assigned (a subset of) these trips, until all trips have been assigned, i.e., $\bar{\bar{J}} = \emptyset$. Note that, in each iteration, if there is more than one possible EVMSP-1 solution (i.e., more than one longest path through the graph as described in Proposition 4.4), we select one at random.

input: an instance of EVMSP
$\bar{J} \coloneqq J$;
$i \coloneqq 1$;
while $\bar{J} \neq \emptyset$ **do**
  $\pi_i \coloneqq$assignment of a subset of trips in $\bar{J}$ to vehicle i via EVMSP-1;
  $\bar{J} \coloneqq \bar{J} \setminus \bigcup_{k=1}^{|\pi_i|}\{\pi_i(k)\}$;
  $i \coloneqq i + 1$;
return EVMSP *solution* $S = \{\pi_1, \ldots, \pi_{i-1}\}$;

Algorithm 1: Openning heuristic.

5.3. Neighborhood search

While it stands to reason that the opening heuristic will output reasonably good solutions, they can probably be improved by neighborhood search. We consider two different neighborhood moves.

Push A trip is pushed from one vehicle to another, i.e., some trip $j \in J$ is removed from its current vehicle's sequence π_i and inserted into some other vehicles sequence $\pi_{i'}$, $i \neq i'$. To minimize overlap, trips are inserted into the target sequence at position $\underset{k=1,\ldots,|\pi_{i'}|}{\operatorname{argmin}} \left\{s_{\pi_{i'}(k)} > s_j\right\}$. Note that it is also possible that i' is a new vehicle that was not part of the solution before. Analogously, if this move removes the last trip from vehicle i, π_i is removed from the solution.

Swap For two distinct trips, $j \in J$ and $j' \in J$, $j' \neq J$ and from different vehicles, switch vehicles.

To evaluate the "fitness" of a solution S, we could use objective function (2). However, that would create a problem as we would be unable to handle infeasible solutions. Consequently, given a solution S with m vehicles in use, we propose the following generalized cost function to evaluate solutions.

$$g(S) = f(S) + \sum_{i=1}^{m}\left(\rho^T . \sum_{k=1}^{|\pi_i|-1} max\left\{0; e_{\pi_i(k)} - s_{\pi_i(k+1)}\right\} + \rho^C . \sum_{k=1}^{|\pi_i|} max\{0; -\bar{C}_i(k)\}\right), \tag{14}$$

where $\bar{C}_i(k)$ is the charge remaining after executing trip $\pi_i(k)$, as calculated by Eq. (1), and ρ^T (ρ^C) is the penalty factor associated with overlapping trips (exceeded battery capacity). These factors are varied during the course of the search: if the last 5 accepted neighbors were infeasible with regard to overlapping trips (exceeded battery capacity), set $\rho^T := 2.\rho^T$ ($\rho^C := 2.\rho^C$); analogously, if the last 5 accepted neighbors were all feasible, set $\rho^T := \rho^T/2$ ($\rho^C := \rho^C/2$).

Given two feasible solutions, g will be lower for the solution using fewer vehicles (provided is great enough). If the number of vehicles is identical, g will give preference to the one that is fairer.

Inserting a trip into a vehicle's trip sequence takes $O(log\, n)$ time using binary search. Calculating the change in overlap penalty for a neighborhood solution can thus be done in logarithmic time if only the overlap with the immediate predecessor and immediate successor of the inserted job is computed. However, calculating the violation of the battery capacity still takes $O(n)$ time in the worst case, because inserting a job into a sequence may well change the available capacities for all following jobs, too. Finally, the change in the fairness objective can be calculated in constant time if the total workload of each vehicle, i.e., $\eta_i := \sum_{j=1}^{|\pi_i|} \left(e_{\pi_i(j)} - s_{\pi_i(j)}\right), \forall i = 1, \dots, m$, is stored. If some job *j* is added to (removed from) some vehicle *i* then $\eta_i := \eta_i + e_{\pi_i(j)} - s_{\pi_i(j)}$ ($\eta_i := \eta_i - e_{\pi_i(j)} + s_{\pi_i(j)}$). Assuming that all η_i are sorted, it can be checked in constant time if the new η_i is the new busiest or least busy vehicle, and modify the objective value accordingly. Consequently, calculating a neighbor's objective value takes $O(n)$ time in total in the worst case.

We embed this neighborhood search strategy into a tabu search framework (Glover and Laguna, 1997). Tabu search is a powerful metaheuristic that has often been applied successfully to complex scheduling problems with difficult feasibility constraints (e.g., Gendreau et al., 1994, Gendreau and Potvin, 2010). The procedure is outlined in Algorithm 2.

Starting from an initial solution generated by Algorithm 1, the entire neighborhood that can be reached by either a push or swap move as described above is investigated, and the best nontabu solution from that neighborhood becomes the new incumbent solution. After some job *j* has been removed from some vehicle π_i (be it through a push or a swap), it cannot be reassigned to π_i again for 10 iterations, i.e., job *j* is tabu for vehicle *i* for 10 iterations (tabu tenure).

input: an instance of EVMSP
$\Theta^{max} := 500$; // max number of iterations before diversification
$\Theta := 0$; // number of iterations since new best solution has been found
$i := 1$; // iteration counter
$S := S^* :=$ solution obtained via Algorithm 1;
while $i \leq 10000$ **and** $f(S^*) > \gamma.m^{LB}$ **do**
    $S :=$ best non-tabu neighbor of S;
    Update tabu list;
    Update penalty parameters if necessary;
    if S is feasible **and** $f(S) < f(S^*)$ **then**
        $\Theta := 0$;
        $S^* := S$;
    else
        $\Theta := \Theta + 1$;
    if $\Theta \geq \Theta^{max}$ **then**

```
        if first diversification phase then
            S := solution where all jobs are assigned to one single vehicle;
            Θ^max := 500;
        else
            if rnd(0,1) < 0.5 then // rnd returns a uniformly distributed random number in [0,1]
                S := random feasible solution via Algorithm 3;
                Θ^max := 250;
            else
                S := random infeasible solution;
                Θ^max := 350;
        reset tabu list;
        Θ := 0;
    i := 1 + 1;
return best found EVMSP solution S*;
```

Algorithm 2: Tabu search for EVMSP.

```
input: an instance of EVMSP
J̄ := J;
m := 0;
while J̄ ≠ ∅; do
    j := one random job from J̄;
    J̄ := J̄\{j};
    while j is unassigned do
        ρ := rnd(1, … , m + 1);
        if ρ = m + 1 then
            Open a new vehicle and assign j to it as the first job;
            m := m + 1;
        else
            Assign j to vehicle ρ if that is possible in terms of battery capacity and overlap;
return feasible EVMSP solution;
```

Algorithm 3: Generate a random feasible solution.

If the best solution S^* has not been updated in the past Θ^{max} iterations, the search is restarted with a new initial solution for diversification purposes. New solutions are generated in one of three ways:

- In the very first diversification phase, a new initial solution is generated by simply assigning all jobs to one single vehicle. This solution will usually be severely infeasible, therefore Θ^{max} is set to 1000, awarding more time to neighborhood search before the next diversification phase is triggered.
- In all further diversification phases (if any), new initial solutions are generated randomly:
  - ✓ With likelihood 0.5, a random feasible solution is created by employing Algorithm 3.
  - ✓ Finally, for even more “aggressive” diversification, with likelihood 0.5, a completely random, most likely infeasible, solution is generated by setting the number of vehicles to the lower bound (obtained via Eq. (3)) and iteratively assigning each job to one randomly chosen vehicle.

Finally, the optimization ends and the best found solution is returned if either 10000 neighborhoods have been investigated or a feasible solution has been found where the number of vehicles is equal to the lower bound (m^{LB}, Eq. (3)) and the fairness is perfect (i.e., $\max_{i=1,\dots,m}\left\{\sum_{j=1}^{|\pi_i|}\left(e_{\pi_i(j)} - s_{\pi_i(j)}\right)\right\} = \min_{i=1,\dots,m}\left\{\sum_{j=1}^{|\pi_i|}\left(e_{\pi_i(j)} - s_{\pi_i(j)}\right)\right\}$).

6 Computational study

To assess the performance of our proposed heuristics, we implemented them in Java 8 and had them solve test instances on an x64-PC equipped with an Intel Core i5-3210M CPU, clocked at 2.5 GHz, and 8 GB of RAM. As a benchmark, we also had a default solver, namely CPLEX 12.6.3, solve the same instances. Since there are no established instances of EVMSP, we will first describe how we generated our test data. Then, we will test the computational performance, discuss our fairness objective, and, finally, derive some insight into the tradeoff between battery capacity and vehicle utilization.

6.1. Instance generation

We model our test instances based on our observations at our OEM partner. A typical tow train moves at a speed of about 10 to 15 kph. The vehicles have to go from the depot to one of several destinations in the final assembly hall and the warehouse. Depending on the distance between depot and destination as well as the time it takes to load/unload the train (semi-automated in the assembly hall, mostly manual in the warehouse), a typical trip takes somewhere between 10 and 25 minutes of time. Consequently, we set the processing time for each trip j to $p_j = rnd^{int}(10,\dots,25)$, where rnd^{int} is a uniformly distributed random integer from the range in the argument.

As the tow trains have to supply the whole assembly plant including dozens of workstations, the tow trains are almost constantly in action, i.e., it is rare that more than a very few minutes pass inbetween consecutive trips. Therefore, we generate the start times of trips as follows. The first trip starts at time $s_1 = 1$. The following trips $j = 2,\dots,n$ start at time $s_j = s_{j-1} + rnd^{int}(1,\dots,3)$. It follows that $e_j = s_j + p, \forall j \in J$.

A typical tow train as it is used at our OEM partner has a battery capacity of about $C = 150$ Ah, of which each trip drains between 25 and 60%, depending mainly on the characteristics of the trip (e.g., a trip with many stops drains the battery faster than steady movement, slopes and uneven floors are harder on the engine than level shopfloors etc.). Hence, we set the battery consumption to $c_j = \lfloor rnd^{cont}(0.25,0.6).C \rceil$, where rnd^{cont} denotes a randomly generated real number from the interval in the argument (continuous uniform distribution), and $\lfloor . \rceil$ denotes rounding to the next integer.

The recharge rate depends somewhat on the technical attributes of the tow trains in question. We therefore generate two different data sets, one with relatively slow-charging batteries ($r = 1$) and one with relatively fast-charging batteries ($r = 5$). Also, for each recharge rate, we generate small and large instances. Large instances require $n = 150$ trips to be scheduled, small instances consist of either $n = 15$ (for $r = 1$) or $n = 25$ (for $r = 5$) trips. Note that the size of the small instances was chosen such that a default solver can still solve them in acceptable time, while the size of the large instances reflects what we observed at our OEM partner. For each tested parameter constellation, we generate 20 instances, leading to a total of $2 * 2 * 20 = 80$. The instances are available from the authors upon request.

6.2. Algorithmic performance

Tables 3 and 4 show the results for the 20 small instances with $n = 15$ and $n = 25$ jobs, respectively. The tables list the lower bound (LB) on the vehicle count as per Eq. (3), as well as the vehicle count (labeled m in the table), the fairness objective value (labeled f^{fair}), and the CPU times in seconds for CPLEX, the opening heuristic from Algorithm 1, and our tabu search scheme. CPU times were omitted for the opening heuristic because it could solve all instances (including the large ones) in negligible time (less than 0.1 seconds). f^{fair} is calculated as per Eq. (2) with $\gamma \coloneqq 0$. In all tables, proven optimal results by the heuristics are in boldface.

The data indicate that our proposed tabu search (TS) scheme is quite successful at solving EVMSP. For the instances where $r = 1$ (Table 3), the optimality gap of TS is 0 without exception, both with regard to the vehicle count and the fairness objective. For the instances where $r = 5$ (Table 4), there are two instances (nos. 11 and 16) where TS returned a solution that, while optimal with regard to *m*, is less fair (by one minute) than the optimum. Otherwise, the optimality gap is 0. The average runtime of TS over all small instances is less than one second, which is in stark contrast to the almost 9 minutes that CPLEX took on average to solve the same instances.

The opening heuristic (OH) also delivers a respectable solution quality, as long as only the fleet size *m* is taken into consideration. In about half of the small instances (17 out of 40), OH found a solution that requires only the minimum number of vehicles. In the other cases, the vehicle count is 1 or 2 above optimal. The real drawback of OH, however, becomes apparent when taking the fairness objective into consideration. OH is not designed to create fair solutions, and it shows. The workload is very unevenly distributed in all cases, the busiest vehicle often doing more than 90 minutes of additional work over the least busy vehicle. On the upside, OH is very quick, generating feasible schedules for a small vehicle fleet in negligible time.

	LB	CPLEX			opening h.		tabu search		
no.	m	m	f^{fair}	CPU sec.	m	f^{fair}	m	f^{fair}	CPU sec.
1	5	6	24	2.86	7	47	**6**	**24**	1.11
2	5	5	10	1.36	**5**	12	**5**	**10**	0.49
3	4	6	8	43.91	8	31	**6**	**8**	0.47
4	4	7	12	1.01	8	35	**7**	**12**	0.52
5	4	6	9	3.87	7	28	**6**	**9**	0.40
6	3	5	9	24.30	**5**	53	**5**	**9**	0.33
7	6	6	5	0.80	8	39	**6**	**5**	0.43
8	5	5	8	0.85	**5**	62	**5**	**8**	0.41
9	4	5	0	1.80	**5**	62	**5**	**0**	0.33
10	4	6	3	12.35	**6**	33	**6**	**3**	0.33
11	6	6	18	7.36	7	43	**6**	**18**	0.47
12	4	5	15	1.25	7	39	**5**	**15**	0.46
13	3	6	5	37.00	7	35	**6**	**5**	0.30
14	4	5	3	4.35	7	35	**5**	**3**	0.39
15	3	5	2	33.95	**5**	42	**5**	**2**	0.26
16	4	6	3	22.79	**6**	39	**6**	**3**	0.31
17	4	5	19	1.03	6	40	**5**	**19**	0.38
18	5	6	5	2.61	**6**	33	**6**	**5**	0.42
19	5	6	3	1.72	7	44	**6**	**3**	0.41
20	4	6	4	34.68	7	56	**6**	**4**	0.34
avg.	4.30	5.65	8.25	11.99	6.45	40.40	5.65	8.25	0.43

Table 3: Results for the small instances with slow recharging (n = 15, r = 1).

	LB	CPLEX			opening h.		tabu search		
no.	m	m	f^{fair}	CPU sec.	m	f^{fair}	m	f^{fair}	CPU sec.
1	4	4	1	6.98	5	105	**4**	**1**	1.07
2	6	6	1	14.42	7	111	**6**	**1**	1.43
3	5	5	1	5.28	**5**	115	**5**	**1**	1.29
4	6	6	1	2.57	**6**	102	**6**	**1**	1.38
5	5	5	1	4.05	6	115	**5**	**1**	1.28
6	5	5	1	17.30	**5**	97	**5**	**1**	1.39
7	7	7	1	2.81	**7**	69	**7**	**1**	1.48
8	4	4	0	2.01	5	139	**4**	**0**	0.01
9	7	7	7	770.59	8	95	**7**	**7**	1.46
10	5	5	1	2.47	6	120	**5**	**1**	1.27
11	4	4	1	5.28	5	120	**4**	2	1.11
12	6	6	1	3.75	7	83	**6**	**1**	1.43
13	5	5	1	5.12	**5**	110	**5**	**1**	1.26
14	4	5	0	28.05	6	96	**5**	**0**	1.18
15	5	5	0	3.37	6	105	**5**	**0**	0.07
16	5	5	1	44.19	6	92	**5**	2	1.27
17	5	5	0	3.15	**5**	78	**5**	**0**	0.02
18	9	9	8	19369.38	**9**	91	**9**	**8**	1.60
19	5	5	1	4.06	**5**	95	**5**	**1**	1.26
20	5	5	1	2.87	**5**	82	**5**	**1**	1.36
avg.	5.35	5.40	1.45	1014.88	5.95	101.00	5.40	1.55	1.13

Table 4: Results for the small instances with slow recharging (n = 25, r = 5).

The final piece of information that can be inferred from the small instances is about the quality of the lower bound (LB). LB relaxes the battery capacity limitation. We can therefore expect LB to be tighter whenever the battery is not a very strong limiting factor to begin with. Our results bear this out, as LB is a lot tighter when $r = 5$ as opposed to $r = 1$. In other words, when the battery can be recharged quickly, LB is a good indicator of the minimum feasible vehicle fleet size, seeing that, for $r = 5$, only in one single instance was there a gap between LB and actual optimal vehicle count as reported by CPLEX.

	LB	opening h.		tabu search		
no.	m	m	f^{fair}	m	f^{fair}	CPU sec.
1	6	12	322	11	1	103.493
2	6	13	306	13	2	102.868
3	6	15	300	13	2	102.575
4	7	12	292	12	1	98.89
5	6	14	294	13	2	99.717
6	7	16	282	15	1	98.73
7	6	11	380	10	1	99.293
8	7	12	327	11	1	98.945
9	7	13	300	12	1	96.976
10	7	13	328	12	1	97.829
11	7	12	328	11	1	112.58
12	8	12	274	12	2	112.38
13	6	12	384	11	1	115.869
14	6	12	316	11	**0**	109.514
15	6	14	319	13	1	99.157
16	7	13	287	13	1	90.851
17	6	12	333	11	1	93.355
18	7	13	290	12	1	91.633
19	8	14	269	14	1	88.988
20	6	11	287	11	1	93.083
avg.	6.60	12.80	310.90	12.05	1.15	100.34

Table 5: Results for the large instances with slow recharging (n=150, r=1).

Tables 5 and 6 contain the results for the large instances ($n = 150$ trips). Note that CPLEX is incapable of solving these instances; even after 48 hours of runtime, no even remotely optimal solution was found. We can, however, use LB to assess the quality of our heuristics, at least in the case of $r = 5$ (Table 6). The results essentially corroborate our findings from the small instances: OH is very quick (negligible CPU times even when $n = 150$) and finds passable solutions with regard to fleet size. Fairness, however, is very far from optimal −in all cases the busiest vehicle's workload is several hours over the least busy one's. TS, on the other hand, matches LB for $r = 5$ in all but two instances, while also balancing the workload almost perfectly. We can hence draw the conclusion that TS is indeed capable of solving EVMSP adequately for realistic problem sizes in less than two minutes of CPU time, which should be acceptable for most practical applications.

	LB	opening h.		tabu search		
no.	m	m	f^{fair}	m	f^{fair}	CPU sec.
1	6	**6**	600	**6**	**0**	0.847
2	8	**8**	628	**8**	1	83.1
3	6	7	647	7	1	95.892
4	6	7	696	**6**	1	95.311
5	6	7	673	**6**	1	86.487
6	6	8	622	**6**	1	94.647
7	9	**9**	653	**9**	1	81.66
8	6	7	645	**6**	1	94.179
9	9	**9**	644	**9**	1	81.229
10	8	8	644	**8**	1	91.645
11	7	8	674	**7**	1	93.618
12	6	7	684	**6**	1	94.807
13	7	**7**	665	**7**	1	93.298
14	6	7	665	**6**	1	93.946
15	6	8	585	8	1	96.149
16	5	6	633	**5**	1	98.37
17	7	9	608	**7**	1	94.638
18	6	7	673	**6**	1	95.142
19	8	**8**	701	**8**	1	83.673
20	6	7	675	**6**	1	96.022
avg.	6.70	7.50	650.75	6.85	0.95	87.23

Table 6: Results for the large instances with slow recharging (n=150, r=5).

6.3. Fairness considerations

Objective function (2) seeks to minimize the number of vehicles and to balance the workload among vehicles. As discussed in Section 1, a balanced schedule may help to improve the expected lifetime of batteries because many batteries age more quickly if they are under heavy use while at a low state of charge. Moreover, it stands to reason that such balanced schedules also improve the perceived fairness for the drivers. This latter aspect is not entirely obvious, however. Drivers need not necessarily be assigned to one specific vehicle for the whole day (or shift) but may switch with other drivers at the depot. Therefore, it may be possible to arrive at a fair schedule (from the drivers' perspective) even if the vehicle schedule is anything but balanced.

To test if it is indeed easy to generate a fair driver assignment, we implement the following simple rule-of-thumb that is uncomplicated to apply in practice: Let all drivers queue up at the depot. Whoever is at the head of the queue takes the next trip. After returning from a trip, the driver joins the queue again at the end. We call this priority rule *first-come-first-to-operate*.

We implement this rule in our computational study as follows.

1. Optimize vehicle schedule via tabu search.
2. Set the number of available drivers to the minimal number of vehicles returned by tabu search.
3. Assign drivers to trips one after another using the *first-come-first-to-operate* rule.

Given this, it is easy to calculate the difference in workload between the busiest and the least busy driver. Table 7 compares the vehicle-based fairness (*vehicle fairness* in the table) as obtained by tabu search using objective function (2) to the driver-based fairness (*driver fairness* in the table) as obtained by using the *first-come-first-to-operate* rule. Clearly, schedules generated by tabu search are significantly more balanced, not only in terms of vehicle workload but also in terms of driver fairness. Indeed, a fairness value of only 1.15 ($r = 1$) and 0.95 ($r = 5$) on average is hard to beat for any driver assignment rule, not just *first-come-first-to-operate*.

	r=1		r=5	
no.	driver fairness	vehicle fairness	driver fairness	vehicle fairness
1	40	1	61.00	0.00
2	45	2	90.00	1.00
3	69	2	72.00	1.00
4	65	1	66.00	1.00
5	64	2	42.00	1.00
6	44	1	58.00	1.00
7	67	1	45.00	1.00
8	60	1	39.00	1.00
9	59	1	65.00	1.00
10	93	1	74.00	1.00
11	79	1	65.00	1.00
12	66	2	101.00	1.00
13	39	1	58.00	1.00
14	86	0	82.00	1.00
15	49	1	82.00	1.00
16	51	1	50.00	1.00
17	47	1	52.00	1.00
18	75	1	70.00	1.00
19	59	1	80.00	1.00
20	50	1	64.00	1.00
avg.	60.35	1.15	65.80	0.95

Table 7: Comparison of fairness as obtained by tabu search and fairness as obtained by the *first-come-first-to-operate* rule (*n*=150).

Given that EVMSP is strongly NP-hard even without the fairness objective (Proposition 4.3), there is little computational benefit in not integrating fairness into the objective during the vehicle scheduling phase. In light of the additional benefits of potentially improved battery life expectancy and easier communication and coordination if drivers do not need to switch vehicles frequently, it may be expedient to simply use the best tabu search schedule and have a fixed one-to-one assignment of drivers to vehicles.

To gain a better intuitive understanding about what such an optimized vehicle schedule looks like, Figure 3 may be helpful. It shows the number of vehicles at the depot for each minute during the planning horizon for the schedule generated by tabu search for instance no. 1 ($n = 150$, $r = 5$).

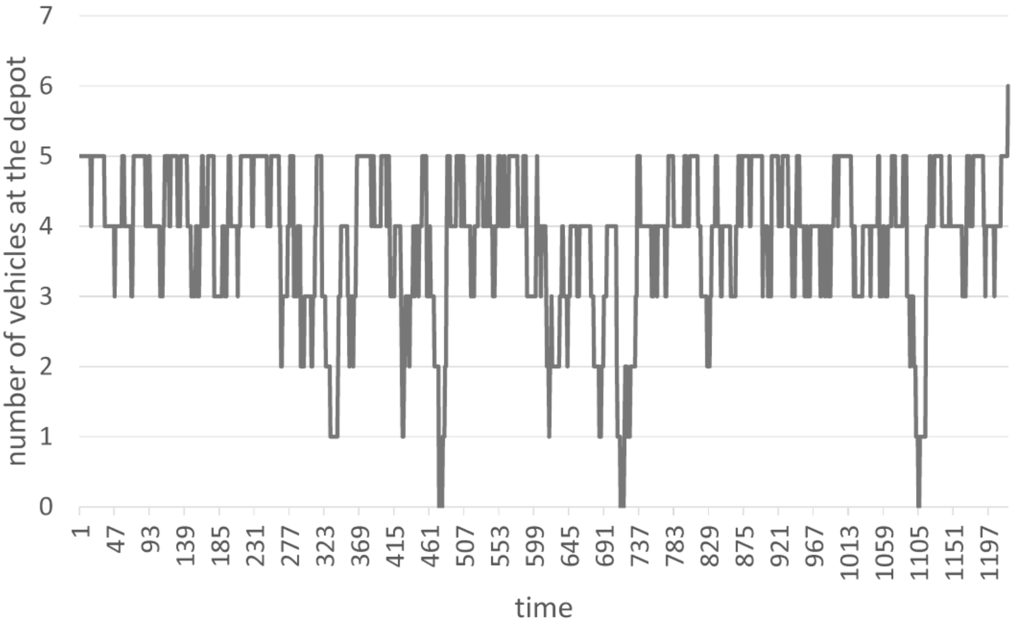


Figure 3: Number of vehicles at the depot as a function of time for large instance no. 1 (n=150, r=5).

6.4. Effect of battery capacity

In the second part of our computational study, we investigate the effect of the battery capacity on the performance of an electric vehicle-based part feeding system. We expect the following tradeoff: If high-capacity batteries are used and recharge times are short, vehicles spend less time recharging and more time actually doing productive work. Hence, the same workload can probably be handled by a smaller vehicle fleet, which is not only advantageous with regard to investment cost but also alleviates shopfloor congestion. On the other hand, batteries are one of the most expensive parts of an electric tugger and also drive cost by adding weight and insurance premium (e.g., Delucchi and Lipman, 2001).

To investigate this tradeoff, we reuse the large instances ($n = 150$ trips) as described in Section 6.1, but instead of just testing the default battery capacity of $C = 150$ Ah, we vary this capacity between 90 and 270, such that $C \in \{90, 120, 150, 180, 210, 240, 270\}$. For each value of C, we minimize the number of vehicles via TS and then average this value over the 20 large instances.

Figure 4 shows the average size of the vehicle fleet depending on the battery capacity. First off, the plots clearly indicate that battery size and/or recharging technology have a very significant effect on vehicle utilization. In the worst case, when batteries are small ($C = 90$ Ah) and recharging is slow ($r = 1$), the average number of required vehicles is more than double that of the best case scenario ($C = 270$, $r = 5$). Shopfloor managers are therefore well advised to consider battery issues when purchasing and installing electric supply vehicles.

As expected, vehicle utilization is better when batteries are large. However, this effect is particularly pronounced when recharge times are slow (Figure 4a). In this case, there is almost a linear decrease of fleet size with rising battery capacity. If batteries can be recharged quickly (Figure 4b), on the other hand, the result is different. Only in case the capacity is very tight ($C \leq 150$) do upgrades make sense. Increasing the size of the battery past a certain point yields basically zero marginal utility. This is because with a fully charged capacious battery, the “naturally occurring” charging breaks in-between regularly scheduled trips are apparently enough to last to the end of the day. Where exactly this point lies is dependent on the recharge rate, but the figures clearly suggest that making large investments to obtain vehicles with excessively capacious batteries may not be advisable.

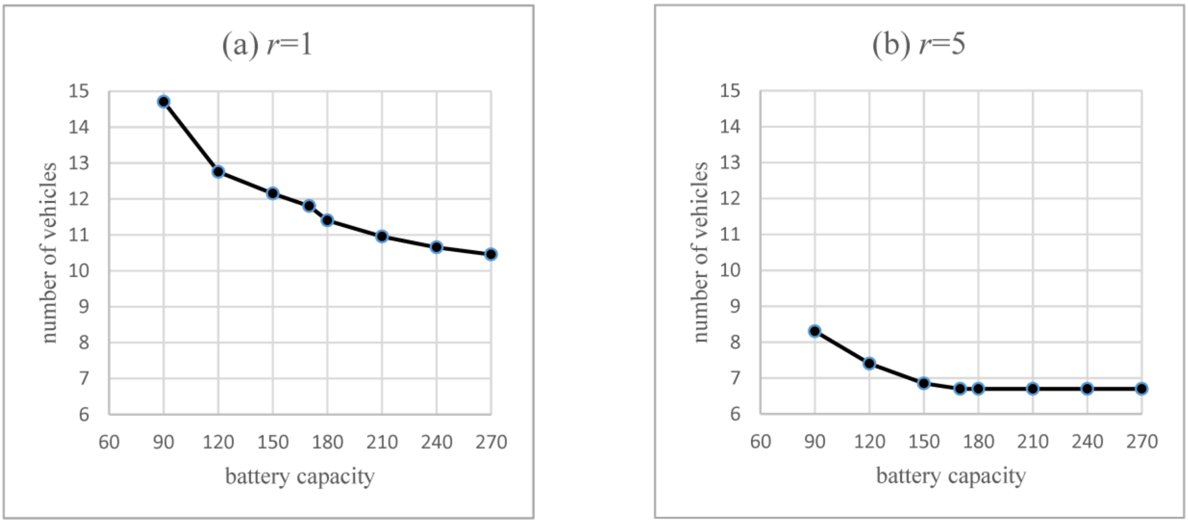


Figure 4: Fleet size vs. battery capacity (n=150).

7 Conclusion

In this paper, we modeled the problem of assigning a set of timetabled milk-run trips to a fleet of electric vehicles such that battery capacities are not exceeded, the fleet size is minimal and fairness is maximal. We analyzed the computational complexity of this problem and some important subproblems, proving that even if the battery restriction is relaxed, the problem remains NP-hard in the strong sense. We also presented two heuristic solution procedures. The main results of this work are as follows.

- The proposed opening heuristic, making use of a subproblem that is solvable in pseudopolynomial time, is very fast, solving even large instances in split seconds. However, the solution quality, especially with regard to fairness, is suboptimal.
- The tabu search scheme proposed in this paper, on the other hand, is capable of solving realistic instances to near-optimality in less than two minutes.
- The battery technology (both capacity as well as recharge rate) has a substantial impact on tow train utilization. Our tests revealed that, in the worst case, the vehicle fleet is more than double the size as compared to the case where capacious, fast-charging batteries are used. However,

improving batteries past a certain point, the exact location of which depends on the recharge rate, is essentially pointless, suggesting that resources might be better spent elsewhere.

Future research should focus on including nonlinear recharging rates and alternative charging technologies such as battery swapping. Moreover, considering robustness objectives when scheduling trips may also be a valuable avenue of research as timetables may occasionally have to be changed on short notice in industrial practice. Another extension of EVMSP could be the case of multiple depots / charging stations. In such scenarios, the vehicles, once they have completed a trip, need not necessarily return to the same depot from which they set off. Finally, the vehicle scheduling decision may also be integrated into a more holistic planning approach that encompasses the timetabling of trips, driver scheduling, as well as the assignment of trips to vehicles.

References

Akillioglu, A.H.; Baydogan, M.G.; Bolath, Y.; Canbaz, D.; Halci, A.; Sezgin, Ö.; Sezgin, Ö.; Özdemirel, N.E.; Türkcan, A. (2006): Pull-based milk-run distribution system design for a firm producing diesel injectors. Journal of Industrial Engineering 17, 2-15.

Battini, D.; Boysen, N.; Emde, S. (2013): Just-in-Time supermarkets for part supply in the automobile industry. Journal of Management Control 24, 209-217.

Bertossi, A.A.; Carraresi, P.; Gallo, G. (1987): On some matching problems arising in vehicle scheduling models. Networks 17, 271-281.

Bodin, L.; Golden, B. (1981): Classification in vehicle routing and scheduling. Networks 11, 97-108.

Bunte, S.; Kliewer, N. (2009): An overview on vehicle scheduling models. Public Transport 1, 299-317.

Chao, Z.; Xiaohong, C. (2013): Optimizing battery electric bus transit vehicle scheduling with battery exchanging: Model and case study. Procedia-Social and Behavioral Sciences 96, 2725-2736.

Cheng, T.E.; Ding, Q.; Lin, B.M. (2004): A concise survey of scheduling with time-dependent processing times. European Journal of Operational Research 152, 1-13.

Delucchi, M.A.; Lipman, T.E. (2001): An analysis of the retail and lifecycle cost of batterypowered electric vehicles. Transportation Research Part D: Transport and Environment 6, 371-404.

Dufo-López, R.; Lujano-Rojas, J.M.; Bernal-Agustín, J.L. (2014): Comparison of different lead-acid battery lifetime prediction models for use in simulation of stand-alone photovoltaic systems. Applied Energy 115, 242-253.

Emde, S.; Fliedner, M.; Boysen, N. (2012): Optimally loading tow trains for JIT-supply of mixed-model assembly lines. IIE Transactions 44, 121-135.

Emde, S.; Boysen, N. (2012): Optimally routing and scheduling tow trains for JIT-supply of mixed-model assembly lines. European Journal of Operational Research 217, 287-299.

Emde, S.; Gendreau, M. (2015): Scheduling In-House Transport Vehicles to Feed Parts to Automotive Assembly Lines. Working paper CIRRELT-2015-50, University of Montreal, QC, Canada.

Faccio, M.; Gamberi, M.; Persona, A.; Regattieri, A.; Sgarbossa, F. (2013): Design and simulation of assembly line feeding systems in the automotive sector using supermarket, kanbans and tow trains: a general framework. Journal of Management Control 24, 187-208.

Freling, R.; Paixao, J.M.P. (1995): Vehicle scheduling with time constraint. In: Computer-Aided Transit Scheduling (pp. 130-144). Springer, Berlin, Germany.

Garey, M. R.; Johnson, D. S. (1979): Computers and intractability: A guide to the theory of NP-completeness. New York, Freeman.

Gendreau, M.; Hertz, A.; Laporte, G. (1994): A tabu search heuristic for the vehicle routing problem. Management Science 40, 1276-1290.

Gendreau, M.; Potvin, J.Y. (2010): Tabu Search. In: Gendreau, M.; Potvin, J.Y. (Eds.): Handbook of Metaheuristics. International Series in Operations Research and Management Science 146. Springer, Berlin, Germany.

Glover, F.; Laguna, M. (1997): Tabu Search. Kluwer, Boston, MA. Golz, J.; Gujjula, R.; Gunther, H.-O.; Rinderer, S.; Ziegler, M. (2012): Part feeding at highvariant mixed-model assembly lines. Flexible Services and Manufacturing Journal 24, 119-141.

Haghani, A.; Banihashemi, M. (2002): Heuristic approaches for solving large-scale bus transit vehicle scheduling problem with route time constraints. Transportation Research Part A: Policy and Practice 36, 309-333.

Kolen, A.W.; Lenstra, J.K.; Papadimitriou, C.H.; Spieksma, F.C. (2007): Interval scheduling: A survey. Naval Research Logistics 54, 530-543.

Kovalyov, M.Y., Ng, C.T., Cheng, T.E. (2007): Fixed interval scheduling: Models, applications, computational complexity and algorithms. European Journal of Operational Research 178, 331-342.

Lawler, E.L. (1976): Combinatorial Optimization: Networks and Matroids. Holt, Rinehart and Winston, New York.

Lee, T.R.; Ueng, J.H. (1999): A study of vehicle routing problems with load-balancing. International Journal of Physical Distribution & Logistics Management 29, 646-657.

Li, J.Q. (2014): Transit bus scheduling with limited energy. Transportation Science 48, 521-539.

Reuer, J.; Kliewer, N.; Wolbeck, L. (2015): The Electric Vehicle Scheduling Problem A study on time-space network based and heuristic solution. In: Proceedings of the Conference on Advanced Systems in Public Transport 2015, Rotterdam, NL.

Saha, J.L. (1970): An algorithm for bus scheduling problems. Operational Research Quarterly 21, 463-474.

Sauer, D.U.; Wenzl, H. (2008): Comparison of different approaches for lifetime prediction of electrochemical systems−Using lead-acid batteries as example. Journal of Power sources 176, 534-546.

Schneider, M.; Stenger, A.; Goeke, D. (2014): The electric vehicle-routing problem with time windows and recharging stations. Transportation Science 48, 500-520.

Vaidyanathan, B.S.; Matson, J.O.; Miller, D.M.; Matson, J.E. (1999): A capacitated vehicle routing problem for just-in-time delivery. IIE Transactions 31, 1083-1092.

Yang, S.J.; Lee, H.T.; Guo, J.Y. (2013): Multiple common due dates assignment and scheduling problems with resource allocation and general position-dependent deterioration effect. The International Journal of Advanced Manufacturing Technology 67, 181-188.

Yin, Y.; Wu, W.H.; Cheng, T.C.E.; Wu, C.C. (2015): Single-machine scheduling with timedependent and position-dependent deteriorating jobs. International Journal of Computer Integrated Manufacturing 28, 781-790.

Paper 5 Scheduling personnel for the build-up of unit load devices at an air cargo terminal with limited space

Authors: Simon Emde, Hamid Abedinnia, Anne Lange, Christoph Glock

Type of publication: Journal article

Publication details: *OR-spectrum*: Under reveiw

Abstract

This paper addresses the preparation of unit load devices (ULD) at an air cargo terminal. This process is difficult to plan for many airlines, which face the challenge of assigning a limited number of workers to a limited number of workspaces available for preparing the ULD, while respecting the requirements imposed by an existing flight schedule. While preparing ULD, the objectives are to comply with the flight schedule, not to exceed the available space at the terminal, and to minimize the maximum workforce employed over time. To support airlines in realizing efficient ULD preparation processes, this paper proposes a mixed-integer programming model as well as a generalized set partitioning reformulation of this problem. Based on the latter formulation, we develop different heuristic strategies, some of which are shown to solve this NP-hard problem to near-optimality in a matter of merely 10 seconds, decisively outperforming the simple rule of thumb frequently used in practice. We also investigate the inherent tradeoff between labor and space utilization as well as the effect of uncertainty about the volume of the cargo to be shipped.

Keywords: *scheduling; air cargo terminal; workforce scheduling; generalized set partitioning; unit load device*

1. Introduction

Air cargo plays a crucial role in modern supply chains. It allows connecting worldwide dispersed supply and demand for products in very short time. As such, air cargo is particularly relevant for time-critical, high-value, and perishable goods. IATA [12] estimates that in 2014, more than 35% of the global trade by value has been transported via air. Air cargo capacity is provided by integrated and non-integrated air carriers. Integrated air cargo providers organize the entire transport, including the air carriage, door-to-door for their customers. UPS, TNT, FedEx and DHL are well-known integrated air cargo providers. The majority of providers are, however, non-integrated. The core function of non-integrated providers are the air carriage and closely related processes. In particular, one differentiates all-cargo airlines from combination carriers. All-cargo airlines operate a fleet of freighter aircraft to transport air cargo. Combination carriers, in contrast, combine passenger and cargo business and make use of the belly-hold capacities of their passenger aircraft for roughly around 50% of the required capacity. Furthermore, some freighter aircraft are operated to provide additional capacity and fill capacity voids in the passenger transportation networks [9, 13].

Air carriers operate air cargo terminals at their major hubs. Operations at the terminals include the receiving of shipments, checking of accompanying documentation and temporary storage. Prior to departure, most shipments are consolidated onto unit load devices (ULD), which may be air cargo pallets or air cargo containers. This allows airfreight to be bundled, making it easier for ground crews to handle and secure cargo onboard the plane. Once the ULD have been built up, they can be transported to the apron for aircraft loading. Upon arrival at the destination airport, the ULD are de-consolidated and handed over to customers.

The focus of our work is on the build-up of ULD. In this context, we tackle the following problem: Given a set of outbound flights for which ULD need to be built up, a warehouse with a limited number of workspaces on which ULD may be placed and a limited workforce which may be flexibly deployed to work on different flights, when should the ULD for each flight be processed such that the available space and workforce is not exceeded at any time and the maximum employed workforce is minimal?

We encountered this problem at the cargo terminal of a major German airfreight carrier. Currently, the shipment consolidation starts at a specific time before the aircraft is scheduled to depart. Thus, ULD build-up follows roughly similar peak patterns to aircraft departures, leading to very unevenly distributed work schedules, where a large number of workers is required at peak times, who are then idle during off-peak hours. Our industry partner is therefore interested in improving schedules such that the workforce requirements are evened out over the day, reducing idle time and hence labor cost.

Workers position ULD on specific workspaces in the terminal and retrieve the shipments to be loaded on the aircraft from a short-term buffer storage facility. There are several ULD per aircraft, depending on its capacity. Workers work on preparing all ULD for that aircraft before moving to the next task (i.e., the next aircraft). Working on several ULD in parallel provides the workers with some degree of freedom as to the order of stacking the shipments, which is important because the exact dimensions of the shipments to be loaded are often not known in advance. Hence, workers tend to open many ULD workspaces in parallel. At the same time, space in the warehouse is limited and the more ULD workspaces are blocked, the fewer aircraft may be handled concurrently. Consequently, choosing when to build up the cargo for what flight is critical for smooth operations. The entire flow of cargo through an airport is schematically depicted in Figure 1.

The main contributions of this paper are as follows. First, we model the novel operational problem of scheduling the build-up of ULD for a specific set of outbound flights as a special type of multi-mode resource-constrained project scheduling problem. Second, we propose a heuristic procedure based on a reformulation of the problem as a generalized set partitioning model, which is shown to perform very well on realistic instances and clearly outperforms simple rules of thumb that we encountered in practice. Third, we derive some managerial insight into the tradeoff between space (i.e., number of workspaces) and size of the workforce as well as the effect of uncertainty about the exact composition of the outbound shipments.

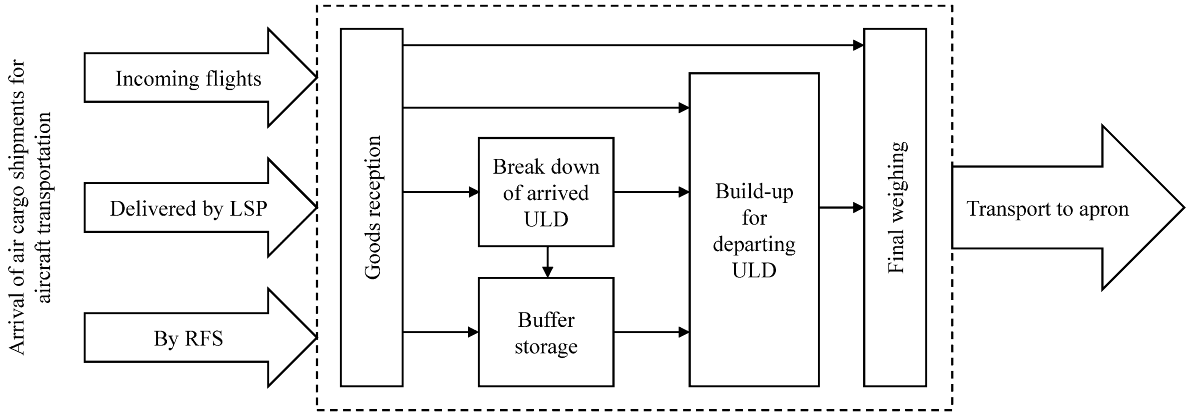


Figure 1: Flow of cargo through an airport.

The remainder of this paper is structured as follows. In Section 2, we review the pertinent literature. In Section 3, we formally define the problem of scheduling the build-up of ULD and propose a mixed-integer linear program. We reformulate the model in Section 4 and propose methods to solve it heuristically, which are tested in a computational study in Section 5. We also compare our optimized schedules to the current status-quo rule-of-thumb we observed at our industry partner and derive some managerial insight into the effect of uncertainty about the exact number of ULD required for some flights and the tradeoff between size of the workforce and space in the warehouse. Finally, Section 6 concludes the paper with an outlook on future research opportunities.

2. Literature review

Regarding air cargo operations in general, Yeung and He [30] review some planning problems with regard to specific applications to the air cargo industry. Feng et al. [9] provide a broader review of the literature. The authors distinguish mainly between problems from the perspective of the airline, the freight forwarder, and the service supply chain. They identify the most important problems from the airlines' perspective – which is also the perspective of this paper – as revenue management [recent contributions, e.g., by 11, 24], fleet routing and flight scheduling [e.g., 1, 15], aircraft loading [e.g., 28, 27], and terminal operations, which we will review in more detail below. By contrasting current business practices and existing literature, the authors identify literature gaps. For the aspect of air cargo terminal operations, the authors highlight the necessity of more integrated models as the decisions taken (such as personnel scheduling and cargo handling process) are tightly interconnected.

Scheduling personnel for break-down and build-up of cargo is a common problem at many airports, and as such it is not surprising that it has received some attention in the literature. A general survey on personnel scheduling problems can be found in Van den Bergh et al. [25]. Recent publications dealing specifically with such scheduling problems in the context of cargo handling at an airport are those of Nobert and Roy [19], Yan et al. [29], and Rong and Grunow [22]. All of these studies, however, take a somewhat aggregate view on the problem, deciding on how many workers should work in what shift. They do not assign workers to individual flights. Consequently, they also do not consider space constraints regarding ULD workspaces which might make it impossible to build up certain flights in parallel.

To include both time and space constraints, the problem can be seen as a kind of resource-constrained project scheduling problem (RCPSP): flights correspond to activities, which take a certain processing time and consume a certain amount of two renewable resources, namely space for ULD on the one hand and workers on the other. Surveys on RCPSP are provided by Brucker et al. [6] and Hartmann and Briskorn [10]. In our problem, it is possible to assign multiple workers to one job to speed up processing, which roughly corresponds to "multi-mode" processing in RCPSP, which is surveyed by Weglarz et al. [26]. Typically, RCPSP aims to minimize the makespan of the schedule, whereas we seek to minimize labor demand fluctuations, which corresponds to "resource leveling" in RCPSP parlance. Publications dealing with the resource leveling problem are comparatively rare. Bandelloni et al. [2], Nudtasomboon and Randhawa [20], and Neumann and Zimmermann [18] are among the few exceptions, none of which deal with multi-mode scheduling, however. Moreover, RCPSP models consider precedence relations between activities, which are immaterial for personnel scheduling. Instead, jobs are constrained by time windows, which is not a feature of classic RCPSP.

Jobs whose processing can be sped up by assigning additional resources to them are also the subject of machine scheduling with malleable tasks [e.g., 16, 3]. Machine scheduling models are also not immediately applicable to our problem, however, as they do not consider the limited space for ULD.

Finally, the problem also bears some resemblance to scheduling problems at other types of transshipment hubs, e.g., at cross docks [surveyed by 5]. However, truck scheduling problems from the literature typically ignore the workforce scheduling aspect. To the best of the authors' knowledge, the only exception is Tadumadze et al. [23], who only consider time-related objectives, however, focusing on synchronizing trucks between inbound and outbound side of a cross dock such that shipments do not miss their designated outbound trucks.

3. Problem description

The problem of scheduling the build-up of unit load devices at an air cargo terminal under space and personnel constraints (ULDSP in the following) can be summarized as follows. Given a set of jobs to perform over a short planning horizon (e.g., one shift or day), where each job consists of the build-up of a given number of unit load devices for a specific outbound flight, *how many workers* should start at *what time* to process each job? At all times, there is a given theoretical maximum number of workers available (typically determined by more long-term personnel shift schedules), which the schedule is not allowed to exceed. Moreover, the number of ULD workspaces where a unit load device may be built up is limited. Furthermore, each job has a given release date (e.g., the time when all cargo destined for a specific flight is available) and a given due date (e.g., the latest time before departure when the completed ULD must be transported to the apron). The goal is to keep the demand for workers just about level at all times, i.e., if possible, there should not be peaks of labor demand at certain times of the day. This should allow shrinking the size of the workforce at least in the medium term and hence lowering labor cost.

To model this problem in a concise fashion, we make the following assumptions.

- All parameters are known with certainty at the time when ULDSP is solved. ULDSP is a short-term problem (typically one shift or day), therefore most information, especially regarding available workers and flight schedules, are indeed quite certain. However, in reality, the exact volume of the shipments to be packed onto ULD can only be estimated. It is therefore possible that a given flight might require a few ULD more or less than originally planned. Note that if not all shipments fit into the allotted number of ULD, that usually means the shipments will be left at the airport for a later flight since aircraft must not be overloaded. These fluctuations are usually not very strong, however, and tend to cancel each other out. We investigate this further in Section 5.
- Once a job is started, space in the warehouse for all ULD associated with that job must be allocated until the job is complete. This is a requirement because of the aforementioned uncertainty regarding the exact volume of individual shipments to be packed. At our industry partner, it is impossible to tell in advance which shipment will fit on which ULD for a given flight, therefore all ULD must be built up in parallel.
- Similarly, a job, once started, must be finished without interruption, i.e., no preemption is allowed. Moreover, a workforce, once assigned to a job, cannot be reassigned before the job is finished.
- All parameters are integer. This is not a very strong assumption because any real-valued parameters can be scaled to integer values to arbitrary precision.
- The number of ULD workspaces in the warehouse constitute the only bottleneck in terms of space. Once all ULD for a flight have been built up, they can always be stored until the flight's departure (without occupying ULD workspaces).
- Once the release date of a job is past, all cargo for the corresponding flight is available. This is not always exactly true in practice, as often cargo trickles in continuously as inbound flights arrive over time. In such cases, the release date can be set to such a time when a sufficient

amount of cargo has arrived to start building up ULD, e.g., when the final shipment will arrive within the processing time.

3.1. Mathematical formulation

Let $j = \{1, \dots, n\}$ be a set of n jobs, where each job corresponds to the build-up of all ULD for one outbound flight. Each job $j \in J$ is associated with a release date a_j, i.e., the earliest time when job j can be started, and a due date d_j, the latest time when job j can be finished. Each job requires $s_j \in \mathbb{N}^{\neq 0}$ workspaces for ULD, of which there is a total of S available. Moreover, each job j takes a processing time of p_{jk}, which depends on the number $k \in K_j$ of workers assigned to that job. The set $K_j \subset \mathbb{N}^{\neq 0}$ contains the potential numbers of workers that can be assigned to a job. Note that, typically, we would expect a job to be performed more quickly the more workers are assigned to it, i.e., $p_{jk} < p_{jk'}$ if $k' < k$. The planning horizon is T periods long; in each period there is a theoretical maximum number of $\underline{K}_t$ workers available.

A schedule Σ for ULDSP is a set of 3-tuples $(j, t, k) \in \Sigma$, indicating that job $j \in J$ starts processing by $k \in K_j$ workers at time $t \in \{a_j, \dots, d_j - p_{jk}\}$. We say that a schedule is feasible if it meets the following conditions.

- Each job is executed exactly once, i.e., for each job $j \in J$, there is exactly one 3-tuple $(j, t, k) \in \Sigma$.
- At no time the total number of assigned workers exceeds the theoretically available workforce, i.e., for all $t = 1, \dots, T$, it must hold that $\sum_{\substack{(j,t',k)\in\Sigma: \\ t \geq t' \geq t - p_{jk}+1}} k \leq \underline{K}_t$.
- At no time the total number of ULD of all active jobs exceeds the available workspace, i.e., for all $t = 1, \dots, T$, it must hold that $\sum_{\substack{(j,t',k)\in\Sigma: \\ t \geq t' \geq t - p_{jk}+1}} s_j \leq S$.

As for the goal of the optimization, one of the major problems with the status quo scheduling of ULD build-up at the air cargo terminal we visited is poor labor utilization due to large demand peaks at certain times of day. Demand levelling is also often seen as a desirable goal in the personnel scheduling literature [e.g., 19]. Among all feasible schedules we therefore seek one where the number of workers in the busiest period is minimal, i.e.,

$$\text{Minimize } f(\Sigma) = \max_{t=1,\dots,T} \left\{ \sum_{\substack{(j,t',k)\in\Sigma: \\ t \geq t' \geq t - p_{jk}+1}} k \right\} \tag{1}$$

Note that minimizing Objective (1) does not automatically ensure that the given maximum number of workers $\underline{K}_t$ is never exceeded or vice versa. Some $\underline{K}_t$ may be lower than the optimal $f(\Sigma)$ (e.g., during lunch breaks or other off-periods) because $f(\Sigma)$ only measures the number of workers in the very busiest period.

Table 1 summarizes the introduced notation, which we use to present the following mixed-integer linear program, which permits the use of default solvers.

T	number of periods, index $t \in \{1, \dots, T\}$
J	set of jobs, index $j \in \{1, \dots, n\}$
S	number of workspaces in the warehouse
K_j	set of processing modes for job *j*
$\underline{K}_t$	maximum number of workers available in period *t*
s_j	space required by job *j*
a_j	release date of job *j*
d_j	due date of job *j*
p_{jk}	processing time of job *j* if *k* workers are assigned to it
x_{jtk}	binary variable: 1, if job *j* is started in period *t* by *k* workers; 0, otherwise
α_t	continuous variable: number of workers busy in period *t*

Table 1: Notation for the MILP model.

$$\text{[ULDSP] Minimize } F(x, \alpha) = \max_{t=1,\dots,T} \{\alpha_t\} \tag{2}$$

subject to

$$\textstyle\sum_{j \in J} \sum_{k \in K_j} \sum_{t'=max\{1; t-p_{jk}+1\}}^{t} s_j . x_{jt'k} \leq S \qquad \forall t \in \{1, \dots, T\} \tag{3}$$

$$\textstyle\sum_{j \in J} \sum_{k \in K_j} \sum_{t'=max\{1; t-p_{jk}+1\}}^{t} k . x_{jt'k} \leq \alpha_t \qquad \forall t \in \{1, \dots, T\} \tag{4}$$

$$\alpha_t \leq \underline{K}_t \qquad \forall t \in \{1, \dots, T\} \tag{5}$$

$$\textstyle\sum_{t=1}^{T} \sum_{k \in K_j} x_{jtk} = 1 \qquad \forall j \in \mathrm{J} \tag{6}$$

$$\textstyle\sum_{t=1}^{T} \sum_{k \in K_j} t . x_{jtk} \geq a_j \qquad \forall j \in \mathrm{J} \tag{7}$$

$$\textstyle\sum_{t=1}^{T} \sum_{k \in K_j} (t + p_{jk}) . x_{jtk} \leq d_j \qquad \forall j \in \mathrm{J} \tag{8}$$

$$x_{jtk} \in \{0,1\} \qquad \forall j \in \mathrm{J}, \forall k \in K_j, \forall t \in \{1, \dots, T\} \tag{9}$$

Objective function (2) minimizes the number of workers active in the busiest period. Constraints (3) ensure that the available space is never exceeded. Constraints (4) in conjunction with (5) limit the number of workers used per period to $\underline{K}_t$. Constraints (6) enforce that each job is executed exactly once, while (7) and (8) render violating any time window impossible. Finally, (9) define the domain of the variables.

Note that it is easy to see that even finding a feasible solution to ULDSP is already strongly NP-hard as it is a generalization of single machine scheduling with time windows, which is well-known to be NP-complete in the strong sense [14]. This leads us to the following proposition.

Proposition 3.1. *Solving ULDSP to feasibility is NP-hard in the strong sense.*

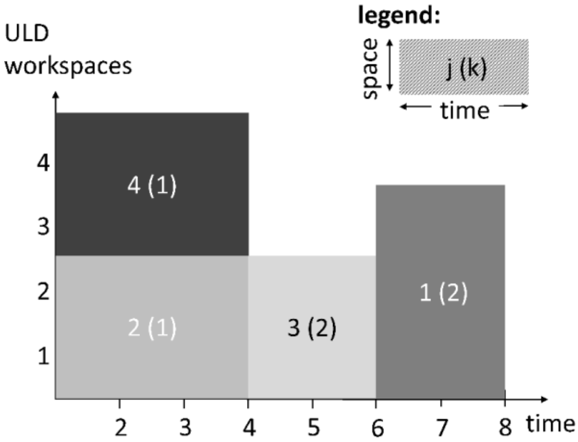
3.2. Example of a ULDSP solution

Consider the example data from Table 2a, consisting of $n = 4$ jobs and $T = 8$ periods in the planning horizon. Each of the 4 jobs can be performed by either 1, 2, or 3 workers (i.e., $K_1 = K_2 = K_3 = K_4 = \{1, 2, 3\}$), and in each period, there are at most 3 workers available (i.e, $\underline{K}_t = 3, \forall t = 1, \dots, 6$), except in the last two periods, where there are only 2 workers, i.e., $\underline{K}_7 = \underline{K}_8 = 2$. Moreover, let the maximum number of ULD that can be built up in parallel be $S = 4$.

A feasible and optimal solution is depicted in Figure 2b as a Gantt chart. Formally, this corresponds to schedule $\Sigma = \{(1,6,2), (2,1,1), (3,4,2), (4,1,1)\}$. The maximum number of workers ever used at any one time is 2, hence $f(\Sigma) = 2$.

j	1	2	3	4
p_{j1}	6	3	3	3
p_{j2}	2	1	2	3
p_{j3}	1	1	1	2
s_j	3	2	2	2
a_j	1	1	4	1
d_j	8	5	8	7

(a) Example problem.



(b) Gantt chart of the example solution.

Figure 2: Example data and solution.

4. Algorithm for the ULDSP

4.1. Reformulation of the problem

While the model from Section 3.1 can be tackled by default solvers, Proposition 3.1 suggests that large instances of realistic size cannot be solved in acceptable time, which is indeed what our computational experiments confirm (see Section 5). To make the problem more tractable, we propose to reformulate it as a type of generalized set partitioning problem. Such models have been used to great success to solve, for instance, discrete berth allocation problems [7] or the train driver recovery problem [21]. Similar procedures have also been proposed to tackle truck scheduling problems that would otherwise be outside the reach of default solvers [4, 23]. Moreover, this approach can also be used as a heuristic, permitting the solution of significantly larger instances at the cost of some solution quality.

In the set partitioning model, each variable corresponds to one feasible assignment of a job j to a number of workers $k \in K_j$ and a starting time $t \in \{a_j, \ldots, d_j - p_{jk}\}$. For brevity of notation, we denote such an assignment by a 3-tuple $\omega = (j, t, k)$, the set of all such assignments as $\Omega = \left\{(j,t,k) | j \in J, k \in K_j, t \in \{a_j, \ldots, d_j - p_{jk}\}\right\}$, and the subset containing only assignments specifically for job $j \in J$ as $\Omega_j = \{(j', t, k) \in \Omega | j' = j\}$. Consequently, once set Ω is given, the scheduling decision consists merely of choosing one $\omega \in \Omega_j$ for each $j \in J$. For each $\omega \in \Omega$, we therefore define binary variable y_ω, which is 1 if and only if assignment $\omega = (j, t, k)$ is in the solution, i.e., job j will be processed in the interval $[t; t + p_{jk}]$ by k workers, requiring s_j ULD workspaces.

Ω_j	set of possible start time and worker assignments for job j
$\sigma_{\omega t}$	number ULD workspaces taken up at time t by assignment $\omega \in \Omega$
$\kappa_{\omega t}$	number of workers required at time t by assignment $\omega \in \Omega$
y_ω	binary variable: 1, if assignment ω is in the solution; 0, otherwise

Table 2: Additional notation for the generalized set partitioning model.

Given set Ω, we can preprocess parameters $\kappa_{\omega t}$ and $\sigma_{\omega t}$. $\kappa_{\omega t}$ denotes the number of workers required at time t by assignment $\omega \in \Omega$. $\sigma_{\omega t}$ stands for the number ULD workspaces taken up at time t by assignment ω. Given these parameters, we can reformulate ULDSP as a generalized set partitioning problem, which we refer to as ULDSP-GSP.

[ULDSP-GSP] Minimize $F(y, \alpha) = \max_{t=1,\ldots,T} \{\alpha_t\}$ (10)

subject to Constraints (5) and

$$\sum_{\omega \in \Omega} \sigma_{\omega t} \cdot y_\omega \leq S \qquad \forall t \in \{1, \ldots, T\} \tag{11}$$

$$\sum_{\omega \in \Omega} \kappa_{\omega t} \cdot y_\omega \leq \alpha_t \qquad \forall t \in \{1, \ldots, T\} \tag{12}$$

$$\sum_{\omega \in \Omega_j} y_\omega = 1 \qquad \forall j \in J \tag{13}$$

$$y_\omega = \{0,1\} \qquad \forall \omega \in \Omega \tag{14}$$

Objective (10) still minimizes the maximum number of workers in any one period. Constraints (11) and (12) replace Constraints (3) and (4) to make sure that neither the available workforce nor space is over-extended. Constraints (13) enforce that each job must be processed according to exactly one 3-tuple from Ω_j, and (14) are the binary constraints.

Example (cont.): Consider the example from Section 3.2. Figure 3 shows all possible assignments (9 in total) for job $j = 3$ in the example.

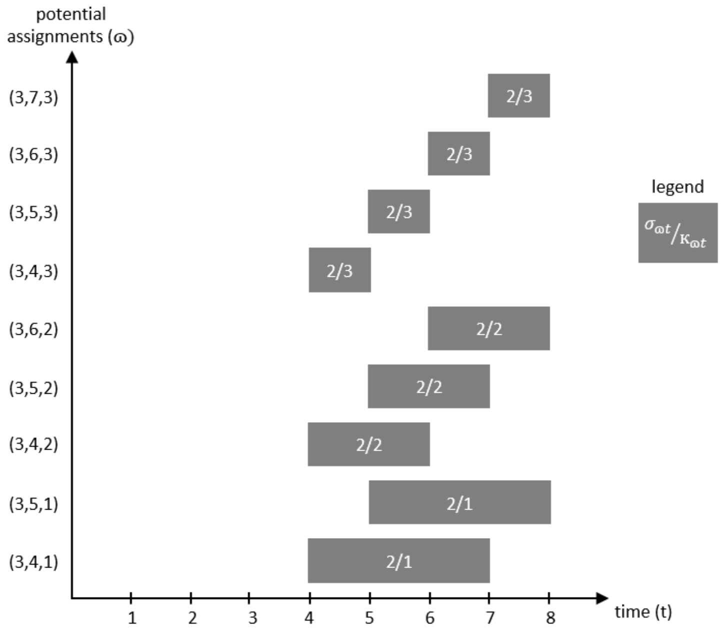


Figure 3: All possible assignments Ω_3 for job $j = 3$ in the example

Model [ULDSP-GSP] is somewhat simpler than the original model [ULDSP]. The number of variables necessary may still be extremely large, however, depending on the number of potential assignments $|\Omega|$. We will investigate the computational performance in Section 5. Regardless, ULDSP-GSP is clearly a difficult problem, as we show in the following.

Proposition 4.1. *ULDSP-GSP is NP-hard in the strong sense even if* $S = \infty$ *and* $\underline{\mathrm{K}}_t = \infty$, $\forall t \in \{1, \dots, T\}$, $|\Omega| = 3.n$, *and the processing time of each job is 2.*

Proof. To show that ULDSP-GSP is NP-hard, we will present a reduction from an interval scheduling problem which has been proven NP-hard in the strong sense by Nakajima and Hakimi [17] (dubbed DST(1) by the authors). Crama and Spieksma [8] tightened this result further by showing strong NP-hardness holds even if the number of intervals is restricted to 3 and the processing time to 2 for each job.

DST(1): Given is a set Q of independent tasks to be scheduled without preemption on a single machine. Each task $q \in Q$ requires a given handling time of 2 and it may be started at any one of its 3 prescribed starting times $c_{q,1}$, $c_{q,2}$, $c_{q,3}$. Is there a conflict-free schedule where all jobs are processed?

We transform any instance of DST(1) to an instance of ULDSP-GSP by considering a problem with a planning horizon of $T = \max_{q \in Q, l \in \{1,2,3\}} \{c_{q,l} + 2\}$ and introducing a job *j* for each task *q*. With each job *j*, we associate a set of assignments $\Omega_j = \{(j, c_{q,2}, 1), (j, c_{q,2}, 1), (j, c_{q,3}, 1)\}$, making the overall set of assignments $\Omega = \bigcup_{j \in J} \Omega_{\mathrm{j}}$. Moreover, the processing time of each job *j* is 2. Note that every assignment only employs one single worker. The question asked is: Is there a feasible schedule where $\max_{t=1,\ldots,T}\{\alpha_t\} \leq 1$?

Clearly, a solution where $\max_{t=1,\ldots,T}\{\alpha_t\} \leq 1$ is one where no two jobs are ever concurrently active. Seeing that start and processing times in the transformed ULDSP-GSP correspond exactly to those in the DST(1) instance, the equivalence of an DST(1) and ULDSP-GSP solution is hence obvious.

4.2. Selecting assignments heuristically

Regarding the generation of sets Ω_j, it would of course be possible to include every possible combination of workers and start times for each job. However, to cut down on computation time, it may be more expedient to only generate a subset $\Omega'_j \subset \Omega_j$ of all potential assignments. Specifically, we restrict the number of assignments for each job by setting a parameter ι. The number of assignments per job is then limited to $|\Omega'_j| = \left\lceil |\Omega_j| / \iota \right\rceil$. Obviously, if $\iota = 1$, then model ULDSP-GSP is equivalent to model ULDSP because all possible assignments are taken into consideration. If $\iota > 1$, only a true subset of assignments is considered, and there is no guarantee that the search space contains the optimal solution. In this case, a subset of assignments needs to be selected heuristically for each job. We propose three different ways to do so.

random For each job j, we create an assignment by drawing a random number k of workers from K_j (discrete uniform distribution) and then a random start time t from $\{a_j, \ldots, d_j - p_{jk}\}$ (also discrete uniform distribution). We add assignment (j, t, k) to set Ω'_j and repeat these steps until we have generate$\left\lceil |\Omega_j| / \iota \right\rceil$d distinct assignments for each job.

semi-random Let $\Omega_{jk} = \{(j, t, k') \in \Omega_j | k' \neq k\}$ be the set of assignments for job j with exactly $k \in K_j$ workers. For each job, we select from each set Ω_{jk} a number of $\left\lceil |\Omega_{jk}| / \iota \right\rceil$ distinct assignments randomly, $\forall k \in K_j$. The rationale behind this strategy is to make the assignment selection somewhat less random by making sure that each potential number of workers is represented in Ω'_j.

smooth Similar to the semi-random strategy, we consider sets Ω_{jk}. For all $j \in J$ and $k \in K_j$, from each set Ω_{jk}, we select a number of $\left\lceil |\Omega_{jk}| / \iota \right\rceil$ assignments such that their start times are as evenly spaced out as possible. That is, for a given job j and number of workers k, we select start times a_j, $a_j + \left\lceil \frac{d_j - p_{jk} - a_j + 1}{\left\lceil |\Omega_{jk}| / \iota \right\rceil} \right\rceil$, $a_j + 2.\left\lceil \frac{d_j - p_{jk} - a_j + 1}{\left\lceil |\Omega_{jk}| / \iota \right\rceil} \right\rceil$, …, $a_j + \left(\left\lceil |\Omega_{jk}| / \iota \right\rceil - 1\right).\left\lceil \frac{d_j - p_{jk} - a_j + 1}{\left\lceil |\Omega_{jk}| / \iota \right\rceil} \right\rceil$.

Regardless of the selection strategy, by varying parameter ι, we can trade off solution quality for computational effort to a lesser or greater degree. We will investigate this in our computational study (Section 5).

Example (cont.): Consider the assignments given in Figure 3 for job $j = 3$. Assume $\iota = 2$. The set of all possible assignments is

$$\Omega_3 = \{(3,4,1),(3,5,1),(3,4,2),(3,5,2),(3,6,2),(3,4,3),(3,5,3),(3,6,3),(3,7,3)\}$$

- Using the random strategy, we pick $\left\lceil |\Omega_j| / \iota \right\rceil = 5$ tuples from Ω_3 at random, e.g.,
$$\Omega'_3 = \{(3,5,1),(3,5,2),(3,6,2),(3,5,3),(3,7,3)\}$$
- Using the semi-random strategy, we first divide set Ω_3 into $|K_j|$ subsets, i.e., $\Omega_{3,1} = \{((3,4,1),(3,5,1)\}$, $\Omega_{3,2} = \{(3,4,2),(3,5,2),(3,6,2)\}$, $\Omega_{3,3} = \{(3,4,3),(3,5,3),(3,6,3),(3,7,3)\}$. Then, we select from each set $\Omega_{3,k}$ $\left\lceil |\Omega_{jk}| / \iota \right\rceil$ elements at random, e.g.,
$$\Omega'_3 = \{(3,5,1),(3,4,2),(3,6,2),(3,5,3),(3,6,3)\}$$
- Using the smooth strategy, we pick from each set $\Omega_{3,k}$ the $\left\lceil |\Omega_{jk}| / \iota \right\rceil$ elements whose start times are evenly spaced out, i.e.,
$$\Omega'_3 = \{(3,4,1),(3,4,2),(3,6,2),(3,4,3),(3,6,3)\}$$

5. Computational study

5.1. Benchmark instances and computational environment

As no benchmark instances for ULDSP exist in the literature, we first explain how we generated test data. To obtain realistic instances, we collaborated with a major German airfreight carrier.

We assume a planning horizon of one day. One period corresponds to 10 minutes of real time, consequently we consider a total of $T = 144$ periods. Space and handling time requirements can vary wildly depending on the flights for which the ULD are to be built up. By Proposition 3.1, even finding a feasible solution for ULDSP is already NP-hard. To ensure that a feasible solution exists at all, we generate workforce and space requirements as well as release and due dates and processing times via the procedure outlined in Algorithm 1.

```
j := 1;
while j ≤ n do
    p_j := rnd(3,36);
    a_j := rnd(1, T − p_j);
    s_j := rnd(1,10);
    k_j := rnd(1, 3.s_j);
    for t = a_j to T − p_j do
        if job j can be scheduled in the interval [t; t + p_j] with regard to already blocked resources then
            η_j := t;
            d_j := rnd(η_j + p_j, T);
            block space s_j and workforce k_j during the interval [η_j, η_j + p_j];
            ρ_j := p_j . β√(k_j);
            for k := 3 to 3.s_j do
                p_jk := ⌈ρ_j / β√(k_j)⌉;
                if p_jk > 3 and a_j + p_jk ≤ T and p_jk ≠ p_j,k−1 then
                    K_j := K_j ∪ {k};
                end
            end
            j := j + 1;
            break;
        end
    end
end
```

Algorithm 1: Generating workforce and space requirements as well as release and due dates and processing times.

The basic idea is that, for each job j, we randomly generate a release date a_j (*rnd()* stands for a uniformly distributed random integer from the interval in the argument), space requirement s_j, and a "reference" processing time p_j when using k_j workers. If it is possible to schedule job j sometime in the interval between a_j and the end of the planning horizon, given that the previous jobs $1, \dots, j-1$ have already been scheduled and may thus block workers and space, we accept the generated parameters, set the due date and calculate the processing times p_{jk}. To get the processing time depending on the number of workers, we first set a baseline processing time ρ_j , which is modified according to the number of workers k as $\left\lceil \frac{\rho_j}{\sqrt[\beta]{k_j}} \right\rceil$. The rationale behind this formula is that we expect the processing time to drop if additional workers are employed. However, the decrease is unlikely to be linear because the labor is rarely perfectly divisible and at some point additional workers will start interfering with each other. The rate of decrease is adjusted through parameter β: the greater β, the more quickly the marginal speedup through additional workers diminishes. Note especially that if $\beta = 1$, the speedup is indeed linear, i.e., doubling the number of workers halves the processing time.

Using this method, we create two data sets, each containing 20 individual instances. One set contains small problems, where $S = 72, n = 20$, and $\underline{K}_t = 100, \forall t = 1, \dots, T$, and one set contains large problems, where $S = 180, n = 100$, and $\underline{K}_t = 200, \forall t = 1, \dots, T$. Moreover we set $\beta = 2$. All instances are available from the authors upon request.

5.2. Computational results

5.2.1. Algorithmic performance

In the first part of our computational study, we investigate the performance of our proposed solution methods. For this purpose, we implemented them in Java 8 and had them solve our test instances on an x64-PC with 12 GB of RAM and an Intel Core i5-7200U 2.5 GHz CPU. To solve models ULDSP and ULDSP-GSP, we used a default solver, namely CPLEX 12.7.

		avg. saved time	avg. opt. gap	max. opt. gap	min. opt. gap
	ULDSP-GSP	-16.86%	0.00%	0.00%	0.00%
random	$\iota = 2$	-83.23%	3.88%	10.00%	0.00%
	$\iota = 3$	-88.66%	8.30%	23.33%	3.23%
	$\iota = 4$	-98.07%	11.50%	23.33%	0.00%
	$\iota = 5$	-98.29%	14.40%	33.33%	6.52%
semi-random	$\iota = 2$	-71.34%	0.78%	4.00%	0.00%
	$\iota = 3$	-95.86%	1.30%	4.00%	0.00%
	$\iota = 4$	-99.73%	2.05%	7.41%	0.00%
	$\iota = 5$	-100.00%	1.80%	6.25%	0.00%
smooth	$\iota = 2$	-84.11%	0.63%	4.00%	0.00%
	$\iota = 3$	-97.27%	0.48%	4.00%	0.00%
	$\iota = 4$	-99.49%	1.60%	5.00%	0.00%
	$\iota = 5$	-100.00%	2.07%	9.38%	0.00%

Table 3: Comparison of the assignment selection strategies (small instances).

The 20 small instances could all be solved to (proven) optimality by CPLEX. Table 3 lists the average, minimum and maximum optimality gaps of the generalized set partitioning-based methods, calculated as $(f^* - f^{opt})/f^*$, where f^{opt} denotes the optimal objective value and f^* denotes the best objective value of the solution method under investigation. ULDSP-GSP stands for the generalized set partitioning model which includes all possible assignments and thus is guaranteed to return an optimal solution. Rows random, semi-random, and smooth denote the heuristic assignment selection strategies discussed in Section 4.2. Moreover, we distinguish between different settings of parameter ι, which influences the size of the subset of assignments taken into consideration when solving [ULDSPGSP]. Finally, column *avg. saved time* shows the average relative reduction in CPU time over CPLEX solving model [ULDSP]. Note that all CPU times for the generalized set partitioning-based methods include the preprocessing time necessary to generate the set of assignments Ω in the first place.

Expectedly, the optimality gap for CPLEX solving model ULDSP-GSP is 0% because this model is equivalent to the original model ULDSP. Interestingly, however, ULDSP-GSP could be solved a little more quickly on average than the original model. To really cut down on CPU times, however, the heuristic assignment selection strategies need to be used. The greater ι, the fewer assignments are added to model ULDSP-GSP, which on the one hand results in shorter runtimes, but on the other hand makes it more likely that the optimal (or even a good) solution is no longer part of the search space. The results clearly show that the former is definitely true: especially with $\iota = 5$, the CPU times become essentially negligible for the small instances, being barely measurable. On the downside, the solution quality suffers somewhat, although the average optimality gaps are still quite low, at least for the *semi-random* and *smooth* selection strategies.

All things considered, the *smooth* selection strategy at $\iota = 3$ presents itself as the most convincing compromise between runtime and solution quality. The optimality gaps are among the best, well below 1% on average, while at the same time reducing CPU times by more than 97% over CPLEX solving the default model. This indicates that it is advantageous to forego the random element in the assignment selection and systematically choose assignments that are spread out over the search space. Note that this is not necessarily true for every single instance – regardless of the selection strategy, if $\iota > 1$, the choice of subsets is always heuristic. For this reason, it is possible in some instances that selection strategies that are poorer on average nonetheless happen to include better assignments and thus yield lower objective values. Seeing that the very maximum optimality gap of the *smooth* strategy with $\iota = 3$ is merely 4%, the performance of this strategy seems quite stable, however. In the following tests, we will therefore only focus on this heuristic.

	ULDSP		ULDSP-GSP	smooth ($\iota = 3$)	
ID	*f* (opt.)	CPU sec.	CPU sec.	rel. opt. gap	CPU sec.
1	20	7	7	0.00%	0
2	60	1	0	0.00%	0
3	22	1	1	0.00%	0
4	27	2	2	3.70%	0
5	55	1	0	1.82%	0
6	26	1	1	0.00%	0
7	19	4	3	0.00%	0
8	30	20	18	0.00%	1
9	32	3	4	0.00%	0
10	45	1	0	0.00%	0
11	35	1	1	0.00%	0
12	22	245	731	0.00%	2
13	29	21	8	0.00%	1
14	25	7	6	4.00%	2
15	30	1	0	0.00%	0
16	31	2	1	0.00%	0
17	12	6	6	0.00%	0
18	46	6	6	0.00%	0
19	20	1	1	0.00%	0
20	25	13	12	0.00%	2
	avg.	17.2	40.4	0.48%	0.4

Table 4: Comparison of solution methods (small instances).

Table 4 lists the test results for each of the 20 small instances in more detail. CPLEX managed to solve model ULDSP-GSP at least as fast as model ULDSP in all but two instances. In one case (instance 12), the picture is reversed, however, and solving the generalized set partitioning model actually took several additional minutes. The real advantage of model ULDSP-GSP is that it can easily be turned into a heuristic. Using the *smooth* selection strategy at $\iota = 3$, the optimality gap is 0 in most instances.

Finally, Table 5 takes a look at the results for the large instances. For these instances, no proven optimal solution is available as CPLEX turned out to be incapable of finishing within a one hour time limit. Instead, we print the best found objective value after one hour of computation in column *f* and use this value as a benchmark. All solution methods are capable of finding a solution whose objective value comes at the very least close to *f* if given one hour of CPU time. However, in light of ULDSP being an operational problem that has to be solved frequently, hour-long computations are most likely not acceptable for practitioners. Therefore, we compare the solution methods by the best found solution within a 10 second time limit. 10 seconds of CPU time should be adequate even for the most demanding applications. Given a 10 second time limit, CPLEX does not find a feasible – let alone optimal – solution when solving the original model ULDSP in most instances. In those few cases where it did find a solution, it is far worse than the best known one. Solving model ULDSP-GSP in 10 seconds, the default solver turns out to be somewhat more successful, at least in finding feasible solutions. The gap is still quite significant in many cases, however. Our best heuristic strategy (*smooth* at $\iota = 3$), on the other hand, is clearly superior at finding good solutions with limited time. The very worst gap to the best known solution is below 4%, indicating a favorable tradeoff of CPU time and solution quality.

	best known solution	ULDSP	ULDSP-GSP	smooth ($\iota = 3$)
ID	*f*	gap	gap	gap
1	94	112.77%	112.77%	2.13%
2	130	-	7.69%	1.54%
3	90	-	122.22%	2.22%
4	89	123.60%	79.78%	3.37%
5	110	-	2.73%	1.82%
6	103	-	-	0.97%
7	100	-	1.00%	1.00%
8	118	-	2.54%	1.69%
9	89	-	11.24%	1.12%
10	108	80.56%	7.41%	1.85%
11	112	76.79%	77.68%	0.89%
12	107	-	8.41%	1.87%
13	86	132.56%	6.98%	1.16%
14	128	-	56.25%	1.56%
15	117	-	43.59%	1.71%
16	119	68.07%	2.52%	0.84%
17	101	-	1.98%	3.96%
18	117	-	11.97%	1.71%
19	100	-	69.00%	0.00%
20	122	-	4.10%	2.46%
	avg.	99.05%	33.15%	1.69%

Table 5: Comparison of solution methods given a 10 second time limit (large instances); a dash (-) denotes that no feasible solution was found within the time limit.

5.3. Practical implications

In the second part of our computational study, we investigate the practical effects of (near-)optimal schedules on day-to-day operations and aim to derive managerial insights. As mentioned in the introduction (Section 1), the current practice at the cargo terminal we visited is to start building up cargo at a predefined time before each flight departs. To analyse to what extent optimized schedules as proposed in this paper can improve on this rule of thumb, we perform the following series of tests.

We implement the rule of thumb we observed in practice as follows, assuming that all jobs start processing at most 6 hours before departure of the corresponding flight.

1. Sort all jobs according to non-decreasing due date.
2. For each job j in sorted order:
    a. Start job j as early as possible within the interval $\left[\max\{0; d_j - 36\}; d_j - \min_{k \in K_j}\{p_{jk}\}\right]$, taking into account previously scheduled jobs, which may block resources (space and workers).
    b. Assign the minimum number $k \in K_j$ of workers to job j such that it finishes not later than its due date d_j.
    c. Block space s_j and workers k during the time the job is active.

Note that when executing step 2, in some cases it might not be possible to schedule a job within its six-hour time window because the previously scheduled jobs do not leave enough available ULD space. In this case, we schedule the job within its interval such that it exceeds the available space as little as possible. Moreover, we relax the bound on the number of workers $\underline{K}_t$, that is, the number of workers assigned to a job is always such that it can be finished by its deadline, even if the total exceeds $\underline{K}_t$.

Table 6 compares the results of the rule of thumb to those of our best heuristic (*smooth* at $\iota = 3$), where *f* denotes the objective value (maximum number of workers) and *max. S* stands for the maximum number of ULD workspaces used at any one time. *Gap* lists the reduction in *f* and *S*, respectively, if the *smooth* heuristic is used, calculated as $(f^{rot} - f^{smooth})/f^{rot}$, where f^{rot} (f^{smooth}) is the objective value returned by the rule of thumb (*smooth* heuristic); analogous for gap S.

Quite remarkably, optimized schedules can substantially cut down on the required workforce. Smartly scheduling jobs to avoid extreme peak hours makes more than half the labor force redundant. More than that, intelligent schedules also reduce the space requirements by more than 20% on average as a side-effect, which may be good news for tight warehouses.

	rule of thumb		smooth ($\iota = 3$)	
ID	f	max. S	gap f	gap S
1	166	193	58.43%	7.25%
2	200	205	47.00%	20.00%
3	240	212	56.67%	20.28%
4	198	199	43.43%	30.15%
5	247	214	55.06%	19.63%
6	204	210	61.27%	15.24%
7	204	192	48.04%	7.29%
8	271	221	50.55%	31.22%
9	292	240	65.07%	25.42%
10	263	232	56.27%	30.17%
11	263	245	58.94%	26.53%
12	200	191	60.00%	5.76%
13	206	210	51.46%	38.10%
14	226	196	47.35%	21.94%
15	211	225	57.82%	27.56%
16	250	219	49.60%	24.20%
17	238	217	63.03%	17.51%
18	249	225	51.41%	27.11%
19	294	253	54.76%	28.85%
20	247	197	53.04%	12.69%
		avg.	54.46%	21.85%

Table 6: Comparison of optimized schedules and a rule-of-thumb (large instances).

To analyse the tradeoff between size of the workforce and space requirements in more detail, we solve large instances generated as above with the *smooth* heuristic at $\iota = 3$, except that we vary the available space S for each instance. The expectation is that the fewer ULD workspaces there are (i.e., the lower S), the more workers need to be hired to quickly build up ULD and release space sooner (i.e., the greater f). Table 7 shows the efficient frontier: the best number of workers for a given warehouse size for each of the 20 large instances. The data confirm our intuition: less space implies additional workers; however, the effect is not always very major. If space is very tight ($S = 90$), it is sometimes not possible to find any feasible solution at all, no matter how many workers are employed. Once space is sufficiently large, however, further increasing it has little effect on f. In our instances this point of diminishing return is reached when $S = 144$. The exact number may vary depending on the setting but our tests clearly indicate that – given good schedules – increasing the size of the warehouse beyond a certain point is almost pointless.

In the final part of our computational study, we consider the effect of uncertainty. As pointed out in Section 3, one assumption of our model is that it is already known with certainty how many ULD will be necessary for a given flight. This is, however, not necessarily true in practice. Since shipment sizes can vary considerably and the exact measurements are usually not known in advance, it is entirely possible that either more or fewer ULD may be required than originally planned. Note that extra shipments that cannot be loaded onto the originally allotted ULD are usually left at the airport because aircraft cannot be overloaded. How these “surplus” ULD are handled is, however, immaterial for our following computational tests.

ID	*S=90*	*S=126*	*S=144*	*S=162*	*S=180*	*S=225*
1	-	129	128	127	127	127
2	134	105	100	96	95	95
3	125	103	100	99	99	100
4	114	108	108	108	108	106
5	200	116	108	105	105	103
6	146	125	122	121	121	121
7	-	129	123	120	119	121
8	200	199	107	106	106	105
9	-	200	136	135	135	135
10	102	95	93	93	94	93
11	118	115	115	115	115	115
12	118	103	101	101	101	101
13	145	101	99	97	98	97
14	-	140	126	121	119	118
15	136	114	112	112	112	112
16	135	105	101	101	102	101
17	120	104	104	104	104	104
18	129	106	106	105	106	105
19	114	104	104	104	103	103
20	199	120	115	104	101	100
avg.	139,69	121,05	110,40	108,70	108,50	108,10

Table 7: Size of workforce required depending on the available space in the warehouse (large instances); dashes (-) denote that no feasible solution could be found.

To test whether these stochastic fluctuations have any effect on the feasibility of a schedule, we solve large instances generated as before using *smooth* at $\iota = 3$, and then modify them ex-post by randomly selecting γ percent of jobs and randomly varying (increasing or decreasing) their space requirement s_j by up to ζ percent. Table 8 contains the total capacity violation for varying levels of uncertainty (i.e., varying γ and ζ). Violations are calculated as the total number of ULD workspaces in excess of the available S workspaces over the entire planning horizon. Obviously, lower values are better.

As can be expected, the more uncertainty there is, the more problematic the feasibility. All things considered, however, schedules generated by our *smooth* heuristic seem to be at least moderately robust. Some few instances (especially no. 19) are apparently sensitive to fluctuations, but in most cases the total violations are 0 even in the worst case, where $\gamma = \zeta = 30\%$. Apparently, positive and negative fluctuations tend to cancel each other out in many cases. This allows drawing the conclusion that our solution methods can be used with a degree of reliability even in cases of uncertainty.

γ	10%			20%			30%		
ζ	10%	20%	30%	10%	20%	30%	10%	20%	30%
ID 1	0	0	0	0	0	0	0	0	0
ID 2	0	0	0	0	0	0	0	0	0
ID 3	0	0	0	0	0	0	0	0	0
ID 4	0	2	0	0	0	11	0	0	0
ID 5	0	0	0	0	0	0	0	0	0
ID 6	0	0	3	0	0	0	0	0	0
ID 7	0	0	0	1	3	0	0	3	67
ID 8	8	4	1	22	0	0	0	22	39
ID 9	0	0	0	0	10	0	12	1	32
ID 10	0	0	0	0	8	0	0	1	34
ID 11	0	0	0	0	0	0	0	0	0
ID 12	0	0	0	0	0	0	0	0	0
ID 13	0	0	0	0	10	55	0	0	24
ID 14	19	48	73	41	11	210	14	17	168
ID 15	0	0	0	0	0	0	0	0	0
ID 16	0	1	123	5	75	87	120	5	238
ID 17	0	0	0	0	0	0	0	0	0
ID 18	0	0	0	0	0	0	0	0	0
ID 19	1	1	4	0	0	10	6	12	3
ID 20	0	6	0	0	8	2	12	0	0
avg.	1,4	3,1	10,2	3,45	6,25	18,75	8,2	3,05	30,25

Table 8: Violation of the space constraint if the number of ULD per flight is uncertain (large instances); γ denotes the percentage of flights affected, ζ is the maximum percentage by which ULD requirements can fluctuate.

6. Conclusion

In this paper, we investigated a problem we encountered at a terminal of a major German airfreight carrier. It consists of scheduling the build-up of unit load devices and assigning a number of workers to each job. The problem is made more complicated by the fact that space for building up ULD is limited. The goal is to keep the peak labor force as small as possible. We proposed two mixed-integer programming models for this novel problem, one of which formed the basis of a heuristic method. In a computational study we showed that, using the right parameter settings, our heuristic performs very well, delivering optimality gaps well below 1% on average.

Regarding managerial insights, we derived the following take-home messages.

- (Near-)optimal schedules as obtained by our heuristics are significantly better at avoiding large peak workloads than the simple rule of thumb we encountered in practice. On average, the required peak workforce could be more than halved.

- It is possible to trade off ULD space in the warehouse for workers. However, there are limits, as too low a number of ULD workspaces may make it impossible to find a feasible solution at all, while increasing the warehouse size hardly diminishes the required workforce after a certain point.
- In reality, the exact number of ULD required for each flight is not always known with certainty. Our tests indicate that schedules generated by our heuristics are fairly robust in the face of unforeseen fluctuations, although in some (few) cases, significant violations of the spatial constraint may occur.

To make the schedules more robust, future research could focus on incorporating the decision of which individual shipments to pack onto which ULD into the problem. At the moment, this seems to be technically impossible, at least with our industry partner, as detailed information on the exact measurements of the shipments is simply not available accurately for sufficient cases. However, it may be possible (and worthwhile) to obtain this data in the future. Another assumption that is currently made in practice but may be relaxed in the future is non-preemption, i.e., it may be unnecessary to reserve space for all ULD during the entire processing time of a job, which would require new algorithms.

References

[1]. Azadian, F., Murat, A. E., and Chinnam, R. B. (2012). Dynamic routing of time-sensitive air cargo using real-time information. Transportation Research Part E: Logistics and Transportation Review, 48(1):355–372.

[2]. Bandelloni, M., Tucci, M., and Rinaldi, R. (1994). Optimal resource leveling using non-serial dyanamic programming. European Journal of Operational Research, 78(2):162–177.

[3]. Bła˙zewicz, J., Machowiak, M., W˛eglarz, J., Kovalyov, M. Y., and Trystram, D. (2004). Scheduling malleable tasks on parallel processors to minimize the makespan. Annals of Operations Research, 129(1-4):65–80.

[4]. Boysen, N., Fedtke, S., andWeidinger, F. (2017). Truck scheduling in the postal service industry. Transportation Science, 51(2):723–736.

[5]. Boysen, N. and Fliedner, M. (2010). Cross dock scheduling: Classification, literature review and research agenda. Omega, 38(6):413–422.

[6]. Brucker, P., Drexl, A., Möhring, R., Neumann, K., and Pesch, E. (1999). Resource-constrained project scheduling: Notation, classification, models, and methods. European journal of operational research, 112(1):3–41.

[7]. Buhrkal, K., Zuglian, S., Ropke, S., Larsen, J., and Lusby, R. (2011). Models for the discrete berth allocation problem: A computational comparison. Transportation Research Part E: Logistics and Transportation Review, 47(4):461–473.

[8]. Crama, Y. and Spieksma, F. C. (1992). The complexity of scheduling short tasks with few starting times. Technical report, University of Limburg.

[9]. Feng, B., Li, Y., and Shen, Z.-J. M. (2015). Air cargo operations: Literature review and comparison with practices. Transportation Research Part C: Emerging Technologies, 56:263 – 280.

[10]. Hartmann, S. and Briskorn, D. (2010). A survey of variants and extensions of the resourceconstrained project scheduling problem. European Journal of operational research, 207(1):1–14.

[11]. Hellermann, R., Huchzermeier, A., and Spinler, S. (2013). Options contracts with overbooking in the air cargo industry. Decision Sciences, 44(2):297–327.

[12]. IATA (2015). IATA cargo strategy. https://www.iata.org/whatwedo/cargo/Documents/cargostrategy.pdf, last accessed 03/23/2017.

[13]. Kupfer, F., Meersman, H., Onghena, E., and Van de Voorde, E. (2017). The underlying drivers and future development of air cargo. Journal of Air Transport Management, 61:6–14.

[14]. Lenstra, J., Rinnooy Kan, A., and Brucker, P. (1977). Complexity of machine scheduling problems. Ann. of Discrete Math., 1:343–362.

[15]. Lin, C.-C., Lin, J.-Y., and Chen, Y.-C. (2012). The capacitated p-hub median problem with integral constraints: An application to a chinese air cargo network. Applied Mathematical Modelling, 36(6):2777–2787.

[16]. Mounie, G., Rapine, C., and Trystram, D. (1999). Efficient approximation algorithms for scheduling malleable tasks. In Proceedings of the eleventh annual ACM symposium on Parallel algorithms and architectures, pages 23–32. ACM.
[17]. Nakajima, K. and Hakimi, S. L. (1982). Complexity results for scheduling tasks with discrete starting times. Journal of algorithms, 3(4):344–361.
[18]. Neumann, K. and Zimmermann, J. (2000). Procedures for resource leveling and net present value problems in project scheduling with general temporal and resource constraints. European Journal of Operational Research, 127(2):425–443.
[19]. Nobert, Y. and Roy, J. (1998). Freight handling personnel scheduling at air cargo terminals. Transportation Science, 32(3):295–301.
[20]. Nudtasomboon, N. and Randhawa, S. U. (1997). Resource-constrained project scheduling with renewable and non-renewable resources and time-resource tradeoffs. Computers & Industrial Engineering, 32(1):227–242.
[21]. Rezanova, N. J. and Ryan, D. M. (2010). The train driver recovery problem—a set partitioning based model and solution method. Computers & Operations Research, 37(5):845–856.
[22]. Rong, A. and Grunow, M. (2009). Shift designs for freight handling personnel at air cargo terminals. Transportation Research Part E: Logistics and Transportation Review, 45(5):725–739.
[23]. Tadumadze, G., Boysen, N., Emde, S., and Weidinger, F. (2017). Truck scheduling with workforce dependent processing times. Working paper TU Darmstadt.
[24]. Totamane, R., Dasgupta, A., and Rao, S. (2014). Air cargo demand modeling and prediction. IEEE Systems Journal, 8(1):52–62.
[25]. Van den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E., and De Boeck, L. (2013). Personnel scheduling: A literature review. European Journal of Operational Research, 226(3):367–385.
[26]. W˛eglarz, J., Józefowska, J., Mika, M., andWaligóra, G. (2011). Project scheduling with finite or infinite number of activity processing modes–a survey. European Journal of Operational Research, 208(3):177–205.
[27]. Wong, W. H., Zhang, A., Van Hui, Y., and Leung, L. C. (2009). Optimal baggage-limit policy: airline passenger and cargo allocation. Transportation Science, 43(3):355–369.
[28]. Yan, S., Shih, Y.-L., and Shiao, F.-Y. (2008). Optimal cargo container loading plans under stochastic demands for air express carriers. Transportation Research Part E: Logistics and Transportation Review, 44(3):555–575.
[29]. Yan, S.-Y., Chen, C.-K., and Chen, C.-H. (2006). Cargo terminal shift setting and manpower supplying in short-term operations. Journal of Marine Science and Technology, 14(2):109–118.
[30]. Yeung, J. H. Y. and He, W. (2012). Shipment planning, capacity contracting, and revenue management in the air cargo industry: A literature review. In Proceedings of the 2012 International Conference on Industrial Engineering and Operations Management.

Paper 6 An integrated model to improve ergonomic and economic performance in order picking by rotating pallets

Authors: **Christoph H. Glock**, **Eric H. Grosse**, **Hamid Abedinnia**, **Simon Emde**

Type of publication: Journal article

Publication details: *European journal of operational research*: Under review

Abstract

This paper studies manual order picking activities in a warehouse where items are stored on pallets in two rows one above the other. Items stored on the pallets may be heavy, and manually picking the items may require excessive bending and stretching, which results in high spinal loads on the order picker associated with high injury risks. For this scenario that can frequently be observed in practice, this paper proposes an integrated model that supports the planning of order picking operations and pallet rotations taking account both of the time required for completing a set of orders as well as the spinal load on the order picker and the consequent injury risks. The results of a numerical experiment indicate that selectively rotating pallets may both reduce order picking time as well as the load on the order picker, leading to a quicker and less risky order picking process. The model proposed in this paper supports the decision of which pallet to rotate (and which not to rotate) against the company's cost objectives and its strive for worker well-being.

Keywords:
Order picking; pallet management; human factors; warehousing; ergonomics; injury risks; spinal load.

1 Introduction

The management of warehousing operations has enjoyed a high popularity in academic research for many years. The attention this research stream has received in the past originates from the high cost impact of warehousing, where order picking accounts for over 55 % of the operating costs (Frazelle 2002; Tompkins et al. 2010; Richards 2014). To support efficient warehousing operations, researchers have developed mathematical models that assist warehouse managers in finding efficient warehouse layouts or storage assignments or in managing retrieval and replenishment processes (see Section 2).

The primary objective of prior research on warehousing operations has been the reduction of travel time or travel distance. In manual order picking, for example, short travel distances can be achieved by calculating short routes for the order picker and/or by assigning products to the storage locations of the warehouse in such a way that frequently requested products are stored close to the depot or in aisles the order picker anyway needs to visit (e.g., Petersen and Aase 2004; Muppani and Adil 2008). In automated order picking (e.g., in AS/RS systems), short travel distances (which correspond to quick item access times) can be achieved by storing products close to the input/output point of the system and by re-sorting products during idle times if changes in the demand patterns occurred (Roodbergen and Vis 2009).

Especially in manual warehousing operations, an exclusive focus on time- and distance-based objectives does not necessarily match the requirements human workers directly or indirectly impose on the system (Grosse et al. 2017a). If items are assigned to storage locations solely with the objective to minimize travel distances, heavy and difficult to handle items may be stored in locations of the warehouse that are hard to access. This, in turn, may necessitate excessive bending and stretching on the part of the operator when retrieving these items from the warehouse, resulting in high loads on the operator's spine (Grosse et al. 2015). It is obvious that the latter may lead to fatigue and high injury risks for warehouse workers (Battini et al. 2016a), and in particular to risks of developing low back disorders. Low back disorders, which are the most common type of musculoskeletal disorders, have been shown to occur especially in risk environments where human workers have to move heavy and difficult to handle items in awkward body postures (Garg 2000; Baldwin 2004; Lavender et al. 2012). Clearly, order picking occurs in such a risk environment.

Injury risks as the ones just referred to are reflected in national statistics on occupational injuries. In the United States, for example, 356,910 cases of musculoskeletal disorders (such as strains or sprains resulting from overexertion in lifting) were reported in 2015, with warehousing being the sector with the highest rate of injuries in private sector industries (BLS 2016). In fact, the incidence rate in warehousing has even increased from 2014 to 2015 to 56,550 days-away-from-work cases, with a high number of musculoskeletal disorders cases (BLS 2016). The economic burden of musculoskeletal disorders is substantial (e.g., with an approximate share of 2.5% of the GDP in Europe and a GDP share of up to 5.7% in the US) and is likely to grow with an advancing demographic change (Coyte et al. 1998; Dagenais et al. 2008; Bevan 2015; HHS 2015).

Even though the manual handling of materials and its impact on human operators is an established and much-noticed stream of research in human factors engineering, it has surprisingly not received much attention in the management-oriented warehousing literature so far (Grosse et al. 2017a; 2017b). Just recently, researchers have started to integrate human factors aspects into mathematical decision support models of warehousing operations to balance the impact of warehousing on the cost objectives of the company and on the well-being of the human operator (e.g., Grosse and Glock 2015; Battini et al. 2016;

Larco et al. 2016; Glock et al. 2017). It is worth noting that prior research has shown that an improvement in the ergonomic conditions of a workplace may also improve worker performance (Neumann and Village 2012; Rose et al. 2013), such that there is not necessarily a trade-off between economic and social objectives.

The paper at hand addresses the manual picking of items from pallets in an order picking warehouse, which is one type of picking operations that very frequently occurs in practice. The scenario considered here assumes that pallets are stored in racks, with two levels of pallets stored above each other. Especially in case the warehouse worker has to pick items from the back part of the pallet, excessive bending and stretching become necessary, which may result in high peak and/or cumulative spinal loads on the order picker. Against this background, this paper investigates the effect that a simple rotation of pallets may have on the warehouse worker. Assuming that the front part of the pallet is more easily accessible to the order picker than the pallet's back part, rotating pallets by 180° after their front part has been emptied helps to reduce bending and stretching on the part of the order picker, which may lower cumulative and peak loads on the low back. Clearly, reduced loads on the order picker may lead to lower levels of fatigue and a lower injury risk for the worker (cf. Ma et al. 2009). Two important questions that arise in this context are, however, how the rotation of pallets interferes with the time- and/or travel distance-based objectives of the company, and whether rotating pallets leads to a positive net effect on the load level of the order picker, given that (manually) rotating the pallet may lead to an additional load. A third question that ultimately follows is which pallets should be rotated to facilitate picking items, and which pallets should be kept in their original position until they have been depleted. The paper at hand contributes to answering these research questions.

The remainder of the paper is structured as follows. The next section reviews the related literature, and Section 3 introduces a practical case that motivated this research. Section 4 describes the problem investigated in this paper in more detail, and Section 5 proposes a mathematical model that optimizes both the routing of order pickers, the sequencing of orders, and the scheduling of pallet rotations in a warehouse. Section 6 presents the results of a comprehensive numerical experiment, and Section 7 concludes the paper.

2 Literature review

To ensure efficient operations in manual order picking, several planning problems need to be solved. These planning problems can roughly be categorized into layout design, routing, storage assignment, and order batching (de Koster et al. 2007; van Gils et al. 2018). To support solving these planning problems in practice, researchers have developed mathematical models in the past whose main objective was the reduction of travel time or space cost. We give a brief overview of these planning problems in the following and then describe the role of human factors in order picking.

Layout design determines the size and shape of the order picking warehouse as well as the number and configuration of aisles and shelves (Roodbergen et al. 2008; Mowrey and Parikh 2014; Roodbergen et al. 2015). Although most researchers focused on rectangular warehouses – either with one (e.g., Petersen and Aase 2004; Thomas and Meller 2014) or more cross aisles (Vaughan and Petersen1999; Roodbergen and de Koster 2001) –, other layouts, such as U-shaped ones, have recently been discussed as well (e.g., Glock and Grosse 2012; Henn et al. 2013).

Routing methods determine the order picker's way through the warehouse and the sequence in which items are retrieved from the shelves. In the one-block rectangular warehouse, the order picker routing problem can be solved optimally in polynomial time as a special case of the Traveling Salesman Problem (Ratliff and Rosenthal 1983; Scholz et al. 2016). Other routing algorithms exist for the case of two or more cross aisles (Roodbergen and de Koster 2001; Theys et al. 2010). In practice, routing heuristics are frequently used that are in many cases more intuitive to the order picker than optimal routing, such as the well-known S-shape strategy, for example (de Koster and van der Poort 1998; Petersen and Aase 2004).

Storage assignment determines how items should be assigned to storage locations. The assignment can either be random or follow item characteristics such as the demand frequency, for example. In practice, items with a high demand frequency are often assigned to storage locations near the depot to reduce travel time (Petersen and Schmenner 1999). Another popular storage assignment method is class-based storage, where items with similar characteristics are assigned to different storage classes, which are then allocated to certain zones of the warehouse. Within each class area, items are assigned randomly to storage locations. The advantages of class-based storage have been highlighted in several studies (e.g., Muppani and Adil 2008; Chackelson et al. 2013; Rao and Adil 2013). In addition, correlated storage assignment, where items that are frequently demanded together are stored next to each other, can save travel time (Glock and Grosse 2012). Apart from that, Petersen et al. (2005) introduced the idea of verticality in manual order picking by considering the extra time it takes to pick items located on the top and lower shelf in a bin-shelving pick operation, which the authors denoted 'golden-zone storage'.

Order batching refers to the consolidation of several customer orders into a single picking order to reduce travel time. Although the order batching problem is NP hard, several algorithms exist that solve the problem in polynomial time under certain assumptions (Gademann and van de Velde 2005). Recently, various heuristic approaches for solving the problem have been developed (Henn and Wäscher 2012; Matusiak et al. 2014; Grosse et al. 2014). Closely related to order batching is zoning, where the warehouse is divided into several zones with an order picker being responsible for a specific zone (e.g., Yu and de Koster 2009).

Apart from the planning problems described above, several other problems that frequently occur in manual order picking warehouses have started to attract the attention of researchers. For example, some researchers have studied congestion in warehouses and the impact of the warehouse system design on the occurrence of blocking (Hong et al. 2012; Hong 2014; Franzke et al. 2017). The intention of works in this area is to develop policies that help to increase the robustness of the warehouse towards picker blocking.

Recently, researchers have started to investigate human factors and ergonomics aspects in warehousing (Grosse et al. 2015; 2017b), which is also the subject of the paper at hand. Interestingly enough, although the negative health impact of manual handling tasks in order picking is undisputed in the ergonomics literature, integrated approaches that take account both of the economic and the ergonomic performance of the warehouse are still rare (Grosse et al. 2017a). Besides considering health and safety issues in planning models, researchers have also called for integrating human factors into order picking models to improve performance and quality (Boysen et al. 2015; Grosse et al. 2015). This call has been addressed by a few works so far. For example, Grosse et al. (2013) and Grosse and Glock (2015) investigated the effect of human learning and forgetting on the performance and quality of order picking. Battini et al. (2016) integrated the concept of energy expenditure into storage assignment models to take into account physical effort, fatigue and discomfort as ergonomic indicators. A similar work is the one of Larco et al. (2016), who used Borg's scale to measure the ergonomic performance of the warehouse. Battini et al. (2017) used the concept of rest allowance to consider ergonomic conditions during order picking and their cost impact. Calzavara et al. (2017) developed mathematical models to evaluate different design options for pallet racks in an order picking zone using both economic (order picking time) and ergonomic (energy expenditure) performance measures. Finally, recent studies experimentally investigated behavioral issues in order picking and their effects on performance (de Vries et al. 2016a; de Vries et al. 2016b; Glock et al. 2017).

3 Case Study

This section presents a practical case that motivated the research at hand. Prior to developing the mathematical model proposed in this paper, we observed order picking operations at a large manufacturer of paint and enamels. The company operates an order picking warehouse with a physical size of 2500 m² and 1800 stock keeping units. The warehouse supplies products mainly to hardware stores, specialized retailers and customers ordering via the company's webshop. Items (mainly buckets, cans and boxes of different sizes and shapes) are stored on pallets in the company's warehouse, and the pick zone includes two layers of pallets stored above each other. Above the pick zone, additional pallets are kept to replenish the lower level pick zone. Items are picked directly from pallets. The order picker can access the lower pallet from the floor of the aisle; the upper pallet is stored approximately 1.60 meters above the floor of the aisle, such that a manlift (integrated in the pick truck in the case warehouse) is required to access this pallet. Figure 1 illustrates the order picking process in the case warehouse.

As can be seen in Figure 1, the order picker has to bend and stretch during the picking of items due to the dimensions of the shelf locations. When picking items from the back part of the pallets, the order picker has to adopt an even more critical body posture than during the picking of items from the pallet's front part. Especially when picking heavy and difficult to handle items, as is the case for some pick positions in the considered warehouse, picking from the back part of the pallets causes a serious injury risk to the order picker. Note that the paint buckets displayed in Figure 1 can weight up to 40 kg and only have a thin wire handle to allow the worker to pick the bucket in this particular application. In the case company described here, as well as in many other order picking warehouses in practice, the load on the order picker could be reduced by removing pallets whose front part has been depleted from the shelf, rotating them by 180°, and pushing them back into the shelf. The rotation of the pallet could be accomplished using a forklift truck, a hand pallet truck, or a specialized device such as the tool for rotating pallets described in Grosse et al. (2015). The impact of a rotation of pallets on the performance of the order picking warehouse and on the load on the order picker is analyzed in the following sections of this paper.

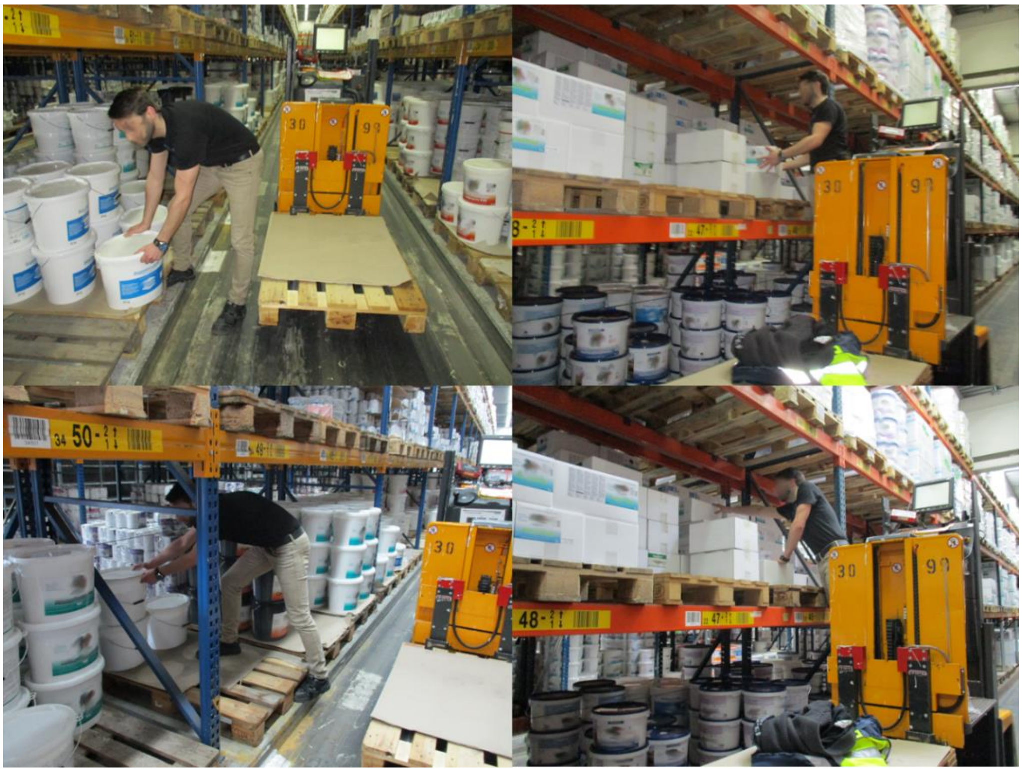


Figure 1: Order picking operations in the case warehouse

4 Problem description

In many order picking warehouses, products are directly picked from pallets or out of containers (e.g., Kadefors and Forsman 2000; Neumann and Medbo 2010; Calzavara et al. 2017). If the suppliers or the manufacturing department of the company deliver products on pallets to the warehouse, for example, then picking products directly from these pallets reduces the handling effort in the warehouse, as it is not necessary to move products from the pallets to the shelves of the warehouse first before they can be picked. In addition, picking products directly from pallets or out of containers may help to reduce investments in warehousing equipment, as fewer and less expensive shelves may be necessary to store the products in the order picking zone. Glock and Grosse (2012), for example, reported the case of a company that used two rows of stillages to form the shelves of the order picking zone, where the upper stillage was directly placed on top of the lower one. In this case, no shelves were necessary in the warehouse, and products could easily be replenished by exchanging stillages. In addition, the layout of the order picking zone could easily be changed if required, as no shelves that are usually fixed to the ground and dedicated to a certain layout were employed. The major disadvantage of this type of warehouse organization is low space utilization, which is usually higher in warehouses where classical racks are used.

If products are directly picked from pallets, then the challenge is to organize the picking process in such a way that no excessive bending, stretching and lifting is required from the order picker, and that simultaneously the costs of picking orders are minimized. This challenge obviously arises since pallets are designed to facilitate transporting the product, and not picking it. If pallets are placed directly on the floor of the warehouse or on a low platform, then the order picker has to bend to pick up items. In addition, the dimensions of the pallet and the storage location may require bending and stretching from the order picker especially to reach products stored on the back part of the pallet. Continuous bending, stretching and lifting of products, however, may lead to a high load on the order picker's musculoskeletal system, which may result in fatigue and/or injuries over time. In particular in the warehousing and storage sector, national statistics report that musculoskeletal disorders account for a high share of all work-related health problems, which renders order picking a high-risk environment (Eurostat 2009; BLS 2016).

In practice, several tools are available today that facilitate picking products from pallets. For example, some companies offer tools that gradually lift pallets as products are picked (and as the pallet's weight reduces) or that rotate pallets once their front part has been emptied (see for example Grosse et al. 2015). The problem associated with using such tools is that they can often only handle a single pallet and that one tool would be required for each pallet. For most applications, and especially for small and medium-sized companies, the associated investment cost is prohibitive.

The paper at hand adopts a different perspective and analyzes the case where pallets can be rotated manually or using a forklift truck once their front part has been emptied. The focus of this paper is on the ergonomic and economic performance impact of rotating pallets. If pallets are rotated after their front part has been emptied, products do not have to be picked from the back part of the pallet anymore, which reduces the load on the order picker and may also help to shorten the actual pick time. The primary research question of our paper is whether the additional time that is required for rotating the pallet, and the possible additional load on the order picker that may result from rotating the pallet, is offset by the positive effects that exclusively picking from the front parts of pallets brings about. Even though the focus of this paper is on the case where products are picked from pallets, the model developed below is also applicable to the case where products are picked out of stillages or containers, provided that the order picker can access products both from the front and back side of the stillage/container.

The model developed in this paper is inspired by the practical case presented in Section 3 and assumes a U-shaped order picking zone as described in Glock and Grosse (2012), Grosse and Glock (2013), and Henn et al. (2013), for example. The order picking zone is illustrated in Figure 2. As can be seen, we assume that the zone consists of two racks with n pallets each, and one rack with m pallets perpendicular to the other two racks. Each of the three segments of the U consists of two pallets, one on the upper and one on the lower level of the rack. We first number the pallets in the lower level of the rack clockwise as shown in Figure 2 and then continue with the upper level in the same manner. The depot, where each order picking tour starts and ends, is located at the open end of the U-zone. Note that Figure 1 shows one possible practical implementation of a U-shaped zone in practice; in the example, the pictures show the open end of the U-zone of a narrow, long U-zone.

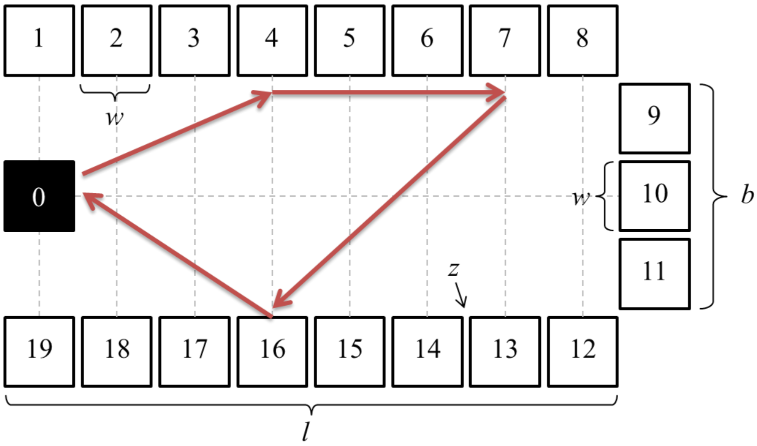


Figure 2: U-shaped zone studied in this paper with *n*=8 and *m*=3

Order picking in this zone works as follows: The order picker receives a customer order at the depot (denoted as "0" in Figure 2) and then starts retrieving items from the pallets until the order has been completed. If one of the pallets has been emptied, it is removed and replaced by a full pallet; the refill process, however, is not studied in this paper, as it is often done independently of the actual order picking process in practice (and in the case company studied in Section 3). The order picker travels along the aisle, possibly afoot pushing or pulling a cart or riding an electric vehicle as illustrated in Figure 1.

In the following, we consider two alternatives for rotating pallets. The first alternative is to use a forklift truck equipped with a special device for rotating pallets, e.g. the device described in Schäfer et al. (2009). This device enables the order picker to rotate the pallet without incurring an additional load; however, it would be necessary to purchase the pallet rotation device in this case. The second alternative considered here is the case where the order picker uses a simple hand pallet truck to rotate the pallet, and it is illustrated in Figure 3. In this case, the order picker pulls out the pallet from the shelf (Steps 1 and 2), moves the pallet truck to the opposite side of the pallet (Step 3) and pushes it back into the shelf again (Steps 4 to 6). Clearly, the second alternative requires a broader aisle than the first one to permit these operations, and it creates an additional load on the order picker. The following section proposes a formal model that optimizes both the sequencing of orders, the routing of the order picker as well as the scheduling of pallet rotation tours. As an economic performance measure, we use the time required to complete all orders, while as an ergonomic performance measure, we use the total peak load on the spine of the order picker. Both performance measures are described in detail in the next section.

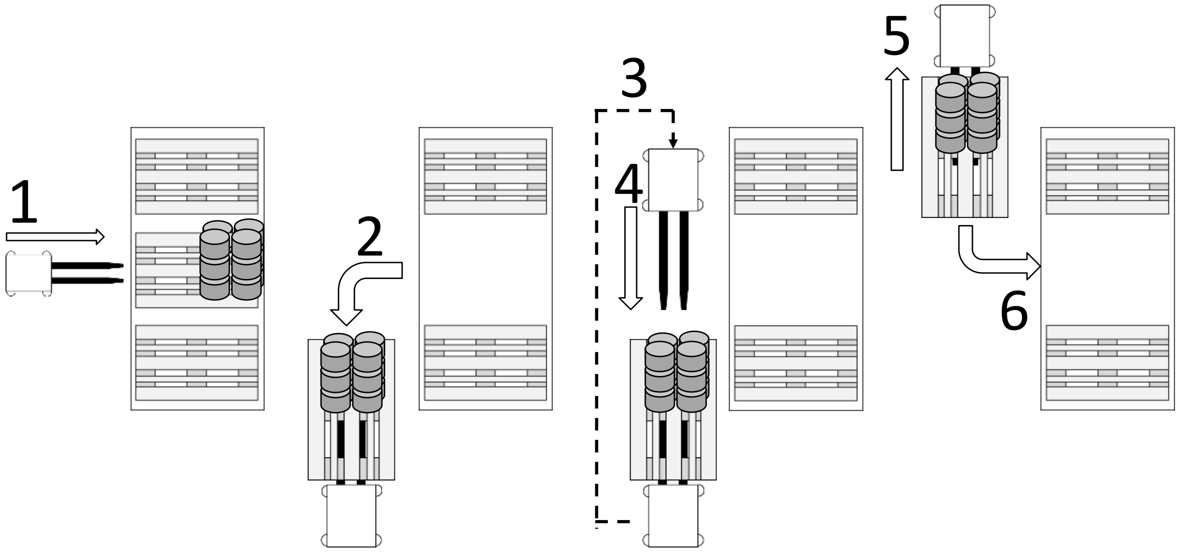


Figure 3: Graphical illustration of the pallet rotation process

5 The Model

5.1. Biomechanical modelling

For the assessment of spinal loading during order picking, a quasi-dynamic two-dimensional biomechanical model (4DWATBAK) was used. 4DWATBAK has been used by ergonomists in various field studies to assess risk factors of manual handling tasks, such as spinal shear, torso angle, or spine compression related to low back injuries (e.g., Daynard et al. 2001; Village et al. 2005; Neumann and Medbo 2010). In particular, the model allows estimating peak and cumulative loads on the lumbar spine, which are important risk indicators that have been validated in epidemiological studies (Norman et al. 1998; Neumann et al. 1999; Kerr et al. 2001; Cole and Grimshaw 2005). During field observations in the case company, data about body postures, pick time and item weight were collected, and the biomechanical model was set up for a male worker with standard measures (178 cm tall, 75 kg mass). Postures observed during picking items from pallets were entered into the biomechanical model (see also Neumann et al. 1999). Figure 4 exemplifies the mannequin from the biomechanical model for a) picking from the back part of the lower level pallet and b) picking from the back part of the upper level pallet (see also Garg 1986).

In our model, the peak L4/L5 spinal compression, which represents the compressive force acting upon the L4/L5 intervertebral joint, was recorded as biomechanical output and calculated as a risk indicator (Schultz and Andersson 1981; Kerr et al. 2001; Neumann and Medbo 2010; Daynard et al. 2001). This measure has been validated in the literature as a suitable indicator for risk of low-back injuries (NIOSH 1981; Norman et al. 1998; Daynard et al. 2001). To take account of item heterogeneity and the different positions of items on the pallet, the biomechanical analysis of peak L4/L5 spinal compression during picking was performed for different positions. The pallet was divided into two halves, front and back, as displayed in Figure 5 (see also Neumann and Medbo 2010). Spinal compression was recorded for each posture observed during the picking of items from the front and back part of the pallet. Depending on how many items are stored above each other, the spinal compression was calculated for the different item layers stored on the front and back part of the lower level pallet. The number of items that can be stored above each other is determined by the item weight and type. For the upper level pallet, only one posture analysis for the front and back part of the pallet was performed as the order picker is able to adjust the height from which the pick is performed with the help of the manlift. As a result, the order picker is able to bring the manlift into a position that allows picking items in upright body posture. Without loss of generality, we considered four different types of items (1 kg, 5 kg, 10 kg and 25 kg buckets) in our analysis to simplify computation. Tables 1 and 2 illustrate the different item characteristics and the biomechanical output for both lower and upper level pallets.

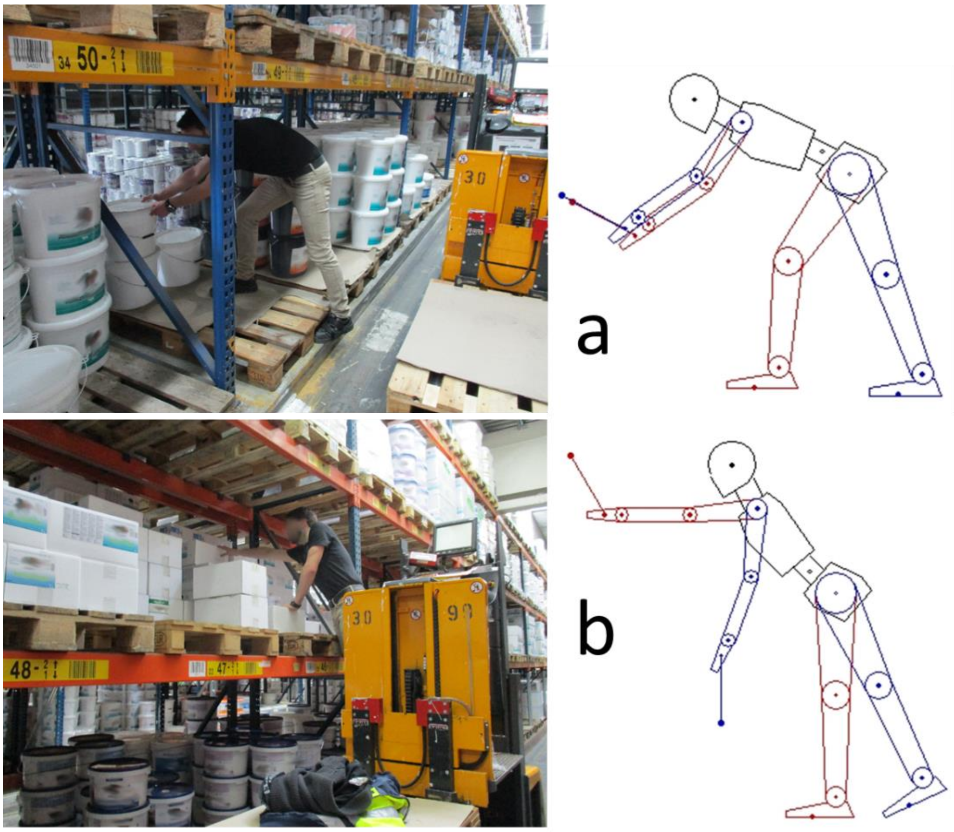


Figure 4: Mannequin from the biomechanical model illustrating example body postures during pick tasks

According to the literature, workers are at increased risk of low back injury if spinal compression exceeds 3400 N (Daynard et al. 2001; NIOSH 1981). The WATBAK model also reports a low back pain index, which represents the likelihood that the worker performing the task under analysis is a low back pain case (Norman et al. 1998). For example, picking the bottommost 25 kg bucket from the lower level pallet at the back leads to a peak spinal compression of 5469 N and a low back pain index of 0.72, which indicates that there is a probability of 72% that this worker would be rated as a low back pain case based solely upon the peak L4/L5 spinal compression (Norman et al. 1998). As can be seen in Tables 1 and 2, the biomechanical analysis revealed a higher peak L4/L5 spinal compression for picking from the lower level pallet than for picking from the upper level pallet. This is due to the fact that bending and stooping is necessary for picking from the ground floor, whereas the manlift allows picking in an almost upright posture on the first floor. In addition, picking from the back part of the pallet is more strenuous than picking from the front part due to the required stretching and straining in awkward postures. Note that solely the peak L4/L5 spinal compression was recorded as risk indicator. Other risk factors, such as potential slips or falls from the first floor pallet, were not considered.

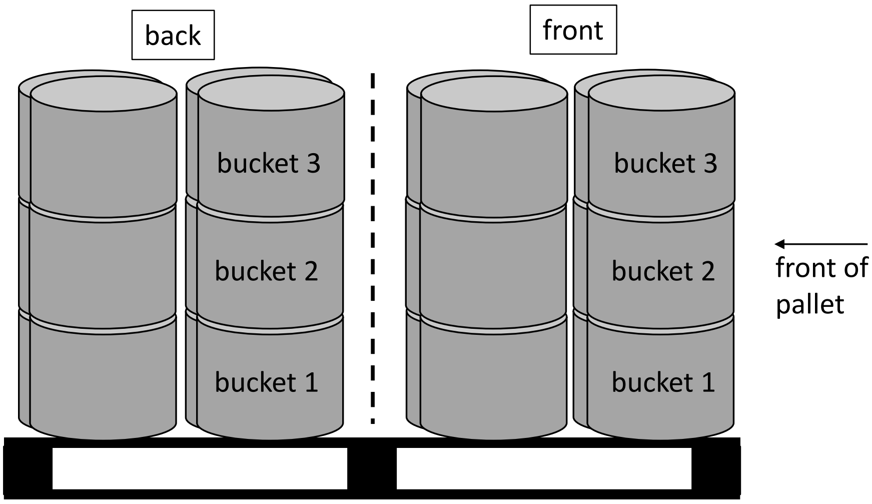


Figure 5: Example of a pallet with three buckets stored above each other (25 kg)

	Bucket weight 1 kg peak L4/L5 compression (N)		**Bucket weight 5 kg** peak L4/L5 compression (N)		**Bucket weight 10 kg** peak L4/L5 compression (N)		**Bucket weight 25 kg** peak L4/L5 compression (N)	
Bucket	front	back	front	back	front	back	front	back
1	1992	2288	2511	2809	2604	3368	3787	5649
2	2193	2404	2457	2875	2651	3650	3211	5320
3	2154	2493	2218	2998	2531	3468	2652	5245
4	1347	2487	1675	2679	2221	3334	-	-
5	1222	2542	1452	2742	-	-	-	-
6	1102	2439	-	-	-	-	-	-

Table 1: Item characteristics and biomechanical output for lower level pallets

item weight (kg)	Peak L4/L5 compression (N)	
	front	back
1	792	1899
5	899	2030
10	1032	2194
25	1432	2687

Table 2: Item characteristics and biomechanical output for upper level pallets

The peak load on the order picker when rotating a pallet was calculated based on the weight of the pallet that needs to be rotated also using a biomechanical model developed in 4DWATBAK, and it is summarized in Table 3. Peak L4/L5 spinal compression was calculated for the task of pulling out the pallet from the rack using a hand pallet truck (Step 2 in Figure 3), where the load was observed the highest for the pallet rotation process. Note that due to the mechanical advantage of the pallet truck used[1], the actual weight acting on the order picker is lower than the combined weight of the pallet and the pallet's load. The reduction in the required movement force resulting from the pallet truck's mechanical advantage depends on the type of device used.

Bucket weight (kg)	# items per pallet (H_i)	Weight of pallet (20 kg) and load of half-full pallet (kg)	Mechanical advantage: 0.0466 (kg)	Peak L4/L5 total compression (N)
25	24 (12)	320	14.9	1989
10	32 (16)	180	8.4	1538
5	60 (30)	170	7.9	1503
1	120 (60)	80	3.7	1187

Table 3: Peak load on the order picker for rotating pallets with different types of items

As an ergonomic performance measure, we calculate the total peak load on the spine of the order picker. The total peak load that results from picking item i, fo_i, depends on the number of times that the corresponding pallet i is rotated ($|Rot_i|$), the number of item layers stored on the pallet ($|l_i|$), as well as the number of picks from the front ($s = f$) and the back side ($s = b$) of layer l of pallet i ($|P_{sli}|$). The total peak load for picking items (Fo) can then be calculated as follows:

$$Fo = \sum_{i=1}^{N} fo_i = \sum_{i=1}^{N} (foR_i * |Rot_i| + \sum_{s=f,b} \sum_{l=1,\ldots,|l_i|} foP_{sli} * |P_{sli}|) \qquad (1)$$

here, foR_i is the compressive force on the L4/L5 intervertebral joint of the order picker for rotating pallet i (see Table 3), and foP_{sli} is the compressive force on the L4/L5 intervertebral joint of the order picker that results from picking a single item from layer level l of pallet i from the front (s=f) or the back (s=b) part of the pallet (see Tables 1 and 2).

5.2. The model of the order picking zone

In developing the proposed model, we assume the following:

1. Pallet locations are numbered as shown in Figure 1, where 0 denotes the depot. Each shelf has an upper (up) and a lower (lo) level, and one pallet is stored in each location.
2. We use Euclidean distances to estimate the travel distance between the depot and a pallet or between any two pallets, as we assume that this is the most intuitive way for the order picker to travel through the order picking zone:
   $$d_{ii'} = \sqrt{(x_i - x_{i'})^2 + (y_i - y_{i'})^2} \qquad (2)$$
   where $d_{ii'}$ is the distance between pallet i and i'.

[1] We use the term mechanical advantage here to refer to the fraction of the actual weight acting on the warehouse worker due to the technical attributes of the hand pallet truck.

Since Goetschalckx and Ratliff (1988a) have shown that the metric used for calculating travel distances in order picking only slightly influences system performance, rectangular distances, for example, could be used as well. This also implies that the triangle inequality holds.

3. Distances are measured from/to the centre of the depot and from/to the centre of the front side of the pallets.
4. There is a gap of width z between each two adjacent pallets to facilitate picking up the pallet, for example with a forklift truck. A part of the gap, or the entire gap, may be filled by vertical components of the racks.
5. The company follows a pick-by-order policy. The order picker uses some type of device, e.g. a hand cart or trolley, to transport items from the shelves of the U-zone to the depot. The device has a transport capacity that is always sufficient to complete the current picklist.
6. The problem of assigning pallets optimally to storage locations is not studied in this paper. We assume that pallets (product types) have been assigned randomly to available storage locations. We will relax this assumption later in the paper and study a weight-based assignment as well. For other policies for assigning products to storage locations, the reader is referred to Petersen and Schmenner (1999) or Petersen et al. (2005).
7. To rotate a pallet, the order picker uses a special device. As described in Section 4, this device can be, e.g., a forklift truck equipped for rotating pallets or a hand pallet truck. The device is only used after the picker has returned to the depot after finishing an order.
8. When setting off from the depot to rotate pallets, the picker rotates all pallets whose front part is empty.
9. When the pallet is emptied completely, the picker (or someone else) swaps it for a new one. There are always enough items in total in the warehouse to satisfy all orders, i.e., there are no stockouts.

The following terminology is used throughout the paper:

n	number of pallets in one of the two parallel shelves of the U-zone [#]
m	number of pallets in the shelve perpendicular to the two parallel shelves [#]
N	total number of pallets in the U-zone (i.e., pallets on the upper and lower level), with $N = 2(2n+m)$ [#]
Ψ	set of pallets with $\Psi = \{1,...,N\}$
$J = \{1, \dots, r\}$	set of orders that need to be processed during the planning horizon
$\Omega_j \subseteq \Psi$	set of pallets that need to be visited for order $j \in J$
Σ	schedule defining the sequence of jobs and the sequence of pallet rotation tours
w	width of a pallet [cm]
z	gap between two pallets [cm]
x_i	coordinate that measures the position of the depot or pallet i along the centre line of the U-zone [cm]
y_i	coordinate that measures the distance of the depot or pallet i from the centre line of the U-zone [cm]
H_i	capacity of the front and back part of pallet i [#]. The total capacity of the pallet is $2H_i$ [#]
$t_f^P(i)$	time required to pick an item from the front part of pallet i [sec]
$t_b^P(i)$	time required to pick an item from the back part of pallet i [sec]

$t_j^R(i)$	time required to rotate pallet i on the lower ($j = low$) or upper level ($j = up$) of the shelves [sec]
$d_{ii'}$	distance between pallets i and i' [cm]
u	time required to travel one centimetre in the warehouse [cm/sec]
$f^D(\Omega_j)$	travel time of the order picker for each order
$R_k \subseteq J$	set of orders that are processed without any rotation of pallets in-between orders
$f^P(R_k)$	total pick time for orders $j \in R_k$
$f^R(R_k)$	total rotation time for orders $j \in R_k$
foR_i	compressive force on the L4/L5 intervertebral joint of the order picker for rotating pallet i
foP_{sli}	compressive force on the L4/L5 intervertebral joint of the order picker for picking a single item from layer level l of pallet i from the front (s=f) or the back (s=b) part of the pallet
fo_i	total peak load for picking item i for all picklists
Fo	total peak load for the order picker for processing all orders
$F(\Sigma)$	total throughput time for a given set of orders J and a schedule Σ
$\lvert P_{sli}\rvert$	number of picks from the front ($s = f$) and the back side ($s = b$) of layer l of pallet i
$\lvert Rot_i\rvert$	number of times that pallet i is rotated
$s(i)$	weight of one unit of an item on pallet i [kg]

5.3. Minimizing the total picking effort – model

The optimization problem consists of the following decisions. Given a set of orders to be processed one after another (pick-by-order),

1. in what sequence should the orders be processed,
2. when should which pallet be rotated, and
3. what route should the picker take to travel from the depot to the pallets and back for each order?

Formally, let $J = \{1, \dots, r\}$ be the set of orders that need to be processed during the planning horizon, and let $\Omega_j \subseteq \Psi$ be the set of pallets that need to be visited for order $j \in J$. Each pallet $i \in \Omega_j$ is associated with a demand q_{ij}, which is the number of items that need to be picked for order j from pallet i, and a capacity H_i, which is the number of items that can be picked from the front part of the pallet before it has to be either rotated or accessed from the back. The back part of a pallet also has a capacity of H_i items, meaning that a pallet holds $2H_i$ items in total. Moreover, let $d_{ii'}$ be the distance from pallet i to pallet i' (or to/from the depot in case of d_{i0}/d_{0i}) as defined by Eq. (2). Each pick takes either $t_f^P(i)$ or $t_b^P(i)$, depending on whether the item can be picked from the front (f) or the back (b) part of the pallet. Note that the pick time also depends on the pallet i per se, specifically on whether it is on the lower or upper level.

After completing an order and returning to the depot, the picker has a choice of either starting on the next order from set J, or alternatively setting off on a tour through the warehouse to rotate pallets. A "rotation tour" starts and ends at the depot just like a "pick tour", hence the distances $d_{ii'}$ are calculated the same way. However, the time to rotate a pallet is given by $t^R(i)$. Note again that the rotation time can vary depending on pallet i.

When a pallet is completely empty, it is (immediately) swapped for a full one. Note that the time it takes to swap empty pallets for full ones is immaterial for our optimization problem as it can be assumed constant over the planning horizon: the number of times each pallet needs to be swapped depends solely on the total amount of items to be picked, regardless of the order sequence.

A schedule Σ is defined by a partition $\{R_1, \dots, R_K\}$ of set J, and a permutation π_k of R_k, $\forall k = 1, \dots, K$, signifying that the picker processes the orders in set R_1 in sequence π_1 first, then orders R_2 in sequence π_2, and so on. Every time the orders in one of the sets R_k are completed, the picker performs a rotation tour to rotate all pallets whose front part is empty. Note that a schedule Σ also implicitly determines when empty pallets are swapped. We say that such a schedule is optimal if it minimizes the overall throughput time, i.e., the last order should be finished as soon as possible. The objective value of a schedule Σ consists of three parts.

- The travel time of the picker for each order, which we denote as $f^D(\Omega_j)$,
- the pick time for the orders, which we denote as $f^P(R_k)$, and
- the time it takes to rotate pallets, which we denote as $f^R(R_k)$.

We only consider solutions feasible if $f^R(R_k) > 0$, $\forall k = 1, \dots, K$, i.e., there are no "empty" rotation rounds, where no pallet is actually rotated. Consequently, we minimize

$$F(\Sigma) = \sum_{j \in J} f^D(\Omega_j) + \sum_{k=1}^{K} \left(f^P(R_k) + f^R(R_k) \right) \tag{3}$$

We now describe how to calculate the individual parts of the objective function for a given schedule Σ. First off, under the assumptions laid out above, the travel time f^D of the picker can easily be determined for each order without solving any kind of optimization problem by considering the following proposition.

Proposition 1. *For a given order $j \in J$, an optimal route of the picker is to visit the pallets in Ω_j in clockwise order from the depot. Let $\langle 0, \omega_1^j, \dots, \omega_{|\Omega_j|}^j, 0 \rangle$ be the clockwise sequence of visits starting and ending at the depot. Then the optimal travel time of the picker is $f^D(\Omega_j) = \sum_{i=0}^{|\Omega_j|} d_{\omega_i^j, \omega_{i+1}^j}$.*

Proof. Considering the layout of the picking area as illustrated in Figure 2, the set of points $\{(x_j, y_j) \mid j \in \Omega_j\}$ to be visited by the picker for any order *j* always makes up a convex polygon. Barachet (1957, Theorem 3) showed that a route that corresponds to this convex polygon is of minimal length. Clearly, cycling in clockwise order through $\{(x_j, y_j) \mid j \in \Omega_j\}$ is equivalent to following a route along the convex polygon and thus optimal. □

Proposition 1 implies that the travel time $\sum_{j \in J} f^D(\Omega_j)$ in the objective is independent of the schedule Σ and can be calculated in polynomial time. The same is not true for the picking effort $f^P(R_k)$, which may vary between schedules because, depending on if and when pallets are rotated, the picker may have to either pick from the front or the back part of a pallet. Note, however, that the exact sequence of orders in R_k (i.e., π_k) is immaterial for the total pick time. Only the total number of items to be picked between refill events is important.

First off, in order to keep track of when pallets need to be swapped, we define η_i^k as the total number of items remaining on pallet i (both on the front and back parts) after orders R_k have been picked, i.e.,

$$\eta_i^k = \eta_i^{k-1} + 2 * H_i * \mu_i^k - \sum_{j \in R_k} q_{ij} \,;\ \forall i \in \Psi, \forall k \in \{1, \dots, r\} \tag{4}$$

For convenience, we define $\eta_i^0 \coloneqq 2H_i, \forall i \in \Psi$. The number of times μ_i^k that pallet i needs to be swapped while processing order set R_k is defined as

$$\mu_i^k = \begin{cases} 1 + \left\lfloor \frac{\sum_{j \in R_k} q_{ij} - \eta_i^{k-1}}{2 * H_i} \right\rfloor, \text{if } \sum_{j \in R_k} q_{ij} > \eta_i^{k-1} \\ 0, \text{otherwise} \end{cases} \tag{5}$$

Now, recall that our assumption is that if the picker makes a rotation tour, all pallets whose front part is empty must be rotated. Let γ_i^k be the number of items on the front part of pallet i after the orders from set R_k have been processed, i.e.,

$$\gamma_i^k = \begin{cases} \eta_i^{k-1} - \sum_{j \in R_k} q_{ij}, \text{if } \gamma_i^{k-1} = \mu_i^k = 0 \\ \max\{0;\ \gamma_i^{k-1} - \sum_{j \in R_k} q_{ij}\}, \text{if } \gamma_i^{k-1} > 0 \wedge \mu_i^k = 0 \\ \max\{0;\ \eta_i^k - H_i\}, \text{otherwise} \end{cases} \tag{6}$$

where $\gamma_i^0 \coloneqq H_i, \forall i \in \Psi$. Moreover, let

$$\breve{\Upsilon}_i^k = \begin{cases} \eta_i^{k-1}, \text{if } \gamma_i^{k-1} = 0 \\ \gamma_i^{k-1}, \text{otherwise} \end{cases} \tag{7}$$

be the number of items on the front part of pallet i after the rotation tour between order sets $k-1$ and k. We can then define the number of items picked from the front part of pallet i for order set k as

$$\zeta_i^k = \begin{cases} \min\{\sum_{j \in R_k} q_{ij}; \breve{\Upsilon}_i^k\}, \text{if } \mu_i^k = 0 \\ \breve{\Upsilon}_i^k + \min\{\mu_i^k * H_i; \sum_{j \in R_k} q_{ij} - \eta_i^{k-1} - (\mu_i^k - 1) * H_i\} \text{ otherwise} \end{cases} \tag{8}$$

The number of items picked from the back part of the same pallet is then $\sum_{j \in R_k} q_{ij} - \zeta_i^k$. The picking effort for one block of orders thus equals

$$f^P(R_k) = \sum_{i \in \Psi} (\zeta_i^k * t_f^P(i) + \left(\sum_{j \in R_k} q_{ij} - \zeta_i^k\right) * t_b^P(i)) \tag{9}$$

This leaves the last term of the objective function, minimizing the time it takes to rotate pallets. Let $\Gamma(k)$ be the set of pallets to be rotated in-between the processing of orders R_k and R_{k+1}, i.e.,

$$\Gamma(k) = \{i \in \Psi | \gamma_i^k = 0\} \tag{10}$$

Note here again that we only consider solutions feasible where $\Gamma(k) \neq \emptyset, \forall k = 1, \dots K$, i.e., each rotation tour must have at least one rotated pallet. Then the total rotation effort for the rotation tour between order set R_k and order set R_{k+1} is

$$f^R(R_k) = \sum_{i \in \Gamma(k)} t^R(i) + f^D(\Gamma(k)) \tag{11}$$

where $f^D(\Gamma(k))$ denotes the travel time for visiting the pallets to be rotated on the optimal route as per Proposition 1.

To illustrate the calculation of the objective function, consider the following small-sized example with $m = n = 1$ and $N = 6$. Table 4 introduces the details of the items stored on these six pallets.

Pallet (i)	H_i	$t_f^P(i)$	$t_b^P(i)$	$t^R(i)$
1	4	6	10	4
2	6	4	8	4
3	2	10	20	4
4	3	8	15	4
5	4	6	10	4
6	2	10	20	4

Table 4: Sample item data used for the illustrative example

We further assume that the order picker needs to collect items according to the three pick lists introduced in Table 5.

Pick lists (j)	q_{1j}	q_{2j}	q_{3j}	q_{4j}	q_{5j}	q_{6j}
1	5	-	3	-	2	-
2	3	4	2	6	2	3
3	1	4	-	-	1	-

Table 5: Sample pick lists used for the illustrative example

We now calculate the total pick and pallet rotation time for the following solutions: $\Sigma_1 = \{\{1,3\}, \{2\}\}$ and $\Sigma_2 = \{\{1,2,3\}\}$. Obviously, solution Σ_2 does not consider pallet rotations in-between orders. Tables 6 to 8 summarize calculations according to Eqs. (3) to (11). To simplify computing and displaying results, we neglect the travel time for rotation tours and for picking items. First, as explained before, the actual travel time for picking items plays no role in the proposed optimization model. Secondly, the effect of the total travel time for rotating pallets in a small-sized example with only three orders and six items is minimal. Therefore, this simplification does not have a big effect on the final result. The total time required for picking items and rotating the pallets for Σ_1 is 325 (with pallet rotation) and 345 for Σ_2 (no pallet rotation).

Σ_1					$k = 1$					
Pallets (i)	$\sum_{j\in R_k} q_{ij}$	η_i^k	μ_i^k	γ_i^k	$\check{\Upsilon}_i^k$	ζ_i^k	$\sum_{j\in R_k} q_{ij} - \zeta_i^k$	$f^P(R_k)$	$t^R(i)$	Total time
1	6	2	0	0	4	4	2	44	4	48
2	4	8	0	2	6	4	0	16	0	16
3	3	1	0	0	2	2	1	40	4	44
4	0	6	0	3	3	0	0	0	0	0
5	3	5	0	1	4	3	0	18	0	18
6	0	4	0	2	2	0	0	0	0	0
							total	118	8	126

Table 6: Results for $\mathbf{\Sigma_1}$ and *k*=1 for the illustrative example

Σ_1					$k = 2$					
Pallets (i)	$\sum_{j\in R_k} q_{ij}$	η_i^k	μ_i^k	γ_i^k	$\check{\Upsilon}_i^k$	ζ_i^k	$\sum_{j\in R_k} q_{ij} - \zeta_i^k$	$f^P(R_k)$	$t^R(i)$	Total time
1	3	7	1	3	2	3	0	18	0	18
2	4	4	0	0	2	2	2	24	4	28
3	2	3	1	1	1	2	0	20	0	20
4	6	6	1	3	3	3	3	69	0	69
5	2	3	0	0	1	1	1	16	4	20
6	3	1	0	0	2	2	1	40	4	44
							total	187	12	199

Table 7: Results for $\mathbf{\Sigma_1}$ and *k*=2 for the illustrative example

Σ_2					$k = 1$					
Pallets (i)	$\sum_{j\in R_k} q_{ij}$	η_i^k	μ_i^k	γ_i^k	$\check{\Upsilon}_i^k$	ζ_i^k	$\sum_{j\in R_k} q_{ij} - \zeta_i^k$	$f^P(R_k)$	$t^R(i)$	Total time
1	9	7	1	3	4	5	4	70	0	70
2	8	4	0	0	6	6	2	40	4	44
3	5	3	1	1	2	3	2	70	0	70
4	6	12	1	6	3	3	3	69	0	69
5	5	3	0	0	4	4	1	34	4	38
6	3	1	0	0	2	2	1	40	4	44
				total				333	12	345

Table 8: Results for $\mathbf{\Sigma_2}$ for the illustrative example

Figure 6 illustrates the inventory level on the front and back part of the pallets for the example illustrated in Tables 6 and 7 (solution Σ_1). In Figure 6, “a” is the inventory level at the beginning of the planning horizon, “b” the inventory level after finishing first order set, “c” the inventory level after the rotation tour in-between the two order sets has been concluded, and finally, “d” the inventory level after the second order set has been completed.

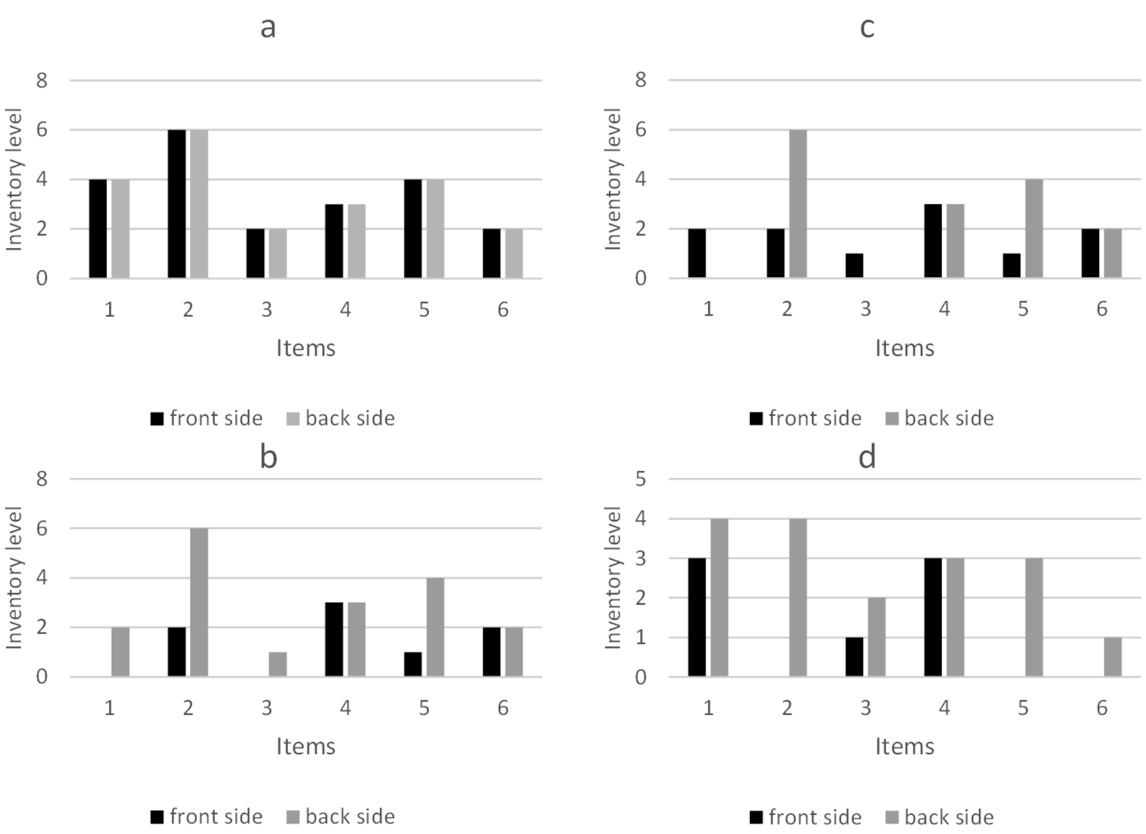


Figure 6: Inventory level of front and back part of the items for a given solution over time

By Proposition 1, the picker routing problem is not computationally hard under our assumptions. However, this is not true for the problem of determining the optimal pallet rotation schedule, as we show in the following.

Proposition 2. Determining an optimal schedule Σ is NP-hard even if there is only one pallet location, i.e., $|\Psi| = 1$.

Proof. We show NP-hardness by reduction from PARTITION, which is well-known to be NP-hard (Garey and Johnson, 1979). An instance of PARTITION is defined as follows. Given e positive integers $a_g (g = 1, \dots, e)$, does there exist a partition $\{A_1, A_2\}$ of the set $\{1, \dots, e\}$ such that $\sum_{g \in A_1} a_g = \sum_{g \in A_2} a_g$?

We transform an instance of PARTITION to an instance of our problem by considering $r = e$ orders and only one pallet, i.e., $|\Psi| = 1$. Each order requires that $q_{1,g} = a_g$, $\forall g = 1, \dots, e$, items be picked from this pallet. The front part of the pallet contains $H_1 = \sum_{g=1}^{e} a_g / 2$ items. We set the picking time from the back part of the pallet to a prohibitively high value, e.g., $t_b^P(1) = \infty$, and the picking speed from the front part of the pallet to a low value, e.g., $t_f^P(1) = 0$. Moreover, the pallet rotation time can assume any value greater than 0, i.e., $t^R(1) > 0$. The travel speed can be any arbitrary value; for simplicity's sake assume that $d_{0,1} = d_{1,0} = 0$. The corresponding instance of PARTITION is a YES-instance if and only if there exists a schedule with $F(\Sigma) \leq t^R(1)$.

Since picking from the back part of the pallet is prohibitively time consuming, the pallet must be rotated before the picker would need to access the back part. In other words, the total demand $\sum_{j\in R_1} q_{1,j}$ before the first (and only) rotation round must be exactly H_1 – otherwise, there would either be no rotation (because only pallets with an empty front part are rotated), or the picker would have to pick from the back part. Moreover, there cannot be more than $K = 2$ order sets R_k in total because the total effort for rotation rounds must be not greater than $t^R(1)$, i.e., there can be only one single rotation tour. It is thus clear that set R_1 must contain orders whose demand $\sum_{j\in R_1} q_{1,j}$ sums up to exactly $H_1 = \sum_{g=1}^{e} a_g/2$. The correspondence with PARTITION is thus obvious. □

5.4. Minimizing the total picking effort – algorithm

By Proposition 2, it is unlikely that a default solver will be able to solve instances of realistic size in acceptable time, which is confirmed by our computational experiments (see Appendix; in our experiments, the default solver was not able to find a solution in one day for problems with 9 or more picklists). To solve large problem instances in reasonable time, we therefore propose a simulated annealing (SA) approach. Simulated annealing, originally proposed by Kirkpatrick et al. (1983), is a metaheuristic modelled after the physical process of cooling a material that has previously been heated above its recrystallization temperature. It has often proven successful in solving difficult scheduling problems (e.g., Van Laarhoven et al., 1992; Bouleimen and Lecocq, 2003; Kim and Moon, 2003).

A solution in our SA is encoded as a vector $\rho = [R_1, \dots, R_K]$ of order sets. All orders in a set R_k are processed without interruption (i.e., without any pallet rotation tours in-between); however, each time the picker finishes all orders from a set R_k, he/she inserts a pallet rotation tour to rotate all pallets in the set R_k whose front part is empty. This information is sufficient to evaluate the objective function (3) for any given vector ρ. To decode ρ to an implementable solution Σ, orders within each set R_k can simply be processed in any arbitrary sequence; this does not influence the overall throughput time. We define $\mathcal{F}(\rho)$ as the objective value as calculated by Eq. (3) for a given ρ. Note that the travel time $f^D(\Omega_j)$, $\forall j \in J$, can be calculated and stored in preprocessing and is then available for all evaluations of $\mathcal{F}$.

We initialize $\rho := [R_1]$ such that it contains only a single order set $R_1 := J$ containing all orders, with no pallet rotation scheduled. A neighbor ρ' of the current incumbent solution ρ is reached by either randomly *pushing* one order from one set R_k to another set $R_{k'}$, $k \neq k'$, or by *swapping* one randomly selected order from set R_k with another order from set $R_{k'}$, $k \neq k'$. Note that for the *push* neighborhood search, the target set $R_{k'}$ may also be a new set that has not previously existed, inserted into vector ρ at some random position $k' \in \{1, \dots, K+1\}$. Similarly, if the last order is pushed from some set R_k, R_k is removed from ρ'. Among the push- and swap-based neighbors, we set ρ' equal to the one that leads to lower $\mathcal{F}(\rho')$. If $\exp\big((\mathcal{F}(\rho) - \mathcal{F}(\rho'))/T\big) > \mathrm{rnd}(0{,}1)$, ρ' is accepted as the new incumbent for the next iteration, i.e., $\rho \coloneqq \rho'$, where $\mathrm{rnd}(0,1)$ is a uniformly distributed random number from the interval in the argument and T is the current temperature. Initially set to $T \coloneqq f^D(\Psi)$, the temperature is lowered every 50 iterations to $T \coloneqq 0.995T$. Once T drops below 0.01, the search stops, and the best found solution is returned. As shown in the Appendix, the SA algorithm found the optimal solution for all small-sized problem instances we investigated, at a 60% lower run-time as compared to the default solver. These results indicate that the SA approach may lead to good solutions.

6 Numerical experiments

This section analyses the influence of a rotation of pallets on the economic and ergonomic performance of an order picking warehouse using the objectives defined in Section 5. In our numerical experiments, we consider a U-zone with 60 pallets in total, where half of the pallets are stored in the lower level of the shelves and the other half in the upper level with $n = 11$ and $m = 8$ determining the layout of the U-zone. We assume standard EURO pallets with $w = 120$ cm. Some data describing the rotation of pallets were taken from Schäfer et al.'s (2009) field study of a pallet rotation device attached to a forklift truck. The authors found that rotating a pallet using this device takes approximately one minute. Our own observations in the case company described in Section 3 showed that approximately the same time is required to rotate pallets on the lower level using a hand pallet truck. Since rotating pallets on the upper level leads to an additional vertical travel of the order picker, we considered an extra 15 sec for rotating pallets on upper shelves, i.e., we assume $t_{up}^R(i) = 75$ and $t_{low}^R(i) = 60$ for both devices. Schäfer et al. (2009) further observed that the time required to empty the front half of a pallet consumes approximately 70% of the time required to empty the back part (see Neumann and Medbo (2010) for similar results). Based on our observation of pick times, the time needed to pick up an item depends on the weight of the item, the level of the pallet (upper vs. lower level) and the part of the pallet the item is stored on (front vs. back). Table 9 summarizes the parameters used for the computational experiment.

Weight of items (kg)	H_i	$t_{f-low}^P(i)$	$t_{f-up}^P(i)$	$t_{b-low}^P(i)$	$t_{b-up}^P(i)$	$t_{low}^R(i)$	$t_{up}^R(i)$
1	60	2	17	4	19	60	75
5	30	5	20	8	23	60	75
10	16	10	25	15	30	60	75
25	12	20	35	30	45	60	75

Table 9: Parameters used for the computational experiment

The remaining parameters are based on our observations in the case company, and they are assumed as follows: $z = 30$ cm, $u = 80$ cm/sec. To simulate demand, 1000 picklists were generated randomly based on our observations at an industry partner. The weight of the items on each pallet were selected randomly from the set {1, 5, 10, 25}. The number of items contained on a picklist was generated using U~(1,10). The required quantity per picklist for the different items was generated using U~(1,8) (1 kg), U~(1,6) (5 kg), U~(1,4) (10 kg) and U~(1,2) (25 kg).

The model proposed in this paper aims on optimizing the total time needed to finish all 1000 picklists (economic objective) as calculated in Eq. (3). In addition, we evaluate the effect of the obtained solution on the total peak load acting on the order picker (ergonomic objective) as calculated in Eq. (1). Figure

7 illustrates the total order completion time per picklist for alternative pallet rotation strategies. In Figures 7 to 12, the abscissa displays the minimum item weight for pallet rotation. For each of the pallet rotation strategies, items that weigh at least the indicated weight are rotated, (e.g., "≥10 kg" means that all pallets with items weighting 10 kg or more are rotated), with "n.R." indicating the case of "no rotation". Figure 7 demonstrates that rotating all pallets would improve the economic objective by 2.05% compared to the n.R. strategy. The average time needed for rotating pallets is identical for both automated and manual pallet rotation cases, as the time needed to rotate a pallet was assumed identical in both cases. Figure 7 also shows that including 25 kg and 1 kg items in the rotation tours leads to a stronger reduction in the total order completion time than including 10 kg and 5 kg items, with 10 kg impacting the total order completion time the least. This result is due to the different net improvement that can be obtained from rotating the different pallets (total pick time saved minus pallet rotation time, see Table 9). The net improvement is 30 sec for 5 kg and 10 kg items in our example, while it is 60 sec for 1 kg and 25 kg items. In addition, the average demand for the different items influences this result, which was assumed higher for 5 kg items than for 10 kg items.

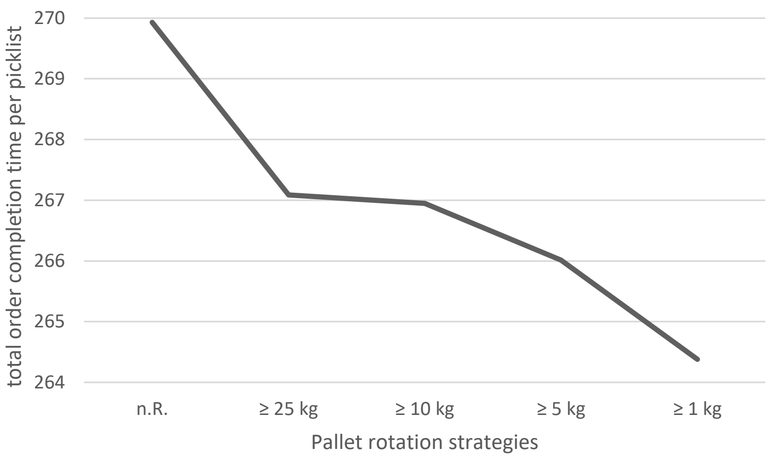


Figure 7: Average time needed to process one pick list for different pallet rotation strategies

To evaluate the effect of an eventual rotation of pallets on the ergonomic objective, we first study the case where rotating pallets does not lead to an additional load on the order picker (i.e., the case where an automated pallet rotation device is used). Figure 8 shows the average load per picklist and the average load per unit of time for different pallet rotation strategies. Note that these two load measures were selected as the different pallet rotation strategies can lead to different order processing times, which would distribute the cumulative peak load over different time intervals. The average load per picklist was calculated by computing the total load on the order picker according to Eq. (1) and by then dividing it by the number of pick lists (1000 in this example). The average load per unit of time, in contrast, was calculated by dividing the total load on the order picker according to Eq. (1) by the total time required to complete all picklists using Eq. (3). As expected, if rotating pallets does not lead to an additional load on the worker, rotating pallets always reduces the load on the order picker and thus improves the ergonomic assessment of the workplace. As can be seen, rotating all pallets once their front part has been depleted leads to a reduction of the average load per picklist on the order picker of 24.54%, and to a reduction of the average load on the order picker per unit of time of 22.95% as compared to the situation where pallets are not rotated at all.

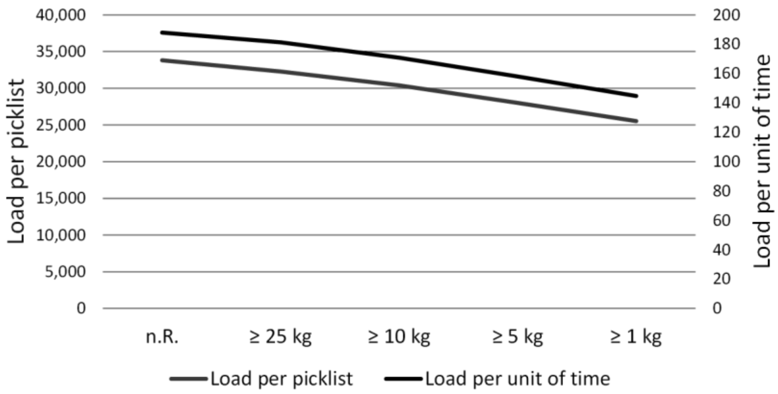


Figure 8: Average load on the order picker for different pallet rotation strategies and automated pallet rotation

Figure 9 illustrates the average time per picklist for different pallet rotation strategies and alternative time differences for picking from the front and the back part of a pallet (referred to as $\Delta_i = t_f^P(i) - t_b^P(i)$ hereafter). Clearly, the quicker the order picker can pick items from the front part of a pallet as compared to items from the pallet's back part, the lower is the average time per picklist for any of the pallet rotation strategies. As expected, for small values of Δ_i, rotating pallets would not be beneficial from a short-term (time-based) economic point of view, as it would lead to an increase in throughput time. Starting from $\Delta_i = 4$, rotating pallets improves the economic objective for the ≥1 kg pallet rotation strategy. A further increase in Δ_i would also make the other rotation strategies profitable.

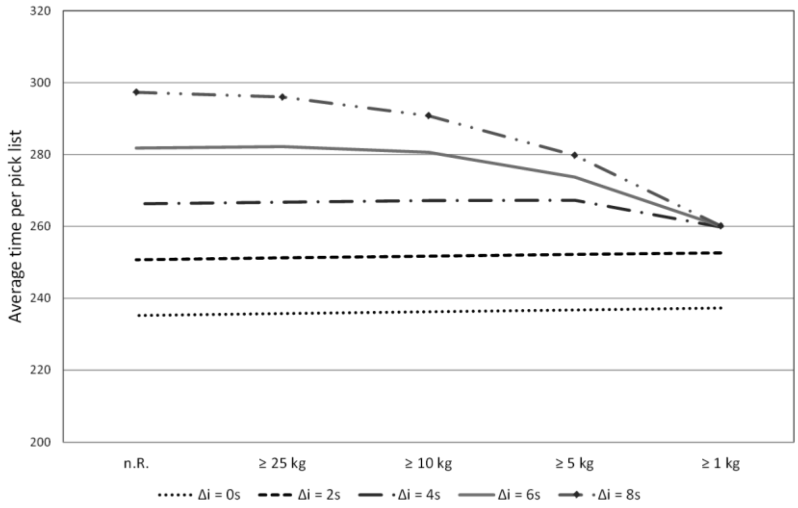


Figure 9: Average time per picklist for different values of Δ_i

We next study the case where rotating pallets leads to an additional load on the order picker (i.e., the case of a manual rotation of pallets). As can be seen in Figure 10, the behavior of the pallet rotation strategies is similar as in the case where rotating pallets does not lead to an additional load on the order picker. Nevertheless, moving from the case where pallets are not rotated ("n.R.") to the case where all pallets are rotated once their front part has been depleted ("≥ 1 kg") still reduces the load on the order picker by 22.65% per pick list and by 21.02% per unit of time.

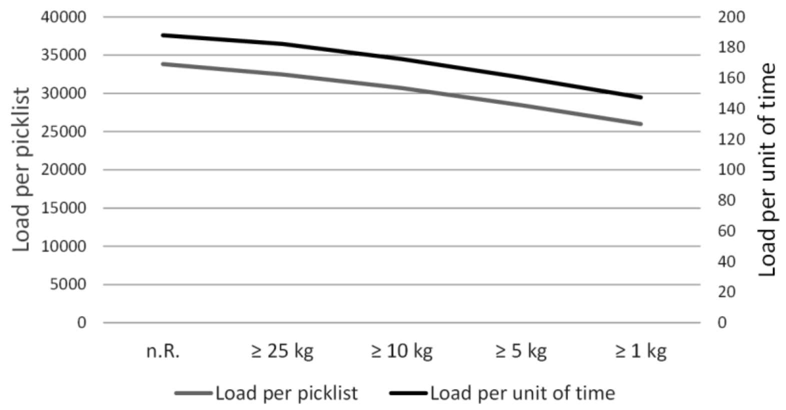


Figure 10: Average load on the order picker for different pallet rotation strategies and manual pal-let rotation

A further aspect we analyze is the number of dangerous picks that have to be performed by the order picker. As explained in Section 5, a pick may be considered dangerous for the order picker if the spinal compression associated with the pick exceeds 3400 N. Figure 11 illustrates that for the pick lists considered in the computational experiment, rotating only the heaviest items (this corresponds to the "≥ 25 kg” strategy) reduces the number of dangerous picks by 29.33% as compared to the case where pallets are not rotated at all (“n.R.” strategy). If we rotate both 25 kg and 10 kg items (this corresponds to the “≥ 10 kg” strategy), the number of dangerous picks would be reduced by 70.58%. As illustrated in Tables 1 and 2, picking from different levels and from the back or front part of the pallet does not lead to a load that exceeds 3400 N for 1 kg and 5 kg items. Thus, rotating them would not reduce the number of dangerous picks as compared to the other two types of items.

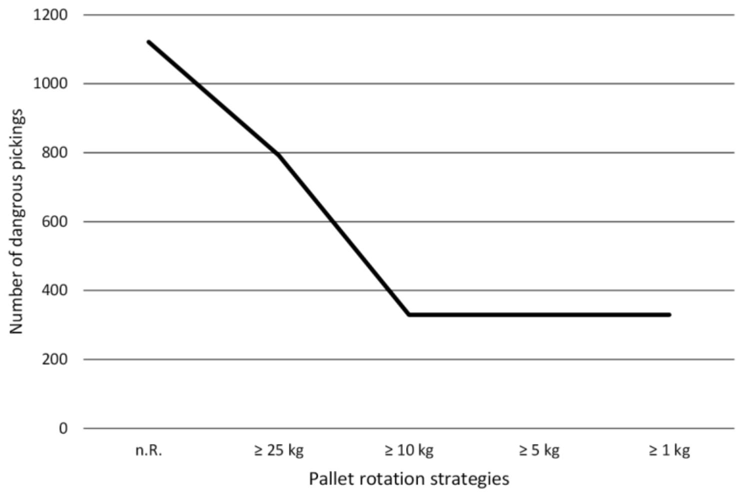


Figure 11: Number of dangerous picks for different pallet rotation strategies

To reduce the load on the order picker, the company could also decide to assign heavy items to storage locations that facilitate picking the products. In the example studied in this paper, items on the upper level can be picked in an upright standing position by the worker, which leads to lower loads on the order picker (see Tables 1 and 2). In the following, we therefore assume that heavy items are assigned to the upper level of the rack (denoted as “hi up” in Figure 12 and 13), while light items are assigned to the lower rack level. In the present example, the assignment procedure starts with assigning 25 kg items to the upper rack level starting with the position closest to the depot and then continues with 10 kg items etc. In the following, this new weight-based assignment method is compared to the “standard” assignment in which the items are assigned randomly to the storage positions.

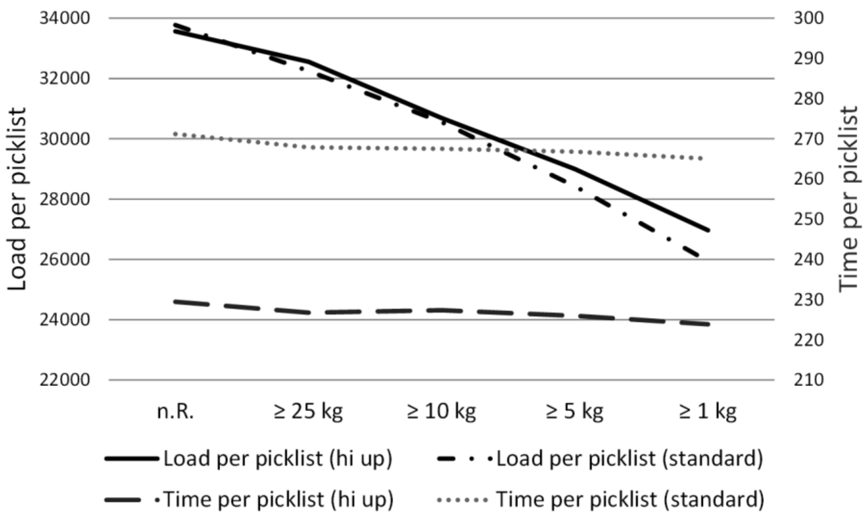


Figure 12: Load and time per picklist for the standard and the weight-based assignment

By storing heavy items in upper shelves, we expect to reduce the overall load on the order picker. However, as Figure 12 illustrates, our numerical experiment revealed that, for the considered problem parameters and for almost all pallet rotation strategies, the hi-up storage assignment increased the load on the order picker as compared to the standard assignment. The highest increase in average load per picklist was realized for the "≥ 1 kg" pallet rotation strategy. Figure 12 also illustrates that changing the storage assignment would consistently lead to a considerable decrease in the average time per picklist for all pallet rotation strategies. The reduction in average time ranged between 15.03% and 15.53%. The highest reduction in the average time per picklist was realized for the "≥ 1 kg" pallet rotation strategy. These counterintuitive results can be explained by the problem parameters used for generating test instances in the numerical experiment. As explained above, it was assumed that light items are demanded more frequently than heavy item, with the demand for 1 kg items being, on average, four times higher than the demand for 25 kg item. Even though the hi-up storage assignment reduces the load on the order picker for heavy items, it increases the load on the order picker for light items that are required much more often than the heavy ones. The net effect on the load the order picker is exposed to is negative in the present example. The same effect explains the difference in average pick times.

The results obtained for the weight-based storage assignment consequently change if the demand structure for the items is adjusted. Figure 13 illustrates the average load on the order picker as well as the average time required for completing a pick list for the case where the demand for all items was generated randomly from U~(1,4). The weight-based storage assignment now consistently outperformed the standard assignment, with reductions in average load ranging between 5.3% and 8.56%. Figure 11 also shows that in this example, the change in the storage assignment entailed a reduction in the average time per picklist that ranged between 1.56% and 1.68%.

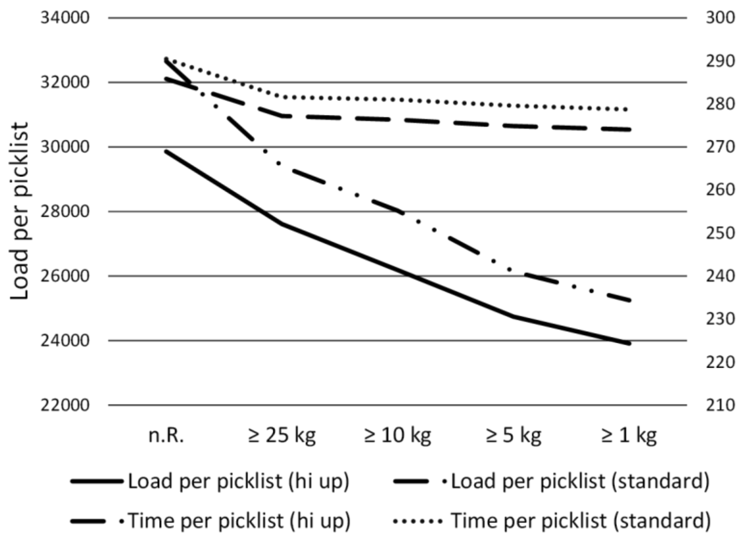


Figure 13: Load and time per picklist for the standard and the weight-based assignment for an adjusted item demand structure

The question of whether rotating a particular pallet improves the time objective of order picking solely depends on $\Delta_i = t_f^P(i) - t_b^P(i)$, the time to rotate a pallet $t^R(i)$, and the number of items stored on the pallet, $2H_i$. If the time difference for picking items from the front and the back part of the pallet is $\Delta_i =$ 2 seconds, for example, and if the pallet fill quantities assumed in Table 8 are given with $t_{low}^R(i) = 60$ and $t_{up}^R(i) = 75$, then rotating 10 kg and 25 kg pallets would result in an increase in order picking time. Considering the problem's parameters displayed in Table 9, which are all based on our observations in practice, we have $\Delta_i * H_i > t_{up/low}^R(i)$ for all items, and therefore rotating all pallets can potentially improve the economic objective.

We now assume that $t_{low}^R(i) = 100$ and $t_{up}^R(i) = 115$, which results in a situation where only rotating 1 kg and 25 kg items can potentially reduce order picking time. A straightforward strategy for the company in this new problem setting could be to only rotate pallets whose rotation leads to a direct improvement of the time objective of order picking, i.e. 1 kg and 25 kg items. This strategy is referred to in the following as "rot. 1&25 kg".

To gain insights into how the new problem setting influences the pallet rotation strategies, we first compare the average number of rotations for the different items in both scenarios. As Figure 14 illustrates, forcing the system to rotate pallets that worsen the time objective, i.e. 5 kg and 10 kg items in this example, would dramatically reduce the total number of pallet rotations for all items. Excluding these items from pallet rotation tours (cf. the "≥ 25 kg" and "rot. 1&25 kg" pallet rotation strategies) leads to a high number of pallet rotations again.

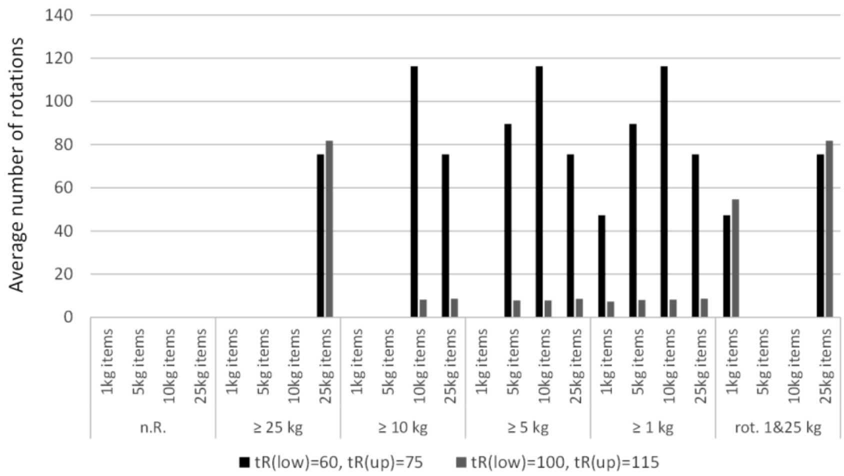


Figure 14: Average number of rotations for different items considering different rotation strategies and under different pallet rotation times

As Figure 15 illustrates, with the new parameter setting, all rotation strategies would slightly increase the time objective, making pallet rotations not worthwhile from an economic point of view. Note that we assumed that for all pallet rotation strategies excluding “n.R.”, at least one pallet rotation tour needs to be scheduled, which is why the “≥25 kg” and “rot. 1&25 kg” strategies led to a slightly higher throughput time than the “n.R.” strategy that does not include any pallet rotations (the increase is less than 0.1%). As a result, for the new parameter settings, a company that is only interested in shortening the throughput time of its order picking operations would select the “n.R.” strategy.

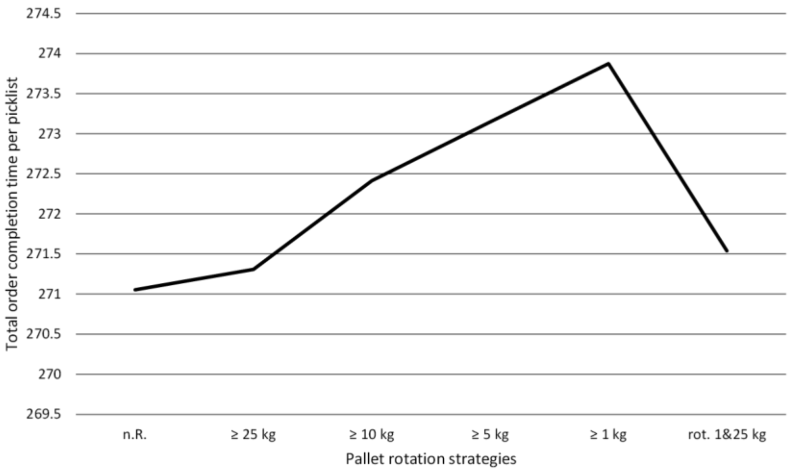


Figure 15: Average time needed to process one pick list for different pallet rotation strategies with the new problem setting

Figure 16, in turn, shows that rotating only pallets with 1 kg and 25 kg items significantly reduces the load on order picker as compared to the other pallet rotation strategies in the new scenario, as the total number of pallet rotations is highest in this case (cf. the grey bars in Figure 14). The highest reduction in the load per picklist with about 9% results, as expected, when comparing this strategy to the case of no rotations. The load per picklist is higher in this example if all pallets are rotated (cf. the "≥ 1 kg" pallet rotation strategy in Figure 16). This result that may seem counterintuitive at first glance can be explained using the results shown in Figure 14, which illustrate that including all items in the pallet rotation tours leads to a significantly lower total number of rotations due to the negative impact a rotation of 5 kg and 10 kg items would have on the total throughput time.

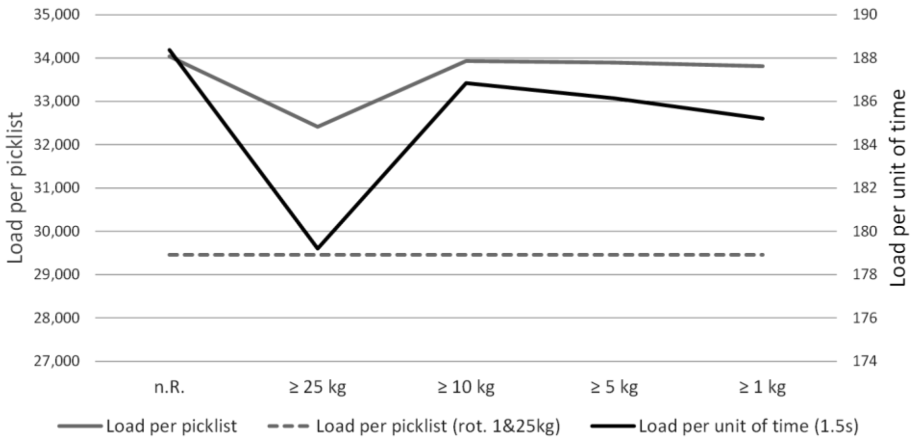


Figure 16: Average load on the order picker for different pallet rotation strategies and automated pallet rotation with the new problem setting

As in the case where the company aims on rotating only pallets whose rotation leads to a direct improvement in order picking time, the objective in the case where rotating pallets leads to an additional load on the order picker could be to rotate only those pallets whose rotation has a positive net effect on the load of the order picker. In the example studied here, the rotation of all pallets has a positive net effect on the load of the order picker as, due to the mechanical advantage of the hand pallet truck studied, a relatively low weight acts on the order picker. The additional load that is caused by rotating the pallet is then directly offset after a single or a few picks. However, in case the company should use a hand pallet truck with a different (less efficient) mechanical advantage than the one studied here, excluding pallets with a negative net effect on the load of the order picker from rotations would make sense from an ergonomics point of view (even though rotating such pallets may still be beneficial from an economic perspective).

7 Discussion and conclusion

This paper studied order picking in a warehouse where products are stored on pallets in two rows one above each other, and it was inspired by a situation observed in practice. Picking products directly from pallets renders order picking a high-risk environment for developing musculoskeletal disorders due to the required handling of heavy loads and continuous bending, stretching and lifting during materials handling. The challenge that arises for warehouse managers in this case is to organize the order picking process as efficiently as possible, simultaneously keeping in mind health and safety issues. The first aspect can be addressed by planning short order picking routes and implementing storage assignment methods that allocate frequently required items close to the depot of the warehouse. Worker well-being, in turn, can be improved by searching for opportunities to reduce the load on the warehouse worker. The paper at hand investigated the case where the company has the opportunity to rotate pallets once their front part has been depleted, which helps to reduce the extent of bending and stretching required on the part of the order picker, and therewith the load on the worker's spine. The load on the spine of a worker is a frequently-used indicator for injury risks in manual materials handling.

After describing the practical case that motivated the research at hand, a biomechanical model was developed to measure the peak L4/L5 spinal compression that acts on the order picker during the picking of items. This approach is common in the human factors engineering literature and well documented as a suitable indicator for risks of low-back injuries. In addition, an economic measure of total order picking time was developed, and a mathematical model was proposed for sequencing orders, routing the order picker through the warehouse, and scheduling pallet rotation tours. The developed model allows studying the impact of rotating pallets on two different measures, order picking time and peak spinal load on the order picker, which has not been addressed in the literature so far.

From a managerial point of view, our analysis revealed that rotating pallets can reduce both order picking time and the spinal load on the order picker. Especially when rotating pallets is associated with no or a low additional load on the order picker, our results indicate that rotating pallets always reduces the load on the order picker and thus improves the ergonomic assessment of the workplace. In particular when picking heavy items from pallets, picking items from the pallet's front part causes a significantly lower load on the order picker, such that a potential additional load that results from rotating pallets is offset already after a few picks. Apart from a reduction in the load on the order picker, our analysis indicated that rotating pallets may also lead to an improvement in the time-objective of order picking. Although rotating pallets consumes a certain amount of time, picking from the front part of a pallet is often quicker than picking from the pallet's back part, such that the time required for rotating pallets can be compensated by the quicker picking activity especially for products where a larger number of items are stored on the pallet.

These results have important managerial implications: First, it could be shown that there is not necessarily a trade-off between economic and ergonomic objectives, as some measures that aim on improving worker well-being also improve the worker's performance. Secondly, companies under strong cost pressure that only aim on improving their economic objectives may still improve worker well-being by rotating only those pallets that lead to a direct reduction of order picking time. Even though such a pallet rotation strategy might exclude pallets with heavy items from being rotated, the selective rotation of pallets still reduces the load on the order picker and improves the ergonomic assessment of the workplace.

Our results further showed that alternative storage assignment methods could be another interesting alternative to reduce the load on the warehouse workers. As our biomechanical analysis indicated that picking from the upper pallet level produces a lower load on the order picker than picking from the lower level, we developed a weight-based storage assignment that assigns heavy items to the upper rack level. Our results indicate that this storage assignment has the potential to reduce the load on the order picker if the demand for these items is high enough; otherwise, it may lead to an increase in the load on the warehouse worker, as items with a lower weight (but a higher demand) are moved to storage positions that enhance the load on the worker. Consequently, alternative storage assignment methods need to be carefully evaluated in light of item and demand characteristics.

This work has limitations. First, the spinal load on the order picker depends on specific worker and workplace attributes, such that the load values obtained from our biomechanical model and the results obtained in the numerical analysis cannot necessarily by generalized. We note, however, that the method proposed in this paper can be applied to any workplace setting where items are picked from pallets, such that our model is well suited to support decisions on the rotation of pallets in practice. Similarly, only two general types of pallet rotation devices were studied in this paper, and a practical application of our model may make it necessary to re-estimate the performance figures of the devices. Thirdly, direct interdependencies between the load level and the performance of the order picker were not considered in this paper. There is, however, evidence that higher load levels increase worker fatigue (e.g., Granata and Marras 1996), leading to a slower order picking process and more pick errors over time. If such a relationship is assumed, rotating pallets becomes even more interesting, as it may help to reduce the fatigue levels of the warehouse workers. In this line of thought, future research could focus on procedures that determine the sequences of heavy picks and work breaks to avoid that several heavy items have to be picked consecutively. This could, for example, be modelled using a fatigue-recovery measure coupled with a maximum endurance time during which a worker is able to complete a certain task (or set of heavy picks) without an increased injury risk. Finally, also the long-term cost impact of high spine loads on the workforce was not considered in this paper. Several studies have shown that high spinal loads may lead to lower back pain resulting in high long-term direct and indirect costs for the company (e.g., Baldwin 2004). Clearly, considering such costs would again increase the economic benefit of rotating pallets in order picking.

Considering the limited number of works that study human factors in order picking from an interdisciplinary perspective, this paper contributes to the development of the research stream of sustainable warehouse management. Future work could study other planning problems in order picking, such as routing or order batching, and integrate ergonomics measures (such as peak spinal load used in this paper) into related decision support models. This could facilitate managerial decision making and help to highlight that considering ergonomics objectives during the planning process does not solely induce costs, but that it can instead contribute to long-term sustainable processes. In case of order picking, this could simultaneously lead to reduced costs of picking as well as reduced injury risks.

Appendix (the default solver)

This section first presents a MILP model for solving the problem under consideration and then we compares its results with the results obtained by our SA approach described in Section 5.4. Table A1 summarizes the notations used in developing the proposed MILP model.

Decision variables		
x_{jk}	1 **if** $order\ j\ is\ included\ in\ set\ k,$ **otherwise** 0	$k, j = 1, \dots, r$
$y_{ii'k}$	1 **if** $items\ i\ \&\ i' > i$ $both\ needed\ in\ set\ k\ (i.e., Q_{ik} > 0\ \&\ Q_{i'k} > 0),$ $means\ has\ \gamma_i^k = 0, \gamma_{i'}^k = 0$ $and\ there\ is\ no\ other\ item\ i > i'' > i\ with\ \gamma_{i''}^k = 0;$ **otherwise** 0	$i = 0..n-1$ $i' = i+1..n$ $k = 1, \dots, r$
Integer variables		
Q_{ik}	$\sum_{j=1}^{r} x_{jk} * q_{ij};$	$\forall i \in \Psi$ $\forall k \in \{1, \dots, r\}$
Binary variables		
a_i^k	**If** $\gamma_i^{k-1} > 0 \rightarrow a_i^k = 0$ **If** $\gamma_i^{k-1} = 0 \rightarrow a_i^k = 1$	$\forall i \in \Psi$ $\forall k \in \{1, \dots, r, r+1\}$
b_i^k	**If** $\gamma_i^{k-1} - Q_{ik} > 0 \rightarrow b_i^k = 0$ **If** $\gamma_i^{k-1} - Q_{ik} \leq 0 \rightarrow b_i^k = 1$	$\forall i \in \Psi$ $\forall k \in \{1, \dots, r\}$
c_i^k	**If** $\mu_i^k > 0 \rightarrow c_i^k = 0$ **If** $\mu_i^k = 0 \rightarrow c_i^k = 1$	
d_i^k	**If** $\eta_i^k - H_i > 0 \rightarrow d_i^k = 0$ **If** $\eta_i^k - H_i \leq 0 \rightarrow d_i^k = 1$	
g_k	**If** $\sum_{j=1}^{r} x_{jk} > 0 \rightarrow g_k = 0$ **If** $\sum_{j=1}^{r} x_{jk} = 0 \rightarrow g_k = 1$	$\forall k \in \{1, \dots, r\}$

Table A 1: The notations (variables) used in MILP model

Objective:

$$\min \sum_{k=1}^{r} f(R_k) = \sum_{k=1}^{r} (f^R(k) + f^P(R_k))$$

$$= (\sum_{k=1}^{r} \left(\sum_{i=1}^{N} a_i^{k+1} * t^R(i) + \sum_{i=0}^{N-1} \sum_{i'=i+1}^{N} y_{ii'k} * \frac{d_{ii'}}{speed} + \sum_{i=1}^{N} y_{i0k} * \frac{d_{i0}}{peed} \right))$$

$$+ \sum_{k=1}^{r} (\sum_{i \in \Psi} (\zeta_i^k * t_f^P(i) + (Q_{ik} - \zeta_i^k) * t_b^P(i)))$$

Constraints:

1) $\sum_{k=1}^{r} x_{jk} = 1 \ \forall j \in \{1, \dots, r\};$ r constraints;

2) $Q_{ik} = \sum_{j=1}^{r} x_{jk} * q_{ij};\ \forall i \in \{1, \dots, N\}, \forall k \in \{1, \dots, r\}$ r*n constraints;

3) $\eta_i^0 = 2 * H_i;\ \gamma_i^0 = H_i;\ \forall i \in \{1, \dots, N\}$ n+n constraints;

4) $\frac{Q_{ik} - \eta_i^{k-1}}{2*H_i} + 1 \geq \mu_i^k > \frac{Q_{ik} - \eta_i^{k-1}}{2*H_i}\ \forall i \in \{1, \dots, N\}, \forall k \in \{1, \dots, r\}$ 2*(n*r) constraints;

5) $\eta_i^k = \eta_i^{k-1} + 2 * H_i * \mu_i^k - Q_{ik};\ \forall i \in \{1, \dots, N\}\Psi, \forall k \in \{1, \dots, r\}$ r*n constraints;

6) constraints for γ_i^k:

The following two constraints will be active when $\gamma_i^{k-1} = \mu_i^k = 0$

6.1. $\gamma_i^k \leq M * (\gamma_i^{k-1} + \mu_i^k) + \eta_i^{k-1} - Q_{ik};\ \forall i \in \{1, \dots, N\}\Psi, \forall k \in \{1, \dots, r\}$ r*n constraints;

6.2 $\gamma_i^k \geq -M * (\gamma_i^{k-1} + \mu_i^k) + \eta_i^{k-1} - Q_{ik};\ \forall i \in \{1, \dots, N\}\Psi, \forall k \in \{1, \dots, r\}$ r*n constraints;

The following three constraints will be active when $\gamma_i^{k-1} > 0\ \&\ \mu_i^k = 0$

6.3. $\gamma_i^k \geq \gamma_i^{k-1} - Q_{ik} - M * (a_i^k + \mu_i^k);\ \forall i \in \{1, \dots, N\}\Psi, \forall k \in \{1, \dots, r\}$ r*n constraints

6.4. $\gamma_i^k \leq \gamma_i^{k-1} - Q_{ik} + M * (a_i^k + \mu_i^k + b_i^k);\ \forall i \in \{1, \dots, N\}\Psi, \forall k \in \{1, \dots, r\}$ r*n constraints

6.5. $\gamma_i^k \leq 0 + M * (a_i^k + \mu_i^k + (1 - b_i^k));\ \forall i \in \{1, \dots, N\}\Psi, \forall k \in \{1, \dots, r\}$ r*n constraints

The following three constraints will be active when $\gamma_i^{k-1} > 0\ \&\ \mu_i^k > 0$

6.6. $\gamma_i^k \geq \eta_i^k - H_i - M * (a_i^k + c_i^k);\ \forall i \in \{1, \dots, N\}\Psi, \forall k \in \{1, \dots, r\}$ r*n constraints

6.7. $\gamma_i^k \leq \eta_i^k - H_i + M * (a_i^k + c_i^k + d_i^k);\ \forall i \in \{1, \dots, N\}\Psi, \forall k \in \{1, \dots, r\}$ r*n constraints

6.8. $\gamma_i^k \leq 0 + M * (a_i^k + c_i^k + (1 - d_i^k));\ \forall i \in \{1, \dots, N\}\Psi, \forall k \in \{1, \dots, r\}$ r*n constraints

The following three constraints will be active when $\gamma_i^{k-1} = 0\ \&\ \mu_i^k > 0$

6.9. $\gamma_i^k \geq \eta_i^k - H_i - M * (\gamma_i^{k-1} + c_i^k);\ \forall i \in \{1, \dots, N\}\Psi, \forall k \in \{1, \dots, r\}$ r*n constraints

6.10. $\gamma_i^k \leq \eta_i^k - H_i + M * (\gamma_i^{k-1} + c_i^k + d_i^k);\ \forall i \in \{1, \dots, N\}\Psi, \forall k \in \{1, \dots, r\}$ r*n constraints

6.11. $\gamma_i^k \leq 0 + M * (\gamma_i^{k-1} + c_i^k + (1 - d_i^k));\ \forall i \in \{1, \dots, N\}\Psi, \forall k \in \{1, \dots, r\}$ r*n constraints

7) constraints for $\check{\Upsilon}_i^k$:

7.1. $\check{\Upsilon}_i^k \leq M * \gamma_i^{k-1} + \eta_i^{k-1}$; $\forall i \in \{1, \ldots, N\}\Psi, \forall k \in \{1, \ldots, r\}$ r*n constraints

7.2. $\check{\Upsilon}_i^k \geq -M * \gamma_i^{k-1} + \eta_i^{k-1}$; $\forall i \in \{1, \ldots, N\}\Psi, \forall k \in \{1, \ldots, r\}$ r*n constraints

7.3. $\check{\Upsilon}_i^k \leq M * a_i^k + \gamma_i^{k-1}$; $\forall i \in \{1, \ldots, N\}\Psi, \forall k \in \{1, \ldots, r\}$ r*n constraints

7.4. $\check{\Upsilon}_i^k \geq -M * a_i^k + \gamma_i^{k-1}$; $\forall i \in \{1, \ldots, N\}\Psi, \forall k \in \{1, \ldots, r\}$ r*n constraints

8) Constraints for ζ_i^k(front picks)

8.1. $\zeta_i^k \leq Q_{ik} + M * \mu_i^k$; $\forall i \in \{1, \ldots, N\}\Psi, \forall k \in \{1, \ldots, r\}$ r*n constraints

8.2. $\zeta_i^k \leq \check{\Upsilon}_i^k + M * \mu_i^k$; $\forall i \in \{1, \ldots, N\}\Psi, \forall k \in \{1, \ldots, r\}$ r*n constraints

8.3. $\zeta_i^k \leq \check{\Upsilon}_i^k + H_i * \mu_i^k + c_i^k * M$; $\forall i \in \{1, \ldots, N\}\Psi, \forall k \in \{1, \ldots, r\}$ r*n constraint

8.4. $\zeta_i^k \leq \check{\Upsilon}_i^k + Q_{ik} - \eta_i^{k-1} - \left(\mu_i^k - 1\right) * H_i + c_i^k * M$; $\forall i \in \{1, \ldots, N\}\Psi, \forall k \in \{1, \ldots, r\}$
r*n constraints

9) Constraints for $y_{ii'k}$(which determines the rotation tours)

9.1. $y_{ii'k} \geq -\left((-a_i^{k+1} + 1\right) + \left(-a_{i'}^{k+1} + 1\right) + \sum_{i''=i+1}^{i'-1} a_{i''}^{k+1}) * M + 1$; $\forall i \in \{1, \ldots, N-1\}, i' \in \{i + 1, \ldots., N\}, k \in \{1, \ldots, r\}$ r*n*(n-1)/2 constraints

9.2. $y_{i0k} \geq -\left((-a_i^{k+1} + 1\right) + \sum_{i''=i+1}^{N} a_{i''}^{k+1}) * M + 1$; $\forall i \in \{1, \ldots, N\}, k \in \{1, \ldots, r\}$

r*n constraints

9.3. $y_{0ik} \geq -\left((-a_i^{k+1} + 1\right) + \sum_{i''=1}^{i-1} a_{i''}^{k+1}) * M + 1$; $\forall i \in \{1, \ldots, N\}, k \in \{1, \ldots, r\}$

r*n constraints

10) feasibility condition:

$g_k * M + \sum_{i=1}^{N} a_i^{k+1} \geq 1$; $\forall k \in \{1, \ldots, r\}$ r constraints

11) auxiliary variables

a: $\frac{-\gamma_i^{k-1}+0.1}{M} + 1 \geq a_i^k \geq \frac{-\gamma_i^{k-1}+0.1}{M}$; $\forall i \in \{1, \ldots, N\}, k \in \{1, \ldots, r+1\}$ 2*(r+1)*n constraints

b: $\frac{-(\gamma_i^{k-1}-Q_{ik})+0.1}{M} + 1 \geq b_i^k \geq \frac{-(\gamma_i^{k-1}-Q_{ik})+0.1}{M}$ 2*r*n constraints c: $\frac{-\mu_i^k+0.1}{M}+1 \geq c_i^k \geq \frac{-\mu_i^k+0.1}{M}$; $\forall i \in \{1,\dots,N\}, k \in \{1,\dots,r\}$ 2*r*n constraints

d: $\frac{-(\eta_i^k-H_i)+0.1}{M}+1 \geq d_i^k \geq \frac{-(\eta_i^k-H_i)+0.1}{M}$; $\forall i \in \{1,\dots,N\}, k \in \{1,\dots,r\}$ 2*r*n constraints

g: $\frac{-\sum_{j=1}^{r} x_{jk}+0.1}{M}+1 \geq g_k \geq \frac{-\sum_{j=1}^{r} x_{jk}+0.1}{M}$; $\forall k \in \{1,\dots,r\}$ 2*r constraints

Boundaries:

$y_{ii'k}$ & x_{jk} $a_i^k, c_i^k, d'^k_i, g_k binary$

$\gamma_i^k, \check{\Upsilon}_i^k, \zeta_i^k, \mu_i^k, d_i^k, integers$

As expected, the default solver was unable to solve even medium-sized problems. For problems with 9 or more than 9 picklists, the solver was unable to find an optimal solution in one day. Therefore, we limited the number of picklists considered in small-sized problems to 8. Furthermore, to have a fair comparison, we changed some other parameters introduced in Section 6 for small-sized problems to ensure that the front side of pallets is also depleted for this low number of picklists from time to time For small-sized problems, we first set new values for H_i as 6, 5, 4, and 3, respectively, for items with 1 kg, 5kg, 10kg, and 25 kg weight. The number of items needed in each picklist was generated by U~(1,6). Finally, we also set $m = 2$ and $n = 1$ (i.e., in total, small-sized problems only consider 10 items). The other problem parameters are the same as the ones introduced in Section 6.

20 small-sized problems were generated according to this procedure to compare the performance of the MILP model and SA optimization algorithm introduced in Section 5.4. Both algorithms were implemented in Java 8. We solved the test instances on a x64-PC with 12 GB of RAM and an Intel Core i5-7200U 2.5 GHz CPU. To solve the MILP model, we used CPLEX 12.7. The average run-time for the default solver was 577 seconds, which was 60.5% higher than the runtime of the SA. As expected, since the size of the problem instances was very small, the default solver was able to find optimal solutions for all problem instances.

References

Baldwin, M. L. (2004): Reducing the costs of work-related musculoskeletal disorders: targeting strategies to chronic disability cases. Journal of Electromyography and Kinesiology, 14 (1), 33-41.

Barachet, L.L. (1957): Graphic solution of the travelling-salesman problem. Operations Research, 5 (6), 841-845.

Battini, D.; Glock, C.H.; Grosse, E.H.; Persona, A.; Sgarbossa, F. (2016): Human energy expenditure in order picking storage assignment: A bi-objective method. Computers & Industrial Engineering, 94, 147-157.

Battini, D., Calzavara, M., Persona, A., & Sgarbossa, F. (2017). Additional effort estimation due to ergonomic conditions in order picking systems. International Journal of Production Research, 55 (10), 2764-2774.

Bevan, S. (2015): Economic impact of musculoskeletal disorders (MSDs) on work in Europe. Best Practice & Research Clinical Rheumatology, 29 (3), 356-373.

Boudreau, J.; Hopp, W.; McClain, J.O.; Thomas, L.J.: (2003): On the interface between operations and human resources management. Manufacturing & Service Operations Management, 5 (3), 179-202.

Bouleimen, K.; Lecocq, H. (2003): A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version. European Journal of Operational Research, 149 (2), 268-281.

Boysen, N.; Emde, S.; Hoeck, M.; Kauderer, M. (2015): Part logistics in the automotive industry: Decision problems, literature review and research agenda. European Journal of Operational Research, 242 (1), 107-120.

Bureau of Labor Statistics (2016): Nonfatal occupational injuries and illnesses requiring days away from work, 2015. USDL-16-2130, November 10, 2016.

Calzavara, M.; Glock, C.H.; Grosse, E.H.; Persona, A.; Sgarbossa, F. (2017): Analysis of economic and ergonomic performance measures of different rack layouts in an order picking warehouse. Computers & Industrial Engineering, 111 (September), 527-536.

Chackelson, C.; Errasti, A.; Ciprés, D.; Lahoz, F. (2013): Evaluating order picking performance trade-offs by configuring main operating strategies in a retail distributor: A Design of Experiments approach. International Journal of Production Research, 51 (20), 6097-6109.

Clarke, G.; Wright, J.W. (1964): Scheduling of vehicles from a central depot to a number of delivery points. Operations Research, 12 (4), 568-581.

Cole, M.H.; Grimshaw, P.N. (2005): Compressive loads on the lumbar spine during lifting: 4D WATBAK versus inverse dynamics calculations. Applied Bionics and Biomechanics, 2 (3, 4), 149-159.

Cordeau, J.F.; Gendreau, M.; Hertz, A.; Laporte, G.; Sormany, J.-S. (2004): New Heuristics for the Vehicle Routing Problem. In: Langevin, A.; Riopel, D. (Eds.): Logistic Systems – Design and Optimization. New York: Springer, 279-297.

Coyte, P.C.; Asche, C.V.; Croxford, R.; Chan, B. (1998): The economic cost of musculoskeletal disorders in Canada. Arthritis & Rheumatism, 11 (5), 315-325.

Dagenais, S.; Caro, J.; Haldeman, S. (2008): A systematic review of low back pain cost of illness studies in the United States and internationally. The Spine Journal, 8 (1), 8-20.

Dantzig, G.B.; Ramser, J.H. (1959): The Truck Dispatching Problem. Management Science, 6 (1), 80-91.

Davarzani, H., & Norrman, A. (2015): Toward a relevant agenda for warehousing research: literature review and practitioners' input. Logistics Research, 8(1), 1-18.

Daynard, D.; Yassi, A.; Cooper, J.E.; Tate, R.; Norman, R.; Wells, R. (2001): Biomechanical analysis of peak and cumulative spinal loads during simulated patient-handling activities: a substudy of a randomized controlled trial to prevent lift and transfer injury of health care workers. Applied Ergonomics, 32 (3), 199-214.

De Koster, R.; Van der Poort, E. (1998): Routing orderpickers in a warehouse: a comparison between optimal and heuristic solutions. IIE Transactions, 30 (5), 469-480.

De Koster, R.; Le-Duc, T.; Roodbergen, K.J. (2007): Design and control of warehouse order picking: A literature review. European Journal of Operational Research, 182 (2), 481-501.

De Vries, J.; de Koster, R.; Stam, D. (2016a): Exploring the role of picker personality in predicting picking performance with pick by voice, pick to light and RF-terminal picking. International Journal of Production Research, 54 (8), 2260-2274.

De Vries, J.; de Koster, R.; Stam, D. (2016b): Aligning Order Picking Methods, Incentive Systems, and Regulatory Focus to Increase Performance. Production and Operations Management. DOI: 10.1111/poms.12547.

Eurostat (2009): European business – facts and figures. Office for Official Publications of the European Communities, Luxembourg.

Franzke, T., Grosse, E. H., Glock, C. H., & Elbert, R. (2017). An investigation of the effects of storage assignment and picker routing on the occurrence of picker blocking in manual picker-to-parts warehouses. The International Journal of Logistics Management, 28 (3), 841-863.

Frazelle, E.A. (2002): World-class warehousing and material handling. McGraw-Hill, New York.

Gademann, N.; Velde, S. (2005): Order batching to minimize total travel time in a parallel-aisle warehouse. IIE Transactions, 37 (1), 63-75.

Garey, M.R.; Johnson, D.S. (1979). A Guide to the Theory of NP-Completeness. WH Freemann, New York.

Garg, A. (1986): Biomechanical and ergonomic stresses in warehouse operations. IIE Transactions, 18 (3), 246-250.

Garg, A. (2000): Ergonomic, Biomechanical and Physiological Stresses from Manual Materials Handling in Grocery Distribution Centers. Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 44 (28), 431-432.

Gillet, B.E.; Miller, L.R. (1974): A heuristic algorithm for the vehicle-dispatch problem. Operations Research, 22 (2), 568-581.

Glock, C.H.; Grosse, E.H. (2012): Storage policies and order picking strategies in U-shaped order-picking systems with a movable base. International Journal of Production Research, 50 (16), 4344-4357.

Glock, C.H.; Grosse, E.H.; Elbert, R.M.; Franzke, T. (2017): Maverick picking: the impact of modifications in work schedules on manual order picking processes. International Journal of Production Research, 55 (21), 6344-6360.

Glock, C.H.; Grosse, E.H.; Neumann, W.P.; Sgarbossa, F. (2017): Editorial: Human factors in industrial and logistic system design. Computers & Industrial Engineering, 111 (September), 463-466.

Goetschalckx, M.; Ratliff, H.D. (1988a): An efficient algorithm to cluster order picking items in a wide aisle. Engineering Costs and Production Economics, 13 (3), 263-271.

Goetschalckx, M.; Ratliff, H.D. (1988b): Order picking in an aisle. IIE Transactions, 20 (1), 53-62.

Granata, K.P.; Marras, W.S. (1996). Changes in trunk and spine biomechanics during repetitive fatiguing exertions. Proceedings of the Fifteenth Biomedical Engineering Conference, 439-442.

Grosse, E.H.; Glock, C.H. (2013): An experimental investigation of learning effects in order picking systems. Journal of Manufacturing Technology Management, 24 (6), 850-872.

Grosse, E.H.; Glock, C.H.; Jaber, M.Y. (2013): The effect of worker learning and forgetting on storage reassignment decisions in order picking systems. Computers & Industrial Engineering, 66 (4), 653-662

Grosse, E.H.; Glock, C.H. (2015): The effect of worker learning on manual order picking processes. International Journal of Production Economics, 170, 882-890.

Grosse, E.H.; Glock, C.H.; Jaber, M.Y.; Neumann, W.P. (2015): Incorporating human factors in order picking planning models: framework and research opportunities. International Journal of Production Research, 53 (3), 695-717.

Grosse, E. H., Glock, C. H., & Neumann, W. P. (2017a). Human factors in order picking: a content analysis of the literature. International Journal of Production Research, 55 (5), 1260-1276.

Grosse, E. H., Calzavara, M., Glock, C. H., & Sgarbossa, F. (2017b). Incorporating human factors into decision support models for production and logistics: current state of research. IFAC-PapersOnLine, 50 (1), 6900-6905.

Henn, S.; Koch, S.; Gerking, H.; Wäscher, G. (2013): A U-shaped layout for manual order-picking systems. Logistics Research, 6 (4), 245-261.

Henn, S.; Wäscher, G. (2012): Tabu search heuristics for the order batching problem in manual order picking systems. European Journal of Operational Research, 222 (3), 484-494.

Hong, S.; Johnson, A.L.; Peters, B.A. (2012): Batch picking in narrow-aisle order picking systems with consideration for picker blocking. European Journal of Operational Research, 221 (3), 557-570.

Hong, S. (2014): Two-worker blocking congestion model with walk speed m in a no-passing circular passage system. European Journal of Operational Research, 235 (3), 687-696.

Kadefors, R.; Forsman, M. (2000): Ergonomic evaluation of complex work: a participative approach employing video–computer interaction, exemplified in a study of order picking. International Journal of Industrial Ergonomics, 25(4), 435-445.

Kerr, M.S.; Frank, J.W.; Shannon, H.S.; Norman, R.W.; Wells, R.P.; Neumann, W.P. (2001): Biomechanical and psychosocial risk factors for low back pain at work. American Journal of Public Health, 91 (7), 1069.

Kim, K.H.; Moon, K.C. (2003): Berth scheduling by simulated annealing. Transportation Research Part B: Methodological, 37 (6), 541-560.

Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. (1983): Optimization by simulated annealing. Science, 220 (4598), 671-680.

Laporte, G. (1986): Generalized Subtour Elimination Constraints and Connectivity Constraints. Journal of the Operational Research Society, 37 (5), 509-514.

Laporte, G. (1992): The Vehicle Routing Problem: An overview of exact and approximate algorithms. European Journal of Operational Research, 59 (3), 345-358.

Larco, J.A.; de Koster, R.; Roodbergen, K.J.; & Dul, J. (2016): Managing warehouse efficiency and worker discomfort through enhanced storage assignment decisions. International Journal of Production Research, 1-16. DOI: 10.1080/00207543.2016.1165880.

Lavender, S.A.; Marras, W.S.; Ferguson, S.A.; Splittstoesser, R.E.; Yang, G. (2012): Developing physical exposure-based back injury risk models applicable to manual handling jobs in distribution centers. Journal of Occupational and Environmental Hygiene, 9 (7), 450-459.

Ma, L.; Chablat, D.; Bennis, F.; Zhang, W. (2009): A new simple dynamic muscle fatigue model and its validation. International Journal of Industrial Ergonomics, 39 (1), 211-220.

Matusiak, M.; de Koster, R.; Kroon, L.; Saarinen, J. (2014): A fast simulated annealing method for batching precedence-constrained customer orders in a warehouse. European Journal of Operational Research, 236 (3), 968-977.

Mowrey, C.H.; Parikh, P.J. (2014): Mixed-width aisle configurations for order picking in distribution centers. European Journal of Operational Research, 232 (1), 87-97.

Muppani, V.R.; Adil, G.K. (2008): Efficient formation of storage classes for warehouse storage location assignment: a simulated annealing approach. Omega, 36 (4), 609-618.

Neumann, W.P.; Wells, R.P.; Norman, R.W. (1999): 4D WATBAK: adapting research tools and epidemiological findings to software for easy application by industrial personnel. In Proceedings of the international conference on computer-aided ergonomics and safety, Barcelona, Spain.

Neumann, W.P.; Medbo, L. (2010): Ergonomic and technical aspects in the redesign of material supply systems: Big boxes vs. narrow boxes. International Journal of Industrial Ergonomics, 40 (5), 541-548.

Neumann, W.P.; Village, J. (2012): Ergonomics action research II: a framework for integrating HF into work system design. Ergonomics, 55 (10), 1140-1156.

NIOSH (1981): A work practices guide for manual lifting. Tech Report No. 81-122. U.S. Dept of Health and Human Services, Cincinnati, OH.

Norman, R.; Wells, R.; Neumann, W.P.; Frank, J.; Shannon, H.; Kerr, M.; Study, T.O. (1998): A comparison of peak vs cumulative physical work exposure risk factors for the reporting of low back pain in the automotive industry. Clinical

Biomechanics, 13 (8), 561-573.

Petersen, C.G.; Schmenner, R.W. (1999): An evaluation of routing and volume-based storage policies in an order picking operation. Decision Sciences, 30 (2), 481.

Petersen, C.G.; Aase, G. (2004): A comparison of picking, storage, and routing policies in manual order picking. International Journal of Production Economics, 92 (1), 11-19.

Petersen, C.G.; Siu, C.; Heiser, D.R. (2005): Improving order picking performance utilizing slotting and golden zone storage. International Journal of Operations & Production Management, 25 (10), 997-1012.

Rao, S.S.; Adil, G.K. (2013): Class-based storage with exact S-shaped traversal routeing in low-level picker-to-part systems. International Journal of Production Research, 51 (16), 4979-4996.

Ratliff, H.D.; Rosenthal, A.S. (1983). Order-picking in a rectangular warehouse: a solvable case of the traveling salesman problem. Operations Research, 31 (3), 507-521.

Richards, G. (2014): Warehouse Management: A complete guide to improving efficiency and minimizing costs in the modern warehouse. Kogan Page Publishers, London.

Roodbergen, K.J.; De Koster, R. (2001): Routing order pickers in a warehouse with a middle aisle. European Journal of Operational Research, 133 (1), 32-43.

Roodbergen, K.J.; Sharp, G.P.; Vis, I.F. (2008): Designing the layout structure of manual order picking areas in warehouses. IIE Transactions, 40 (11), 1032-1045.

Roodbergen, K.J.; Vis, I.F. (2009): A survey of literature on automated storage and retrieval systems. European Journal of Operational Research, 194 (2), 343-362.

Roodbergen, K.J.; Vis, I.F.; Taylor Jr, G.D. (2015): Simultaneous determination of warehouse layout and control policies. International Journal of Production Research, 53 (11), 3306-3326.

Rose, L.M.; Orrenius, U.E.; Neumann, W.P. (2013). Work Environment and the Bottom Line: Survey of Tools Relating Work Environment to Business Results. Human Factors and Er-gonomics in Manufacturing & Service Industries 23 (5), 368-381.

Schäfer, K.; Schick, R.; Rokosch, F.; Schwarzmann, K.; Karny, H.-P.; Weiner, U. (2009): Verbesserung der Arbeitsbedingungen beim Kommissionieren durch Einsatz eines Palettenwendegerätes. Zentralblatt für Arbeitsmedizin, Arbeitsschutz und Ergonomie, 59 (4), 106-118.

Scholz, A.; Henn, S.; Stuhlmann, M.; Wäscher, G. (2016): A new mathematical programming formulation for the Single-Picker Routing Problem. European Journal of Operational Research, 253 (1), 68-84.

Schultz, A.B.; Andersson, G.B. (1981): Analysis of loads on the lumbar spine. Spine, 6 (1), 76-82.

Theys, C.; Bräysy, O.; Dullaert, W.; Raa, B. (2010): Using a TSP heuristic for routing order pickers in warehouses. European Journal of Operational Research, 200 (3), 755-763.

Thomas, L.M.; Meller, R.D. (2014): Analytical models for warehouse configuration. IIE Transactions, 46 (9), 928-947.

Tompkins, J.A.; White, Y.A.; Bozer, E.H.; Tanchoco, J.M.A. (2010): Facilities planning, 4th ed. John Wiley & Sons, Hoboken, N.J.

U.S. Department of Health & Human Services (2015): Medical Expenditure Panel Survey (2015): http://meps.ahrq.gov/mepsweb/

Van Gils, T.; Ramaekers, K.; Caris, A.; de Koster, R.B. (2018). Designing Efficient Order Picking Systems by Combining Planning Problems: State-of-the-art Classification and Review. European Journal of Operational Research, 267 (1), 1-15.

Van Laarhoven, P.J.; Aarts, E.H.; Lenstra, J.K. (1992): Job shop scheduling by simulated annealing. Operations Research, 40 (1), 113-125.

Vaughan, T.S.; Petersen, C.G. (1999): The effect of warehouse cross aisles on order picking efficiency. International Journal of Production Research, 37 (4), 881-897.

Village, J.; Frazer, M.; Cohen, M.; Leyland, A.; Park, I.; Yassi, A. (2005): Electromyography as a measure of peak and cumulative workload in intermediate care and its relationship to musculoskeletal injury: an exploratory ergonomic study. Applied ergonomics, 36 (5), 609-618.

Wren, A. (1971): Computers in Transport Planning and Operation. London: Ian Allan.

Wren, A.; Holliday, A. (1972): Computer Scheduling of Vehicles from One or More Depots to a Number of Delivery Points. Operational Research Quarterly, 23 (3), 333-344.

Yu, M.; De Koster, R.B. (2009): The impact of order batching and picking area zoning on order picking system performance. European Journal of Operational Research, 198 (2), 480-490.