Universität
Augsburg
University

Serial-batch scheduling – the special case of laser-cutting machines

Cumulative Doctoral Dissertation
at the Faculty of Business Administration and Economics
of the University of Augsburg

submitted in partial fulfillment of the requirements for the degree of
Doctor of Economic Sciences (Dr. rer. pol.)

submitted by
Stefan Wahl, M.Sc.

Augsburg, Germany, 2023

First reviewer: Prof. Dr. Axel Tuma

Second reviewer: Prof. Dr. Robert Klein

Chairman of the oral exam: Prof. Dr. Marco C. Meier

Date of oral exam: 22 June 2023

Table of Contents

List of Scientific Contributions

The following published and submitted scientific contributions are presented within this doctoral dissertation. The articles are sorted following their order of publication. The journal rankings correspond to VHB-JOURQUAL3, published by the German Academic Association for Business Research (VHB).

Contribution C1

(Published in the International Journal of Production Research, ranked **B**)

Gahm, C., Wahl, S., & Tuma, A. (2022b). Scheduling parallel serial-batch processing machines with incompatible job families, sequence-dependent setup times and arbitrary sizes. *International Journal of Production Research, 60(17), 5131‑5154*. doi:10.1080/00207543.2021.1951446.

Contribution C2

(Published in the European Journal of Operational Research, ranked **A**)

Gahm, C., Uzunoglu, A., Wahl, S., Ganschinietz, C., & Tuma, A. (2022a). Applying machine learning for the anticipation of complex nesting solutions in hierarchical production planning. *European Journal of Operational Research, 296(3), 819‑836*. doi:10.1016/j.ejor.2021.04.006.

Contribution C3

(Published in Computers and Operations Research, ranked **B**)

Uzunoglu, A., Gahm, C., Wahl, S., & Tuma, A. (2023). Learning-augmented heuristics for scheduling parallel serial-batch processing machines. *Computers & Operations Research, 151, 106122*. doi:10.1016/j.cor.2022.106122.

Contribution C4

(Submitted to the European Journal of Operational Research, ranked **A**)

Wahl, S., Gahm, C., & Tuma, A. (2023). Serial-batch scheduling: a systematic review and future research directions. *European Journal of Operational Research*.

LIST OF APPENDICES AND ELECTRONIC SUPPLEMENTARY FILES

Contribution C4

- The submitted manuscript

Electronic supplementary material:

- **sb- and hb-literature classification matrix** *(MS Excel file)*
  This file includes the classification of 118 research articles on sb- and hb-scheduling according to the SPCS and SACS.

1

INTRODUCTION

Batching is a commonly used concept when work needs to be done. Regardless of the specific use case, batching means combining or grouping two or more tasks. In other words, it is a productivity strategy in which tasks are grouped into batches and processed together. In everyday life, instead of going to the supermarket five times a week, these tasks may be batched and reduced to one or perhaps two visits. The two visits will be slightly more laborious, but batching the visits will save a lot of time. Batching, therefore, helps to increase effectiveness and efficiency.

Depending on the specific use case, it is not only the batching decision that is crucial but also the decision as to when to process a batch – known in operations management as “scheduling”. To return to the example, the visits to the supermarket are not likely to be random but are scheduled with other pending tasks. Obviously, there exist interdependencies between batching and scheduling decisions.

One of the most prominent examples of such a specific use case comes from the manufacturing industry – the scheduling of laser-cutting machines. Here, we often find machines that achieve higher productivity by processing jobs in batches. The operational production planning of such batch processing machines (“batch scheduling”) also comprises the two interdependent decisions: the batching decision (assigning jobs, production orders, products, etc. to batches) and the scheduling decision (assigning batches to machines – in the case of multiple machines – and sequencing these batches). Since the batch composition strongly influences the scheduling decision (and vice versa), the batching and scheduling decisions should be made taking into account their interdependencies.

The interdependencies between the batching and the scheduling decisions can vary and are strongly related to the specific problem context, i.e., the machine characteristics, the job and processing characteristics, and the scheduling objectives. In terms of job and processing characteristics – in this respect most batch scheduling problems differ strongly from classical scheduling problems – batch scheduling problems can be classified according to some main characteristics (cf., Wahl et al., 2023): (a) batching type, e.g., parallel batching and serial batching, (b) job availability, i.e., batch availability and item availability, and (c) batch capacity, i.e., unbounded batch capacity and bounded batch capacity. “Parallel batching” means that the jobs within a batch are processed simultaneously (in parallel, e.g., burn-in operations in the semiconductor industry), while “serial batching” means that the jobs within a batch are processed sequentially (e.g., laser cutting in the metalworking industry). “Batch availability” means that a job is not available until its entire batch has been completed, while “item availability” means that a job is available immediately after it has been processed. The characteristic “unbounded batch capacity” means that any number of jobs can be grouped into one batch, while “bounded batch capacity” (*bc*) indicates restrictions on the batches (e.g., a limited number, size, or volume of the jobs per batch). In addition to these main characteristics, there are several others, such as “release dates” or “setup times” (*s*), which can make two batch scheduling problems very different from each other (also speaking in terms of complexity).

For example, consider a single batch-processing machine with bounded batch capacity (e.g., by the number of jobs) and batch availability. While the batch completion time for parallel batching problems is calculated from the maximum processing time of the jobs in a batch (cf., Figure 1), the batch completion time for serial batching problems is calculated from the sum of the processing times of the jobs in a batch. In addition, serial batching problems usually include setup times or setup costs (cf., Figure 2).

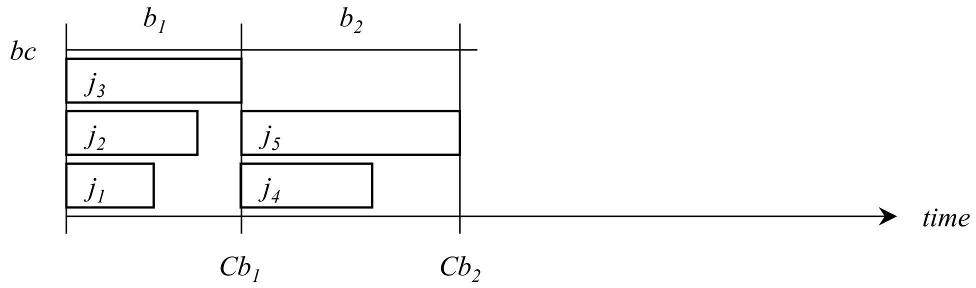

Figure 1: Batch scheduling with parallel batching

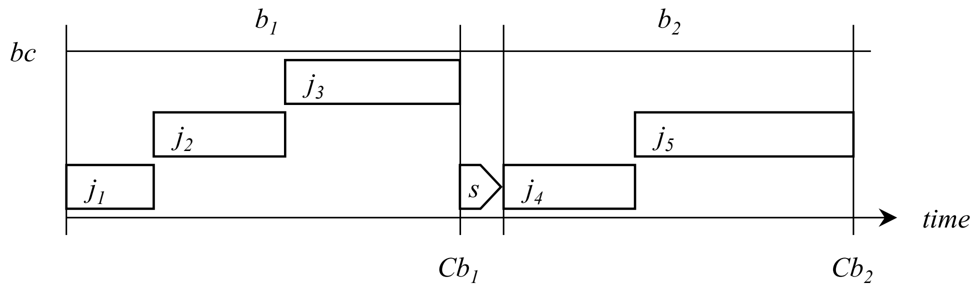

Figure 2: Batch scheduling with serial batching

The interdependencies between the batching and the scheduling decisions can be seen in Figure 1 and Figure 2: moving a job to a different batch (e.g., job j_3 from batch b_1 to b_2) – to achieve a better objective value – can have an impact on each job and the entire schedule (because the composition of the batches determines the processing time of the batches and thus their completion time (Cb_1 and Cb_2) – in the case of batch availability). Because of the interdependencies, batch scheduling problems are particularly challenging combinatorial optimization problems.

When solving combinatorial optimization problems in general, we typically seek optimum solutions. Standard solver engines (e.g., CPLEX Optimizer or Gurobi Optimizer) or exact solution methods (e.g., exact algorithms or dynamic programs) are capable of computing such optimum solutions. However, very often these solution methods are only able to solve "small" batch scheduling problem instances (i.e., with a small number of jobs) in a reasonable time, and such instances often do not reflect real-world application cases. For the solving of real-world application cases (i.e., problem instances with a larger number of jobs – speaking in the range of hundreds and thousands), the employment of heuristics or metaheuristics is recommended or even necessary (especially for NP-hard problems). While heuristics and metaheuristics do not guarantee to find optimum solutions, they commonly provide good results in a reasonable amount of time. This solving efficiency is usually sufficient for most cases, but it is not without limitations: heuristics and metaheuristics are very often problem-specific developments. They have to be adapted to the problem. Thereby, the quality of solutions often depends on parameters that control the behavior of heuristics and metaheuristics. The tuning of such parameters is therefore of particular importance.

In summary, there are three challenges in solving batch scheduling problems: First, the interdependencies between the batching decisions and the scheduling decisions should be respected. Second, the selection or development of appropriate solution methods, and third, the determination of appropriate parameters for the solution methods (if necessary).

This dissertation is based on a real-world application case from the metal processing industry, namely the scheduling of laser cutting machines. The problem can be classified as "Scheduling parallel serial-batch processing machines with incompatible job families, sequence-dependent setup times and arbitrary sizes" (PSBIJF) and is defined and elaborated across several conjoining scientific contributions.

The structure of this thesis is as follows: Section 2 highlights each scientific contribution and explains the overall context. Section 3 provides an overview of the bibliographic data and the abstract of each contribution – the complete manuscripts can be found in the appendix. Finally, Section 4 summarizes the results of the four contributions, discusses the added value to the field of batch scheduling research, and gives some future research directions.

2

CONSTITUENT ELEMENTS OF THE DISSERTATION

Contribution C1

"Scheduling parallel serial-batch processing machines with incompatible job families, sequence-dependent setup times and arbitrary sizes". Gahm et al. (2022b)

In the metal processing industry, many companies produce customer-specific parts, that are either finished products or undergo further processing. The manufacturing process for all of these parts usually starts with the cutting of raw shapes from metal sheets. The raw shapes have several characteristics such as material type, material thickness, and the shape itself (CAD data). As an efficient planning strategy, jobs of the same job family (i.e., jobs of the same material type and material thickness) are grouped together. The decision to group jobs together is called "batching" and a resulting group of jobs is called a "batch". The batching decision must ensure that the jobs can be placed on the metal sheet, i.e., that they do not exceed the sheet boundaries and that they do not overlap. The fulfillment of these constraints together with an intelligent spatial arrangement of the jobs (depending on the objective, e.g., minimizing waste or energy consumption) is also known in the literature as the "complex nesting problem" (CNP) (or "two-dimensional, highly irregular strip packing problem", cf., e.g., Wäscher et al., 2007). In addition, the planned batches must be allocated to the available laser-cutting machines and placed in an advantageous sequence (depending on the objective, e.g., minimizing tardiness). This scheduling decision must take into account setup times that depend on the sequence and job family. The complete serial-batch scheduling problem for the special case of laser-cutting machines is illustrated in Figure 3 (along with the decisions to be made).

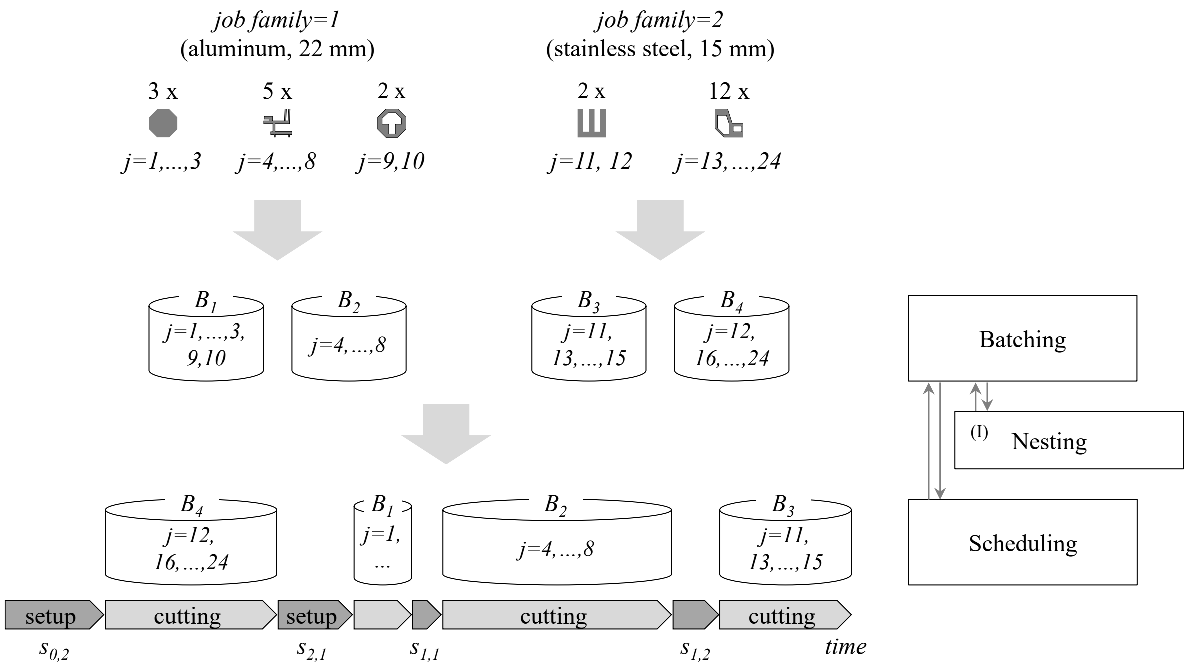

Figure 3: The serial-batch scheduling problem and its decisions

The experimental study shows that the standard solver Gurobi (solving the MILP) is basically able to solve problem instances with up to 60 jobs and 5 machines. When the number of jobs increases, heuristics are used. The unique feature of the presented heuristics is the controllable batch utilization, which allows the creation of smaller batches without non-urgent jobs. The experimental results show that this feature leads to a general superiority of the proposed heuristics compared to heuristics from the literature.

Contribution C2

"Applying machine learning for the anticipation of complex nesting solutions in hierarchical production planning". Gahm et al. (2022a)

Both the described batch scheduling problem (PSBIJF) and the complex nesting problem (CNP) are NP-hard (cf., Gahm et al., 2022b and Gahm et al., 2022a). Since the complexity of NP-hardness suggests that any optimization program or exact solution method is likely to run into an extremely high computational effort as the number of jobs increases, and since solving the PSBIJF would require solving the CNP several times, an approximation approach is developed. This means that the capacity requirements of the jobs are only roughly estimated (i.e., via the sum of the content areas of the raw shapes) and the sum of the capacity requirements of the jobs grouped in a batch must not exceed the available capacity (i.e., the area of the metal sheet). The nesting decision is thus replaced by a simple capacity check (cf., Figure 4, III). Obviously, the use of such a simple approximation is subject to error (and not as accurate as solving the CNP) but is very fast to compute.

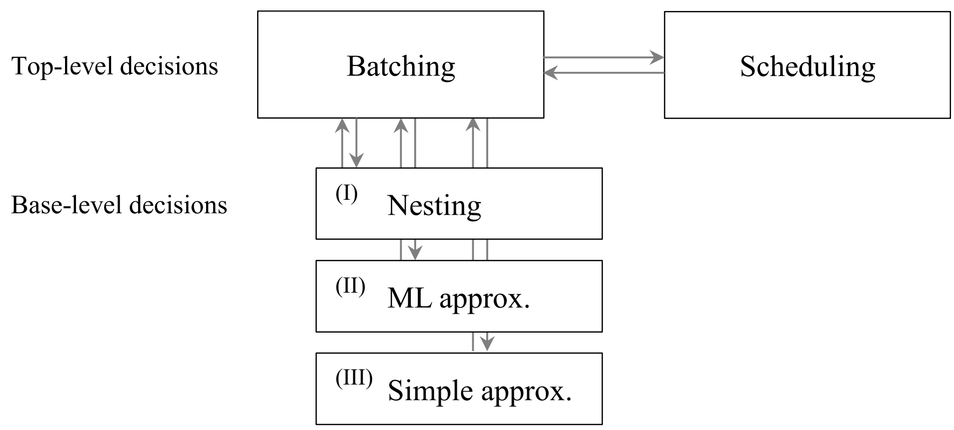

Figure 4: The BSP with approximation

Contribution C2 explores the use of simple approximations from the perspective of hierarchical production planning. Hierarchical production planning distinguishes between superior top-level decisions and subordinate base-level decisions (cf., e.g., Schneeweiss, 2003). There are interdependencies between these decision levels so that the decisions cannot be made in isolation from each other. In general, the top level makes the first decision, gives instructions to the base level, and incorporates the reaction into its own decision. With the batching decision and the scheduling decision constituting the top level and the nesting decision constituting the base level, the coordination of the decisions becomes difficult because each base-level reaction requires solving a CNP. In this case, the anticipation of base-level reactions (bottom-up feedback) by top-level decisions is highly recommended. The use of simple approximations as an anticipation function is possible (cf., Contribution C1), but it is subject to error. At the same time, better approximations (e.g., packing the minimum bounding rectangles instead of nesting the original shapes) might be very time-consuming. Therefore, a highly accurate and also efficient anticipation method for complex nesting solutions would be valuable to improve solution quality and/or reduce computation time.

For solving the PSBIJF it is important to plan only with feasible batches (i.e., that the batched jobs do not exceed the sheet boundaries and do not overlap). Actually, a real nesting is not necessary, but the decision (or even a good suggestion) as to whether a batch is feasible or not is sufficient. Therefore, we propose the anticipation of base-level reactions by using machine learning (ML) to approximate the feasibility of batches (cf., Figure 4, II). In addition, Contribution C2 presents a prediction framework to identify the most promising machine learning method, with particular emphasis on the entire process of data preparation, learning pipeline configuration, hyperparameter tuning, validation, and testing to ensure transparency and reproducibility.

The experimental results show the great potential and suitability of the proposed ML-based anticipation approach and its superiority over simple approximations as anticipation functions.

Contribution C3

"Learning-augmented heuristics for scheduling parallel serial-batch processing machines".
Uzunoglu et al. (2023)

Contribution C3 focuses on the development and improvement of the solution methods. Two main aspects are addressed:

Due to the specific problem context (in particular the serial batching characteristic and the weighted tardiness objective), it is not necessarily appropriate to maximize batch utilization. As a result, batches with a lower utilization result in more batches with a shorter processing time and thus, batches can be scheduled more flexibly according to the due dates of the jobs. Of course, more batches also result in additional setups that delay batch start and completion times. This relationship is exploited by Contribution C1, wherein the β-parameter is used to control the batch utilization. Contribution C3 presents another batching strategy by controlling the jobs in a batch according to their "urgency", i.e., the priority of a job. For this purpose, a parameter δ is used, again in combination with the weighting of processing time (κ_1) and setup time (κ_2).

Since the setting of the heuristics' parameters (i.e., $\beta, \kappa_1, \kappa_2$ and $\delta, \kappa_1, \kappa_2$) has a significant influence on the solution quality, multi-start heuristic approaches with full-grid search are used. This leads to very competitive results in terms of solution quality, but also to high computation times for large-scale problem instances with hundreds or thousands of jobs (as can be found in some industries). Contribution C3 proposes new solution methods that are competitive in terms of solution quality and particularly efficient in solving large-scale problem instances. The main driver thereby is the use of machine learning, which significantly shortens the time-consuming enumerative search for promising parameter configurations. Again, special emphasis is placed on the complete data preparation, learning pipeline configuration, hyperparameter tuning, validation, and testing processes to ensure transparency and reproducibility.

Contribution C4

"Serial batch scheduling: a systematic review and future research directions". Wahl et al. (2023)

Contribution C4 has a broader view of batch scheduling problems compared to the other contributions. Its purpose is to structure the research field of (serial) batch scheduling problems, based on the authors' scheduling expertise and the experience and insights gained in Contributions C1, C2, and C3. The main goals of Contribution C4 are to establish a conceptual framework based on classification schemes for batch scheduling research and to provide a comprehensive knowledge base for theory and practice.

As discussed in the introduction, one of the challenges of batch scheduling problems is the interdependencies that arise from the relationship between the grouping of jobs into batches and the scheduling of those batches. In this context, batching and scheduling characteristics can lead to very different problems. This makes it difficult to identify relevant literature and to keep track of the state of the art. To overcome this difficulty, two classification schemes are developed: the "Scheduling Problem Classification Scheme" (SPCS) and the "Scheduling Article Classification Scheme" (SACS). As it is common in the context of scheduling problem classification, the SPCS – with the aim of comprehensively specifying batch scheduling problems – comprises three fields, namely "A - Machine characteristics" (e.g., machine environment), "B - Job and processing characteristics" (e.g., batching type or job availability), and "C - Objective system" (e.g., objective criteria). The SACS – for the purpose of describing research articles in detail – consists of five fields, namely "D - Theoretical insights" (e.g., problem complexity), "E - Model type" (e.g., mixed-integer linear program), "F - Solution method" (e.g., heuristic), "G - Experimental evaluation" (e.g., problem instance data), and "H - Application case" (e.g., manufacture of fabricated metal products). The development of the two classification schemes is based on existing classification schemes from the literature and, in particular, on an iterative process of literature search, analysis, and synthesis. This process identified 425 high-quality research articles on batch scheduling, the analysis of which led to iterative adaptations of the schemes and the detailed classification of the articles.
Based on the classification schemes and the classified articles, a systematic review of serial-batch scheduling problems is conducted. As the first review of serial-batch scheduling problems in the literature, 118 articles are analyzed, providing some interesting insights into the current state of research, identifying research gaps, and suggesting several future research directions.

3

SCIENTIFIC CONTRIBUTIONS

CONTRIBUTION C1

Scheduling parallel serial-batch processing machines with incompatible job families, sequence-dependent setup times and arbitrary sizes

Christian Gahm, Stefan Wahl and Axel Tuma*

Published in the International Journal of Production Research, 2021, 60(17), 5131–5154.

** Corresponding author*

Available via: https://doi.org/10.1080/00207543.2021.1951446

Abstract:

The scheduling of (parallel) serial-batch processing machines is a task arising in many industrial sectors. In the metal-processing industry for instance, cutting operations are necessary to fabricate varying metal pieces out of large base slides. Here, the (cutting) jobs have individual, arbitrary base slide capacity requirements (sizes), individual processing times and due dates, and specific material requirements (i.e. each job belongs to one specific job family, whereby jobs of different families cannot be processed within the same batch and thus are incompatible). In addition, switching of base metal slides and material dependent adjustments of machine parameters cause sequence-dependent setup times. All these conditions need to be considered while minimising total weighted tardiness. For solving the scheduling problem, a mixed-integer program and several tailor-made construction heuristics (enhanced by local search mechanisms) are presented. The experimental results show that problem instances with up to five machines and 60 jobs can be tackled using the optimisation model. The experiments on small and large problem instances (with up to 400 jobs) show that a purposefully used batch capacity limitation improves the solution quality remarkably. Applying the best heuristic to the data of two real-world application cases shows its huge potential to increase delivery reliability.

CONTRIBUTION C2

Applying machine learning for the anticipation of complex nesting solutions in hierarchical production planning

Christian Gahm, Aykut Uzunoglu, Stefan Wahl, Chantal Ganschinietz and Axel Tuma*

Published in the European Journal of Operational Research, 2022, 296(3), 819–836.

** Corresponding author*

Available via: https://doi.org/10.1016/j.ejor.2021.04.006

Abstract:

In hierarchical production planning, the consideration of interdependencies between superior top-level decisions and subordinate base-level decisions is essential. In this respect, the anticipation of base-level reactions is highly recommended. In this paper, we consider an example from the metal-processing industry: a serial-batch scheduling problem constitutes the top-level problem and a complex nesting problem constitutes the base-level problem. The top-level scheduling decision includes a batching decision, i.e., the determination of a set of small items to be cut out of a large slide. Thus, to evaluate the feasibility of a batch, the base-level nesting problem must be solved. Because solving nesting problems is time consuming even when applying heuristics, it is troublesome to solve it multiple times during solving the top-level scheduling problem. Instead, we propose an approximative anticipation of base-level reactions by machine learning to approximate batch feasibility. To that, we present a prediction framework to identify the most promising machine learning method for the prediction (regression) task. For applying these methods, we propose new feature vectors describing the characteristics of complex nesting problem instances. For training, validation, and testing, we present a new instance generation procedure that uses a set of 6,000 convex, concave, and complex shapes to generate 88,200 nesting instances. The testing results show that an artificial neural network achieves the lowest expected loss (root mean squared error). Depending on further assumptions, we can report that the approximate anticipation based on machine learning predictions leads to an appropriate batch feasibility decision for 98.8% of the nesting instances.

CONTRIBUTION C3

Learning-augmented heuristics for scheduling parallel serial-batch processing machines

Aykut Uzunoglu, Christian Gahm, Stefan Wahl and Axel Tuma*

Published in Computers & Operations Research, 2023, 151, 106122.

** Corresponding author*

Available via: https://doi.org/10.1016/j.cor.2022.106122

Abstract:

The addressed machine scheduling problem considers parallel machines with incompatible job families, sequence-dependent setup times, limited batch capacities, and arbitrary sizes combined with the serial-batch processing characteristic (i.e., the processing time of a batch is equal to the sum of processing times of all jobs grouped in a batch). The primary objective is the minimization of the total weighted tardiness, and a subordinate (secondary) objective is the minimization of the flow time. This scheduling problem arises in many production environments like cutting operations (metal-processing industry or garment industry) or in industrial 3D printing. For solving this problem, we propose a new multi-start construction heuristic with controlled batch urgencies. Furthermore, to improve solution efficiency, we use machine learning methods that are appropriate for multi-target regression with dependent outputs (i.e., Neural networks) to minimize the number of starts by predicting the most suitable heuristic parameters. Hereby, different learning aspects and pipeline parameters must be considered. Additionally, we apply a mixed-integer linear program and a local search mechanism with advanced termination criteria for solution improvement.

To evaluate the performance of the new heuristic, we use an exhaustive set of small, large, and very large instances (with symmetric Euclidean, asymmetric Euclidean, and arbitrary sequence-dependent setup times) and heuristics from the literature. The results indicate the superiority of the new, learning-augmented heuristics in terms of solution quality and computation times.

CONTRIBUTION C4

Serial-batch scheduling: a systematic review and future research directions

Stefan Wahl, Christian Gahm and Axel Tuma*

Submitted to the European Journal of Operational Research, 2023.

** Corresponding author*

The manuscript printed here represents the submitted version of Contribution C4
to the European Journal of Operational Research.

Abstract:

Cutting operations (e.g., laser cutting) and especially industrial 3D printing are two production processes that are becoming increasingly important in many manufacturing companies. The corresponding scheduling problems can be classified as serial-batch scheduling problem and hierarchical-batch scheduling problem. In serial-batch scheduling, the processing time of a batch is the sum of the processing times of the jobs grouped in a batch. In hierarchical-batch scheduling, which can be seen as a special form of serial-batch scheduling, the processing time of a batch results from sub-level decisions, such as solving a 3-dimensional nesting problem. Both scheduling problems are challenging because they involve considering the interdependencies between two major decisions: the grouping of jobs into batches and the scheduling of these batches. In addition, small variations in process characteristics can make one batch scheduling problem very different from another.

In this paper, we present a systematic review of the current state of the literature on serial-batch and hierarchical-batch scheduling. Systematic means that a structured, traceable, and repeatable search process has been applied and that each relevant, high-quality scientific article has been classified using two classification schemes (taxonomies). The used classification schemes allow an objective comparison of the literature, empirical analyses, the traceable elaboration of research gaps, and facilitate the identification of relevant research for new batch scheduling problems (for academics and practitioners). The analysis of 118 classified articles on serial- and hierarchical-batch scheduling provides some interesting insights into the current state of research and draws attention to various aspects that should be considered to improve research articles. Based on the analysis, we highlight several topics that might be interesting to explore in the future.

4

CONCLUSION

Key results and added value

A problem of high theoretical and practical relevance, which has not yet been considered in the literature, has been studied. The problem is specified and solved by a mixed-integer linear program, and several heuristic solution methods are developed. The main goal in the development of the solution methods was to achieve high solution efficiency, i.e., solution methods with competitive solution quality and low computation time. Regarding the solution quality, the heuristics employ different parameters that control the weighting of the processing time (κ_1) and the setup time (κ_2), as well as the batch utilization (β) and the urgency of the jobs (δ), respectively. Using multi-starts with full-grid search to find suitable parameter configurations for $\beta, \kappa_1, \kappa_2$ and $\delta, \kappa_1, \kappa_2$ leads to very competitive results, but also to high computation times for large-scale problem instances. Therefore, in terms of computation time, we propose to use machine learning to predict promising parameter configurations, which significantly shortens the time-consuming enumerative search. In general, the experimental results show a broad superiority of the newly developed heuristics compared to heuristics from the literature.

Solving the introduced PSBIJF actually requires solving a subordinate CNP several times. Since both problems are NP-hard, all presented solution methods use a simple approximation of the CNP. This approach is effective, but simple approximations are error-prone. To address this challenge, we propose to use machine learning to approximate the CNP. The experimental results show the great potential and suitability of the proposed ML-based anticipation approach and its superiority over simple approximations.

The proposed optimization program and the heuristic solution methods are not only applicable to the described laser cutting use case, but also to many other cutting or packing problems, and even to industries such as additive manufacturing (3D printing). Similarly, the application of machine learning techniques as an approximate anticipation function is not only applicable to the described use case of production scheduling but also to other hierarchical decision environments.

One of the challenges of batch scheduling problems is the interdependencies between the batching and the scheduling decisions. These interdependencies can vary and are strongly related to the specific problem context. To enable a comprehensive specification of batch scheduling problems and a detailed description of the research articles, two extensive classification schemes were developed, and 425 high-quality research articles were classified.
Based on the classification schemes and the classified articles, the first literature review on serial-batch scheduling problems is carried out. It includes 118 scientific articles, which are systematically analyzed. The analysis reveals several research gaps and points to many important and interesting future research directions.

The classification schemes used allow an objective comparison of the literature, empirical analysis, the elaboration of research gaps, and facilitate the identification of relevant research for new batch scheduling problems in a very efficient way. The classification schemes, together with the classified articles, provide a comprehensive and freely accessible knowledge base for researchers and industrial decision-makers.

Outlook and future research

Even though the developed heuristic solution methods achieve a high solution efficiency, sophisticated improvement methods or metaheuristics such as Variable neighborhood search or Genetic algorithms should be developed. The use of metaheuristics would also facilitate the investigation of different objective functions. In particular, since the production processes under consideration require substantial amounts of resources, the efficient use of resources (e.g., material or energy) and the elimination of waste should be investigated. The different objectives could be investigated individually, or in a multi-criteria approach to optimize multiple objectives simultaneously.

The laser cutting application case is very similar to the additive manufacturing application case. The results and findings from laser cutting should be applied to additive manufacturing. One of the first steps should be to synthesize and consolidate batch scheduling and additive manufacturing scheduling research to avoid reinventing scheduling methods and to unify terms and notations.

A common problem with knowledge bases (and literature reviews in particular) is that they quickly become out of date. They need to be maintained and updated. For this reason, it would be very useful to establish a collaborative web-based knowledge base for batch scheduling problems. Authors would contribute their own classified articles (according to standardized classification schemes) and a review process could ensure the required scientific quality of the knowledge base.

REFERENCES

Gahm, C., Uzunoglu, A., Wahl, S., Ganschinietz, C., & Tuma, A. (2022a). Applying machine learning for the anticipation of complex nesting solutions in hierarchical production planning. *European Journal of Operational Research*, *296*(*3)*, *819–836.* doi:10.1016/j.ejor.2021.04.006.

Gahm, C., Wahl, S., & Tuma, A. (2022b). Scheduling parallel serial-batch processing machines with incompatible job families, sequence-dependent setup times and arbitrary sizes. *International Journal of Production Research*, *60*(*17)*, *5131–5154.* doi:10.1080/00207543.2021.1951446.

Schneeweiss, C. (2003). Distributed decision making—a unified approach. *European Journal of Operational Research*, *150*(*2)*, *237–252.* doi:10.1016/S0377-2217(02)00501-5.

Uzunoglu, A., Gahm, C., Wahl, S., & Tuma, A. (2023). Learning-augmented heuristics for scheduling parallel serial-batch processing machines. *Computers & Operations Research*, *151*, *106122.* doi:10.1016/j.cor.2022.106122.

Wahl, S., Gahm, C., & Tuma, A. (2023). Serial batch scheduling: a systematic review and future research directions. *European Journal of Operational Research*, *(submitted)*.

Wäscher, G., Haußner, H., & Schumann, H. (2007). An improved typology of cutting and packing problems. *European Journal of Operational Research*, *183*(*3)*, *1109–1130.* doi:10.1016/j.ejor.2005.12.047.

APPENDIX

Serial-batch scheduling: a systematic review and future research directions

Stefan Wahl[a]
Christian Gahm[a*]
Axel Tuma[a]

[a] Chair of Business Administration, Production & Supply Chain Management, University of Augsburg, 86159 Augsburg, Germany

* Corresponding author:
E-mail: christian.gahm@wiwi.uni-augsburg.de; Phone: +49-821-598-4359

Serial-batch scheduling: a systematic review and future research directions

Abstract:

Cutting operations (e.g., laser cutting) and especially industrial 3D printing are two production processes that are becoming increasingly important in many manufacturing companies. The corresponding scheduling problems can be classified as serial-batch scheduling problem and hierarchical-batch scheduling problem. In serial-batch scheduling, the processing time of a batch is the sum of the processing times of the jobs grouped in a batch. In hierarchical-batch scheduling, which can be seen as a special form of serial-batch scheduling, the processing time of a batch results from sub-level decisions, such as solving a 3-dimensional nesting problem. Both scheduling problems are challenging because they involve considering the interdependencies between two major decisions: the grouping of jobs into batches and the scheduling of these batches. In addition, small variations in process characteristics can make one batch scheduling problem very different from another.

In this paper, we present a systematic review of the current state of the literature on serial-batch and hierarchical-batch scheduling. Systematic means that a structured, traceable, and repeatable search process has been applied and that each relevant, high-quality scientific article has been classified using two classification schemes (taxonomies). The used classification schemes allow an objective comparison of the literature, empirical analyses, the traceable elaboration of research gaps, and facilitate the identification of relevant research for new batch scheduling problems (for academics and practitioners). The analysis of 118 classified articles on serial- and hierarchical-batch scheduling provides some interesting insights into the current state of research and draws attention to various aspects that should be considered to improve research articles. Based on the analysis, we highlight several topics that might be interesting to explore in the future.

Keywords:

Scheduling, serial-batching, review, classification scheme

1 Introduction

Batch processing machines are widely used in many manufacturing industries, such as the semiconductor industry (e.g., burn-in operations), metalworking (e.g., laser cutting), or in industrial 3D printing. In general, batch processing is an important technique for improving production efficiency by processing jobs in groups rather than individually. The decision to group jobs together is called "batching" and a resulting group of jobs is called a "batch". The scheduling of batch processing machines (batch scheduling) involves two decisions: the batching decision (grouping jobs into batches) and the scheduling decision (assigning batches to machines – in the case of multiple machines – and sequencing the batches). Since the composition of the batches strongly influences the scheduling decision, the batching and scheduling decisions should be made considering their interdependencies. These interdependencies can vary and are strongly related to the respective problem context, i.e., the machine characteristics, the job and processing characteristics, and the scheduling objectives. In terms of processing characteristics, batching problems can typically be described by "batching type", "job availability", "batch capacity" and "job families":

i. batching type

- parallel batching (*pb*): the processing of jobs is performed in parallel, and the processing time of a batch is equal to the longest processing time of the jobs in the batch. (e.g., parallel burn-in operations in the semiconductor industry, cf., e.g., Yang et al., 2022).
- serial batching (*sb*): the processing of jobs is performed in series and the processing time of a batch is the sum of the processing times of the jobs in the batch (e.g., laser cutting in the metalworking industry; cf., e.g., Gahm et al., 2022b).
- fixed batching (*fb*): the processing time of a batch is independent of the jobs in a batch, i.e., given by a "fixed" value (cf., e.g., Sung & Kim, 2003).
- mixed batching (*mb*): the processing time of a batch is composed by a *pb*-component and a *sb*-component (Wang et al., 2020).
- hierarchical batching (*hb*): the batch processing time depends on the result of a sub-level decision (e.g., in the case of industrial 3D printing, the batch processing time depends on the nesting of the parts assigned to a batch; cf., e.g., Zehetner & Gansterer, 2022). Since the sub-level decision, and thus the resulting batch processing time, most often depends on job characteristics, *hb* can be seen as a special type of *sb*. However, we introduce this type of batching to highlight the particular hierarchical dependency between the top-level decision and the base-level decision (on such a hierarchical dependency see e.g., Gahm et al., 2022a).

ii. job availability

- batch availability (*ba*): a job is not available until its entire batch is completed.
- item availability (*ia*): a job is available immediately after its processing is completed (cf., e.g., Shen & Buscher, 2012).

iii. batch capacity

- unbounded batch capacity: any number of jobs can be grouped into a batch.
- bounded batch capacity: the grouping of jobs can be restricted in different ways, e.g., by a limited number of jobs (cf., e.g., Li, 2017), where the batch capacity requirement of a job is equal to one (*cr1*), or by a maximum total size per batch (cf., e.g., Muter, 2020), where job-dependent batch capacity requirements are given (*crJ*). In both cases, further characteristics such as family-related (*bF*) or machine-related (*bM*) upper bounds and batch lower bounds (*bLb*) can be given.

iv. job families

- compatible job families (*cf*): jobs belonging to different compatible job families may be processed in the same batch (cf., e.g., Bellanger & Oulamara, 2009).
- incompatible job families (*if*): jobs belonging to different incompatible job families cannot be processed in the same batch (cf., e.g., Chakhlevitch et al., 2011).

In addition to these main characteristics, there are several others, such as "release dates" or "setup times", which can make two batch scheduling problems very different. Overall, this leads to a wide variety of batch scheduling problems discussed in the literature.

To analyze the current state of research, we use classification schemes. Classification schemes offer several advantages, especially in the area of scheduling (cf., e.g., Herroelen et al., 1999). They are particularly helpful in enabling an objective comparison of the literature, conducting empirical analyses, elaborating research gaps, and facilitating the identification of relevant research for new batch scheduling problems (for academics and practitioners). For this purpose, two classification schemes are used: The first classification scheme is an adapted and extended scheduling problem classification scheme (SPCS) to comprehensively specify batch scheduling problems. The second classification scheme is a completely new scheduling article classification scheme (SACS), developed to describe theoretical insights, solution methods, and other important aspects. The development of both classification schemes is based on existing classification schemes from the literature and on an iterative process of literature search, analysis, and synthesis. In this process, all types of batching (cf., i.) are considered, and 425 research articles have been analyzed in detail. In the analytical part of this paper (starting with section 5) we focus on *sb*- and *hb*-scheduling research. For *pb*-scheduling research, please see the recently published survey by Fowler & Mönch (2022). Since *fb*- and *mb*-scheduling play a minor role in theory and practice, these are not part of our analysis. However, the entire data and analysis of all 425 classified articles is provided for download at Mendeley Data (along with detailed descriptions of the methodology and the classification schemes; see Wahl et al., 2023).

In summary, our article makes the following contributions to literature:

- We present two comprehensive classifications schemes for specifying batch scheduling problems and scheduling research articles.
- We provide the first systematic review of *sb*- and *hb*-scheduling research. The analysis of 118 scientific articles gives insights and elaborates several future research directions.

- The detailed documentation of the applied methodology provides the foundation for high-quality reviews as it ensures transparency and reproducibility.

The remainder of this paper is organized as follows: In section 2, we document the methodology and scope of the review. Section 3 describes basics and the related literature on classification schemes and batch scheduling. Section 4 briefly introduces the proposed classification schemes. In section 5, we analyze the current state of *sb*- and *hb*-scheduling research and elaborate future research directions that are summarized in section 6. The paper closes with final remarks in section 7.

2 Methodology and scope

The methodology of the systematic literature review, the development of the classification schemes and the classification of the articles is illustrated in Figure 1. It is based on general guidelines for literature reviews and guidelines for the design of research frameworks described in Webster & Watson (2002), vom Brocke et al. (2009), and Gahm et al. (2016). The implementation of these guidelines is carried out in a six-step process, which is briefly explained in the following (the complete literature search process is described in full detail in Wahl et al., 2023).

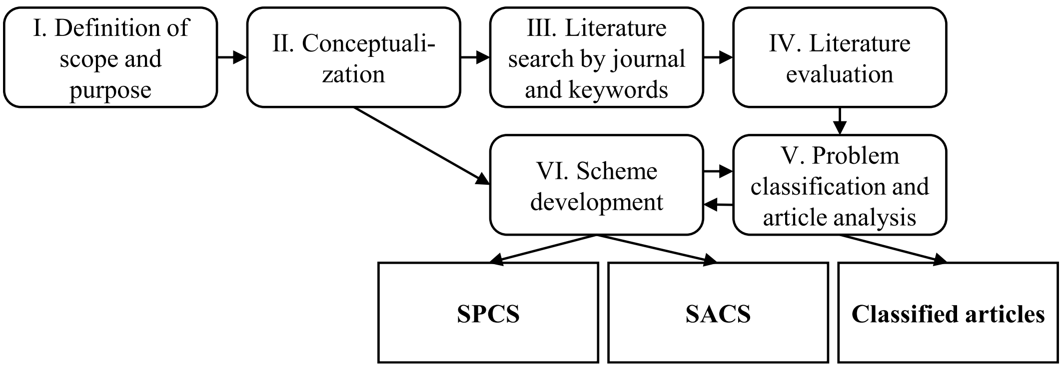

Figure 1: Methodology of the classification scheme development

In the first step (I.), we define scope and purpose: The scope of our analysis is problems with integrated, active batching and scheduling decisions (e.g., the scheduling of given batches is not included in the scope; please note, that we do not restrict our literature search to *sb*- and *hb*-scheduling research, but also consider *pb*-, *fb*-, and *mb*-scheduling). The purpose is to provide a systematic review and analysis of the current state of research on *sb*- and *hb*-scheduling problems.

In the deductive conceptualization step (II.), we examine a basic literature sample consisting of the reviews discussed in section 3. This literature sample also builds the groundwork for identifying most relevant journals and keywords, which are used to structure the subsequent literature search.

For the literature search by journal and key words (III.), we only consider scientific articles that are published in the English language, have undergone a peer review process, and that can be found in renowned journals (with an SJR-index not less than 1.00 or an h-index not less than 75). The literature search resulted in 1,319 initial “hits” in 60 journals (last update: February 2023).

In the literature evaluation (IV.), we examine the search results for compatibility with the scope. For this purpose, we check title, keywords, abstract (and full text) and then decide whether an article matches the scope and will be analyzed further or whether it will be discarded. Altogether, a total of 425 articles in 33 journals were identified as relevant and analyzed in detail. Due to the large number of relevant articles, we refrain from an explicit forward and backward search.

Step five (V.) comprises the problem classification and article analysis. Here, we analyzed the relevant articles in detail and classified them according to both classification schemes. Whenever appropriate, we adjusted and adapted both schemes by adding or rearranging dimensions, categories, and attributes (VI.). The resulting versions of both schemes are then used to finally complete and update the classification of all relevant articles.

These six steps lead to a comprehensive knowledge base consisting of the SPCS, the SACS, and the fully classified research articles.

3 Basics and related literature

Generally, scheduling problems belong to the class of combinatorial search problems. Hereby, a combinatorial search problem Π is a set of pairs (*I, S*), where *I* represents a problem instance (i.e., a finite set of parameters with given values) and *S* a (feasible) solution for instance *I* (Błażewicz et al., 2019). For calculating solutions (scheduling), exact algorithms, heuristics, metaheuristics, and other solution methods are applied to assign scarce resources (machines, processors, workers, etc.) to jobs (production orders, products, etc.). This may require interdependent decisions, as in the case of batch scheduling. Whenever more than one processing step is required to execute a job, several operations (tasks, processing steps) must be scheduled, and the term "station" is used to indicate the machines (resources) dedicated to a subset of operations (belonging to the different jobs). Furthermore, jobs, operations, and resources can be grouped in "families" if they have similar characteristics that are relevant for scheduling.

Regarding the research area of batch scheduling, several reviews exist: Potts & van Wassenhove (1992) describe a general model to capture the notions of batching and lot-sizing. For this purpose, they adapt the $\alpha|\beta|\gamma$ classification scheme of a technical report published by Lawler et. al in 1989 to their general model and review articles on problems that integrate scheduling with batching or lot-sizing, respectively. Their focus is on the specification of problems, and the analysis of problem and solution method complexity. Webster & Baker (1995) review the three basic models "family scheduling with item availability", "family scheduling with batch availability", and "batch processing" with a single machine. Their family scheduling models correspond to an unbounded serial-batch scheduling model with setup times. The batch processing model investigates constant batch processing times and parallel batching. The authors provide several insights concerning the objectives minimizing total weighted flowtime and minimizing maximum lateness (amongst others). To indicate different types of batching, Jordan (1996) introduces an entry in the β-field differentiating between "family scheduling", "item availability-preemptive batching", "item availability-nonpreemptive batching", and “batch availability”. Based on this differentiation, the author discusses solution methods for the single machine and parallel machine case. Potts & Kovalyov (2000) update the reviews of Potts & van Wassenhove (1992) and Webster & Baker (1995). They adapt the $\alpha|\beta|\gamma$ classification scheme of Graham et al. (1979) and detail the investigation of the family scheduling models and the batch processing model. For a selection of problems, they analyze their complexity and solution methods, and their efficiency and effectiveness. They also take up approximation algorithms and their worst-case performance. Mathirajan & Sivakumar (2006) classify and analyze literature dealing with the scheduling of batch processing machines in semiconductor manufacturing. They introduce two schemes that are different from the three-field classification. The first scheme explicitly addresses problem characteristics from semiconductor manufacturing, whereby the second scheme is related to solution methods. In a meta-analysis, they match articles from the literature (published between 1986 and October 2004) according to their proposed schemes and identify potential research areas. Allahverdi et al. (2008) provide an extensive survey on scheduling literature that explicitly considers setup times not already classified in Allahverdi et al. (1999) or Potts & Kovalyov (2000). They review static, dynamic, deterministic, and stochastic problems and categorize them according to shop environments, non-batching and batching considerations, and sequence-independent and sequence-dependent setup times. Regarding setup times, they adapt the three-field notation of Graham et al. (1979). In their survey, the authors give concise summaries of the problem, the objective criteria, and the solution approach or result. Mönch et al. (2011) present a survey on scheduling semiconductor manufacturing operations. They focus on very specific process characteristics and introduce corresponding entries in their β-field. However, also batching related characteristics are depicted. Wu (2014) proposes a taxonomy of batch queuing models that differentiates between transfer and process batches, whereby the latter one is further separated into “parallel batch” and “serial batch” problems. Altogether, the author considers eight different problem settings and discusses queuing models and simulation results according to those settings. In a third survey on scheduling with setups, Allahverdi (2015) reviews about 500 papers including static, dynamic, deterministic, and stochastic settings. However, problems with batching decisions are explicitly excluded. To structure their literature review, Gahm et al. (2022b) extend and adapt the classification scheme of Potts & Kovalyov (2000) by several new, batching-related attributes in the β-field. This enables the authors to efficiently describe and compare the most relevant literature in the context of scheduling parallel serial-batch processing machines with incompatible job families, sequence-dependent setup times and arbitrary sizes. In a recent survey, Fowler & Mönch (2022) discuss literature on parallel batch processing with bounded batch capacities and a focus on offline deterministic scheduling approaches. Their survey considers the makespan and flow time- and due date-related performance measures but also multi-criteria settings. For classifying a problem, a taxonomy based on the notations developed by Graham et al. (1979) and Pinedo (2016) is used and their literature analysis is structured according to the type of job families, machine environment, and performance measure.

As this summary of the literature review shows, there is no current and comprehensive analysis of *sb*- and *hb*-scheduling research. As a result, our review serves to fill this research gap.

4 Applied classification schemes

To analyze the current state of research in a specific area, literature reviews based on classifications schemes (taxonomies or research frameworks) are very helpful as they can serve different purposes (cf., e.g., Herroelen et al., 1999 and Ganschinietz, 2021):

- A classification scheme greatly simplifies the presentation, discussion, and comparison of scheduling problems by immediately highlighting the basic problem characteristics in an objective manner (through well-defined terms and notations).
- A classification scheme facilitates the identification of relevant research for new (batch) scheduling problems (for academics and practitioners).
- A classification scheme allows for an easy and traceable elaboration of viable research topics by the identification of open problems.
- A classification scheme comprises the most relevant aspects to be considered and helps to analyze and structure individual scheduling problems and research articles.

Summarizing, the usage of up-to-date classification schemes is important and thus, we present an adapted and extended "Scheduling Problem Classification Scheme" (SPCS) to comprehensively specify (batch) scheduling problems. It provides several extensions and adaptations of existing classification schemes to reflect recent trends and developments in (batch) scheduling. Furthermore, we propose a more generic concept to simplify further extensions and adaptations.

Besides the SPCS, we propose a second, completely new classification scheme to describe research articles in detail, the "Scheduling Article Classification Scheme" (SACS).

4.1 Basic structures

Both classification schemes are structured in a strong hierarchical tree structure using different elements: "fields" are used to separate different aspects (dimensions), "categories" are used to structure the fields, and "sub-categories" are introduced to flexibly structure categories whenever helpful. For instance, the field "C - Objective system" contains the category "C1 Objective criteria" that is further divided into the sub-categories "C1.1 Time-oriented" criteria, "C1.2 Resource-oriented" criteria, and "C1.3 Finance-oriented" criteria. For the classification of a scheduling problem (or a research article) itself, we provide "binary" attributes. Binary means that it should be possible to clearly decide if an attribute is "true" for a scheduling problem (or a research article) or if an attribute is "false". Hereby, it is important that each attribute has a unique and meaningful name.

For the SPCS, we additionally define unique abbreviations that allow scientists and practitioners to specify a (batch) scheduling problem by a single "classification string" (as it is possible with the existing classification schemes).

Unlike existing classification schemes, we do not use Greek letters, subscripts, superscripts, or any special characters in the attribute abbreviations to make them easier to type (especially for practitioners). Furthermore, we introduce the convention that abbreviations of attributes belonging to the same category should start with the same lowercase character (e.g., "p" for processing time or "s" for setups) whenever reasonable. In addition, we use specific capital letters to indicate specific relations (dependencies) of an attribute to a single object or a set of objects (e.g., jobs or machines): "J" indicates a relationship to jobs, "O" indicates a relationship to operations, "B" indicates a relationship to batches, "F" indicates a relationship to job families, "M" indicates a relationship to machines (or stations), "S" signals a dependence on the sequence, and "C" stands for a common property. These letters can also be combined to express multiple dependencies. By using these conventions, it is easy to understand that the attribute "pF" marks processing times that are family related or that "sMS" indicates machine- and sequence-dependent setup times. In consequence, new attributes can be easily defined, and their context (category and dependencies) can be derived from their reasonably defined abbreviation.

4.2 Scheduling problem classification scheme (SPCS)

The first three-field ($\alpha|\beta|\gamma$) classification scheme to specify scheduling problems was proposed by Graham et al. in 1979. They use the α -field to define the "machine environment", the β-field to define the "job characteristics", and the γ-field to define the "optimality criteria". Blazewicz et al. (1983) extend this classification scheme by allowing the jobs to demand for additional scarce resources. Lawler et al. (1993) take up the idea of the three fields: While α and γ remain largely the same (this generally applies for most notations), some changes take place around the characteristics of the β-field. The three-field classification approach also found its way into standard text books like the ones of Brucker (2007), Pinedo (2016), and Błażewicz et al. (2019) and many extensions and adaptations for different research areas have been made: e.g., for project scheduling (cf., e.g., Brucker et al., 1999 or Herroelen et al., 1999) or scheduling semiconductor manufacturing operations (cf., Mönch et al., 2011). Because most readers are familiar with the general classification schemes cited above, we omit their detailed description here.

Based on the analysis of previously published reviews and classification schemes in (batch) scheduling, we follow the traditional and established approach and propose three fields to classify (batch) scheduling problems. However, we break with the tradition of using α, β, and γ, but use A, B, and C to represent the three fields "A - Machine characteristics", "B - Job and processing characteristics", and "C - Objective system". This change is made for two reasons: First, we generally omit Greek letters in both classification schemes to make the typing easier. Second, the SACS includes the fields D to H and thus, a consistent naming is achieved.

To keep this paper to a manageable size, we omit a detailed description of the SPCS here and refer the reader to the full description in Wahl et al. (2023). However, we list the fields, categories, sub-categories, and attributes in the following sections. Note that most categories have a default attribute (abbreviated as "–"), indicating either that the problem does not have a special characteristic in this category, or that the category is not relevant at all. In both cases, the abbreviation is not part of the classification string.

4.2.1 A - Machine characteristics

The field A - Machine characteristics is not only used to specify the basic machine environment and the number of machines but also to inform about their availability. Therefore, we use three categories for the classification:

- A1 Machine environment {*S, P, Q, R, F, HF, PF, J, HJ, O*}
- A2 Number of machines {–, *m=2, m=3*}
- A3 Machine availability {–, *avSt, avDyn, avFlx, avPer, avStoc, avState*}

The newly introduced category A3 provides attributes for machine availability characteristics: e.g., dynamic unavailability (*avDyn*; e.g., due to shift calendars), flexible unavailability (*avFlx*; e.g., maintenance activities to be scheduled). Also, machine state characteristics (*avState;* e.g., idle or processing) can be characterized here.

4.2.2 B - Job and processing characteristics

The B-field specifies job (operation) and processing characteristics. The categories B1 to B6 provide job attributes and the categories B7 to B14 provide processing attributes.

- B1 Processing times and intensities {–, *pF, pC, p1, pJM, pCM, pDet, pLe, pCo, …*}
- B2 Release dates {–, *rJ, …*}
- B3 Due dates and deadlines {–, *dJ, dC, da, daC, dlJ, dlC, …*}
  Note that *da* and *daC* are used to classify due date assignment problems.
- B4 Precedence relations {–, *chain, in-tree, out-tree, net, alt, re*}
  Note that *alt* and *re* are used to classify alternative and re-entrant routings, respectively.
- B5 Job families {–, *if*, *if2*, *cf*, …}
- B6 Batch capacity requirement {–, *crO*, *crJ*, *crF*, *crC*, *cr1*}
- B7 Setup and removal {–, *s*, *sF*, *sM*, *sS*, *sFM*, *sFS*, *sFMS*, *sDet*, *sCo*, …}
- B8 Timing constraints {–, *minL, maxL, now, …*}
  Note that *minL* (*maxL*) are used to classify minimum (maximum) time lags and *now* for no-wait.
- B9 Execution characteristics {–, *blk, buf, cb, dcr, elig, mode, res, …*}
  The execution characteristics contain specials like blocking (*blk*), the requirement of consistent batches (*cb*) on all stages of a flow shop, or machine eligibility constraints (*elig*).
- B10 Batching type {–, *pb, sb, fb, mb, hb*}
- B11 Job availability {–, *ia*}
  Note that batch availability (*ba*) is considered the default case here and is therefore omitted.
- B12 Batch capacity {–, *bF, bM, bFM, bLb*}
- B13 Transportation {–, *tInb, tInt, tOub*}

- B14 Information type {–, *fuzzP, on, on-sc, …*}

  The information type contains characteristics like fuzzy processing times (*fuzzP*) or on-line scheduling (*on*).

4.2.3 **C - Objective system**

Since the consideration of multiple objective criteria has received increasing attention in recent years, we propose to specify not only the objective criterion, such as makespan or weighted tardiness, but also the model that is used in the case of multiple criteria. In the “C1 Objective criteria” category, we use three sub-categories to simplify the identification of attributes, i.e., the specific objective criteria: “C1.1 Time-oriented” (e.g., *wT* for the total weighted tardiness), “C1.2 Resource-oriented” (e.g., *Cmax* for the makespan), “C1.3 Finance-oriented” (e.g., *cRc* for resource consumption costs), and “C1.4 Other” (e.g., *FY* for a feasibility problem). In the “C2 Multi-criteria model” category, the way in which the objective system handles multiple criteria can be specified (e.g., *ParA* indicates Pareto optimization with two agents, cf., Feng et al., 2013).

4.3 Scheduling article classification scheme (SACS)

Besides the detailed specification of batch scheduling problems by the SPCS, further aspects are important for structuring and analyzing scheduling research. For this purpose, we propose the SACS, which consists of five fields: “D - Theoretical insights”, “E - Model type”, “F - Solution method”, “G - Experimental evaluation”, and “H - Application case”. These fields are used because they represent the most important aspects for researchers (e.g., to identify the complexity of a problem, which is classified in field D) and industry decision-makers (e.g., to identify whether a suitable solution method exists for solving problem instances of a certain size, which is specified by fields F and G) to assess the relevance of an article.

4.3.1 **D - Theoretical insights**

Field D provides several categories and attributes to describe different findings common to (batch) scheduling problems. For example, the complexity of a problem (*NP* or *sNP*), dominance properties, *lower bounds*, or the definition of a *worst-case performance* or *competitive ratio*.

4.3.2 **E - Model type**

Besides the basic model type (e.g., mixed-integer linear program *MILP*, mixed-integer non-linear program *MINLP*, or queuing model *QM*), also the type of the main decision variables can be classified in this field: sequence-based, position-based, time-indexed, time point-based or network (graph) model.

4.3.3 **F - Solution method**

The F-field is structured by ten categories: F1 Standard solver, F2 Exact solution method, F3 Heuristic, F4 Metaheuristic, F5 Matheuristic, F6 Simulation, F7 Robust scheduling method, F8 Learning-augmented scheduling method, F9 Parameter tuning, and F10 Advanced computing.

4.3.4 G - Experimental evaluation

The experimental investigation and evaluation of specific problem characteristics, optimization programs, solution methods, instance parameters, etc. is an essential part of most research articles dealing with (batch) scheduling problems. Three categories are used to analyze and structure this aspect: G1 Evaluation scope (*Solution method(s), Objective system, Lower/upper bounds, Optimization models*), G2 Performance assessment, and G3 Instance data.

4.3.5 H - Application case

Many research articles discuss (batch) scheduling problems that are related to specific application cases. To enable a consistent specification of application cases, "The International Standard Classification of All Economic Activities" (ISIC; cf., United Nations, Department of Economic and Social Affairs, 2008) is used. In addition to the division related attributes of *Section C - Manufacturing*, we add the attributes *Additive manufacturing (3D printing)* and *Cloud manufacturing and collaborative production* as these applications are not included in the most recently published ISIC version.

5 Current state of *sb*- and *hb*-scheduling research

In this part of the literature review, we analyze the current state of research based on both classification schemes. The analysis is based on the 118 *sb*- and *hb*-scheduling related articles listed in Table 1 through Table 5 in Appendix A. These tables report the complete problem specification according to the SPCS and some major SACS-attributes (the complete classifications of all 118 *sb*- and *hb*-articles are provided by spreadsheets in the supplementary material).

Since research gaps build the basis for identifying further research directions, we focus not only on problems and characteristics that have already been investigated, but also on those that have not yet been considered. In the following, we focus on what we consider to be the most interesting topics. For further and individual analysis, the data made available for download can be used.

5.1 Analysis of problem characteristics

The temporal development of *sb*- and *hb*-scheduling research, illustrated in Figure 2, shows that the number of articles published has remained almost constant over the last years (the values in brackets indicate the total number).

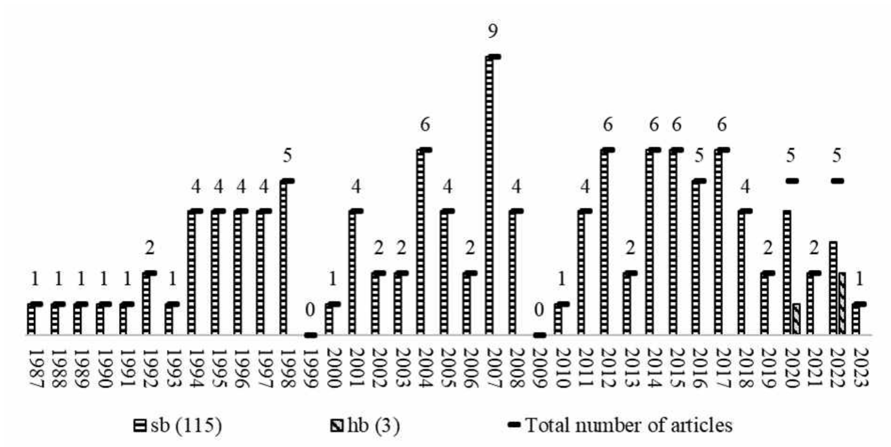

Figure 2: Temporal development of the number of articles published (per year)

All three *hb*-scheduling related articles are dedicated to the use case of *Additive manufacturing* (AM). This research stream has only recently started, but due to the increasing importance of AM, this research area will be of great importance in the future (see R1 in section 6 Further research directions).

5.1.1 Machine environment (A1) and Machine availability (A3)

Machine environment and related machine availability characteristics are presented in Figure 3. Most approaches consider single machine environments (*S*) and parallel identical machines (*P*), flow shops (*F*), and hybrid flow shops (*HF*) are also very often considered. However, research on job shop (*J*) or hybrid job shop (*HJ*) machine environments is relatively sparse and offers potential for further research (R6).

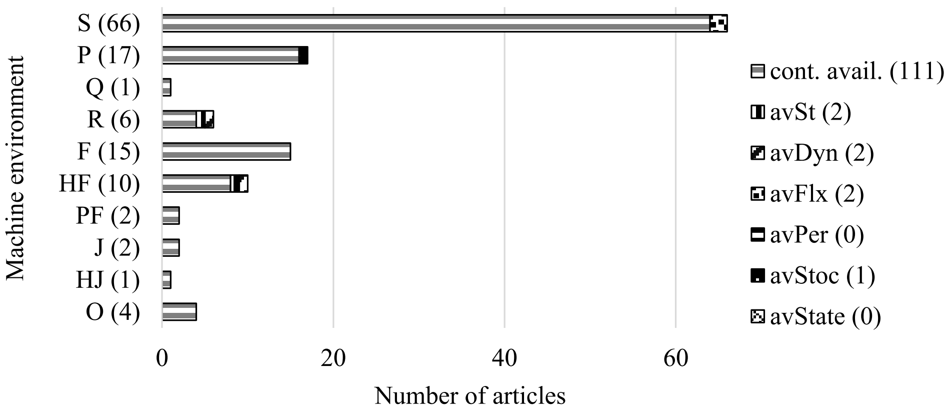

Figure 3: Machine environment (A1) and Machine availability (A3)

Most authors addressing a multi-stage machine environment (*F*, *HF*, *PF*, *J*, *HJ*, *O*) consider batch-processing machines at more than one stage: *F2* – Glass et al. (2001), Lin & Cheng (2001), Kovalyov et al. (2004), *F* – Ng & Kovalyov (2007), Shen & Gupta (2018), Quadt & Kuhn (2007), *HF2* – Yu et al. (2017), *HF* – Voß & Witt (2007), Shahvari & Logendran (2016), *PF* – Hakim Halim & Ohta (1993), Mosheiov & Oron (2005), *J2* – Mosheiov & Oron (2011), *J* – Shen & Buscher (2012), *HJ* – Castillo & Gazmuri (2015), *O2* – Gribkovskaia et al. (2006), and *O* – Mosheiov & Oron (2008b), Lin & Cheng (2011). The special case of combining different batching types per stage is considered by

Oulamara (2007), Muthuswamy et al. (2012), and Zhou et al. (2016), who all consider two-machine flow-shops (*F2*) with one *pb*-stage and one *sb*-stage.

Regarding the machine availability characteristics, Figure 3 shows that almost all authors consider a continuous machine availability from the beginning of the planning horizon. Machine availabilities related to given start times (*avSt*) or dynamic unavailability (*avDyn*; e.g., due to break calendars) are rarely considered. The scheduling of maintenance activities, represented by flexible unavailability (*avFlx*) and periodic repetitive unavailability (*avPer*), may become more important in the future as "predictive maintenance" is an emerging topic (R7). Overall, only Pei et al. (2016) consider stochastic machine unavailability (*avStoc*; caused by machine breakdowns). An aspect not yet been considered, but particularly important for energy-efficient scheduling (cf., Gahm et al., 2016), is machine state characteristics (*avState;* e.g., idle or processing). In general, too little attention is paid to the topic of energy-related scheduling. Therefore, it is an important research topic for the future (R2).

5.1.2 Job availability (B11) and Setup and removal (B7)

The job availability (B11) characteristics, which include batch availability (*ba*) and item availability (*ia*), are shown in Figure 4. As setups are related to this aspect, the setup types considered are also shown. Note that since some attributes are non-exclusive (*sDet* - setup times with deterioration effect; *sCo* - setup times are controllable or "compressible"; *sAnt* - anticipatory setups), the total amount per row is not equal to the amount per job availability type.

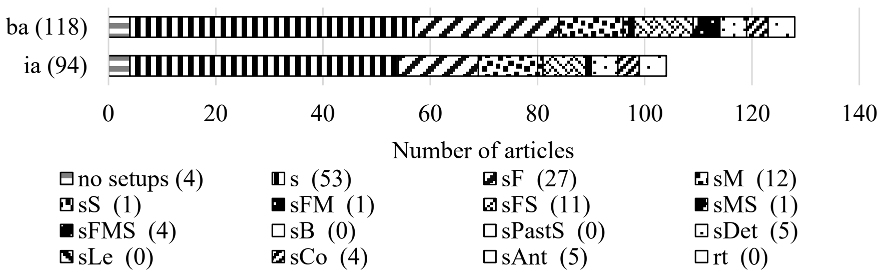

Figure 4: Job availability (B11) and Setup and removal (B7)

The strong relationship between batch scheduling and setups can be clearly seen in Figure 4 (only four articles do not consider setups at all), and the variety of considered setup types indicates their importance for *sb*- and *hb*-scheduling. It also shows that item availability is always related to setups (otherwise no batching decisions are necessary). Currently, neither batch-related setups (*sB*; e.g., related to the batch size), past sequence-dependent setups (*sPastS*), setups with learning effects (*sLe*), nor removal times (*rt*) are investigated.

5.1.3 Batch capacity (B12) and Batch capacity requirement (B6)

Regarding batch capacity, we can report that most articles consider problems with unbounded batches (80). Regarding bounded batches, most of the 42 articles consider maximum capacity bounds (38), but only Castillo & Gazmuri (2015) consider a job family-related bound (*bF*) and Muthuswamy et al. (2012) consider a machine-related bound (*bM*). Bounds related to job families and to machines (*bFM*) are not considered at all and seven articles account for a lower bound on the batch capacity (*bLb*).

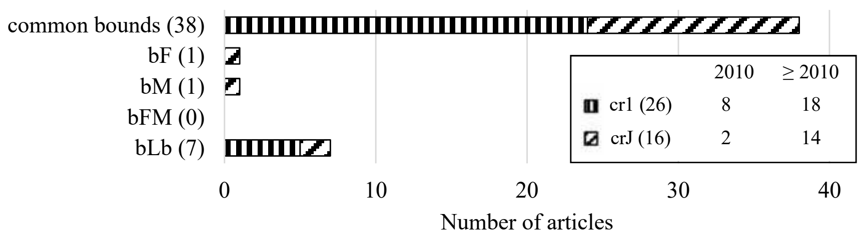

Figure 5: Bounded batch capacities (B12) and Batch capacity requirement (B6)

As far as bounds on the batch capacity are concerned, we observe that the consideration of the number of jobs in a batch (with a batch capacity requirement of one; *cr1*) and individual batch capacity requirements per job (*crJ*) has increased since 2010 (about 76% of these 42 articles have been published since then; cf., Figure 5). This suggests that bounded batch capacity is of increasing interest.

5.1.4 Job families (B5)

Job families, both compatible (*cf*) and incompatible (*if*), play an important role in batch scheduling: of the 118 articles analyzed, 42% (49) consider job families, of which 39% (46) consider incompatible job families and 3% (3) consider compatible job families. Regarding the articles that consider exactly two incompatible job families (*if2*), we observe that they are mostly (6 out of 7) related to problems with two "competing agents" (e.g., representing two types of customers; see, e.g., Mor & Mosheiov, 2011).

In general, job families are a very flexible concept that can be used to represent not only customer groups, but also, for example, part types (Cheng et al., 2004), material types (Uzunoglu et al., 2023), or recipes (Castillo & Gazmuri, 2015).

5.1.5 Job and processing characteristics (B)

The large variety of job and processing characteristics examined is best illustrated by Figure 6: leaving aside *sb* and *hb*, 53 different job and processing characteristics from field B are considered by at least one article. Furthermore, only four pairs of articles deal with the same scheduling problem (according to their SPCS-classification). We can therefore conclude that the research field of *sb*- and *hb*-scheduling is very heterogenous.

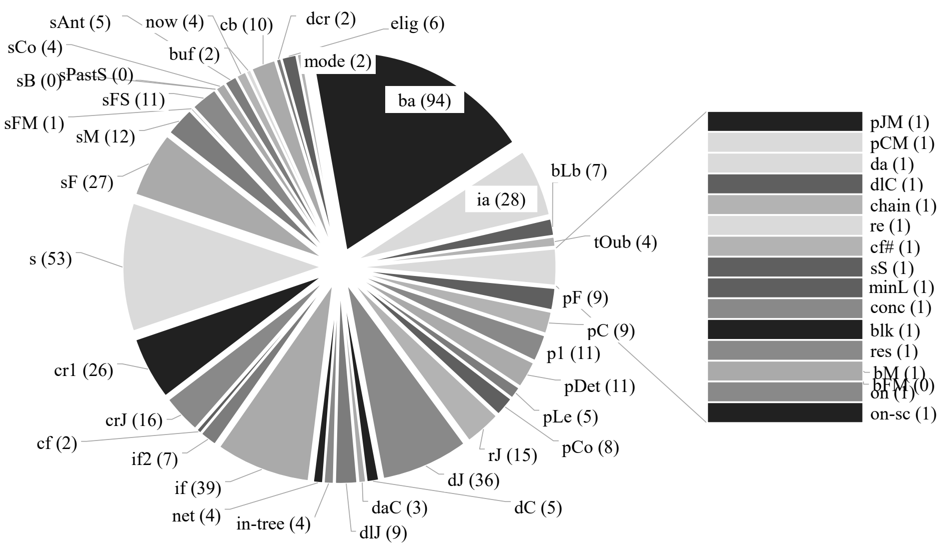

Figure 6: Considered problem characteristics (number of articles)

However, we can also see that some job characteristics are rarely considered, e.g., minimum time lags (*minL*) or blocking (*blk*), and others are not considered at all, e.g., maximum time lags (*maxL*) or work contents (also called "work packages" or "variable intensity operations"). The latter aspect in particular may be of interest in the future (R9). A hardly considered topic is the assembly production environment represented by the *in-tree* attribute. This is somewhat surprising as *sb*-processing machines are often used to produce parts that need to be assembled into components or final products in subsequent production stages. So far, only Kovalyov et al. (2004), Lin et al. (2007), Hwang & Lin (2012), and Liao et al. (2015) have considered assembly processes (all in two-stage flow-shop production systems). Therefore, we conclude that this is another interesting research topic (R5).

Another topic that is hardly considered is transportation. Only four articles consider outbound transports (*tOub*; e.g., transport from the production to the customer, Pei et al., 2015). Inbound transports (e.g., transports from the warehouse to the shop floor) and intermediate transports (e.g., between machines) are not considered. Consequently, the integrated consideration of scheduling and transport planning is another topic for future research (R11).

A further observation is that almost all articles assume a deterministic environment (116): only one article considers on-line scheduling (*on*; clairvoyant, i.e., all the parameters of a job are revealed at the stochastic arrival of a job; Giglio, 2015); one article considers semi-clairvoyant on-line scheduling (*on-sc*; i.e., only some parameters or approximate knowledge are known at the stochastic arrival of a job; Wu, 2014), and fuzzy parameters or stochastic parameters are not considered at all. This observation, and the fact that only one article considers a stochastic machine availability (cf., Figure 3), leads to the conclusion that the consideration of uncertainty is also an important future research direction (R4).

5.1.6 **Objective system (C)**

The diversity of *sb*- and *hb*-problems is underlined by the large number of objective criteria considered (32). The tree map diagram in Figure 7 illustrates the objective criteria used, grouped by the categories *Time-oriented*, *Resource-oriented*, *Finance-oriented*, and *Other*.

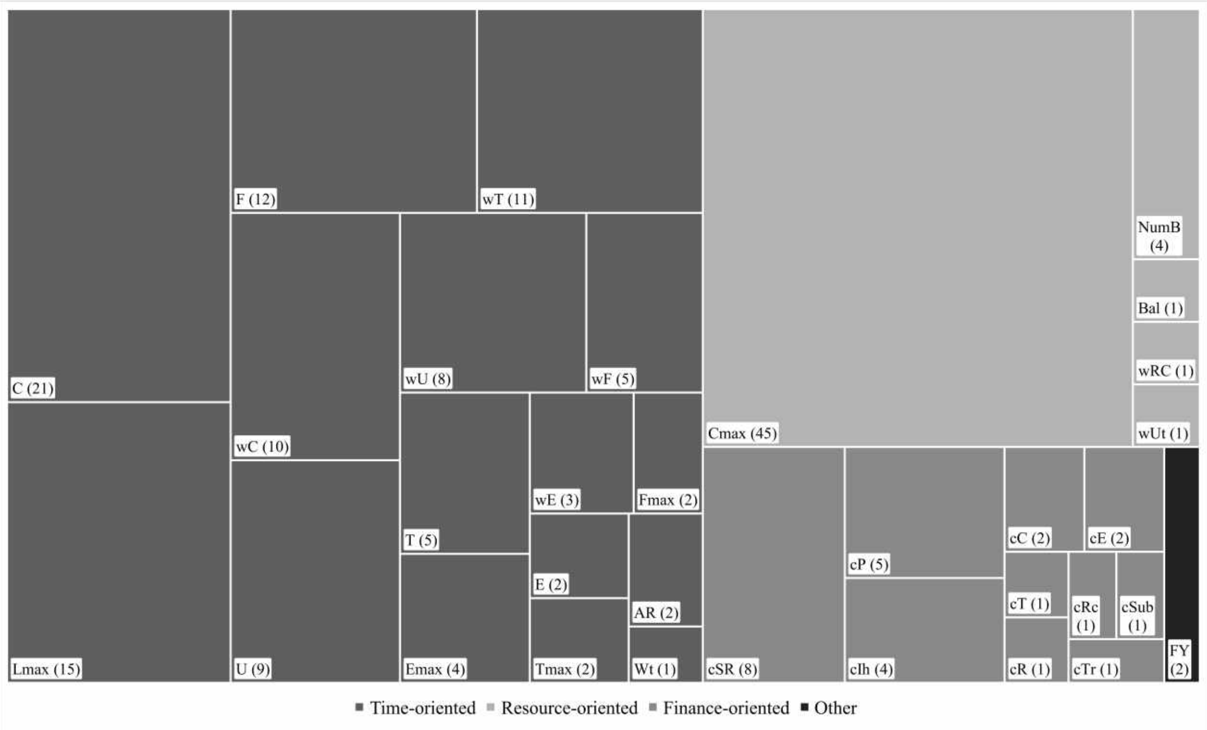

Figure 7: Used Objective criteria (C1)

Figure 7 shows that the makespan objective (*Cmax*) dominates all others (45). In the group of time-oriented criteria, the completion time related criteria (such as total completion time - *C*, total weighted completion time - *wC*, total flow time - *F*, and total weighted flow time - *wF*) and the due date related criteria (such as maximum lateness - *Lmax*, total weighted tardiness - *wT*, and number of tardy jobs - *U*) are considered with about the same frequency (50 and 58, respectively).

The importance of on-time delivery is reflected in the steady increase in the number of articles with objective criteria related to due dates (cf., Figure 8) and the interest in such criteria is likely to continue to grow.

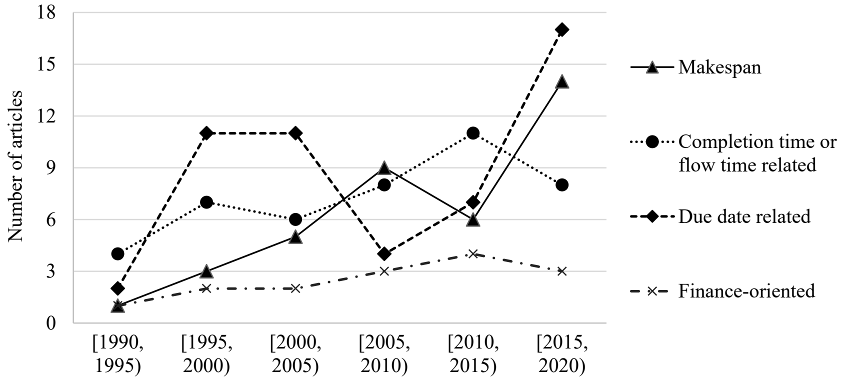

Figure 8: Temporal development of the used objective criteria

It should be noted that the maximum lateness criterion is the only criterion used that directly measures both earliness and tardiness, thus reflecting the just-in-time principle. Criteria such as quadratic lateness are not considered at all and only Hazır & Kedad-Sidhoum (2014) and Yin et al. (2021) consider earliness and tardiness in a weighted sum multi-criteria approach. Thus, the optimization of just-in-time related criteria might be a topic for further research (R8).

The analysis of the objective criteria also reveals that no article addresses energy-related objectives. This aspect clearly indicates a future research topic (R2), especially since many *sb*- and *hb*-scheduling problems are related to energy-intensive production processes (e.g., heat-treatment or industrial 3D printing).

Regarding multi-criteria decision making, we can report that 15 articles examine multiple criteria independently from each other. In contrast, 29 articles use a multi-criteria model, most commonly weighted sum (17), Pareto optimization (7), and single optimization criterion combined with satisficing constraints (9). Also, two articles consider a lexicographical ordering of objectives. Note that some articles consider more than one multi-criteria model.

Analyzing the objective criteria used by the multi-criteria models, we can report that 25 different criteria from all three categories are used. However, the application of multi-criteria models seems to be a valuable research topic for the future, especially when social, environmental, and economic criteria are combined to form a sustainable objective system (R3).

5.2 Analysis of article characteristics

In this section, we present various analyses with respect to the article classification scheme SACS to gain insights into theoretical findings, applied model types and solution methods, experimental evaluation, and considered application cases. We focus not only on the current-state and further research topics, but also derive some points to improve research articles.

5.2.1 Theoretical insights (D) and Model type (E)

In total, 107 articles explicitly provide theoretical insights by reporting the complexity of their problems (33 claim *NP* hardness and 19 *sNP* hardness), dominance properties (79), lower bounds (21), upper bounds (3), and worst-case performances (9).

Regarding model types, we observe that 32 optimization programs (6 *LP*, 24 *MILP*, and 2 *MINLP*) have been developed. 10 of them use sequence-based variables (e.g., to indicate that one batch is being processed before another; Shen et al., 2013), 22 use position-based assignment variables (e.g., to assign jobs to batches having a fixed position on a machine; Gahm et al., 2022b), and 17 use timepoint-based variables (e.g., to Guo et al., 2020). Only Voß & Witt (2007) use a time-indexed formulation with binary variables. Note that many of the proposed models use more than one type of decision variables and the type was characterized by the "main" decision variables (if possible). To date, there is no article that compares different optimization models for the same *sb*- or *hb*-problem or the performance of different solver engines. This seems to merit further research (R12).

5.2.2 **Solution methods (F)**

As far as the use of standard solvers is concerned, we would like to point out that not all developed optimization models are used, e.g., to compute reference solutions, and that in some articles the termination criteria (e.g., MIP-gap or maximum runtime) are not correctly or not completely specified. The wide range of maximum runtimes (from 30 minutes to more than 720 minutes) indicates that this parameter seems to be very problem specific.

In Figure 9, the applied solution methods are depicted according to the main categories Exact solution method (*ExSM*; comprising *DP* - Dynamic programming, *ExAlg* - Exact algorithm, and *EnB* - Enumeration-based methods such as Branch-and-bound), Heuristic (*Heu*; comprising *AS* - Approximation schemes, *CH* - Construction heuristics, *LS* - Local search, and *oHeu** - Other heuristics), Metaheuristic (*Meta*, comprising *MetaS* - Single-solution metaheuristic, *MetaP* - Population-based metaheuristic, and *MetaHyb* - Hybrid metaheuristic), and Other (comprising *MatH* - Matheuristic, *Sim* - Simulation, *RoSM* - Robust scheduling method, and *LaSM* - Learning-augmented method).

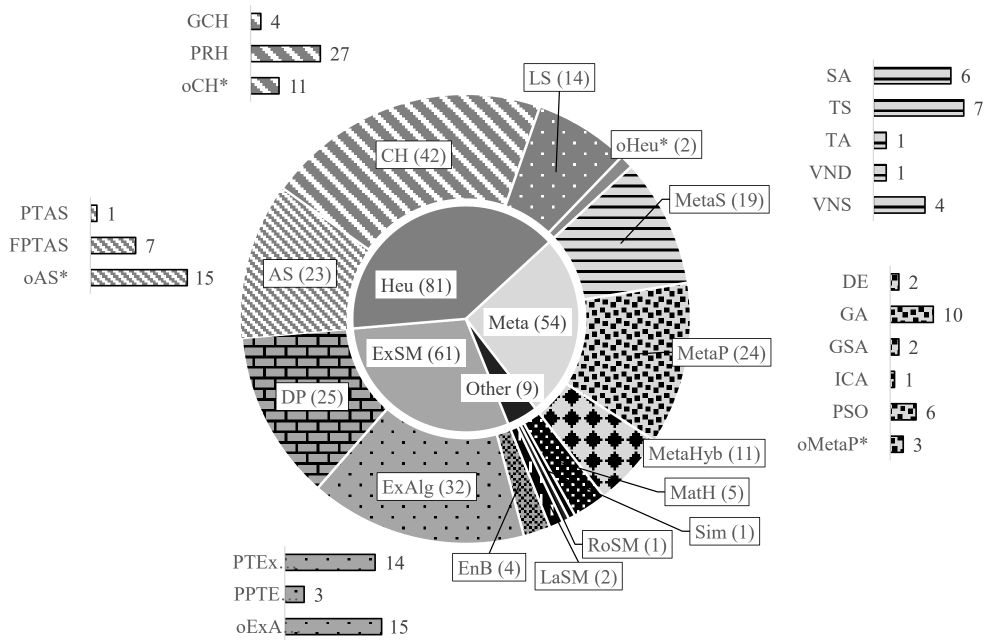

Figure 9: Applied Solution methods (F) (number of articles)

As can be seen in Figure 9, a wide variety of solution methods have been developed for *sb*- and *hb*-scheduling. Enumeration-based methods (*EnB*) such as Branch-and-bound or Branch-and-price are seldom used (R13). As there is only one problem with stochastic parameters investigated, there is also only one *RoSM* applied: Completely reactive scheduling (cf., Pei et al., 2016). This clearly indicates the need for further research on non-deterministic scheduling problems (R4). Furthermore, although some authors consider multi-criteria problems (cf., section 5.1.6), so far none use multi-criteria metaheuristics such as NSGA-II (Deb et al., 2002) or SPEA2 (Zitzler et al., 2001), which are commonly used to compute Pareto solutions for comparisons (R14). Multi-criteria models and metaheuristics should be given more attention, especially for *hb*-scheduling, where aspects such as resource consumption (base metal sheets in the case of laser cutting or polymers in the case of additive manufacturing) and energy consumption are important (R3).

The emerging topic of learning-augmented scheduling methods (i.e., a method/model from the field of machine learning is used to improve scheduling methods) is rarely addressed by the current literature. Only two articles have been published on this topic: Shahvari et al. (2022) use Random forest classification to estimate lower bounds on the number of jobs in a batch to reduce the solution space; Uzunoglu et al. (2023) use Artificial neural networks to estimate a set of most preferred parameters for a multi-start construction heuristic to reduce the computational effort. Both articles demonstrate the potential of integrating machine learning with scheduling, so this integration is worth exploring further in the future (R15).

The performance of many heuristics, metaheuristics and other solution methods depends strongly on appropriate parameters, and thus their determination (tuning) is important. However, many authors only report the parameters they use without giving any reasons. When using solution methods from the literature for comparisons, even given parameter values from the literature are used without any problem specific adjustment. As also stated by Neufeld et al. (2022), this "simple" application of benchmark methods to show the superiority of newly proposed solution methods is not appropriate, as it is not clear whether the superiority is only due to the tuned parameters. Furthermore, we observe that most (73%) of the parameters used are independent of the characteristics of a specific problem instance (e.g., the number of jobs), and that only six articles use online tuning, where the parameters are tuned while solving a problem instance. Overall, these observations show that more attention should be paid to the topic of parameter tuning in *sb*- and *hb*-scheduling research (R16).

In terms of advanced computing capabilities such as parallel or distributed computing, we found that only Uzunoglu et al. (2023) explore and exploit the capabilities of a parallel implementation (shared-memory multiprocessor computing). The authors report remarkable speedups and efficiency gains. As the benefits of parallel processing for optimization algorithms are also emphasized in the review by Schryen (2020), we conclude that this topic is also of further interest for *sb*- and *hb*-scheduling (R17).

5.2.3 **Experimental evaluation (G)**

59 of the 118 analyzed articles include an experimental evaluation. The scope of most of them (58) is dedicated to the investigation of solution methods. Only two articles (also) investigate different optimization programs: Shahvari & Logendran (2017) compare a complete model with the performance of a restricted model and Yin et al. (2016) analyze several models with different objective criteria. Lower/upper bounds or objective functions are not studied at all (note that here, investigating objective functions means analyzing the impact of different operational criteria on business goals). Regarding the performance assessment, we observe a wide variety of performance indicators and aggregation functions. Unfortunately, the descriptions of the indicators, the reference values used, the problem instances solved, and the aggregation functions are not clearly and

comprehensively described in all the articles examined. Another aspect related to writing articles is that the number of experiment repetitions is not always clearly stated when stochastic solution methods are used.

Another point for improvement would be the application of (the same) statistical methods to analyze the performance of solution methods. Currently only three articles use Wilcoxon signed-rank test (e.g., Zhang et al., 2020 or Toksarı & Toğa, 2022), two articles use Paired t-test (Suppiah & Omar, 2014 and Gahm et al., 2022b), one article uses multivariate test (Shin et al., 2020), and one article uses Confidence intervals (Zehetner & Gansterer, 2022). We conclude that it would be very welcome to see a greater use of statistical methods as well as the use of the same (standardized) methods.

The analysis of problem instance sources shows that 92% (54) of the articles use self-generated instances, whereas only three use real-world instances and four use instances from the literature. These values indicate a high non-uniformity of the problem instances. Although many of the investigated *sb*- and *hb*-problems are very different und thus the individual generation of problem instances is reasonable, the development of benchmark instances (or at least the basic data for specific problem classes) would be very helpful to improve the performance assessment of new solution methods (R18). Another way for improvement would be to provide problem instances for download, as these instances could then be used for appropriate comparisons. Currently, only three articles use a data repository to provide their instances for permanent download (Gahm et al., 2022b, Zehetner & Gansterer, 2022, and Uzunoglu et al., 2023).

The number of articles using a certain number of instances is shown in Figure 10. As can be seen, many authors examine only a small number of instances, making it difficult to derive statistically valid statements about the performance of solution methods.

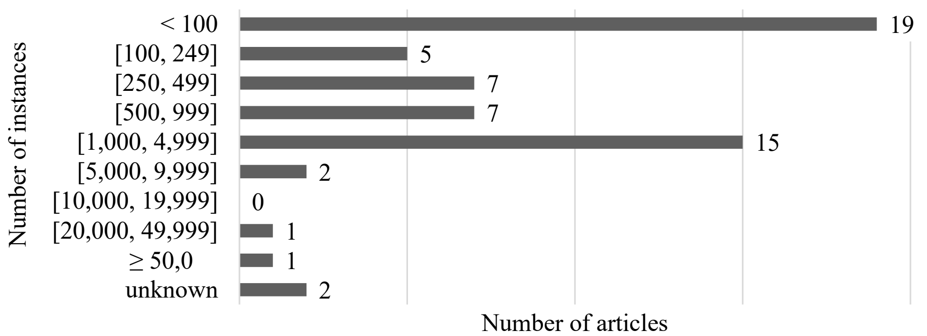

Figure 10: Number of solved instances

The diversity of problem instances investigated is illustrated by the frequency of use (in terms of number of jobs and machines/stations) shown in Figure 11. Note that for problems with more than one production stage (*F*, *HF*, *PF*, *J*, *HJ*, or *O*) the number of jobs is depicted and not the number of operations.

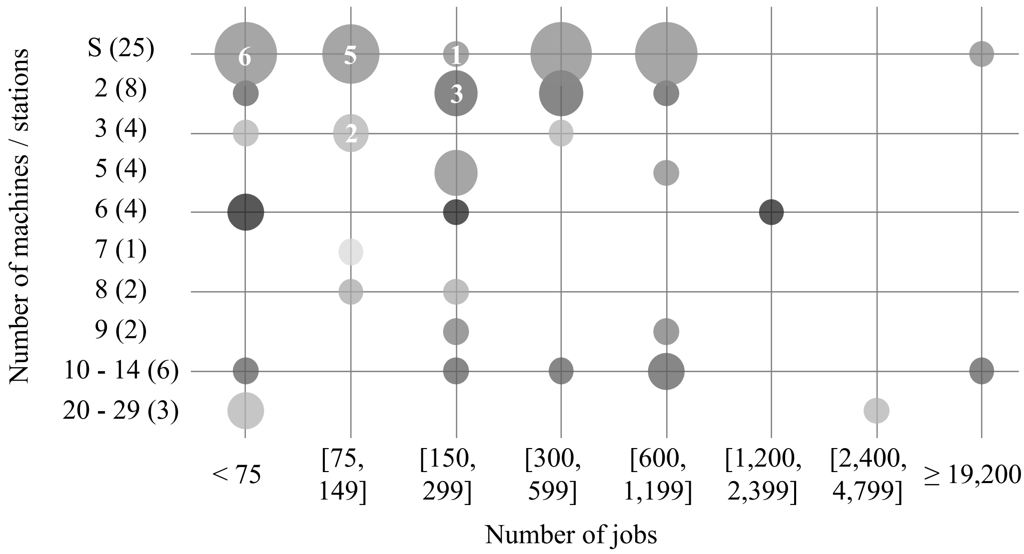

Figure 11: Number of articles in relation to the number of jobs and machines/stations (S representing single machine problems)

As can be seen in Figure 11, the combination of the two main characteristics, number of jobs and number of machines (stations), varies widely and the future development of benchmark problem instances should take this into account. For *sb*- and *hb*-scheduling, particularly instances with a very large number of jobs and/or machines (stations) can be very interesting to evaluate solution methods with respect to real-world applications. For further details on main instance characteristics that should be considered during instance generation, see Table 6 in Appendix B.

During the analysis of the used instances, we observe that many descriptions of instance generation procedures or the instance declarations are unclear or incomplete (cf., “unknown” classes in Figure 10 and Table 6). To improve the transparency and the reproducibility of experimental evaluations, authors should pay more attention to a complete documentation of their experimental studies.

5.2.4 Application case (H)

57 articles directly refer to one or more real-world application cases. Figure 12 shows the number of application cases related to ISIC-divisions (cf., 4.3.5) and decade. Almost all of theme refer to manufacturing, only Mor & Mosheiov (2014b) address a problem from Telecommunication (i.e., a problem to satisfy the service requirements of a content provider that uses a commercial satellite to transfer voice, image and text files).

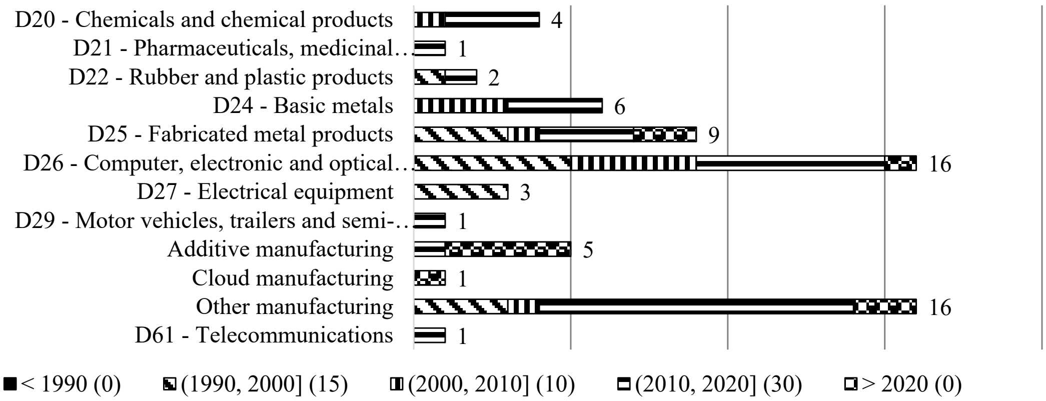

Figure 12: Number of application cases by ISIC-divisions

Very interesting for further research are the application areas of *Cloud manufacturing* and *Additive manufacturing*, for which an increasing interest can be assumed. Zehetner & Gansterer (2022) are the only ones who explicitly consider *Cloud manufacturing* and *sb*- or *hb*-scheduling, so this seems to be a promising topic (R10).

Of the articles dealing with *Additive manufacturing* (AM), Gahm et al. (2022b) and Uzunoglu et al. (2023) formulate a *sb*-problem with considering the subordinate nesting problem by a single one-dimensional batch capacity constraint (with *crJ*). Zhang et al. (2020) formulate an *hb*-problem with three constraints to respect the length, width, and height of the Stereo Lithography Appearance printer. The processing time depends, among other things, on the scanning time per part and the recoating time for multiple parts, which depends on the number of layers determined by the maximum height of all parts assigned to a batch. Toksarı & Toğa (2022) also formulate an *hb*-problem. The authors use a one-dimensional batch capacity constraint that guarantees that the total area of parts assigned to a batch cannot be greater than the production area of the AM machine. The batch processing time depends on the maximum height and the total volume of all parts assigned to a batch. Zehetner & Gansterer (2022) investigate an *hb*-problem for a powder bed laser machine. The authors use several constraints to guarantee that all the rectangular cubic parts are positioned in the geometric boundaries of the print chamber and to prevent part overlapping. What all these articles have in common is that they simplify, to a greater or lesser extent, the characteristics of the subordinate nesting problem. Whether the applied anticipations are appropriate to reflect the subordinate nesting problem or whether more advanced methods should be applied is a topic for further research (R1).

The above list of AM scheduling approaches is by no means complete, but they are the only ones that directly link AM and batch scheduling. This leads to the future research topic of synthesizing and consolidating batch scheduling and AM scheduling research (R1).

6 Further research directions

The preceding analysis of the current state of the literature on *sb*- and *hb*-scheduling has led to several starting points for future research. The following list of research topics summarizes the potential directions and the references listed here either come to similar conclusions in other scheduling contexts or provide further detail on the topic.

6.1 Topics related to (new) problem settings

R1 In many application cases and particularly in AM, *sb*- and *hb*- scheduling problems are strongly coupled with subordinate nesting problems (as also highlighted by Neufeld et al., 2022). Their integration by anticipation (in a hierarchical production planning system) generally depends on simplifying estimations and it needs to be investigated whether these are appropriate or whether more advanced methods should be used. A first step in this direction is made by Gahm et al. (2022a), who use Artificial neural networks to anticipate complex nesting solutions. A further topic to be investigated is the synthesis and consolidation of batch scheduling research and AM scheduling research to avoid the reinvention of scheduling methods (Fowler & Mönch, 2022) and to unify terms and notations.

R2 Energy-oriented (or even sustainability-oriented) aspects are not considered in *sb*- and *hb*-scheduling. Despite the fact that many of the production processes considered require substantial amounts of energy, none of the aspects discussed in the energy-oriented scheduling reviews by Biel & Glock (2016) and Gahm et al. (2016) have been considered so far. Fowler & Mönch (2022) and Neufeld et al. (2022) also conclude that energy considerations will become more important in the future.

R3 In general, more attention should be paid to multi-criteria problems. Not only energy-awareness, but also general limitations in the availability of resources make the efficient use of (material) resources (e.g., base metal sheets in the case of laser cutting or polymers in the case of AM) and the elimination of waste more important in the future. Therefore, appropriate objective criteria should be developed and integrated into multi-criteria models to find the most suitable tradeoffs.

R4 Also, more attention needs be paid to information uncertainty and dynamics. Only two articles address on-line scheduling problems and only Pei et al. (2016) consider stochastic machine breakdowns and use a reactive scheduling approach to deal with them. The low consideration of uncertainty is also highlighted by Fowler & Mönch (2022) and Neufeld et al. (2022).

R5 A rarely considered topic is assembly production environments, where parts are produced in a first production stage (e.g., by a laser cutting machine) and then assembled into components or final products in subsequent production stages (for more details on assembly production environments, cf., e.g., Komaki et al., 2019).

R6 The study of job shop (*J*) and hybrid job shop (*HJ*) machine environments, particularly in the context of assembly processes, is a promising research topic, as those are the prevailing production environments in special part manufacturing and only three articles have addressed such environments so far (cf., Mosheiov & Oron, 2011, Shen & Buscher, 2012, and Castillo & Gazmuri, 2015).

R7 Since new machine learning methods provide the information needed to schedule "predictive maintenance" operations, the scheduling of maintenance activities will become more important in the future.

R8 Just-in-time related criteria and objective systems are rarely considered. Their study should be intensified in the future, as production processes getting more dispersed, and their efficient timely coordination becomes more important.

R9 Not only, but also in the context of energy-oriented scheduling, the potential of variable-intensity operations (work contents) should be investigated (for more details on this topic, cf., e.g., Fündeling & Trautmann, 2010).

R10 Cloud manufacturing (CM) is an emerging manufacturing paradigm with the goal to deliver on-demand manufacturing services to customers (Liu et al., 2019). As scheduling is of critical means in this context and only Zehetner & Gansterer (2022) have addressed CM so far, it is a promising future research topic.

R11 The combined consideration of scheduling and transportation (distribution) planning is somehow related to CM. Whenever production stages are located in different geographical locations or when customer delivery depends on production location decisions, transport and production should be planned together. Inbound, outbound, and intermediate transportation should be investigated. Also Fowler & Mönch (2022) and Neufeld et al. (2022) highlight this aspect for further research.

6.2 Topics related to solution methods and their assessment

R12 Although different optimization models (with different main decision variables) can be found in the literature, there is no article that compares different models for the same *sb*- or *hb*-problem. Therefore, the analysis of different types of models and also the performance of different solver engines should be addressed in the future.

R13 In principle, greater efforts should be made to develop enumeration based-based exact solution methods like branch-and-bound or branch-and-cut (at least for calculating benchmark solutions).

R14 The application and development of multi-criteria metaheuristics should be addressed. Hereby, special attention should be paid to the metrics used to evaluate Pareto fronts (cf., Neufeld et al., 2022).

R15 A current research trend that has received too little attention in the past is the enhancement of scheduling methods by Machine learning. In this regard, approaches such as heuristic generation (e.g., Genetic programming), scheduling method (algorithm) selection, or operator and neighborhood selection should be explored. In addition, online learning and Reinforcement learning seem to be promising topics for the future.

R16 For many solution methods, parameter tuning is very important. However, many authors do not use appropriate methods to determine suitable parameters. In this context, it would be helpful to develop standard procedures or at least basic requirements for appropriate parameter tuning. Also, the application of automated parameter tuning by analytical methods should be investigated.

R17 The use of advanced computing architectures, features, or capabilities (e.g., parallel or distributed computing or GPUs/CUDA) should be explored and exploited in the future (see also Fowler & Mönch, 2022).

R18 The development and provision of benchmark problem instances for "standard" problems would be very welcome to enable a more valid and traceable evaluation of newly developed solution methods (see also Fowler & Mönch, 2022 and Neufeld et al., 2022).

7 Final remarks

The literature review and analysis presented here are based on two classification schemes (SPCS and SACS), as these are perfectly suited for conducting objective reviews, empirical analyses, and developing further research directions. The analysis of the literature has led to several topics for future research but is of course not exhaustive. For example, research on new solution methods is only addressed to a limited extent, since their development is usually very problem-specific and corresponding analyses have to be carried out in relation to the specific problem. However, the classified articles help to identify the most relevant literature in the most efficient way.

A common problem with literature reviews is that they are out of date by the time they are published. Therefore, we propose to develop and establish a collaborative online knowledge base for batch scheduling problems, where the current state of research is "automatically" updated as researchers contribute their own classified articles. The review of the submitted articles will lead to an appropriate scientific quality of the knowledge base.

References

Allahverdi, A. (2015). The third comprehensive survey on scheduling problems with setup times/costs. *Eur J Oper Res*, *246*(*2*), *345–378.* doi:10.1016/j.ejor.2015.04.004.

Allahverdi, A., Gupta, J. N., & Aldowaisan, T. (1999). A review of scheduling research involving setup considerations. *Omega*, *27*(*2*), *219–239.* doi:10.1016/S0305-0483(98)00042-5.

Allahverdi, A., Ng, C. T., Cheng, T. C. E., & Kovalyov, M. Y. (2008). A survey of scheduling problems with setup times or costs. *Eur J Oper Res*, *187*(*3*), *985–1032.* doi:10.1016/j.ejor.2006.06.060.

Bellanger, A., & Oulamara, A. (2009). Scheduling hybrid flowshop with parallel batching machines and compatibilities. *Comput Oper Res*, *36*(*6*), *1982–1992.* doi:10.1016/j.cor.2008.06.011.

Biel, K., & Glock, C. H. (2016). Systematic literature review of decision support models for energy-efficient production planning. *Comput Ind Eng*, *101*, *243–259.* doi:10.1016/j.cie.2016.08.021.

Blazewicz, J., Lenstra, J. K., & Kan, A. (1983). Scheduling subject to resource constraints: classification and complexity. *Discrete Appl Math*, *5*(*1*), *11–24.* doi:10.1016/0166-218X(83)90012-4.

Błażewicz, J., Ecker, K., Pesch, E., Sterna, M., Ecker, K. H., Schmidt, G., Sterna, M., & Weglarz, J. (2019). *Handbook on Scheduling: From Theory to Practice* (Second edition). *International Handbooks on Information Systems*. Cham: Springer International Publishing.

Brucker, P. (2007). *Scheduling Algorithms* (Fifth edition). Berlin Heidelberg: Springer-Verlag GmbH.

Brucker, P., Drexl, A., Möhring, R., Neumann, K., & Pesch, E. (1999). Resource-constrained project scheduling: Notation, classification, models, and methods. *Eur J Oper Res*, *112*(*1*), *3–41.* doi:10.1016/S0377-2217(98)00204-5.

Castillo, F., & Gazmuri, P. (2015). Genetic algorithms for batch sizing and production scheduling. *Int J Adv Manuf Technol*, *77*(*1-4*), *261–280.* doi:10.1007/s00170-014-6456-5.

Chakhlevitch, K., Glass, C. A., & Kellerer, H. (2011). Batch machine production with perishability time windows and limited batch size. *Eur J Oper Res*, *210*(*1*), *39–47.* doi:10.1016/j.ejor.2010.10.033.

Cheng, T. C. E., Chen, Z.-L., Kovalyov, M. Y., & Lin, B. M. T. (1996a). Parallel-Machine Batching and Scheduling to Minimize Total Completion Time. *IIE Trans*, *28*(*11*), *953–956.* doi:10.1080/15458830.1996.11770748.

Cheng, T. C. E., Chen, Z.-L., & Oguz, C. (1994). One-machine batching and sequencing of multiple-type items. *Comput Oper Res*, *21*(*7*), *717–721.* doi:10.1016/0305-0548(94)90001-9.

Cheng, T. C. E., & Gordon, V. S. (1994). Batch Delivery Scheduling on a Single Machine. *J Oper Res Soc*, *45*(*10*), *1211–1216.* doi:10.1057/jors.1994.191.

Cheng, T. C. E., Gordon, V. S., & Kovalyov, M. Y. (1996b). Single machine scheduling with batch deliveries. *Eur J Oper Res*, *94*(*2*), *277–283.* doi:10.1016/0377-2217(96)00127-0.

Cheng, T. C. E., Janiak, A., & Kovalyov, M. Y. (2001). Single machine batch scheduling with resource dependent setup and processing times. *Eur J Oper Res*, *135*(*1*), *177–183.* doi:10.1016/S0377-2217(00)00312-X.

Cheng, T. C. E., & Kovalyov, M. Y. (1995). Single machine batch scheduling with deadlines and resource dependent processing times. *Oper Res Lett*, *17*(*5*), *243–249.* doi:10.1016/0167-6377(95)00011-8.

Cheng, T. C. E., & Kovalyov, M. Y. (1996). Batch scheduling and common due-date assignment on a single machine. *Discrete Appl Math*, *70*(*3*), *231–245.* doi:10.1016/0166-218X(96)80468-9.

Cheng, T. C. E., & Kovalyov, M. Y. (1998). An exact algorithm for batching and scheduling two part types in a mixed shop: A technical note. *Int J Prod Econ*, *55*(*1*), *53–56.* doi:10.1016/s0925-5273(98)00039-5.

Cheng, T. C. E., & Kovalyov, M. Y. (2001). Single machine batch scheduling with sequential job processing. *IIE Trans*, *33*(*5*), *413–420.* doi:10.1023/A:1011005314354.

Cheng, T. C. E., Kovalyov, M. Y., & Chakhlevich, K. N. (2004). Batching in a two-stage flowshop with dedicated machines in the second stage. *IIE Trans*, *36*(*1*), *87–93.* doi:10.1080/07408170490247368.

Cheng, T. C. E., Lin, B. M. T., & Toker, A. (2000). Makespan minimization in the two-machine flowshop batch scheduling problem. *Nav Res Log*, *47*(*2*), *128–144.* doi:10.1002/(SICI)1520-6750(200003)47:2<128::AID-NAV4>3.0.CO;2-%23.

Cheng, T. C. E., & Wang, G. (1998). Batching and scheduling to minimize the makespan in the two-machine flowshop. *IIE Trans*, *30*(*5*), *447–453.* doi:10.1080/07408179808966485.

Chrétienne, P., Hazır, Ö., & Kedad-Sidhoum, S. (2011). Integrated batch sizing and scheduling on a single machine. *J Sched*, *14*(*6*), *541–555.* doi:10.1007/s10951-011-0229-x.

Crauwels, H. A. J., Hariri, A., Potts, C. N., & van Wassenhove, L. N. (1998). Branch and bound algorithms for single-machine scheduling with batch set-up times to minimize total weighted completion time. *Ann Oper Res*, *83*, *59–76.* doi:10.1023/A:1018920416308.

Crauwels, H. A. J., Potts, C. N., van Oudheusden, D., & van Wassenhove, L. N. (2005). Branch and Bound Algorithms for Single Machine Scheduling with Batching to Minimize the Number of Late Jobs. *J Sched*, *8*(*2*), *161–177.* doi:10.1007/s10951-005-6365-4.

Crauwels, H. A. J., Potts, C. N., & van Wassenhove, L. N. (1996). Local search heuristics for single-machine scheduling with batching to minimize the number of late jobs. *Eur J Oper Res*, *90*(*2*), *200–213.* doi:10.1016/0377-2217(95)00349-5.

Crauwels, H. A. J., Potts, C. N., & van Wassenhove, L. N. (1997). Local search heuristics for single machine scheduling with batch set-up times to minimize total weighted completion time. *Ann Oper Res*, *70*, *261–279.* doi:10.1023/A:1018978322417.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Computat*, *6*(*2*), *182–197.* doi:10.1109/4235.996017.

Dobson, G., Karmarkar, U. S., & Rummel, J. L. (1987). Batching to Minimize Flow Times on One Machine. *Manage Sci*, *33*(*6*), *784–799.* doi:10.1287/mnsc.33.6.784.

Dobson, G., Karmarkar, U. S., & Rummel, J. L. (1989). Batching to Minimize Flow Times on Parallel Heterogeneous Machines. *Manage Sci*, *35*(*5*), *607–613.* doi:10.1287/mnsc.35.5.607.

Dobson, G., & Khosla, I. (1995). Simultaneous resource scheduling with batching to minimize weighted flow times. *IIE Trans*, *27*(*5*), *587–598.* doi:10.1080/07408179508936774.

Erel, E., & Ghosh, J. B. (2007). Batch scheduling to minimize the weighted number of tardy jobs. *Comput Ind Eng*, *53*(*3*), *394–400.* doi:10.1016/j.cie.2007.03.006.

Fan, W., Pei, J., Liu, X., Pardalos, P. M., & Kong, M. (2018). Serial-batching group scheduling with release times and the combined effects of deterioration and truncated job-dependent learning. *J Glob Optim*, *71*(*1*), *147–163.* doi:10.1007/s10898-017-0536-7.

Feng, Q., Yuan, J., Liu, H., & He, C. (2013). A note on two-agent scheduling on an unbounded parallel-batching machine with makespan and maximum lateness objectives. *App Math Model*, *37*(*10-11*), *7071–7076.* doi:10.1016/j.apm.2013.03.013.

Fowler, J. W., & Mönch, L. (2022). A survey of scheduling with parallel batch (p-batch) processing. *Eur J Oper Res*, *298*(*1*), *1–24.* doi:10.1016/j.ejor.2021.06.012.

Fündeling, C.-U., & Trautmann, N. (2010). A priority-rule method for project scheduling with work-content constraints. *Eur J Oper Res*, *203*(*3*), *568–574.* doi:10.1016/j.ejor.2009.09.019.

Gahm, C., Denz, F., Dirr, M., & Tuma, A. (2016). Energy-efficient scheduling in manufacturing companies: A review and research framework. *Eur J Oper Res*, *248*(*3*), *744–757.* doi:10.1016/j.ejor.2015.07.017.

Gahm, C., Uzunoglu, A., Wahl, S., Ganschinietz, C., & Tuma, A. (2022a). Applying machine learning for the anticipation of complex nesting solutions in hierarchical production planning. *Eur J Oper Res*, *296*(*3*), *819–836.* doi:10.1016/j.ejor.2021.04.006.

Gahm, C., Wahl, S., & Tuma, A. (2022b). Scheduling parallel serial-batch processing machines with incompatible job families, sequence-dependent setup times and arbitrary sizes. *Int J Prod Res*, *60*(*17*), *5131–5154.* doi:10.1080/00207543.2021.1951446.

Ganschinietz, C. (2021). Design of on-site energy conversion systems for manufacturing companies – A concept-centric research framework. *J Clean Prod*, *310*, *127258.* doi:10.1016/j.jclepro.2021.127258.

Geng, Z., Yuan, J., & Yuan, J. (2018). Scheduling with or without precedence relations on a serial-batch machine to minimize makespan and maximum cost. *App Math Comp*, *332*, *1–18.* doi:10.1016/j.amc.2018.03.001.

Gerstl, E., & Mosheiov, G. (2013). A two-stage flow shop batch-scheduling problem with the option of using Not-All-Machines. *Int J Prod Econ*, *146*(*1*), *161–166.* doi:10.1016/j.ijpe.2013.06.018.

Giglio, D. (2015). Optimal control strategies for single-machine family scheduling with sequence-dependent batch setup and controllable processing times. *J Sched*, *18*(*5*), *525–543.* doi:10.1007/s10951-015-0440-2.

Glass, C. A., Potts, C. N., & Strusevich, V. A. (2001). Scheduling Batches with Sequential Job Processing for Two-Machine Flow and Open Shops. *INFORMS J Comput*, *13*(*2*), *120–137.* doi:10.1287/ijoc.13.2.120.10521.

Graham, R. L., Lawler, E. L., Lenstra, J. K., & Rinnooy Kan, A. H. G. (1979). Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey. In B. H. Korte, E. L. Johnson, & P. L. Hammer (Eds.), *Annals of Discrete Mathematics: Vol. 5. Discrete optimization II* (, 287–326), Amsterdam, New York, New York: North-Holland Pub. Co.

Gribkovskaia, I. V., Lee, C.-Y., Strusevich, V. A., & Werra, D. de. (2006). Three is easy, two is hard: open shop sum-batch scheduling problem refined. *Oper Res Lett*, *34(4)*, *459–464.* doi:10.1016/j.orl.2005.07.006.

Guo, Q., Tang, L., Liu, J., & Zhao, S. (2020). Continuous-time formulation and differential evolution algorithm for an integrated batching and scheduling problem in aluminium industry. *Int J Prod Res*, *1–16.* doi:10.1080/00207543.2020.1747656.

Hakim Halim, A., Miyazaki, S., & Ohta, H. (1994). Batch-scheduling problems to minimize actual flow times of parts through the shop under JIT environment. *Eur J Oper Res*, *72(3)*, *529–544.* doi:10.1016/0377-2217(94)90421-9.

Hakim Halim, A., & Ohta, H. (1993). Batch-scheduling problems through the flowshop with both receiving and delivery just in time. *Int J Prod Res*, *31(8)*, *1943–1955.* doi:10.1080/00207549308956833.

Hazır, Ö., & Kedad-Sidhoum, S. (2014). Batch sizing and just-in-time scheduling with common due date. *Ann Oper Res*, *213(1)*, *187–202.* doi:10.1007/s10479-012-1289-9.

He, C., Lin, H., & Han, X. (2022). Two-agent scheduling on a bounded series-batch machine to minimize makespan and maximum cost. *Discrete Appl Math*, *322*, *94–101.* doi:10.1016/j.dam.2022.08.001.

He, C., Xu, C., & Lin, H. (2020). Serial-batching scheduling with two agents to minimize makespan and maximum cost. *J Sched*, *609–617.* doi:10.1007/s10951-020-00656-5.

Herroelen, W., Demeulemeester, E., & Reyck, B. de. (1999). A Classification Scheme for Project Scheduling. In F. S. Hillier & J. Węglarz (Eds.), *International Series in Operations Research & Management Science. Project Scheduling* (, 1–26), Boston, MA: Springer US.

Hidayat, N. P., Cakravastia, A., Samadhi, T. A., & Hakim Halim, A. (2016). A batch scheduling model for m heterogeneous batch processor. *Int J Prod Res*, *54(4)*, *1170–1185.* doi:10.1080/00207543.2015.1056322.

Hochbaum, D. S., & Landy, D. (1994). Scheduling with batching: minimizing the weighted number of tardy jobs. *Oper Res Lett*, *16(2)*, *79–86.* doi:10.1016/0167-6377(94)90063-9.

Hochbaum, D. S., & Landy, D. (1997). Scheduling with batching: two job types. *Discrete Appl Math*, *72(1-2)*, *99–114.* doi:10.1016/S0166-218X(96)00039-X.

Hwang, F. J., & Lin, B. M. T. (2012). Two-stage assembly-type flowshop batch scheduling problem subject to a fixed job sequence. *J Oper Res Soc*, *63(6)*, *839–845.* doi:10.1057/jors.2011.90.

Ji, M., & Cheng, T. C. E. (2010). Batch scheduling of simple linear deteriorating jobs on a single machine to minimize makespan. *Eur J Oper Res*, *202(1)*, *90–98.* doi:10.1016/j.ejor.2009.05.021.

Jordan, C. (1996). *Batching and Scheduling: Models and Methods for Several Problem Classes* (Vol. 437). Berlin, Heidelberg: Springer Berlin Heidelberg.

Jordan, C. (1998). A two-phase genetic algorithm to solve variants of the batch sequencing problem. *Int J Prod Res*, *36(3)*, *745–760.* doi:10.1080/002075498193679.

Jordan, C., & Drexl, A. (1998). Discrete Lotsizing and Scheduling by Batch Sequencing. *Manage Sci*, *44(5)*, *698–713.* doi:10.1287/mnsc.44.5.698.

Komaki, G. M., Sheikh, S., & Malakooti, B. (2019). Flow shop scheduling problems with assembly operations: a review and new trends. *Int J Prod Res*, *57(10)*, *2926–2955.* doi:10.1080/00207543.2018.1550269.

Kovalyov, M. Y. (1997). Batch scheduling and common due date assignment problem: An NP-hard case. *Discrete Appl Math*, *80(2-3)*, *251–254.* doi:10.1016/S0166-218X(97)00092-9.

Kovalyov, M. Y., Oulamara, A., & Soukhal, A. (2015). Two-agent scheduling with agent specific batches on an unbounded serial batching machine. *J Sched*, *18(4)*, *423–434.* doi:10.1007/s10951-014-0410-0.

Kovalyov, M. Y., Potts, C. N., & Strusevich, V. A. (2004). Batching decisions for assembly production systems. *Eur J Oper Res*, *157(3)*, *620–642.* doi:10.1016/S0377-2217(03)00250-9.

Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H., & Shmoys, D. B. (1993). Chapter 9 Sequencing and scheduling: Algorithms and complexity. In S. C. Graves, A. H. G. Rinnooy Kan, & P. H. Zipkin (Eds.), *Handbooks in Operations Research and Management Science. Logistics of Production and Inventory* (, 445–522), North-Holland Pub. Co.

Leung, J. Y., Ng, C. T., & Cheng, T. C. E. (2008). Minimizing sum of completion times for batch scheduling of jobs with deteriorating processing times. *Eur J Oper Res*, *187*(*3*), *1090–1099.* doi:10.1016/j.ejor.2006.03.067.

Li, S. (2017). Parallel batch scheduling with inclusive processing set restrictions and non-identical capacities to minimize makespan. *Eur J Oper Res*, *260*(*1*), *12–20.* doi:10.1016/j.ejor.2016.11.044.

Li, S., Cheng, T. C. E., Ng, C. T., & Yuan, J. (2017). Two-agent scheduling on a single sequential and compatible batching machine. *Nav Res Log*, *64*(*8*), *628–641.* doi:10.1002/nav.21779.

Li, X., Ishii, H., & Masuda, T. (2012). Single machine batch scheduling problem with fuzzy batch size. *Comput Ind Eng*, *62*(*3*), *688–692.* doi:10.1016/j.cie.2011.12.021.

Liao, B., Song, Q., Pei, J., Yang, S., & Pardalos, P. M. (2020). Parallel-machine group scheduling with inclusive processing set restrictions, outsourcing option and serial-batching under the effect of step-deterioration. *J Glob Optim*, *78*(*4*), *717–742.* doi:10.1007/s10898-018-0707-1.

Liao, C.-J., Lee, C.-H., & Lee, H.-C. (2015). An efficient heuristic for a two-stage assembly scheduling problem with batch setup times to minimize makespan. *Comput Ind Eng*, *88*, *317–325.* doi:10.1016/j.cie.2015.07.018.

Lin, B., & Jeng, A. (2004). Parallel-machine batch scheduling to minimize the maximum lateness and the number of tardy jobs. *Int J Prod Econ*, *91*(*2*), *121–134.* doi:10.1016/j.ijpe.2003.07.003.

Lin, B. M., & Cheng, T. C. E. (2001). Batch scheduling in the no-wait two-machine flowshop to minimize the makespan. *Comput Oper Res*, *28*(*7*), *613–624.* doi:10.1016/S0305-0548(99)00138-0.

Lin, B. M. T., & Cheng, T. C. E. (2005). Two-Machine Flowshop Batching and Scheduling. *Ann Oper Res*, *133*(*1-4*), *149–161.* doi:10.1007/s10479-004-5029-7.

Lin, B. M. T., & Cheng, T. C. E. (2011). Scheduling with centralized and decentralized batching policies in concurrent open shops. *Nav Res Log*, *58*(*1*), *17–27.* doi:10.1002/nav.20437.

Lin, B. M. T., Cheng, T. C. E., & Chou, A. (2007). Scheduling in an assembly-type production chain with batch transfer. *Omega*, *35*(*2*), *143–151.* doi:10.1016/j.omega.2005.04.004.

Liu, Y., Wang, L., Wang, X. V., Xu, X., & Zhang, L. (2019). Scheduling in cloud manufacturing: state-of-the-art and research challenges. *Int J Prod Res*, *57*(*15-16*), *4854–4879.* doi:10.1080/00207543.2018.1449978.

Lu, L. F., & Yuan, J. J. (2007). The single machine batching problem with identical family setup times to minimize maximum lateness is strongly NP-hard. *Eur J Oper Res*, *177*(*2*), *1302–1309.* doi:10.1016/j.ejor.2005.12.027.

Luu, D. T., Bohez, E. L. J., & Techanitisawad, A. (2002). A hybrid genetic algorithm for the batch sequencing problem on identical parallel machines. *Prod Plan Control*, *13*(*3*), *243–252.* doi:10.1080/09537280110065517.

Mathirajan, M., & Sivakumar, A. I. (2006). A literature review, classification and simple meta-analysis on scheduling of batch processors in semiconductor. *Int J Adv Manuf Technol*, *29*(*9-10*), *990–1001.* doi:10.1007/s00170-005-2585-1.

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. *J Sched*, *14*(*6*), *583–599.* doi:10.1007/s10951-010-0222-9.

Mor, B., & Mosheiov, G. (2011). Single machine batch scheduling with two competing agents to minimize total flowtime. *Eur J Oper Res*, *215*(*3*), *524–531.* doi:10.1016/j.ejor.2011.06.037.

Mor, B., & Mosheiov, G. (2012a). Batch scheduling on uniform machines to minimize total flow-time. *Comput Oper Res*, *39*(*3*), *571–575.* doi:10.1016/j.cor.2011.05.014.

Mor, B., & Mosheiov, G. (2012b). Batch scheduling with step-deteriorating processing times to minimize flowtime. *Nav Res Log*, *59*(*8*), *587–600.* doi:10.1002/nav.21508.

Mor, B., & Mosheiov, G. (2014a). Batch scheduling of identical jobs with controllable processing times. *Comput Oper Res*, *41*, *115–124.* doi:10.1016/j.cor.2013.08.007.

Mor, B., & Mosheiov, G. (2014b). Batch scheduling with a rate-modifying maintenance activity to minimize total flowtime. *Int J Prod Econ*, *153*, *238–242.* doi:10.1016/j.ijpe.2014.03.004.

Mosheiov, G., & Oron, D. (2005). A note on flow-shop and job-shop batch scheduling with identical processing-time jobs. *Eur J Oper Res*, *161*(*1)*, *285–291.* doi:10.1016/j.ejor.2003.09.010.

Mosheiov, G., & Oron, D. (2008a). A single machine batch scheduling problem with bounded batch size. *Eur J Oper Res*, *187*(*3)*, *1069–1079.* doi:10.1016/j.ejor.2006.01.052.

Mosheiov, G., & Oron, D. (2008b). Open-shop batch scheduling with identical jobs. *Eur J Oper Res*, *187*(*3)*, *1282–1292.* doi:10.1016/j.ejor.2006.03.068.

Mosheiov, G., & Oron, D. (2011). Batch scheduling on a two-machine jobshop with machine-dependent setup times. *Nav Res Log*, *58*(*7)*, *676-684.* doi:10.1002/nav.20473.

Mosheiov, G., Oron, D., & Ritov, Y. (2004). Flow-shop batch scheduling with identical processing-time jobs. *Nav Res Log*, *51*(*6)*, *783–799.* doi:10.1002/nav.20028.

Mosheiov, G., Oron, D., & Ritov, Y. (2005). Minimizing flow-time on a single machine with integer batch sizes. *Oper Res Lett*, *33*(*5)*, *497–501.* doi:10.1016/j.orl.2004.09.007.

Muter, İ. (2020). Exact algorithms to minimize makespan on single and parallel batch processing machines. *Eur J Oper Res*, *285*(*2)*, *470–483.* doi:10.1016/j.ejor.2020.01.065.

Muthuswamy, S., Vélez-Gallego, M. C., Maya, J., & Rojas-Santiago, M. (2012). Minimizing makespan in a two-machine no-wait flow shop with batch processing machines. *Int J Adv Manuf Technol*, *63*(*1-4)*, *281–290.* doi:10.1007/s00170-012-3906-9.

Naddef, D., & Santos, C. (1988). One-pass batching algorithms for the one-machine problem. *Discrete Appl Math*, *21*(*2)*, *133–145.* doi:10.1016/0166-218X(88)90049-2.

Neufeld, J. S., Schulz, S., & Buscher, U. (2022). A systematic review of multi-objective hybrid flow shop scheduling. *Eur J Oper Res.* doi:10.1016/j.ejor.2022.08.009.

Ng, C. T., Cheng, T. C. E., Yuan, J. J., & Liu, Z. H. (2003a). On the single machine serial batching scheduling problem to minimize total completion time with precedence constraints, release dates and identical processing times. *Oper Res Lett*, *31*(*4)*, *323–326.* doi:10.1016/S0167-6377(03)00007-5.

Ng, C. T., & Kovalyov, M. Y. (2007). Batching and scheduling in a multi-machine flow shop. *J Sched*, *10*(*6)*, *353–364.* doi:10.1007/s10951-007-0041-9.

Ng, C. T. D., Cheng, T. C. E., & Kovalyov, M. Y. (2003b). Batch scheduling with controllable setup and processing times to minimize total completion time. *J Oper Res Soc*, *54*(*5)*, *499–506.* doi:10.1057/palgrave.jors.2601537.

Ng, C. T. D., Cheng, T. C. E., & Kovalyov, M. Y. (2004). Single machine batch scheduling with jointly compressible setup and processing times. *Eur J Oper Res*, *153*(*1)*, *211–219.* doi:10.1016/S0377-2217(02)00732-4.

Ng, C. T. D., Cheng, T. C. E., & Yuan, J. J. (2002). A note on the single machine serial batching scheduling problem to minimize maximum lateness with precedence constraints. *Oper Res Lett*, *30*(*1)*, *66–68.* doi:10.1016/S0167-6377(01)00105-5.

Oulamara, A. (2007). Makespan minimization in a no-wait flow shop problem with two batching machines. *Comput Oper Res*, *34*(*4)*, *1033–1050.* doi:10.1016/j.cor.2005.05.028.

Pei, J., Cheng, B., Liu, X., Pardalos, P. M., & Kong, M. (2019a). Single-machine and parallel-machine serial-batching scheduling problems with position-based learning effect and linear setup time. *Ann Oper Res*, *272*(*1-2)*, *217–241.* doi:10.1007/s10479-017-2481-8.

Pei, J., Liu, X., Fan, W., Pardalos, P. M., & Lu, S. (2019b). A hybrid BA-VNS algorithm for coordinated serial-batching scheduling with deteriorating jobs, financial budget, and resource constraint in multiple manufacturers. *Omega*, *82*, *55–69.* doi:10.1016/j.omega.2017.12.003.

Pei, J., Liu, X., Fan, W., Pardalos, P. M., Migdalas, A., Goldengorin, B., & Yang, S. (2016). Minimizing the makespan for a serial-batching scheduling problem with arbitrary machine breakdown and dynamic job arrival. *Int J Adv Manuf Technol*, *86*(*9-12)*, *3315–3331.* doi:10.1007/s00170-016-8408-8.

Pei, J., Liu, X., Liao, B., Pardalos, P. M., & Kong, M. (2018). Single-machine scheduling with learning effect and resource-dependent processing times in the serial-batching production. *App Math Model*, *58*, *245–253.* doi:10.1016/j.apm.2017.07.028.

Pei, J., Liu, X., Pardalos, P. M., Fan, W., & Yang, S. (2017a). Scheduling deteriorating jobs on a single serial-batching machine with multiple job types and sequence-dependent setup times. *Ann Oper Res*, *249*(*1-2)*, *175–195.* doi:10.1007/s10479-015-1824-6.

Pei, J., Liu, X., Pardalos, P. M., Migdalas, A., & Yang, S. (2017b). Serial-batching scheduling with time-dependent setup time and effects of deterioration and learning on a single-machine. *J Glob Optim*, *67*(*1-2)*, *251–262.* doi:10.1007/s10898-015-0320-5.

Pei, J., Pardalos, P. M., Liu, X., Fan, W., & Yang, S. (2015). Serial batching scheduling of deteriorating jobs in a two-stage supply chain to minimize the makespan. *Eur J Oper Res*, *244*(*1*), *13–25.* doi:10.1016/j.ejor.2014.11.034.

Pei, J., Song, Q., Liao, B., Liu, X., & Pardalos, P. M. (2021). Parallel-machine serial-batching scheduling with release times under the effects of position-dependent learning and time-dependent deterioration. *Ann Oper Res*, *298*(*1-2)*, *407–444.* doi:10.1007/s10479-020-03555-2.

Pinedo, M. L. (2016). *Scheduling: Theory, Algorithms, and Systems* (Fifth Edition). Cham: Springer International Publishing.

Potts, C. N., & Kovalyov, M. Y. (2000). Scheduling with batching: A review. *Eur J Oper Res*, *120*(*2*), *228–249.* doi:10.1016/S0377-2217(99)00153-8.

Potts, C. N., & van Wassenhove, L. N. (1992). Integrating Scheduling with Batching and Lot-Sizing: A Review of Algorithms and Complexity. *J Oper Res Soc*, *43*(*5)*, *395–406.* doi:10.1057/jors.1992.66.

Qi, X., & Yuan, J. (2017). A further study on two-agent scheduling on an unbounded serial-batch machine with batch delivery cost. *Comput Ind Eng*, *111*, *458–462.* doi:10.1016/j.cie.2017.07.029.

Quadt, D., & Kuhn, H. (2007). Batch scheduling of jobs with identical process times on flexible flow lines. *Int J Prod Econ*, *105*(*2)*, *385–401.* doi:10.1016/j.ijpe.2004.04.013.

Schryen, G. (2020). Parallel computational optimization in operations research: A new integrative framework, literature review and research directions. *Eur J Oper Res*, *287*(*1)*, *1–18.* doi:10.1016/j.ejor.2019.11.033.

Shabtay, D. (2014). The single machine serial batch scheduling problem with rejection to minimize total completion time and total rejection cost. *Eur J Oper Res*, *233*(*1*), *64–74.* doi:10.1016/j.ejor.2013.08.013.

Shabtay, D., & Steiner, G. (2007). Single machine batch scheduling to minimize total completion time and resource consumption costs. *J Sched*, *10*(*4-5)*, *255–261.* doi:10.1007/s10951-007-0025-9.

Shahvari, O., & Logendran, R. (2016). Hybrid flow shop batching and scheduling with a bi-criteria objective. *Int J Prod Econ*, *179*, *239–258.* doi:10.1016/j.ijpe.2016.06.005.

Shahvari, O., & Logendran, R. (2017). An Enhanced tabu search algorithm to minimize a bi-criteria objective in batching and scheduling problems on unrelated-parallel machines with desired lower bounds on batch sizes. *Comput Oper Res*, *77*, *154–176.* doi:10.1016/j.cor.2016.07.021.

Shahvari, O., Logendran, R., & Tavana, M. (2022). An efficient model-based branch-and-price algorithm for unrelated-parallel machine batching and scheduling problems. *J Sched.* doi:10.1007/s10951-022-00729-7.

Shallcross, D. F. (1992). A polynomial algorithm for a one machine batching problem. *Oper Res Lett*, *11*(*4)*, *213–218.* doi:10.1016/0167-6377(92)90027-Z.

Shen, L., & Buscher, U. (2012). Solving the serial batching problem in job shop manufacturing systems. *Eur J Oper Res*, *221*(*1*), *14–26.* doi:10.1016/j.ejor.2012.03.001.

Shen, L., & Gupta, J. N. D. (2018). Family scheduling with batch availability in flow shops to minimize makespan. *J Sched*, *21*(*2*), *235–249.* doi:10.1007/s10951-017-0529-x.

Shen, L., Mönch, L., & Buscher, U. (2013). An iterative approach for the serial batching problem with parallel machines and job families. *Ann Oper Res*, *206*(*1*), *425–448.* doi:10.1007/s10479-013-1339-y.

Shin, J.-H., Yu, J.-M., Doh, H.-H., Kim, H.-W., & Lee, D.-H. (2020). Batching and scheduling for a single-machine flexible machining cell with multi-fixturing pallets and controllable processing times. *Int J Prod Res*, *58*(*3*), *863–877.* doi:10.1080/00207543.2019.1602742.

Sung, C. S., & Joo, U. G. (1997). Batching to minimize weighted mean flow time on a single machine with batch size restrictions. *Comput Ind Eng*, *32*(*2*), *333–340.* doi:10.1016/S0360-8352(96)00300-2.

Sung, C. S., & Kim, Y. H. (2003). Minimizing due date related performance measures on two batch processing machines. *Eur J Oper Res*, *147*(*3*), *644–656.* doi:10.1016/S0377-2217(02)00352-1.

Suppiah, Y., & Omar, M. K. (2014). A hybrid tabu search for batching and sequencing decisions in a single machine environment. *Comput Ind Eng*, *78*, *135–147.* doi:10.1016/j.cie.2014.10.010.

Tamer Unal, A., & Kiran, A. S. (1992). Batch Sequencing. *IIE Trans*, *24*(*4*), *73–83.* doi:10.1080/07408179208964235.

Tang, C. S. (1990). Scheduling batches on parallel machines with major and minor set-ups. *Eur J Oper Res*, *46*(*1*), *28–37.* doi:10.1016/0377-2217(90)90295-M.

Tanrisever, F., & Kutanoglu, E. (2008). Forming and scheduling jobs with capacitated containers in semiconductor manufacturing: Single machine problem. *Ann Oper Res*, *159*(*1*), *5–24.* doi:10.1007/s10479-007-0273-2.

Toksarı, M. D., & Toğa, G. (2022). Single batch processing machine scheduling with sequence-dependent setup times and multi-material parts in additive manufacturing. *CIRP J Manuf Scien Tech*, *37*, *302–311.* doi:10.1016/j.cirpj.2022.02.007.

United Nations, Department of Economic and Social Affairs. (2008). *International Standard Industrial Classification of All Economic Activities (ISIC)* (4th ed.). *Statistical papers. Series M*. New York: United Nations Publications.

Uzunoglu, A., Gahm, C., Wahl, S., & Tuma, A. (2023). Learning-augmented heuristics for scheduling parallel serial-batch processing machines. *Comput Oper Res*, *151*, *106122.* doi:10.1016/j.cor.2022.106122.

vom Brocke, J., Simons, A., Niehaves, B., Riemer, K., Plattfaut, R., & Cleven, A. (2009). RECONSTRUCTING THE GIANT: ON THE IMPORTANCE OF RIGOUR IN DOCUMENTING THE LITERATURE SEARCH PROCESS. In L. Mathiassen (Ed.), *Proceedings of the 17th European Conference on Information Systems (ECIS 2009)* .

Voß, S., & Witt, A. (2007). Hybrid flow shop scheduling as a multi-mode multi-project scheduling problem with batching requirements: A real-world application. *Int J Prod Econ*, *105*(*2*), *445–458.* doi:10.1016/j.ijpe.2004.05.029.

Wahl, S., Gahm, C., & Tuma, A. (2023). Knowledge base for batch-processing machine scheduling research, *Mendeley Data (V3)*. doi:10.17632/7cv58py5hk.3.

Wang, J.-Q., Fan, G.-Q., & Liu, Z. (2020). Mixed batch scheduling on identical machines. *J Sched*, *23*(*4*), *487–496.* doi:10.1007/s10951-019-00623-9.

Webster, J., & Watson, R. T. (2002). Analyzing the Past to Prepare for the Future: Writing a Literature Review. *MIS Quarterly*, *26*(*2*), *13–23*. Retrieved from https://www.jstor.org/stable/4132319

Webster, S., & Baker, K. R. (1995). Scheduling Groups of Jobs on a Single Machine. *Oper Res*, *43*(*4*), *692–703.* doi:10.1287/opre.43.4.692.

Wu, K. (2014). Taxonomy of batch queueing models in manufacturing systems. *Eur J Oper Res*, *237*(*1*), *129–135.* doi:10.1016/j.ejor.2014.02.004.

Yang, F., Davari, M., Wei, W., Hermans, B., & Leus, R. (2022). Scheduling a single parallel-batching machine with non-identical job sizes and incompatible job families. *Eur J Oper Res*, *303*(*2*), *602–615.* doi:10.1016/j.ejor.2022.03.027.

Yin, Y., Li, D., Wang, D., & Cheng, T. C. E. (2021). Single-machine serial-batch delivery scheduling with two competing agents and due date assignment. *Ann Oper Res*, *298*(*1-2*), *497–523.* doi:10.1007/s10479-018-2839-6.

Yin, Y., Wang, Y., Cheng, T. C. E., Wang, D.-J., & Wu, C.-C. (2016). Two-agent single-machine scheduling to minimize the batch delivery cost. *Comput Ind Eng*, *92*, *16–30.* doi:10.1016/j.cie.2015.12.003.

Yu, J.-M., Huang, R., & Lee, D.-H. (2017). Iterative algorithms for batching and scheduling to minimise the total job tardiness in two-stage hybrid flow shops. *Int J Prod Res*, *55*(*11*), *3266–3282.* doi:10.1080/00207543.2017.1304661.

Yuan, J. J., Lin, Y. X., Cheng, T. C. E., & Ng, C. T. (2007). Single machine serial-batching scheduling problem with a common batch size to minimize total weighted completion time. *Int J Prod Econ*, *105*(*2*), *402–406.* doi:10.1016/j.ijpe.2004.04.014.

Yuan, J. J., Liu, Z. H., Ng, C. T., & Cheng, T. C. E. (2006). Single machine batch scheduling problem with family setup times and release dates to minimize makespan. *J Sched*, *9*(*6*), *499–513.* doi:10.1007/s10951-006-8776-2.

Yuan, J. J., Yang, A. F., & Cheng, T. C. E. (2004). A note on the single machine serial batching scheduling problem to minimize maximum lateness with identical processing times. *Eur J Oper Res*, *158*(*2*), *525–528.* doi:10.1016/S0377-2217(03)00361-8.

Zarook, Y., Rezaeian, J., Tavakkoli-Moghaddam, R., Mahdavi, I., & Javadian, N. (2015). Minimization of makespan for the single batch-processing machine scheduling problem with considering aging effect and multi-maintenance activities. *Int J Adv Manuf Technol*, *76*(*9-12*), *1879–1892.* doi:10.1007/s00170-014-6342-1.

Zdrzałka, S. (1991). Approximation algorithms for single-machine sequencing with delivery times and unit batch set-up times. *Eur J Oper Res*, *51*(*2*), *199–209.* doi:10.1016/0377-2217(91)90250-Y.

Zdrzałka, S. (1995). Analysis of approximation algorithms for single-machine scheduling with delivery times and sequence independent batch setup times. *Eur J Oper Res*, *80*(*2*), *371–380.* doi:10.1016/0377-2217(93)E0160-Y.

Zehetner, D., & Gansterer, M. (2022). The collaborative batching problem in multi-site additive manufacturing. *Int J Prod Econ*, *248*, *108432.* doi:10.1016/j.ijpe.2022.108432.

Zhang, J., Yao, X., & Li, Y. (2020). Improved evolutionary algorithm for parallel batch processing machine scheduling in additive manufacturing. *Int J Prod Res*, *58*(*8*), *2263–2282.* doi:10.1080/00207543.2019.1617447.

Zhou, S., Li, X., Chen, H., & Guo, C. (2016). Minimizing makespan in a no-wait flowshop with two batch processing machines using estimation of distribution algorithm. *Int J Prod Res*, *54*(*16*), *4919–4937.* doi:10.1080/00207543.2016.1140920.

Zitzler, E., Laumanns, M., & Thiele, L. (2001). *SPEA2: Improving the strength pareto evolutionary algorithm* (TIK Report), ETH Zurich, Computer Engineering and Networks Laboratory. doi:10.3929/ethz-a-004284029.

Appendix A

Table 1: Classified literature with *sb*, *S*, and unbounded batch capacity (46)

A	B	C	D	E & F	Reference
S	pF, if, s, sb, ia	F	-	MILP, PRH	Dobson et al. (1987)
S	pF, if, sFS, sb	C	-	oCH*	Naddef & Santos (1988)
S	dJ, if, sF, sb, ia	Lmax	-	oAS*	Zdrzałka (1991)
S	pC, s, sb	C	-	PTExAlg	Shallcross (1992)
S	pF, dlJ, if, sF, sb, ia	FY	-	oExAlg*, GCH	Tamer Unal & Kiran (1992)
S	if, sF, sb	F	-	DP	Cheng et al. (1994)
S	dJ, sb	NumB, cE, wSum	NP	DM, DP	Cheng & Gordon (1994)
S, R	dJ, dC, s, sb	F	-	LP, oExAlg*, PRH, oHeu*	Hakim Halim et al. (1994)
S	pC, dJ, dlJ, s, sb	wU	NP	PTExAlg, PPTExAl, DP	Hochbaum & Landy (1994)
S	pCo, dlJ, s, sb	wUt	NP	DP, FPTAS	Cheng & Kovalyov (1995)
S	rJ, dJ, if, sF, sb, ia	Lmax, Cmax	-	oAS*	Zdrzałka (1995)
S, P	s, sb	E, wE, Emax, NumB, wSum	NP	DP	Cheng et al. (1996b)
S	pF, daC, if, s, sb, ia	wU	NP	DP, FPTAS	Cheng & Kovalyov (1996)
S	dJ, if, sF, sb, ia	U	NP	PRH, LS, SA, TS, VND, GA	Crauwels et al. (1996)
S	if, sF, sb	wC	-	BnB, GCH, PRH, LS, SA, TS, TA, GA	Crauwels et al. (1997)
S	pC, s, sb	wC	-	PTExAlg	Hochbaum & Landy (1997)
S	pF, daC, if, sF, sb, ia	wT, wU, wSum	NP	-	Kovalyov (1997)
S	if, sF, sb, ia	wC	-	BnB, oCH*, LS	Crauwels et al. (1998)
S	dlJ, if, sFS, sb, ia	wE, cSR, wSum	-	DM, PRH, Heu-ExSM*	Jordan & Drexl (1998)
S	pC, pCo, dJ, dlJ, sM, sCo, sb	Lmax, wRC	-	PTExAlg	Cheng et al. (2001)
S	dJ, dlJ, cr1, s, sb	C, wC, T, Tmax, wT, Lmax, U, wU, FY	NP	oExAlg*	Cheng & Kovalyov (2001)
S	dJ, net, s, sb	Lmax	-	PTExAlg	Ng et al. (2002)
S	pC, rJ, net, s, sb	C	-	DP	Ng et al. (2003a)
S	pCo, sCo, sb	C, cC, wSum, OS	-	-	Ng et al. (2003b)
S	pCo, sCo, sb	C	-	PTExAlg, DP	Ng et al. (2004)
S	rJ, net, s, sb	Lmax	-	PTExAlg	Yuan et al. (2004)
S	dJ, if, sF, sb, ia	U	NP	BnB	Crauwels et al. (2005)
S	p1, s, sb	F	-	LP, oExAlg*	Mosheiov et al. (2005)
S	rJ, sF, sb	Cmax	sNP	DP, PTAS, oHeu*	Yuan et al. (2006)
S	dJ, if, sF, sb, ia	wU	NP	DP, FPTAS	Erel & Ghosh (2007)
S	dJ, if, sF, sb, ia	Lmax	NP	-	Lu & Yuan (2007)
S	pCo, s, sCo, sb	C, cIh, cP	-	oExAlg*	Shabtay & Steiner (2007)
S	pDet, s, sb, ia	Cmax	sNP	DP, FPTAS	Ji & Cheng (2010)
S	p1, if2, sF, sb	F, OSA	-	LP, LS	Mor & Mosheiov (2011)
S	p1, pDet, s, sb	C	-	oAS*	Mor & Mosheiov (2012b)
S	p1, pCo, s, sb	C, cC, wSum, OS	-	LP, oAS*	Mor & Mosheiov (2014a)
S, avFlx	p1, s, sb	C	-	oAS*	Mor & Mosheiov (2014b)
S	s, sb	C, cR, Par, wSum, OS	NP	PTExAlg, PPTExAl	Shabtay (2014)
S	pCo, dJ, if, sFS, sb, ia, on	wT, cRc, cSR, wSum	NP	DM, DP	Giglio (2015)
S	dJ, if2, sF, sb	wC, Lmax, wU, Cmax, cP, OSA	NP	oExAlg*	Kovalyov et al. (2015)

S	dJ, if2, sF, sb	C, Lmax, wU, NumB, OSA	NP	MILP, DP	Yin et al. (2016)
S	dJ, cf#, s, sb	AR, OSA	NP	PPTExAl, FPTAS	Li et al. (2017)
S	dJ, if2, sF, sb	C, Lmax, cSR, ParA, OSA	-	DP	Qi & Yuan (2017)
S	dJ, chain, cr1, s, sb	Fmax, Cmax, Par	sNP	PTExAlg	Geng et al. (2018)
S	dJ, if2, cr1, s, sb	Lmax, AR, Cmax, ParA	-	-	He et al. (2020)
S	da, daC, if2, sF, sb, ia, tOub	U, cIh, cE, cT, wSumA, OSA	NP	DP, FPTAS	Yin et al. (2021)

Table 2: Classified literature with *sb*, *S*, and bounded batch capacity (19)

A	B	C	D	E & F	Reference
S	rJ, dJ, if, crJ, cr1, sF, pb, sb, fb, ia	wF, Lmax	-	-	Webster & Baker (1995)
S	if, cr1, sF, sb, bLb	wF	-	DP, oCH*, Heu-ExSM*	Sung & Joo (1997)
S	cr1, s, sb	wC	sNP	PTExAlg	Yuan et al. (2007)
S	p1, cr1, s, sb, bLb	F	-	LP, oExAlg*	Mosheiov & Oron (2008a)
S	crJ, sb	C	sNP	MINLP, DP, PRH, LS	Tanrisever & Kutanoglu (2008)
S	dJ, dC, crJ, s, sb, bLb	cT, cSR, wSum	-	DM, DP, PRH	Chrétienne et al. (2011)
S	p1, cr1, s, sb	F, Cmax, Par	-	oAS*	Li et al. (2012)
S	p1, dC, cr1, s, sb, bLb	E, T, cSR, wSum	-	oCH*	Hazır & Kedad-Sidhoum (2014)
S	dJ, if, cr1, sFS, sb	wT	-	MILP, PRH, TS	Suppiah & Omar (2014)
S	cr1, s, pb, sb, ia, on-sc	Wt	-	QM	Wu (2014)
S	pDet, cr1, s, blk, buf, sb, tOub	Cmax	-	MILP, oExAlg*, PRH	Pei et al. (2015)
S, avFlx	pDet, rJ, crJ, sb	Cmax	NP	MINLP, PRH, GA, ICA	Zarook et al. (2015)
S	pDet, dJ, if, cr1, sFS, sb	C, Emax, Tmax, Lmax, U, Cmax	-	PTExAlg	Pei et al. (2017a)
S	pDet, pLe, dJ, cr1, sDet, sb	Emax, U, Cmax	-	PTExAlg	Pei et al. (2017b)
S	pDet, pLe, rJ, if, cr1, sF, sDet, sb	Cmax	-	PRH, SA, VNS, PSO, Meta-LS	Fan et al. (2018)
S	pLe, pCo, cr1, sDet, sb	Cmax	-	SA, GSA, Meta-Meta	Pei et al. (2018)
S, P	pLe, dC, dlC, cr1, sDet, sb	Emax, U	-	MILP, PRH, LS, SA, VNS, GSA, Meta-Meta	Pei et al. (2019a)
S	rJ, dJ, crJ, buf, mode, sb	T, cP, Par	NP	PRH	Shin et al. (2020)
S	cr1, s, sb	Cmax, cP, ParA	-	PTExAlg	He et al. (2022)

Table 3: Classified literature with *sb* and parallel machines (21)

A	B	C	D	E & F	Reference
R	sM, sb, ia	F	-	DM, oCH*	Dobson et al. (1989)
P	pF, cf, sF, sb, ia	Cmax	-	oCH*	Tang (1990)
S, R	dJ, dC, s, sb	F	-	LP, oExAlg*, PRH, oHeu*	Hakim Halim et al. (1994)
P	pF, if, sF, res, sb	wF	NP	MILP, oCH*	Dobson & Khosla (1995)
P	cr1, s, sb	C	-	DP	Cheng et al. (1996a)
S, P	s, sb	E, wE, Emax, NumB, wSum	NP	DP	Cheng et al. (1996b)
P	dlJ, if, sF, sb	cIh	NP	GCH, GA, Meta-LS	Luu et al. (2002)
P	dJ, s, sb	Lmax, U	-	DP, PRH	Lin & Jeng (2004)
P	pDet, s, sb, ia	C	sNP	oExAlg*, oAS*	Leung et al. (2008)

Q2	pCM, sM, sb	C	-	LP, oAS*	Mor & Mosheiov (2012a)
P	pF, if, sFS, sb	C	sNP	MILP, PRH, LS, VNS	Shen et al. (2013)
R	pC, dlJ, cr1, sM, sb	F	-	MILP, oCH*	Hidayat et al. (2016)
P2, avStoc	rJ, crJ, s, sb, tOub	Cmax	sNP	MILP, PRH	Pei et al. (2016)
R, avDyn	rJ, dJ, if, cr1, sFMS, elig, sb, ia, bLb	wC, wT, wSum	sNP	MILP, PRH, TS, Meta-Meta	Shahvari & Logendran (2017)
S, P	pLe, dC, dlC, cr1, sDet, sb	Emax, U	-	MILP, PRH, LS, SA, VNS, GSA, Meta-Meta	Pei et al. (2019a)
P	pDet, cr1, sM, sDet, dcr, sb	Cmax	NP	LS, PSO, oMetaP*, Meta-LS	Pei et al. (2019b)
P	pDet, if, crJ, sF, elig, sb	Cmax, cSub, wSum	-	oCH*, VNS, Meta-Meta	Liao et al. (2020)
P	pDet, pLe, rJ, cr1, s, sb	Cmax	NP	PRH, LS, DE, GA, PSO, oMetaP*, Meta-Meta	Pei et al. (2021)
P	dJ, if, crJ, sFS, sb	wF, wT, Lex	NP	MILP, PRH, LS	Gahm et al. (2022b)
R, avSt	rJ, dJ, if, sMS, elig, sb, ia	wC, wT, wSum	-	MILP, BnP	Shahvari et al. (2022)
P	dJ, if, crJ, sFS, sb	wF, wT, Lex	NP	MILP, PRH, LS, Heu-Opt*	Uzunoglu et al. (2023)

Table 4: Classified literature with *sb* and multi-stage machine environment (32)

A	B	C	D	E & F	Reference
PF	dC, sM, sb	F	-	MILP, LS, Heu-Opt*	Hakim Halim & Ohta (1993)
F2	if2, sFM, sb, ia	Cmax	-	DP	Cheng & Kovalyov (1998)
F2	s, sAnt, sb	Cmax	NP	DP	Cheng & Wang (1998)
F2	dlJ, if, sF, sb, ia	wE, cSR, wSum	-	PRH, GA	Jordan (1998)
F2	s, sb	Cmax	sNP	PRH, LS	Cheng et al. (2000)
F2, O2	sM, sAnt, sb	Cmax	NP, sNP	FPTAS, PRH	Glass et al. (2001)
F2	s, now, sb	Cmax	NP	oAS*	Lin & Cheng (2001)
HF2	pF, if, sF, sb	Cmax	-	oExAlg*	Cheng et al. (2004)
F2, HF2	in-tree, cr1, sM, pb, sb	Cmax	sNP	oAS*	Kovalyov et al. (2004)
F	p1, s, sb	C	-	oAS*	Mosheiov et al. (2004)
F2	s, sAnt, sb	Cmax	sNP	oAS*	Lin & Cheng (2005)
PF	pC, s, cb, sb	Cmax	-	oExAlg*	Mosheiov & Oron (2005)
O2	s, cb, sb	Cmax	-	oExAlg*	Gribkovskaia et al. (2006)
HF2	in-tree, s, sb	Cmax	sNP	oCH*	Lin et al. (2007)
F	sM, sAnt, cb, sb	Cmax	sNP	DP, oAS*	Ng & Kovalyov (2007)
F2	cr1, s, sAnt, now, cb, pb, sb	Cmax	sNP	oAS*	Oulamara (2007)
HF	pC, if, sF, sb, ia	F, cSR	-	GA	Quadt & Kuhn (2007)
HF, avDyn	dJ, net, if, sFS, minL, elig, mode, sb, ia	wT, Cmax, cSR, wSum	-	MILP, PRH	Voß & Witt (2007)
O	pC, s, cb, sb	F, Cmax	-	oExAlg*, oAS*	Mosheiov & Oron (2008b)
O	dJ, re, sM, conc, sb	wC, Lmax, wU	-	oExAlg*	Lin & Cheng (2011)
J2	p1, sM, cb, sb	Cmax	-	oExAlg*	Mosheiov & Oron (2011)
HF2	dJ, in-tree, s, sb	C, Fmax, T, U	-	DP	Hwang & Lin (2012)
F2	rJ, crJ, s, now, cb, pb, sb, bM	Cmax	NP	MILP, PSO	Muthuswamy et al. (2012)
J	if, sFMS, sb, ia	Cmax	NP	PRH, TS	Shen & Buscher (2012)
HF2	p1, sM, cb, sb	Cmax	-	PTExAlg, DP	Gerstl & Mosheiov (2013)
HJ	if, crJ, sFMS, cb, sb, bF, bLb	Cmax	NP	SA, GA	Castillo & Gazmuri (2015)
F2	in-tree, if, s, sb	Cmax	-	MILP, PRH	Liao et al. (2015)
HF, avSt	pJM, rJ, dJ, if, cr1, sFMS, elig, sb, ia, bLb	wC, wT, wSum	sNP	MILP, GCH, TS, GA, Meta-Meta	Shahvari & Logendran (2016)

F2	rJ, crJ, s, now, pb, sb	Cmax	-	MILP, PSO, oMetaP*, Meta-LS	Zhou et al. (2016)
HF2	dJ, s, sb	T	NP	MILP, oCH*, LS	Yu et al. (2017)
F	if, sFS, sb	Cmax	-	MILP, TS	Shen & Gupta (2018)
HF	crJ, sS, dcr, sb	Cmax	-	MILP, DE	Guo et al. (2020)

Table 5: Classified literature with *hb*

A	B	C	D	E & F	Reference
S	cf, crJ, s, sFS, hb	Cmax	sNP	MILP	Toksarı & Toğa (2022)
P	pB, crJ, s, hb	Cmax, Bal	NP	DM, PRH, GA, PSO, Meta-LS	Zhang et al. (2020)
P	dJ, if, crJ, s, elig, hb, tOub	cIh, cP, cSR, cTr	sNP	MILP, Meta-ExSM*	Zehetner & Gansterer (2022)

Appendix B

Table 6: Instance properties

Max. number of machines/stations	#	Max. number of jobs	#	Max. batch capacity	#	Max. number of job families	#
2	8	< 75	13	< 10	7	2	2
3	4	btw. 75 to 149	9	btw. 10 to 19	3	3	3
4	0	btw. 150 to 299	11	btw. 20 to 49	6	4	1
5	4	btw. 300 to 599	11	btw. 50 to 99	2	5	1
6	4	btw. 600 to 1,199	9	btw. 100 to 249	0	6	0
7	1	btw. 1,200 to 2,399	1	btw. 250 to 499	0	7	0
8	2	btw. 2,400 to 4,799	1	≥ 500	1	8	0
9	2	btw. 4,800 to 9,599	0	unknown	4	9	0
btw. 10 to 14	7	btw. 9,600 to 19,199	0			btw. 10 to 19	15
btw. 15 to 19	0	≥ 19,200	2			btw. 20 to 49	6
btw. 20 to 29	3	unknown	2			≥ 50	2
btw. 30 to 49	0					unknown	3
btw. 50 to 99	0						
≥ 100 or more	0						
unknown	1						