Solving large scale industrial production scheduling problems with complex constraints: an overview of the state-of-the-art

Article *in* Procedia Computer Science · January 2023

DOI: 10.1016/j.procs.2022.12.301

2 authors:

Manuel Schlenkrich
WU Vienna University of Economics and Business

Sophie N. Parragh
Johannes Kepler University of Linz

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 217 (2023) 1028–1037

Procedia Computer Science

www.elsevier.com/locate/procedia

4th International Conference on Industry 4.0 and Smart Manufacturing

Solving large scale industrial production scheduling problems with complex constraints: an overview of the state-of-the-art

Manuel Schlenkrich[a,b,*], Sophie N. Parragh[a]

[a] *Johannes Kepler University Linz, Altenbergerstraße 69, 4040 Linz, Austria*
[b] *RISC Software GmbH, Softwarepark 32a, 4232 Hagenberg im Mühlkreis, Austria*

Abstract

Production scheduling is challenging and the body of literature addressing various variants of the problem is large. It can roughly be divided into two streams: The first stream addresses and generalizes established scheduling problems, being general in the sense that they are not only applicable in a particular industry. The second stream works on less generic scheduling approaches for real industry cases by enriching standard models with all the required realistic aspects, such as process overlapping or sequence dependent setup times. Furthermore, different approaches have different limitations in terms of the problem size that they can tackle. The rise of Industry 4.0 has lead to a significant increase in data collection activities and the gathered information is used to build larger and more complex models. Industrial use cases may consist of several thousand operations on a large variety of machines, while classical benchmark instances tend to range up to only a few hundred of operations. It is therefore necessary to identify and highlight approaches, that can meet the challenges of scheduling in the era of Industry 4.0 and are suitable to tackle large scale problems.

In this work, we conduct a structured literature review on scheduling problems incorporating several real world aspects among a broad range of use cases. Based on the identified publications we find that advanced solution approaches for large scale scheduling problems usually belong to one out of three categories, namely metaheuristic methods, constraint programming and machine learning. Our review shows that comparably few contributions tackling (very) large scale problems exist, emphasizing the need for additional research in this field. We identify promising approaches for further research, such as powerful metaheuristics combining concepts of tabu search and genetic algorithms. We further discuss the possibility to enhance solution methods by integrating constraint programming concepts and investigating problem decomposition.

Peer-review under responsibility of the scientific committee of the 4th International Conference on Industry 4.0 and Smart Manufacturing

Keywords: Large scale scheduling; literature review; complex constraints; Industry 4.0

* Corresponding author. Tel.: +43 732 2468 5503
E-mail address: manuel.schlenkrich@jku.at

Peer-review under responsibility of the scientific committee of the 4th International Conference on Industry 4.0 and Smart Manufacturing
10.1016/j.procs.2022.12.301

1. Introduction

Scheduling refers to the sequencing of tasks and their assignment to suitable resources. Efficient production scheduling is crucial for manufacturers in order to maintain high resource utilization, minimize delays or keep production costs as low as possible, to mention just three of the commonly applied objectives. Applications of production scheduling arise in many different industries, such as the chemical-, textile-, steel- or electronics sector. Fuchigami and Rangel [10] provide a survey on different scheduling use cases in industry.

The rise of Industry 4.0 and newly emerging technologies, such as the internet of things or cloud computing, directly impact the planning and scheduling frameworks used in the production industry. The trend towards digitalization, automation and interconnection of systems operating at the manufacturing floor has significantly increased data collection activities [29]. Information becomes available in real-time and complex relationships within the production environment can be mapped to increasingly large models including various types of constraints. This trend towards larger and more complex types of scheduling problems arising in the era of Industry 4.0 motivates to highlight research that is facing these challenges.

There exists a large body of literature investigating different problem formulations and solution approaches, such as mathematical programming techniques, (meta-) heuristics and constraint programming. While mathematical programming techniques seek to solve the underlying scheduling problem to optimality, heuristics and metaheuristics aim at finding good solutions in reasonable computation times. The constraint programming paradigm leverages different ingredients, like domain reduction and constraint propagation with tailored heuristic approaches so as to efficiently reduce the search space. Finally, very recently also ideas from machine learning have been used to tackle complex scheduling problems.

A lot of effort has been put into closing the gap between ”academic” scheduling problems and real world use cases, by gradually enriching mathematical models, such that they are able to capture all the required features. In the literature, this is usually either done in a general way that allows their adaptation to different problems or in a very specific way, solving a concrete real world problem. Furthermore, academic benchmark problem instances are often limited to a few hundred operations whereas practical problems range up to several thousands, even hundreds of thousand operations.

Since the body of literature in this field is very large and the range of investigated problems in terms of included real world aspects and instance size is broad, the process of identifying promising approaches can be very time consuming. This paper aims at giving an overview of selected state-of-the-art scheduling approaches and their respective limitations in order to support the development of tailored solution methods for practical scheduling problems. This review differentiates itself from existing surveys by explicitly taking into account the size of the investigated scheduling problems and therefore provides valuable insights on how suitable different approaches are for adaptation to real-world problems.

The existing literature reviewing scheduling approaches can be separated into two streams. The first stream of surveys summarizes work solving generic extensions of classical scheduling problems. Literature reviews of this type for example investigate the flexible job shop scheduling problem [5, 40], assembly flow shop scheduling [17], non-permutation flow shop scheduling [31] or resource constrained project scheduling [15, 30]. The second stream of reviews deals with approaches to tailor solution methods to a certain problem at hand, oftentimes motivated by a specific industry. These surveys on specific practical use cases summarize, e.g., work on chemical production scheduling [25].

The remaining paper is structured as follows. Section 2 introduces the main challenges that manufacturers face, Section 3 presents the research methodology of the conducted structured literature review, Section 4 categorizes the investigated papers according to the developed solution methods and limitations in instance size and Section 5 summarizes the main findings.

2. Industrial scheduling problems

At the core of every scheduling problem there are usually two major tasks - the assignment of operations to suitable resources and defining their processing sequence. These tasks need to be performed by a scheduling method satisfying a set of constraints, such that an objective function, e.g., the makespan or the total tardiness, is optimized. The set of constraints can vary significantly from one scheduling problem to another and reflect different aspects of the underlying real-world problem.

Many classical academic scheduling problems, such as the job shop scheduling problem (JSSP), only consider comparably few and simple restrictions: In job-shop scheduling n jobs need to be processed on m machines and the processing times are known. The sequences in which the machines need to be visited can differ from job to job, but are also fixed beforehand. A job can only be started at the subsequent machine, if processing on the predecessor machine is finished and preemption is not allowed. Even though the classical job shop problem is already a challenging optimization problem, practical problems usually require the consideration of a number of additional constraints. Changing the setup of machines between jobs usually takes time and may depend on the processing sequence, machines might have alternatives and processing times could vary between them or multiple resources could be needed in parallel.

In the following, we briefly discuss several of these extensions that frequently appear in real world scheduling problems. These concern the characteristics of *resources* used to perform a given task, the *sequence* in which tasks are processed, complex task *relationships* and constraints on their *timing*.

Resources are usually not unique and there are multiple different options for a task to be processed. This is true for machines, as well as for workers, who perform the respective production step. Especially for staff there can be a significant difference in experience and skill level, which results in resource dependent processing times. For machines this difference might result from variations in the available technology. Instead of initially allocating one fixed resource it is reasonable to define a set of suitable resources for each task and providing information on the respective processing times. The flexible job shop scheduling problem (FJSSP) [40] has been proposed to capture this more realistic aspect and extends the standard job-shop problem by the possibility to choose the processing machine from a list of suitable machines. Kress et al. [18], e.g., incorporate heterogeneous machine operator qualifications by taking into account machine- and operator-dependent processing times. Another aspect of real world use cases is the differentiation of various kinds of resources. In classical scheduling problems, a resource can usually only process one task at a time. In reality this might be the case for a part of the resources such as tools or workers. However there might also be resources, such as shared workplaces, that have a scalable capacity depending on the present workforce, as for instance in [4] or even unlimited resources such as an outdoor space to dry freshly painted material. Some very specific machines might also be designed to produce two products in parallel, such as the left and the right part in an injection molding machine. This motivates the incorporation of batch production constraints to the scheduling models that make sure that some tasks can only be processed in parallel. Ham [12] for instance investigate the FJSSP with parallel batch processing machines and compatible job families. Further examples of this type of problem extension can be found, e.g., in Mahmoodjanloo et al. [24], who present an approach to model reconfigurable machine tools for FJSSP or in [45], where the flexible assembly job shop problem with sequence dependent setup times and part sharing is discussed.

In order to process an operation, the necessary resources need to be set up. This might be attaching a certain tool to a machine or cleaning a tank. In many practical cases the *sequence* of tasks at the resource has a significant influence on the setup time needed between two production tasks. For a machine responsible for coloring parts of a product, the setup time between two tasks requiring the same color might be close to zero, while for tasks with different colors, it may take up to several minutes to clean the machine and change the cartridge. This information needs to be considered in order to obtain feasible production schedules. For instance, Shen et al. [34] develop a metaheuristic, namely a tabu search algorithm, to tackle this aspect in the context of the flexible job shop scheduling problem with sequence dependent setup times.

In many classical scheduling problems the *relationship* between different tasks is given by a simple sequence, where one task can be started, as soon as its predecessor task has ended. In practice, however, these relationships, which impact the timing of the different tasks, can have more complex properties. In industries like steel production, where products need to have a certain temperature in order to be processed, minimal and maximal overlap times between two tasks can be relevant. Overlapping of operations in flexible job shop scheduling is studied in [27]. It might also be the case, that the successor task needs to be processed immediately after the predecessor has been finished. This is referred to as a so called no-wait constraint and is for example investigated in [2] and [36]. On the other hand, also transportation times may have to be considered. Transportation times for a multi-objective version of the FJSSP are introduced e.g., in [7].

Usually, in industrial cases, the scheduling unit receives information from the internal ERP system concerning lot sizes and due dates. In many cases it also informs about specific time windows that should be met, affecting the *timing* of the different operations in production. So, additionally to the sequence of tasks to be performed, there are also earliest and latest start or finish times that need to be met. An overall due date for a job usually comes with a tardiness cost, that needs to be paid when the due date is violated. In some cases also earliness is associated with a cost. The consideration of these types of time windows can be found, e.g., in [16, 3].

3. Research methodology: structured literature review

The main contribution of this work is to provide an overview of the state of the art in solving (very) large scale scheduling problems. This goal has been pursued by performing a structured literature review. In a first step an appropriate search string, capturing the main aspects of interest, was developed. In order to restrict search results to publications within the investigated research subject the keywords *scheduling*, *job*, *shop* and *optimization* need to be included within title, abstract or keywords. Moreover either the combination *large scale* instance** or *large size* instance** need to be present in order to account for the main focus on solving very large problems. The search was performed on journal and conference proceedings papers available on Scopus published between 2022 and 2017 and resulted in a total of 81 publications. These papers were then, in a next step, filtered manually in 2 phases. In the first phase the papers were checked for the actual type of investigated problem. Scheduling problems, such as job shop, flexible job shop, open shop or resource constrained project scheduling in general allow for situations in which several jobs are processed on multiple different machines in a different sequence. Therefore they all reflect an abstraction of industrial scheduling problems and are considered within the review. Flow shop, hybrid flow shop, parallel machine scheduling, single machine scheduling or other types of problems, such as assignment problems, were classified as out of scope and have not been considered further. In this phase 41 papers out of 81 were removed. The remaining 40 papers were filtered manually in a second phase, checking for significant size of the investigated problem instances. A paper was classified as large scale, if instances consisting of least 1000 operations are tackled. In this phase 26 papers were sorted out, resulting in 14 extant publications. In order to also account for publications published before 2017, all cited papers of the resulting 14 publications were filtered by the search string *large scale* instance** or *large size* instance** within title, abstract or keywords. As a result of this secondary level filtering 20 additional publications between 2021 and 2001 were considered. After removing duplicates and repeating the manual two phase selection explained earlier, 6 papers remained. Combining the 14 identified papers of the first stage and the 6 papers of the secondary stage, a total of 20 papers are considered in our structured literature review and can be found in Table 1. The selected papers are further discussed in Section 4.

4. Solution methods: a state-of-the-art

Advanced solution approaches to solve scheduling problems considering complex real-world constraints belong to four major categories, namely mixed integer programming, metaheuristics, constraint programming, and machine learning (we exclude simple construction heuristics which can potentially provide solutions for problems of any size). Within the conducted literature review presented in Section 3 there was however no paper identified, that applies mixed integer programming to problem instances considered as large scale. In the following, we provide an overview of the in the review identified publications addressing large scale scheduling problems, meaning that problem instances with at least 1000 operations are tackled. For each contribution, we discuss the considered scheduling problem, the developed solution approach and the maximum problem size that has been tackled.

4.1. Metaheuristics

The first category of solution methods to tackle large scale production scheduling problems concerns the application of metaheuristic approaches. Metaheuristic methods are designed to deliver good solutions for the investigated problems, however do not guarantee global optima. Nevertheless in industry cases a good feasible solution for a large problem instance that captures all required features may already be sufficient. Within the conducted review presented

Table 1. Reviewed papers on large scale scheduling.

Paper	Problem type	Objective	Solution approach	Instance size (#ops)
Ali et al. [1]	JSSP	Makespan	Metaheuristic (Genetic Algorithm)	6000
Da Col and Teppan [6]	JSSP	Makespan	Constraint Programming	1,000,000
Defersha and Rooyani [8]	FJSSP with features	Makespan	Metaheuristic (Genetic Algorithm)	5600
El-Kholany et al. [9]	JSSP	Makespan	Metaheuristic (Decomposition)	2000
Hajibabaei and Behnamian [11]	FJSSP with features	Composite[1]	Metaheuristic (Tabu Search)	1000
Han and Yang [13]	JSSP	Makespan	Deep Reinforcement Learning	2000
Lei et al. [21]	FJSSP	Makespan	Deep Reinforcement Learning	2000
Lunardi et al. [22]	FJSSP with features	Makespan	Constraint Programming	1000
Lunardi et al. [23]	FJSSP with features	Makespan	Metaheuristic (Various)	2600
Matta [26]	Multiprocessor OSSP	Makespan	Metaheuristic (Genetic Algorithm)	1792
Mogali et al. [28]	Blocking JSSP	Makespan	Metaheuristic (Tabu Search)	2000
Schwenke et al. [32]	JSSP with features	Tardiness	Metaheuristic (Tabu Search)	40,000
Shao and Kim [33]	JSSP	Makespan	CNN and Iterative LS	1000
Singer [35]	JSSP	Tardiness	Metaheuristic (Decomposition)	1000
Sobeyko and Mönch [37]	FJSSP	Tardiness	Metaheuristic (Iterative LS)	1600
Teppan [38]	JSSP	Makespan	Dispatching Rules	100,000
Yang et al. [41]	FJSSP	Makespan	Metaheuristic (Dragonfly Algorithm)	4000
Zhang et al. [42]	JSSP	Makespan	Deep Reinforcement Learning	2000
Zhang et al. [43]	FOSSP	Makespan	Metaheuristic (Various)	1500
Zhang et al. [44]	FAJSSP	Multi-objective[2]	Metaheuristic (Ant Colony System)	1336

CNN = convolutional neural network, FAJSSP = flexible assembly job shop scheduling problem, FJSSP = flexible job shop scheduling problem, FOSSP = flexible open shop scheduling problem, JSSP = job shop scheduling problem, LS = local search, OSSP = open shop scheduling problem

[1]Inventory cost, total tardiness, delivery time and makespan

[2]Makespan, total tardiness and total workload

in Section 3 various types of metaheuristics have been identified, i.e. population based and nature inspired methods, trajectory based approaches, decomposition algorithms or combinations of these.

Population based and nature inspired metaheuristics follow principles that can be observed in phenomena such as animal behaviour and natural selection in evolution. Genetic algorithms are popular examples for population based metaheuristic that successively modify a set of solutions (also called population) under the principle of survival of the fittest. Selection, crossover and mutation operators are applied in order to achieve intensification and diversification of the solution candidates. Defersha and Rooyani [8] propose a two-stage genetic algorithm for the FJSSP with additional features, such as sequence dependent attached or detached setup time, release dates and lag time. In the first stage of the algorithm the order in which operations are considered for assignment is decided in a greedy manner, while the second stage promotes solution diversification in a common genetic algorithm fashion. With this approach the authors tackle flexible job shop problems with up to 140 jobs, consisting of 20 to 40 operations each, on up to 80 machines, where each operation has between 4 and 15 alternative machine options. They demonstrate superiority of the two-stage approach by providing solution quality of the largest problem instance within 30 minutes, that was not reached within 72 hours by a regular genetic algorithm. Also Matta [26] presents a genetic algorithm for the multiprocessor open shop scheduling problem. Problem instances up to a size of 112 jobs to be processed on 16 stages, leading up to 1792 operations within the open shop environment, are investigated. Ali et al. [1] develop a genetic algorithm approach based on new virtual crossover operators for the dynamic job shop scheduling problem. Instances with up to 300 jobs on 25 machines are tackled.

Zhang et al. [43] propose three metaheuristic optimization approaches for medical examination scheduling, which can be represented as a flexible open shop, namely hybrid particle swarm optimization, a genetic algorithm and simulated annealing. Particle swarm optimization, similar to genetic algorithms, modifies a population of solutions, in this context also called particles, in order to find the best solution within the search space. Simulated annealing is inspired by the process of annealing in metallurgy. The solution candidate is iteratively improved, while there is an over time decreasing probability to accept worse candidates to foster diversification. The genetic algorithm is shown to perform best out of the three on a variety of large scale test instances ranging up to 150 clients to be examined in 10 departments, resulting in 1500 operations in total.

Zhang et al. [44] investigate multi-objective optimization in flexible assembly job shop scheduling aiming at minimizing makespan, total tardiness and total workload. The authors propose a distributed ant colony system, which is a metaheuristic approach, that is inspired by the behaviour of ants travelling between their nest and potential food sources. With this approach the authors tackle problem instances with up to 1336 operations on 40 machines. Yang et al. [41] present a so called dragonfly metaheuristic, which is a method performing the phases of exploration and exploitation in a manner that is inspired by the behaviour of insects during their hunt for prey. The authors incorporate a dynamic opposite learning strategy into the algorithm in order to enhance the population initialization and generation jumping stage. Dynamic opposite learning is a tool to avoid getting stuck in local optima. Problem instances with up to 200 jobs, consisting of 10 to 20 operations per job and 120 machines are investigated.

Next to population based methods, also several trajectory based approaches have been identified to be suitable for large scale scheduling problems. Instead of combining a population of solutions, trajectory based methods modify a single solution candidate that moves through the search space. In [11] a tabu search approach for flexible job shop scheduling with unrelated parallel machines and resource dependent processing times is proposed. The presented metaheuristic method works on a matrix based solution representation containing information on the operation sequence, assignment of machines and flexible resources. An initial solution is then improved iteratively by performing solution modifications, also called neighborhood operations, while keeping track of previous modifications in the tabu list in order to prevent getting stuck in local optima. Problem instances with up to 100 jobs on 10 machines are investigated and within a computational experiment the proposed tabu search outperforms a genetic algorithm, especially on the largest instances. Mogali et al. [28] investigate the blocking job shop problem, which is an extension of the job shop problem, in which jobs continue to block the used machine after completion until the following machine becomes available. A tabu search algorithm based on neighborhood operations for the regular job shop problem, which were modified for the blocking case, is presented. The largest tackled problem instances consist of 100 jobs to be processed on 20 machines, resulting in 2000 operations. Also [32] develop a tabu search for job shop scheduling problems with routing, batching and release dates arising from wafer fabrication. Different neighborhood structures are validated and combined in order to efficiently solve large scale instances with more than 40,000 operations to be scheduled. Sobeyko and Mönch [37] develop an iterative local search method using a simulated annealing acceptance criterion and hybridize the approach by means of the shifting bottleneck heuristic and a variable neighborhood search. The authors aim at minimizing the total weighted tardiness of flexible job shops and investigate problem instances up to a size of 20 jobs, with 80 to 120 operations each, that are supposed to be processed on 23 to 32 machines.

Solution approaches do not necessarily need to belong to a single category of metaheuristic. Advanced combinations of population- and trajectory based methods can make use of the benefits from both worlds. Lunardi et al. [23] investigate the online printing shop scheduling problem, a challenging scheduling problem incorporating many real world restrictions, such as sequence flexibility, resumable operations, sequence dependent setup times, partial overlapping between operations, unavailability of machines or fixed operations. They present four different metaheuristic solution approaches to tackle the problem, namely a genetic algorithm, differential evolution, tabu search and iterated local search. The authors also propose to combine the metaheuristics in order to create more advanced ones. The introduced methods are based on a solution representation scheme using two structures for the assignment of operations to machines and the sequence of the non-fixed operations. With the presented metaheuristics problems with up to 2600 operations on 100 machines are tackled and for several instances known lower bounds have been improved. A combination of tabu search and differential evolution appears to be the most efficient method and outperforms the other heuristics and a constraint programming approach. In this approach the tabu search is performed until no improvement can be reached within a predefined time threshold. Then the incumbent tabu search solution is used to generate several pertubated solutions, on which another local search is performed. The thus obtained local minima of the pertubated solutions are then used as the initial population of the proposed differential evolution algorithm, which proceeds until a final time limit is reached. The combination of a good starting point resulting from a trajectory based metaheuristic - in this case a tabu search - and a broad diversification from the population based method appears well suited to solve large problem instances of the complex scheduling problem.

Another stream of metaheuristics approaches large problem instances through decomposition by solving smaller subproblems efficiently and combining the results to an overall solution. El-Kholany et al. [9] present a method for decomposing a job shop scheduling problem into time windows, whose operations are then scheduled using multi-shot Answer Set Programming (ASP), which is a form of declarative programming. With this approach problem instances with up to 100 jobs on up to 20 machines resulting in 2000 operations are tackled. The decomposition approach with multi-shot ASP is compared to the single-shot version of ASP, as well as to a-state-of-the-art Constraint Programming (CP) solver by OR-tools. After a timeout of 1000 seconds the CP solver gave the best solution quality of the three methods, with the multi-shot version clearly outperforming the single-shot version. It could be observed that with larger instance size, the decomposition method delivered results closer to the results provided by the CP solver, which however remained best over all size categories. Singer [35] decompose a job shop scheduling problem into multiple time windows and solve the resulting sub problems by means of a shifting bottleneck heuristic. The objective to be optimized is the total weighted tardiness and problem instances with up to 1000 operations on 10 machines are tackled. In [38] different dispatching rules are evaluated and combined in order to produce efficient schedules for the production of semiconductors. The rules are tested on very large problem instances with up to 100,000 operations on 1000 machines.

4.2. Constraint programming

Constraint programming (CP) is a programming paradigm for combinatorial optimization problems that uses constraint propagation in order to reduce variable domains. It has received considerable attention in the scheduling literature, due to its capability of solving extremely large scheduling problems. Da Col and Teppan [6] compare the leading constraint programming solvers from IBM and Google. Within this study the IBM CP Optimizer was able to solve some instances with up to 1 million operations to optimality and also the open source solver by Google could solve problems with up to 100,000 operations. The time limit for these experiments was set to 6 hours. The sheer size of the solved instances motivates a closer investigation of the techniques used within these solvers. Laborie et al. [20] present the main components used within the IBM CP Optimizer. Two of them, namely the "Self adapting large neighbourhood search" and the "Failure directed search", are further elaborated in [19] and [39].

Basically these two heuristics pursue different goals. "Self adapting large neighbourhood search" aims at finding good schedules in a very broad area of the solution space by modifying large parts of the schedule. This is done by fixing parts of the solution and removing others. The partial schedule is then completed to a full solution by using one out of several available completion strategies. The method is self adapting in the sense, that the performance of these completion strategies is updated in each iteration, such that successful strategies are used more often in the future. The second heuristic "Failure directed search" tries to reach infeasible solutions as quickly as possible, to reduce the search space. In order to do this, the method branches on variables, that are likely to lead to infeasible solutions. Also for "Failure directed search" the most successful, meaning early detection of infeasible solutions, choices are tracked and chosen with higher frequency in the next iterations. "Failure directed search" is especially useful in situations, where the "Self adapting large neighbourhood search" has difficulties to find better solutions. In the IBM CP Optimizer framework these two search algorithms are combined with constraint propagation in order to shrink variable domains and therefore further enhance the solution procedure. As it appears that the propagation of scheduling constraints works particularly well, it can be considered to adapt constraint propagation also to other heuristic frameworks in order to find promising solutions and discard neighborhoods that do not lead to an improvement.

Another successful application of constraint programming to challenging scheduling problems can be found in [22]. The authors present mixed integer programming models and constraint programming formulations for the online printing shop scheduling problem, which is an extension of the flexible job shop scheduling problem and has already been described in the metaheuristic section for the discussion of [23]. The CP approach, represented by the commercial IBM CP Optimizer, was shown to outperform mixed integer programming, represented by IBM CPLEX, by a large extent. While MIP models reached their limitations already at problem instances with 20 operations, the CP Optimizer was able to tackle instances with around 1000 operations.

4.3. Machine learning

The third category of solution approaches for scheduling problems consists of different learning based methods, such as reinforcement learning and neural networks. Driven by the aim of Industry 4.0 to automate processes and the availability of large amounts of data, machine learning in the production context is a fast growing field. It receives more and more attention in the scheduling literature, however it is also the most recent type of solution approach in this area and more to come is expected.

In the context of machine learning, oftentimes dynamic scheduling environments are considered, where frequently new jobs appear or machines fail and scheduling is done dynamically. Reinforcement learning is the most popular approach to solve problems of this type. In reinforcement learning an agent repeatedly performs actions impacting an environment and receiving respective reward signals in order to learn a scheduling policy. Lei et al. [21] apply deep reinforcement learning, which is a special sort of reinforcement learning where the agent is represented by a deep neural network, to the flexible job shop scheduling problem. Policies for operation and machine actions are learned by the framework and provide production schedules for problem instances with up to 2000 operations. The resulting schedules are shown to be superior to solutions obtained by handcrafted heuristic rules.

Simple dispatching rules, i.e. earliest due date or shortest processing time, can be used in order to prioritize tasks and schedule them accordingly. Zhang et al. [42] use deep reinforcement learning for priority dispatching in a classical job shop environment. They develop a Markov Decision Process model and represent the scheduling policy based on Graph Neural Network (GNN). A GNN is a certain type of deep neural networks, that is especially well suited to leverage the properties of graphs. Their investigated instances are of significant size reaching up to 2000 operations. Also Han and Yang [13] present a deep reinforcement learning framework to solve job shop scheduling problems. They combine deep convolutional neural networks and reinforcement learning and test the developed scheduling framework on benchmark instances with up to 100 jobs to be performed on 20 machines. The framework was shown to outperform traditional dispatching rules, especially for the largest instances. Finally, [33] propose a hybrid method combining convolutional neural networks and iterative local search for the job shop scheduling problem. Instances with up to 50 jobs to be processed on up to 20 machines are tackled.

5. Discussion and conclusion

The era of Industry 4.0 with the comprising stronger interconnection of operating systems, advancing process automation and increasing data availability leads to more complex scheduling problems. Within the conducted structured literature review three categories of advanced solution approaches for large scale scheduling problems with real world extension have been discussed in order to highlight research facing the challenges that result from this trend. Publications on approaches using metaheuristic methods, constraint programming, and machine learning, that are able to tackle problem instances including more than 1000 operations, have been identified.

Metaheuristic methods are able to tackle problems of significant size, even when considering a variety of complex constraints within the investigated problems. Trajectory based heuristics, i.e. tabu search, as well as population based methods, such as genetic algorithms, appear to be promising approaches. Especially combinations of different metaheuristics have shown great potential in terms of intensification and diversification by leveraging the benefits of contrasting approaches. Also attempts to decompose large sized problems into smaller sub problems, that can be solved more efficiently, were demonstrated to work well and encourage further research in this direction.

For extremely large problems, constraint programming is a very interesting and successful approach. Since constraint programming solvers are able to tackle problem instances with hundreds of thousand tasks, closer consideration of this programming paradigm can be recommended. State-of-the-art constraint programming solvers work extraordinary well for scheduling problems and their key ingredients could be incorporated into newly designed solution methods. Machine learning approaches for scheduling currently appear to be mainly used for dynamic problem settings, where reinforcement learning methods show a lot of potential. Especially deep reinforcement learning approaches, using neural networks as learning agents, have been demonstrated to be well suited for large problem settings.

Even though a large part of the literature investigates the fourth category of solution methods for scheduling problems, namely mixed integer programming (MIP), it appears to be not suitable for larger sized problems. During our review there was no paper identified that is approaching large scale scheduling problems by means of mixed integer programming. We conclude that there is a strong limitation in problem size that can be tackled, however want to point out that a lot of effort has been put in extending MIP models by additional aspects of real world applications. Examples can be found e.g. in [34], where a mixed integer linear programming (MILP) model for the flexible job shop scheduling problem with sequence dependent setup times is presented or Hansmann et al. [14] who develop an exact Branch & Bound method for the FJSSP with blockages. Exemplary also Ham [12] can be mentioned, who presents a modified mixed integer programming formulation for the flexible job shop scheduling problem with parallel batch processing and compatible job families or Kress et al. [18] who investigate flexible job shop scheduling with sequence dependent setup times and incorporate heterogeneous machine operator qualifications by taking account of machine- and operator-dependent processing times. Despite the fact that the considered problem extensions are of interest, the instance size solvable with mixed integer programming approaches is significantly smaller than desired by industrial use cases. The complexity of the mathematical models increases very fast and therefore an application to industrial use cases does not appear to be very promising. However, we acknowledge their value for benchmarking approximate methods against optimal solutions on small problem instances. Furthermore, it could be interesting to combine mixed integer programming with efficient metaheuristics in order to create advanced matheuristics. MIP models can, e.g., be leveraged to solve smaller subproblems to optimality, guiding a metaheuristic component through the search space in an efficient manner.

Summarizing these findings, we conclude that future research in the development of solution methods for practical large scale production scheduling applications should leverage tabu search ideas in combination with genetic algorithms and integrate elements from the constraint programming domain, which are often again powerful metaheuristics, such as self adapting large neighborhood search. These advanced methods could be further enhanced by means of decomposition and integration of exact approaches to solve subproblems providing a strengthened guidance for the metaheuristic approaches.

Acknowledgements

The research was supported by the Austrian Research Promotion Agency (FFG) (grant # FO999890318). This support is gratefully acknowledged.

References

[1] Ali, K., Telmoudi, A., Gattoufi, S., 2020. Improved Genetic Algorithm Approach Based on New Virtual Crossover Operators for Dynamic Job Shop Scheduling. IEEE Access 8, 213318–213329. doi:10.1109/ACCESS.2020.3040345.

[2] Belkhamsa, M., Jarboui, B., Masmoudi, M., 2018. Two metaheuristics for solving no-wait operating room surgery scheduling problem under various resource constraints. Computers & Industrial Engineering 126, 494–506. doi:10.1016/j.cie.2018.10.017.

[3] Berndorfer, J., Parragh, S.N., 2022. Modeling and solving a real world machine scheduling problem with due windows and processing set restrictions. Procedia Computer Science 200, 1646–1653. doi:10.1016/j.procs.2022.01.365.

[4] Bögl, M., Gattinger, A., Knospe, I., Schlenkrich, M., Stainko, R., 2021. Real-life scheduling with rich constraints and dynamic properties – an extendable approach. Procedia Computer Science 180, 534–544. doi:https://doi.org/10.1016/j.procs.2021.01.272.

[5] Chaudhry, I.A., Khan, A.A., 2016. A research survey: review of flexible job shop scheduling techniques. International Transactions in Operational Research 23, 551–591. doi:10.1111/itor.12199.

[6] Da Col, G., Teppan, E., 2022. Industrial-size job shop scheduling with constraint programming. Operations Research Perspectives 9. doi:10.1016/j.orp.2022.100249.

[7] Dai, M., Tang, D., Giret, A., Salido, M.A., 2019. Multi-objective optimization for energy-efficient flexible job shop scheduling problem with transportation constraints. Robotics and Computer-Integrated Manufacturing 59, 143–157. doi:10.1016/j.rcim.2019.04.006.

[8] Defersha, F., Rooyani, D., 2020. An efficient two-stage genetic algorithm for a flexible job-shop scheduling problem with sequence dependent attached/detached setup, machine release date and lag-time. Computers and Industrial Engineering 147. doi:10.1016/j.cie.2020.106605.

[9] El-Kholany, M., Gebser, M., Schekotihin, K., 2022. Problem Decomposition and Multi-shot ASP Solving for Job-shop Scheduling. Theory and Practice of Logic Programming 22, 623–639. doi:10.1017/S1471068422000217.

[10] Fuchigami, H.Y., Rangel, S., 2018. A survey of case studies in production scheduling: Analysis and perspectives. Journal of Computational Science 25, 425–436. doi:10.1016/j.jocs.2017.06.004.

[11] Hajibabaei, M., Behnamian, J., 2021. Flexible job-shop scheduling problem with unrelated parallel machines and resources-dependent processing times: a tabu search algorithm. International Journal of Management Science and Engineering Management 16, 242–253. doi:10.1080/17509653.2021.1941368.

[12] Ham, A., 2017. Flexible job shop scheduling problem for parallel batch processing machine with compatible job families. Applied Mathematical Modelling 45, 551–562. doi:10.1016/j.apm.2016.12.034.

[13] Han, B.A., Yang, J.J., 2020. Research on adaptive job shop scheduling problems based on dueling double DQN. IEEE Access 8, 186474–186495. doi:10.1109/ACCESS.2020.3029868.

[14] Hansmann, R.S., Rieger, T., Zimmermann, U.T., 2014. Flexible job shop scheduling with blockages. Mathematical Methods of Operations Research 79, 135–161. doi:10.1007/s00186-013-0456-3.

[15] Hartmann, S., Briskorn, D., 2010. A survey of variants and extensions of the resource-constrained project scheduling problem. European Journal of Operational Research 207, 1–14. doi:https://doi.org/10.1016/j.ejor.2009.11.005.

[16] Huang, R.H., Yang, C.L., Cheng, W.C., 2013. Flexible job shop scheduling with due window—a two-pheromone ant colony approach. International Journal of Production Economics 141, 685–697. doi:10.1016/j.ijpe.2012.10.011.

[17] Komaki, G.M., Sheikh, S., Malakooti, B., 2019. Flow shop scheduling problems with assembly operations: a review and new trends. International Journal of Production Research 57, 2926–2955. doi:10.1080/00207543.2018.1550269.

[18] Kress, D., Müller, D., Nossack, J., 2019. A worker constrained flexible job shop scheduling problem with sequence-dependent setup times. OR Spectrum 41, 179–217. doi:10.1007/s00291-018-0537-z.

[19] Laborie, P., Godard, D., 2007. Self-Adapting Large Neighborhood Search: Application to single-mode scheduling problems. Proceedings MISTA-07, Paris , 8.

[20] Laborie, P., Rogerie, J., Shaw, P., Vilím, P., 2018. IBM ILOG CP optimizer for scheduling: 20+ years of scheduling with constraints at IBM/ILOG. Constraints 23, 210–250. doi:10.1007/s10601-018-9281-x.

[21] Lei, K., Guo, P., Zhao, W., Wang, Y., Qian, L., Meng, X., Tang, L., 2022. A multi-action deep reinforcement learning framework for flexible Job-shop scheduling problem. Expert Systems with Applications 205. doi:10.1016/j.eswa.2022.117796.

[22] Lunardi, W., Birgin, E., Laborie, P., Ronconi, D., Voos, H., 2020. Mixed Integer linear programming and constraint programming models for the online printing shop scheduling problem. Computers and Operations Research 123. doi:10.1016/j.cor.2020.105020.

[23] Lunardi, W., Birgin, E., Ronconi, D., Voos, H., 2021. Metaheuristics for the online printing shop scheduling problem. European Journal of Operational Research 293, 419–441. doi:10.1016/j.ejor.2020.12.021.

[24] Mahmoodjanloo, M., Tavakkoli-Moghaddam, R., Baboli, A., Bozorgi-Amiri, A., 2020. Flexible job shop scheduling problem with reconfigurable machine tools: An improved differential evolution algorithm. Applied Soft Computing 94, 106416.

[25] Maravelias, C.T., 2012. General framework and modeling approach classification for chemical production scheduling. AIChE Journal 58, 1812–1828. doi:https://doi.org/10.1002/aic.13801.

[26] Matta, M., 2009. A genetic algorithm for the proportionate multiprocessor open shop. Computers and Operations Research 36, 2601–2618.

[27] Meng, T., Pan, Q.K., Sang, H.Y., 2018. A hybrid artificial bee colony algorithm for a flexible job shop scheduling problem with overlapping in operations. International Journal of Production Research 56, 5278–5292. doi:10.1080/00207543.2018.1467575.

[28] Mogali, J., Barbulescu, L., Smith, S., 2021. Efficient primal heuristic updates for the blocking job shop problem. European Journal of Operational Research 295, 82–101. doi:10.1016/j.ejor.2021.02.051.

[29] Parente, M., Figueira, G., Amorim, P., Marques, A., 2020. Production scheduling in the context of industry 4.0: review and trends. International Journal of Production Research 58, 5401–5431. doi:10.1080/00207543.2020.1718794.

[30] Pellerin, R., Perrier, N., Berthaut, F., 2020. A survey of hybrid metaheuristics for the resource-constrained project scheduling problem. European Journal of Operational Research 280, 395–416. doi:https://doi.org/10.1016/j.ejor.2019.01.063.

[31] Rossit, D.A., Tohmé, F., Frutos, M., 2018. The non-permutation flow-shop scheduling problem: A literature review. Omega 77, 143–153. doi:https://doi.org/10.1016/j.omega.2017.05.010.

[32] Schwenke, C., Blankenstein, H., Kabitzsch, K., 2018. Large-Scale Scheduling with Routing, Batching and Release Dates for Wafer Fabs using Tabu Search, pp. 472–479. doi:10.1109/ETFA.2018.8502606.

[33] Shao, X., Kim, C., 2022. An Adaptive Job Shop Scheduler Using Multi-Level Convolutional Neural Network and Iterative Local Search. IEEE Access , 1–1doi:10.1109/ACCESS.2022.3188765.

[34] Shen, L., Dauzère-Pérès, S., Neufeld, J.S., 2018. Solving the flexible job shop scheduling problem with sequence-dependent setup times. European Journal of Operational Research 265, 503–516. doi:10.1016/j.ejor.2017.08.021.

[35] Singer, M., 2001. Decomposition methods for large job shops. Computers and Operations Research 28, 193–207.

[36] Smutnicki, C., Pempera, J., Bocewicz, G., Banaszak, Z., 2022. Cyclic flow-shop scheduling with no-wait constraints and missing operations. European Journal of Operational Research 302, 39–49. doi:10.1016/j.ejor.2021.12.049.

[37] Sobeyko, O., Mönch, L., 2016. Heuristic approaches for scheduling jobs in large-scale flexible job shops. Computers and Operations Research 68, 97–109. doi:10.1016/j.cor.2015.11.004.

[38] Teppan, E., 2019. Dispatching Rules Revisited-A Large Scale Job Shop Scheduling Experiment, pp. 561–568. doi:10.1109/SSCI.2018.8628827.

[39] Vilím, P., Laborie, P., Shaw, P., 2015. Failure-Directed Search for Constraint-Based Scheduling, in: Michel, L. (Ed.), Integration of AI and OR Techniques in Constraint Programming. Springer International Publishing, Cham. volume 9075, pp. 437–453. doi:10.1007/978-3-319-18008-3_30. series Title: Lecture Notes in Computer Science.

[40] Xie, J., Gao, L., Peng, K., Li, X., Li, H., 2019. Review on flexible job shop scheduling. IET Collaborative Intelligent Manufacturing 1, 67–77. doi:10.1049/iet-cim.2018.0009.

[41] Yang, D., Wu, M., Li, D., Xu, Y., Zhou, X., Yang, Z., 2022. Dynamic opposite learning enhanced dragonfly algorithm for solving large-scale flexible job shop scheduling problem. Knowledge-Based Systems 238. doi:10.1016/j.knosys.2021.107815.

[42] Zhang, C., Song, W., Cao, Z., Zhang, J., Tan, P.S., Xu, C., 2020a. Learning to dispatch for job shop scheduling via deep reinforcement learning, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, Curran Associates Inc., Red Hook, NY, USA.

[43] Zhang, J., Wang, L., Xing, L., 2019. Large-scale medical examination scheduling technology based on intelligent optimization. Journal of Combinatorial Optimization 37, 385–404. doi:10.1007/s10878-017-0246-6.

[44] Zhang, S., Li, X., Zhang, B., Wang, S., 2020b. Multi-objective optimisation in flexible assembly job shop scheduling using a distributed ant colony system. European Journal of Operational Research 283, 441–460. doi:10.1016/j.ejor.2019.11.016.

[45] Zhang, S., Wang, S., 2018. Flexible Assembly Job-Shop Scheduling With Sequence-Dependent Setup Times and Part Sharing in a Dynamic Environment: Constraint Programming Model, Mixed-Integer Programming Model, and Dispatching Rules. IEEE Transactions on Engineering Management 65, 487–504. doi:10.1109/TEM.2017.2785774.