European Journal of Operational Research 295 (2021) 399–426

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Invited Review

Four decades of research on the open-shop scheduling problem to minimize the makespan

Mohammad Mahdi Ahmadian[a], Mostafa Khatami[a], Amir Salehipour[a,*], T. C. E. Cheng[b]

[a] School of Mathematical and Physical Sciences, University of Technology Sydney, Australia
[b] PolyU Business School, The Hong Kong Polytechnic University, Hong Kong

ARTICLE INFO

Article history:
Received 28 May 2020
Accepted 12 March 2021
Available online 24 March 2021

Keywords:
Scheduling
Open-shop
Review
Makespan

ABSTRACT

One of the basic scheduling problems, the open-shop scheduling problem has a broad range of applications across different sectors. The problem concerns scheduling a set of jobs, each of which has a set of operations, on a set of different machines. Each machine can process at most one operation at a time and the job processing order on the machines is immaterial, i.e., it has no implication for the scheduling outcome. The aim is to determine a schedule, i.e., the completion times of the operations processed on the machines, such that a performance criterion is optimized. While research on the problem dates back to the 1970s, there have been reviving interests in the computational complexity of variants of the problem and solution methodologies in the past few years. Aiming to provide a complete road map for future research on the open-shop scheduling problem, we present an up-to-date and comprehensive review of studies on the problem that focuses on minimizing the makespan, and discuss potential research opportunities.

1. Introduction

The open-shop scheduling is one of the basic scheduling problems, in which a set of jobs are processed on a set of different machines such that a performance criterion is optimized. The open-shop scheduling problem has a broad range of applications across different sectors. An example of the problem in the health care sector is as follows: Scheduling patients for diagnosing the coronary heart disease. A patient needs to undergo the diagnosis in three stages, namely blood testing, ultrasonic cardiogramming, and coronary computed tomography scanning, in any order. Each stage requires multiple facilities and/or medical personnel to conduct the diagnosis. In general, the open-shop problem concerns scheduling a set of jobs, each with a set of operations, on a set of different machines, where each machine can process at most one operation at a time and no order for processing the jobs on the machines is given. The scheduler wishes to construct a feasible schedule (containing the completion times of all the operations on the machines) such that a performance criterion is optimized. A schedule is feasible if the operations of each job do not overlap and each machine executes at most one operation at any point in time.

While research on the open-shop scheduling problem dates back to the 1970s, there have been reviving interests in the computational complexity of variants of the problem and solution methodologies in the past few years. We set out to provide a complete road map for future research on the open-shop scheduling problem, presenting an up-to-date and comprehensive review of studies on the problem that focuses on minimizing the makespan, i.e., the maximum completion time of all the jobs, and discussing potential research opportunities.

As the structure of the present review paper, in the remainder of Section 1 we detail the scope and coverage of the paper, and position the paper within the shop scheduling domain (Section 1.1), followed by an analysis of the publications distribution in different time periods and the publication outlets (Section 1.2). Then, in Section 1.3, we present the open-shop scheduling problem and the major notations that we use in the paper, and in Section 1.4, we discuss a number of applications of the problem.

We devote Sections 2 to 4 to discussing and reviewing the available studies on the problem. We propose three sections of classical open-shop (Section 2), the non-classical open-shop focusing on resource (machine) settings (Section 3), and the non-classical

* Corresponding author.
E-mail addresses: mohammadmahdi.ahmadian@student.uts.edu.au (M.M. Ahmadian), mostafa.khatami@student.uts.edu.au (M. Khatami), pamir.salehipour@uts.edu.au (A. Salehipour), edwin.cheng@polyu.edu.hk (T.C.E. Cheng).

https://doi.org/10.1016/j.ejor.2021.03.026

open-shop focusing on the job settings (Section 4). In the classical open-shop, we aim to cover the papers on the standard open-shop scheduling problem without specific settings for the machines or jobs. We also review the available exact and heuristic methods that generate a schedule for the open-shop scheduling problem in the same section. In Section 3, we focus on studies that investigate the open-shop scheduling problem under various settings for processing resources, e.g., availability and renewablility of the resources. In Section 4, we focus on studies that consider the open-shop scheduling problem with different constraints on the jobs, including the precedence, release and processing times. The section also reviews the studies on job batching and rejection.

We highlight the main open problems and potential areas for future research on the open-shop scheduling problem in Section 5, and we conclude the present review paper in Section 6.

1.1. Scope and classification

We present a detailed literature review of the classical open-shop scheduling problem. Unless otherwise stated, we assume that all the parameters, e.g., the processing times of job operations on the machines, the deadlines, the release times etc, are known and deterministic.

We confine the scope of the paper to the performance criterion (or objective function) of minimizing the makespan. Therefore, we do not review the other performance criteria. This is because of several reasons. First, makespan minimization is the most studied performance criterion in the open-shop problem that has many real-world applications, some of which are discussed in Section 1.4. Second, as we discussed in the following, the existing review papers on the open-shop problem with makespan minimization are neither up-to-date nor comprehensive. Third, given that a large number of papers have been published on performance criteria other than the makespan, including all these studies will result in a lengthy review that will exceed the journal's page limit. However, a few variants of the problem concern performance criteria that are related to the makespan even though they consider slightly different objective functions. To provide a through review, we discuss those studies as well. We only focus on the open-shop environment, so do not review studies on generalization of the open-shop scheduling environment, e.g., the multi-processor open-shop (Adak, Akan, & Bulkan, 2020), where parallel identical machines process the jobs in every stage. We do not review studies on hybridization of the open-shop scheduling problem with other shop scheduling environments (e.g., job-shop or flow-shop). We do not report results on mixed-shop scheduling (Shakhlevich, Sotskov, & Werner, 2000), in which some jobs have fixed machine orders (as in the job-shop), whereas the operations of the other jobs may be processed in an arbitrary order (as in the open-shop). Neither do we study the super-shop setting (Strusevich, 1991), in which the jobs may have same (as in the flow-shop), fixed, or arbitrary processing routes.

On the open-shop scheduling problem to minimize the makespan, we present a comprehensive review of the pertinent studies that include various constraints and jobs' execution settings. While we have made a meticulous effort to include all the related studies that we are aware of, we might have missed some papers. In particular, we are only aware of a few review papers on the open-shop scheduling problem. One extensive review paper on deterministic machine scheduling problems that also includes the open-shop problem is due to Chen, Vestjens, and Woeginger (1998). The authors reviewed more than 550 papers across topics of single and parallel machines, and flow-shop, job-shop and open-shop settings. The paper reviews the complexity results, and exact and heuristic algorithms for a number of performance criteria such as maximum weighted earliness and tardiness, total weighted completion times and weighted number of late jobs. Since the publication of that paper in 1998, almost 200 papers on the open-shop problem with the makespan minimization have been published. The review papers by Gonzalez (2004), Prins (2010) and Anand, Panneerselvam et al. (2015) on the open-shop scheduling are limited in many aspects. The brief survey by Gonzalez (2004) merely covers 46 papers and spans as late as 2002. While the criteria of completion (mean flow) and total completion times, in addition to the makespan, were considered, the survey only discusses the well-known results for some restricted settings, e.g., the two-machine and non-preemptive schedules. The paper by Prins (2010) reviews about 45 papers on the complexity and algorithmic developments for the open-shop with the makespan minimization. An interesting aspect of that review includes performance comparison of a number of heuristic and meta-heuristic methods. Nonetheless, that review did not go beyond the year 2007 and since then almost 100 papers on the open-shop with the makespan minimization were published. The most recent review paper is due to Anand et al. (2015). That review study is limited to the papers published between 1981 and 2013. Despite reviewing various objective functions that intuitively leads to the expectation that a large number of papers were reviewed, Anand et al. (2015) covered only 100 papers, implying that many studies between 1981 and 2013 were missing from their review. Notably, we review 263 papers on makespan minimization. Also, we include all 55 relevant studies conducted between 2014 and 2020, which account for more than half of the papers on various performance criteria reviewed in Anand et al. (2015). In addition, we provide an analysis of the publication rate of papers (on open-shop scheduling to minimize the makespan) over the years and the publication outlets.

We organize the rest of the paper as follows: In the next subsection we present an analysis of the reviewed papers. In Section 1.3 we formally define the problem and introduce the main notation that we use throughout the paper, followed by discussion of a number of applications of the open-shop scheduling problem in Section 1.4. In Section 2 we review studies on the open-shop scheduling problem in its basic setting. We review studies on the problem in specific settings of the machines or resources, and jobs in Sections 3 and 4, respectively. We discuss the potential research opportunities in Section 5, followed by concluding the paper in Section 6.

1.2. Analysis of the reviewed papers

In this survey, we review all the available studies on the open-shop scheduling problem, including the early results published in 1970s. We collect and review 263 papers in total. Eight papers were published before 1980, followed by a total of 23 papers that were published in the 1980s. The next decade witnessed a large increase in the interest in the open-shop scheduling, with a total of 82 papers published in the 1990s. We also observe that 65 papers appeared in the 2000s and 85 papers in the 2010s and afterwards. The trend shows that even though open-shop is a classical scheduling problem and that significant results were presented in the 1980s and 1990s, the problem's properties and the solutions are yet to be fully discovered, so there is still much interest in the problem. Figure 1 shows the number of papers published over time, from 1970 up to the present time. Table 1 reports the outlets that have published more than two papers on open-shop scheduling, where the entries are sorted in non-increasing order of number of published papers in major journals, followed by book chapters, technical reports, theses and conference proceedings. The table shows that *European Journal of Operational Research*, *Journal of Scheduling*, *Annals of Operations Research*, *Discrete Applied Mathematics*, *Computers & Operations Research*, and *Naval Research Logis-*

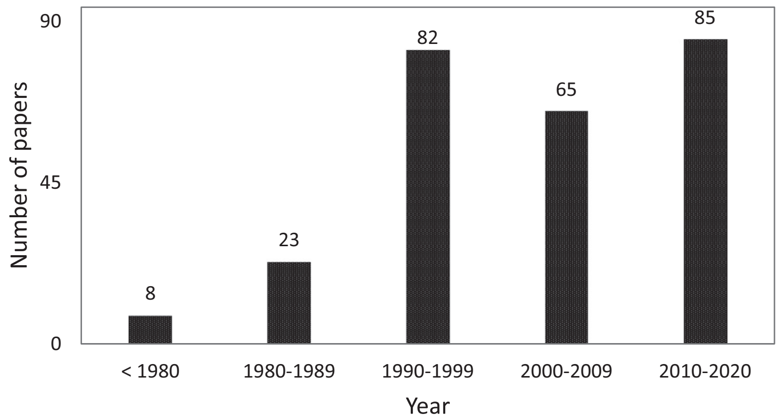

Fig. 1. Distribution of the number of papers published in the area of open-shop scheduling in different time periods.

Table 1
Distribution of papers by the major publication outlets that have published more than two papers.

Outlet	Number of papers (%)
European Journal of Operational Research	26 (9.89)
Journal of Scheduling	16 (6.08)
Annals of Operations Research	14 (5.32)
Discrete Applied Mathematics	14 (5.32)
Computers & Operations Research	11 (4.18)
Naval Research Logistics	10 (3.80)
Operations Research	8 (3.04)
Operations Research Letters	8 (3.04)
Mathematics of Operations Research	6 (2.28)
The Journal of the Operational Research Society	5 (1.90)
IIE Transactions	4 (1.52)
Theoretical Computer Science	4 (1.52)
INFOR: Information Systems and Operational Research	3 (1.14)
The International Journal of Advanced Manufacturing Technology	3 (1.14)
Mathematical Methods of Operations Research	3 (1.14)
Zeitschrift für Operations Research	3 (1.14)
Book chapters, reports, Ph.D. theses, and 65 other journals	95 (36.12)
Proceedings	30 (11.41)
Total	263 (100)

Table 2
Mathematical notation used throughout the paper.

Notation	Description
Set:	
N	Jobs, $N = \{1, \dots, n\}$.
M	Machines, $M = \{1, \dots, m\}$.
Parameter:	
n	Number of jobs.
m	Number of machines.
p_{ij}	Processing time of job j on machine i, $p_{ij} \in \mathbb{Z}^+$.
$p_{\max}$	The longest processing time among all the operations.
$p_{\min}$	The smallest processing time among all the operations.
$p(p_i)$	A common processing time (a common processing time on machine i).
d_j	Deadline of job j.
r_j	Release time of job j.
e_j	Rejection cost of job j.
s_{ij}	Setup time of job j on machine i.
s_f	Setup time of batch f.
$M_{\max}$	Maximal machine.
l_i	Load of machine i ($l_i = \sum_j p_{ij}$).
$l_{\max}$	Maximum machine load ($l_{\max} = \max\{l_i\}$).
$L(\bar{L})$	Exact (minimum) amount of delay.
α_{ij}	Basic processing time of job j on machine i (in the time-dependent setting).
β_{ij}	Deterioration rate of job j on machine i (in the time-dependent setting).

tics have published almost 35% of the papers, suggesting that they are the ideal outlets for future research on the problem.

Next, we formally define the open-shop scheduling problem and introduce the major notation that we use throughout the paper.

1.3. Problem statement and notation

We define the "classical" open-shop scheduling problem as follows: There is a set $N = \{1, \dots, n\}$ of jobs, each of which has a set of operations, to be processed on a set of different machines $M = \{1, \dots, m\}$. The operation of job j on machine i is denoted by O_{ij} and its duration, i.e., its processing time, is p_{ij}. If job j is given a release time and a deadline, we let r_j and d_j denote them, respectively. Deadline d_j for operation j means that the operation must be completed by its given deadline. The processing order of each job on the machines is immaterial, i.e., it has no implication for the scheduling outcome. Each machine can process at most one operation at a time. The aim is to determine a feasible schedule, i.e., the completion times of the job operations on the machines, such that a performance criterion is optimized. A schedule is feasible if the operations of each job do not overlap and each machine processes at most one operation at any point in time. In this study we focus on minimizing the makespan, as well as on the performance criteria related to the makespan. It is clear that in the open-shop setting, one can swap the jobs and machines since they play equivalent roles. It should be noted that there might be additional parameters that are defined in specific settings. We define all the major notation in Table 2.

We use the standard three-field notation $\alpha|\beta|\gamma$ introduced by Graham, Lawler, Lenstra, and Rinnooy Kan (1979) for describing scheduling problems throughout the paper, where α, β, and γ represent the scheduling environment, the job characteristics, and the performance criterion, respectively. With respect to the field α, the open-shop scheduling problem is denoted by O. For two, three or a fixed number of machines m, we use $O2$, $O3$, and Om, respectively. However, if the number of machines is a part of the input, i.e., given, we use O. Regarding the field β, various characteristics have been considered in the existing literature that we will review throughout the paper. For example, if preemption of the operations is allowed, i.e., the execution of the operations can be interrupted and will be resumed or re-started later, it is shown by *prmp*. Considering the field γ, we confine to stud-

European Journal of Operational Research 295 (2021) 399–426

Table 3
Details of the discussed applications of the open-shop scheduling problem.

Application	Job	Machine	Operation	Reference
Timetabling:				
Job fair meeting scheduling	Firms	Students	Meetings	Bartholdi and McCroan (1990)
National hockey league scheduling	Home games	Road games	Games	Costa (1995)
University educational programme scheduling	Teachers	Classes	Lectures	de Werra et al. (2002, 2000)
Trade fair meeting scheduling	Buyers	Sellers	Meetings	Ernst et al. (2003)
School meeting scheduling	Parents	Teachers	Meetings	Rinaldi and Serafini (2006)
Workplace training scheduling	Apprentices	Practice placements	Trainings	Czibula et al. (2016)
Airplane garage task scheduling	Technicians	Airplanes	Checking or maintenance operations	Grinshpoun et al. (2014, 2017)
Satellite communication:				
Time slot assignment	A group of packets	Satellite repeaters	Sending packets	Inukai (1979); Prins (1994)
Health care management:				
Endoscopy scheduling	Patients	Endoscopy operating rooms	Gastroscopy and colonoscopy tests	Fei et al. (2009)
Laboratory scheduling	Patients	Place or staff in the laboratory	Medical tests	Azadeh et al. (2014)
Coronary heart disease diagnosis	Patients	Medical equipments	Medical checks	Bai et al. (2016)
Medical clinic scheduling	Patients	Diagnostic stations	Medical tests	Baron et al. (2016)
Rehabilitation scheduling	Patients	Therapists	Therapeutic processes	Zhao et al. (2018)
Transport:				
Crane scheduling	Holds	Cranes	Unloading holds	Daganzo (1989)
Truck scheduling	Trucks	Dock doors	Delivering parts	Cankaya et al. (2019)
Port traffic scheduling	Chemical tankers	Port terminals	Loading and unloading cargoes	Cankaya et al. (2019)
Robot task scheduling	Transport robots	Pick robots	Order picking and transporting	Wang et al. (2020)
Tourism:				
Museum visitor routing	Visitor groups	Exhibit rooms	Visits	Chou and Lin (2007); Vincent et al. (2010)
Event-bus scheduling	Buses	Locations	Visits	Brandinu and Trautmann (2014)

ies that minimize the makespan, i.e., the maximum completion time of the jobs. Let C_{ij} denote the completion time of job j on machine i. Then, the makespan is defined as $C_{\max} = \max_{j\in N}\{C_j\}$, where $C_j = \max_{i\in M}\{C_{ij}\}$.

1.4. Applications

One of the basic scheduling problems, open-shop has a broad range of applications across different sectors. In this section we discuss several applications of this scheduling model. For each application, we provide the descriptions of the jobs, operations, and machines in Table 3.

Consider a large automobile workshop with specialized shops and a set of automobiles that require different types of repairs, ranging from general services such as changing oil, rotating tires, and checking electrical parts to specialized services, e.g., painting. No two operations on an automobile can be executed simultaneously due to the different locations of the shops and the characteristics of the operations. It is also clear that the order of performing the operations by the mechanics is immaterial (Adiri & Amit, 1983; Kubzin, Strusevich, Breit, & Schmidt, 2006).

In satellite communication, two or more earth stations may communicate by exchanging data packets. The transmitting earth station sends a packet to the satellite in a specified frequency, i.e., the uplink frequency. The packet is then converted into a different frequency, i.e., the downlink frequency, by the satellite repeater and is sent to the receiving station. The incoming packets from the earth stations are transmitted cyclically and each cycle, called a "frame", takes typically two milliseconds. The order of sending the packets is immaterial. An on-board $n \times n$ switch is used to connect n transmitting stations and n receiving ones. The time needed to transmit data from a transmitting station i to a receiving station j is denoted by p_{ij}, while transmitting (receiving) more than one packet at a time is prohibited. In addition, two stations cannot send a packet at the same time to the same repeater. The problem is then to schedule the transmission of all the data within a short cycle. A schedule is constructed by assigning packets to the repeaters and setting the transmission times. Therefore, p_{ij} is sent through different modes of an on-board switch and each time a fraction of it will be sent over. It is not hard to see that this problem is equivalent to the open-shop scheduling problem (Prins, 1994).

The timetabling problem can also be modelled and solved as the open-shop scheduling problem. de Werra, Asratian, and Durand (2002); de Werra, Hertz, Kobler, and Mahadev (2000) discussed the timetabling issue arising in universities and educational centres, where there are group lectures (given by one teacher to a group of classes) and individual lectures (given by one teacher to one class). Group lecturing takes place in the basic programmes of the universities and schools. In this setting, a class is defined as a group of students following the same programme, and the set of classes is partitioned into a collection of groups such that each class belongs to exactly one group. However, there is a possibility that several groups contain exactly one class. In this context, machines represent the teachers and jobs denote the classes. The jobs are allowed to be preempted and must be processed within k time units. In the context of sport scheduling, Costa (1995) discussed the resemblance between the problem of national hockey league scheduling and the preemptive open-shop scheduling problem. Teams in the league can be split into two sets of M and N. Teams in M only play on the road, whereas teams in N always stay at home. Therefore, M and N may represent machines and jobs, respectively. A game is defined when a team from M matches up with a team from N. Other timetabling applications that can be modelled as the open-shop scheduling problem include scheduling a job fair (Bartholdi & McCroan, 1990), a trade fair (Ernst, Mills, & Welgama, 2003), school meetings (Rinaldi & Serafini, 2006) and workplace training (Czibula, Gu, & Zinder, 2016).

The airplane garage task scheduling is another application of the open-shop problem. Consider a fleet of airplanes. A set of operations need to be performed on each plane in order to prepare it for take-off. Each operation is performed by a technician (a machine) that has the expertise only for that operation. While some operations can be performed simultaneously on a plane, e.g., one technician checks the engine while the other technician inspects the wings, there are operations that cannot be performed concurrently, leading therefore to the classical open-shop setting (Grinshpoun, Ilani, & Shufan, 2014; 2017). We note that it is called the concurrent open-shop if all the operations of a job can be performed in parallel and it is called the partially concurrent open-shop if only some of the operations of a job can be performed in parallel.

Askin, Dror, and Vakharia (1994) modelled the problem of assembling k types of printed circuit boards (PCB) as the open-shop scheduling problem. First, the electrical components to be placed on PCBs by machines are assigned to the machines. Then, the PCBs with the same requirements are grouped into families. The objectives include minimizing the makespan of the assembly time and minimizing the flow time. Daganzo (1989) studied the problem of crane scheduling to unload holds for a set of ships at berth. Considering each hold as a single-operation job and each crane as a machine, and assuming that two or more cranes are not allowed to work on a hold simultaneously, the problem can be modelled as classical open-shop scheduling. Another interesting application of the open-shop in this area includes the problem of scheduling chemical tankers' arrivals in ports. An arriving tanker must visit multiple terminals in a port in any order. The aim is to determine the order in which a tanker visits a terminal and the order in which a given terminal services the tankers (Cankaya, Wari, & Tokgoz, 2019). This is an important application. For example, Cankaya et al. (2019) pointed out that almost half of the traffic in the Houston ship channel is due to chemical tanker movements that carry liquid cargoes between multiple terminals.

The open-shop scheduling problem can be used to model a range of problems arising in the health care sector. Consider scheduling patients for diagnosing the coronary heart disease. A patient needs to undergo the diagnosis in three stages, namely blood testing, ultrasonic cardiogramming, and coronary computed tomography (CT) scanning, in any order. Each stage requires multiple facilities and/or medical personnel to conduct the diagnosis (Bai, Zhang, & Zhang, 2016). Other problems in health care sector that have been modelled as the open-shop scheduling problem include the endoscopy scheduling problem (Fei, Meskens, Combes, & Chu, 2009), patient scheduling in emergency department laboratories (Azadeh, Farahani, Torabzadeh, & Baghersad, 2014), patient scheduling in medical clinics (Baron, Berman, Krass, & Wang, 2016), and the rehabilitation scheduling problem (Zhao, Chien, & Gen, 2018).

The open-shop scheduling problem is also applicable to the transport sector. For example, Chou and Lin (2007) and Vincent, Lin, and Chou (2010) introduced the museum visitor routing problem. By treating the jobs and machines as visitor groups and exhibition rooms, respectively, the problem aims to ensure that each visitor group visits each exhibition room exactly once, and at any order, such that the time by which the last visitor group leaves the museum, i.e., the makespan, is minimized. Brandinu and Trautmann (2014) discussed the event-bus scheduling problem that can be modelled as the open-shop problem. Given a number of buses carrying tourists that visit the sites of famous Bollywood movies, the problem's characteristics include parking at most one bus at any location at a time and not having a fixed route for the buses. Wirth and Emde (2018) showed that scheduling trucks on factory premises may be modelled as the open-shop scheduling problem, where a fleet of trucks must visit and deliver parts to a number of premises in a factory in any order. In addition, no two trucks can be handled simultaneously at any destination. In a recent attempt, Ahmadian, Salehipour, and Kovalyov (2020) introduced the following application: There are n trucks each delivering a consolidated cargo to m destinations in any order. No two trucks can be handled simultaneously at any destination. The travel time is negligible compared with the service time at any destination. The heterogeneous robot task scheduling, which arises in order fulfillment systems on logistic networks, is another application of the open-shop problem. Consider two types of robots, namely, the transport robots that are responsible for transporting items, and the pick robots that pick items from storage locations and place them into the transport robots. Here, the transport robots represent jobs in the open-shop setting and the pick robots present machines (Wang, Chen, & Wang, 2020).

Next, we introduce the classical open-shop scheduling problem under, e.g., various numbers of machines and preemption. We also review the major solution methods developed to deal with the classical open-shop scheduling problem in Section 2.

2. The classical open-shop scheduling problem

In this section we focus on the classical open-shop scheduling problem. We review variants of the problem in Sections 3 and 4. In the classical variant, we review the basic and fundamental characteristics of the open-shop scheduling problem, such as the number of machines and the preemption/non-preemption conditions. We also review the general solution methods, ranging from exact to heuristic approaches, that have been developed to generate a schedule for the open-shop scheduling problem.

2.1. The two-machine open-shop

The number of machines vastly impacts the computational complexity of the open-shop scheduling problem. For example, the well-known two-machine problem to minimize the makespan is polynomially solvable (see, e.g., Gonzalez & Sahni, 1976) whereby the same problem under an arbitrary number of machines is strongly *NP*-hard (Williamson et al., 1997). Therefore, in this section we review studies that consider the two-machine setting, and in the next section we review studies that consider an arbitrary number of machines.

Interestingly, Dell'Amico and Martello (1996), and Martello (2010) argued that the algorithm of Gonzalez and Sahni (1976) for the two-machine open-shop to minimize the makespan, which runs in $O(n)$, is indeed an implementation of the results of Egerváry (1931) published in the early 1930s. Cheng and Shakhlevich (2007) stated that the algorithm constructs an optimal schedule such that each machine has at most one idle time in the interval $[0, C_{\max}]$. In addition to the linear-time algorithm of Gonzalez and Sahni (1976), other $O(n)$ algorithms exist for the two-machine open-shop problem to minimize the makespan. One example is the longest alternative processing time (LAPT) dispatching rule of Pinedo and Schrage (1982). The second algorithm includes the de Werra (1989)'s method that partitions the jobs into three batches. An interesting characteristic of the method lies in the fact that the jobs of a batch may have their processing order changed without impacting the optimal makespan. Soper (2015) generalized the method to the case with more than three batches. The generalization solves the two-machine n-job open-shop problem through solving the two-machine flow-shop problem with $n-1$ jobs. The one "omitted" job is then added to the constructed flow-shop schedule, where it is first processed on the second machine before processing all the other jobs, and then on the first machine after processing all the other jobs.

Shakhlevich and Strusevich (1993) presented an $O(n)$-time algorithm for solving the two-machine open-shop problem with arbitrary regular penalty functions associated with "machine usage", i.e., the penalty is a function of the length of the time during which the machines operate. Van Den Akker, Hoogeveen, and Woeginger (2003) used the algorithm of Shakhlevich and Strusevich (1993) and investigated the existence of a feasible schedule such that all the jobs can be completed before k_1 and k_2, where $[0, k_1]$ and $[0, k_2]$ are given time windows during which machines 1 and 2 are available. It is evident that their study aims to verify whether the given time windows can lead to a feasible schedule. It is also evident that the case of $k_1 = k_2$ leads to makespan minimization.

2.2. The m-machine open-shop

In reviewing studies on the classical m-machine open-shop scheduling problem, we first consider the studies that focus on the complexity analysis of the problem. Then we report the papers that study the problem with the so-called "dense" schedules, followed by works that aim at reducing the solution (feasible search) space.

Complexity

Contrary to the case with two machines, the open-shop scheduling problem with an arbitrary number of machines, i.e., $m \geq 3$, to minimize the makespan is strongly *NP*-hard (see Gonzalez, 1982; Lawler, Lenstra, Kan, & Shmoys, 1993). However, if the number of machines is fixed, i.e., it is not part of the input, and the problem instance includes at least one job with three operations, it is ordinary *NP*-hard (Gonzalez & Sahni, 1976). Another ordinary *NP*-hard case includes four machines with two operations per job, where each job is processed on only two machines (Gonzalez & Sahni, 1976). In general, the complexity of the three-machine problem is still open if each job has exactly two operations. An easy problem is $O|UET|(\sum C_j \cap C_{\max})$, i.e., the open-shop with unit execution time (UET) jobs to minimize the makespan and total completion time that can be solved by the $O(mn)$-time algorithm by Adiri and Amit (1984) where in the optimal schedule of the algorithm both $\sum C_j$ and $C_{\max}$ are minimized simultaneously. Also, problems $O|n = 2|C_{\max}$ and $O|n = k, prmp|C_{\max}$ are polynomially solvable, whereas problem $O|n = 3|C_{\max}$ is *NP*-hard (Brucker, Sotskov, & Werner, 2007).

Williamson et al. (1997) showed that no polynomial-time ρ-approximation algorithm, $\rho < \frac{5}{4}$, exists for problem $O||C_{\max}$ unless $P = NP$. When m is fixed, Sevast'yanov and Woeginger (1998) showed that there exists a $(1 + \varepsilon)$-approximation scheme for the problem that is polynomial in the size of the instance, but exponential in m and $1/\varepsilon$. It has also been conjectured that the optimal solution for $O||C_{\max}$ is at most $\frac{3}{2}$ larger than that of $O|prmp|C_{\max}$ (Schuurman & Woeginger, 1999). It is known, due to Bárány and Fiala (1982), that the makespan of a solution obtained by any list scheduling heuristic is at most two times greater than the optimal makespan (because Bárány and Fiala (1982) is in Hungarian, we refer the interested reader to Shmoys, Stein, and Wein (1994)). The results also hold when each job has a release time r_j at which it becomes available for processing (Wein, 1991).

Certain studies apply the mass polynomial-time reduction from the UET open-shop to the parallel-machine case, and vice versa, in order to determine the complexity of a large number of open-shop scheduling problems. For example, Brucker, Jurisch, and Jurisch (1993) showed that the complexity of solving a number of m-machine open-shop problems with UET operations is $O(k + n^2m)$ or $O(k + nm(\log nm)^2)$ if the corresponding parallel-machine problem can be solved in $O(k)$ time. Their proposed algorithm transforms the problem to an m-identical parallel machine under the condition that all the problem's parameters, including the start times, completion times, preemption times, and re-start times of the jobs only take integer values. While the overall complexity of their algorithm is $O(n^2m)$, it can be improved to $O(nm(\log nm)^2)$ by using edge colouring algorithm of Gabow and Kariv (1982) in bipartite graph. Timkovsky (2003) proposed two mass reductions, namely the "open-shopping" and the "open-shop paralleling", which can be applied for any criterion except the total completion time. The former maps the preemptive identical parallel-machine problem with arbitrary jobs' processing times to the UET open-shop, and the latter reduces the UET open-shop problem to the non-preemptive identical parallel-machine problem with UET jobs.

Using a specific matrix of "latin rectangle" (a latin rectangle $LR[n, m, k]$ is a matrix in the format $[n \times m]$ with the entries taken from the set $\{1, \ldots, k\}$, where every entry occurs at most once in every row and column) to present a schedule, Tautenhahn (1994) proposed a polynomial algorithm for the open-shop problem with UET operations, in which every operation must be completed by its given deadline. They proposed an $O(n^2m)$ algorithm that delivers the optimal schedule for $O|UET, C_j \leq d_j|C_{\max}$ and also for $O|UET, C_j \leq d_j|\sum C_j$, and showed that there is a common optimal schedule for both problems. In addition, they used the same algorithm to solve the problem to minimize the number of tardy jobs and the maximum lateness, and improved the earlier result of Liu and Bulfin (1988).

Impagliazzo and Paturi (2001) introduced the exponential time hypothesis (ETH) as follows: The 3-SAT (a very strict form of the satisfiability problem) with the parameter n, where n is the number of variables, has no sub-exponential algorithm. That is, 3-SAT cannot be solved in $2^{O(n)}$ time. This conjecture is important because one may use it to derive lower bounds on the running times of algorithms for other combinatorial or computational problems. For example, Jansen, Land, and Land (2016) showed that $O3||C_{\max}$ cannot be decided in $2^{O(n)}$ time and $O||C_{\max}$ does not admit any ρ-approximation algorithm for any $\rho \leq \frac{5}{4}$ in $2^{O(n)}$ time unless ETH is invalid. Sevastianov (2005) presented a 4-parameter complexity analysis of the open-shop scheduling problem. In this approach, instead of establishing the complexity result for a problem based on a single parameter, say, the number of machines or jobs, a set of important parameters are considered and the complexity analysis of sub-problems can be studied. A sub-problem is characterized by each combination of the constraints. That study investigated a complete basis system of the sub-problems of the open-shop with the four parameters of number of jobs, maximum number of operations per job, number of machines, and maximum number of operations on a machine. Using a basis system, one can determine the complexity of any other sub-problems. Kononov, Sevastyanov, and Sviridenko (2012) studied the complexity of an infinite class of shop scheduling problems, including the open-shop, with combinations of constraints imposed on the processing times of the operations and the maximum number of operations per job, and an upper bound on the length of the schedule. They presented a finite basis system of sub-problems for these problems that includes ten problems, five of which are polynomially solvable, and the other five are *NP*-complete. They showed that the problem of deciding the existence of a schedule with a length of at most four for the open-shop with processing times one and two, with at most two operations per job and at most three operations per machine, i.e., $O|p_{ij} \in \{1, 2\}, k_1 \leq 2, k_2 \leq 3|C_{\max} \leq 4$, is *NP*-complete, where k_1 and k_2 represent the maximum number of operations per job and per machine, respectively.

Dense schedules

In the open-shop with "dense" schedule, the machine is idle only if there is no job that is ready to be processed on that machine (Bárány & Fiala, 1982). Aksjonov (1988) and Shmoys et al. (1994) showed that the worst-case performance ratio for any dense open-shop schedule is 2, since the makespan of a dense schedule cannot be more than the maximum of jobs' durations plus the maximum of machines' loads.

For the m-machine open-shop problem, Bárány and Fiala (1982) proved that any dense schedule is at most $(m-1)p_{\max}$ larger than the optimal solution, where $m \geq 1$ and $p_{\max}$ is the longest processing time among all the operations, and that the bound is tight. It should be noted that a general conjecture on the worst-case ratio of the dense schedule is $2-(\frac{1}{m})$ due to Chen and Strusevich (1993). The conjecture has not yet been proved for arbitrary m. Nonetheless, Chen and Strusevich (1993) proved the conjecture for $m = 3$ and proposed two approximation algorithms. The first algorithm is a greedy one that has an approximation bound equal to $\frac{5}{3}$ of the optimal schedule. The second algorithm improves the bound to $\frac{3}{2}$, through reducing the idle time on the third machine and by moving the jobs (other than the last two jobs) on the first two machines to the end of the schedule. Sevast'yanov and Tchernykh (1998) provided an $O(n)$ algorithm to improve the bound to $\frac{4}{3}$, where they showed that any instance with n jobs can be transformed to an instance with five "aggregated" jobs. The conjecture was later proved for $m = 4, 5, 6$ by Chen and Yu (2001, 2003, 2000), and for $m = 7, 8$ by Chen, Huang, Men, and Tang (2012). In addition, Chen et al. (2012) presented certain types of dense schedules under which the conjecture is always true.

Solution space reduction

A few studies have attempted to identify subsets of the solution space that contains at least one optimal schedule. Two such subsets include the dense schedules and the so-called "rank-minimal" schedules discussed in Bräsel, Tautenhahn, and Werner (1993). If we present the problem by a rank matrix, the rank-minimal schedules are those that the largest value of the rank for all the operations is minimal and is equal to $\max\{n, m\}$. Bräsel and Kleinau (1992) established a one-on-one mapping between the feasible schedules of $O|UET|C_{\max}$ and a special latin rectangle called "plan" (here, for every entry $a_{ij} > 1$, there exists element a_{ij} in row i or in column j), and determined the numbers of optimal solutions for $O2|UET|C_{\max}$ and $O3|UET|C_{\max}$. For small n and m, they proposed an enumeration algorithm that delivers the exact number of feasible schedules for $O||C_{\max}$. We note that the rank matrix is a special latin rectangle. Bräsel and Kleinau (1996) showed that the set of irreducible sequences contains at least one optimal sequence. They also showed that the active (a schedule is active if no operation can be started earlier without delaying another operation), rank-minimal, and irreducible sets are neither disjoint nor equivalent. Later, Bräsel, Harborth, Tautenhahn, and Willenius (1999b) and Bräsel, Harborth, Tautenhahn, and Willenius (1999a) presented necessary and sufficient conditions for feasibility that can be verified in polynomial time, and developed enumeration algorithms to identify the set of irreducible sequences. Andresen and Dhamala (2012) attempted to answer the open question of whether a given open-shop sequence is irreducible. They proposed three conjectures that if any of them holds, the given open-shop sequence can be polynomially verified to be irreducible. Nevertheless, the problem whether a given sequence is irreducible remains an open question. For a given sequence under certain assumptions the conjectures of either a reducible sequence exists or a sequence is feasible can be verified in $O(n^9m^9)$ and $O(n^5m^52^{n^2m^2})$ time, respectively. Next, we review the major results on the preemptive open-shop scheduling problem.

2.3. The preemptive open-shop

In scheduling theory, preemption refers to the situation where the execution of a job or operation can be interrupted so that another job or operation is executed. Depending on the setting, the preempted job may be resumed or re-started. Minimizing the makespan in the two-machine open-shop scheduling problem leads to identical optimal makespan for both the preemptive and non-preemptive variants (Gonzalez & Sahni, 1976). In addition, when $m > 2$ the optimal makespan for the preemptive variant can be obtained by either of the two polynomial algorithms by Gonzalez and Sahni (1976), which have the time complexity of $O(k^2)$ and $O(k(\min\{k, m^2\} + m\log n))$, where k is the number of non-zero operations (with non-zero processing time). Gabow and Kariv (1982) later improved the complexity of those algorithms for certain special cases. A similar polynomial algorithm to that of Gonzalez and Sahni (1976) was independently proposed for the preemptive m-machine open-shop by Inukai (1979). Lawler and Labetoulle (1978) proposed a linear program to solve the preemptive open-shop scheduling problem.

Assuming at most $\min\{kn, km, n^3, m^3\}$ preemptions, i.e., an upper bound on the number of preemptions, Gonzalez (1979) showed that the problem can be solved in $O(k + \min\{m^4, n^4, k^2\})$ time. However, if there is a lower bound on the number of preemptions, the problem is NP-hard (Gonzalez & Sahni, 1976). Baptiste et al. (2011) proved the integer preemption property for the two-machine problem. Under that property, if the input data for an instance are integral, then there exists an optimal preemptive schedule where all the interruptions, and start and completion times occur at integral times. Shchepin and Vakhania (2015; 2008; 2011) attempted to minimize the number of preemptions for the problem, such that the problem remains polynomially solvable. The problem can be mapped to an acyclic machine dependency graph in which each machine is denoted by a node and each job is represented by an edge to be processed by the pair of nodes (machines) connected by the edge. They proposed an $O(nm)$-time algorithm if at most $m-2$ preemptions are allowed, and showed that the problem with at most $m-3$ preemptions is NP-hard. They showed that the 2-approximation algorithm for the classical open-shop problem also holds for the non-preemptive acyclic problem, i.e., where the associate graph is acyclic.

Due to the absence of processing routes among the operations in the open-shop setting, it is possible to interrupt a job on a machine and resume it on another one, although such a feature might be undesirable for some real-world applications. Therefore, the case where the interrupted job must wait because it cannot be processed on another machine has also been studied. We may distinguish the machine-preemption-only and the job-preemption-only cases. In the former, interrupted job j on machine i cannot be transferred to other machines, while machine i can process other jobs during the interruption of job j. In the latter, interrupted job j on machine i can be processed on other machines, but machine i cannot process other jobs until job j resumes its processing. We note that due to symmetry of the jobs and machines in the open-shop scheduling environment, both cases, i.e., machine-preemption-only and job-preemption-only, are equivalent. The case of machine-preemption-only is also known as "open-shop with no passing" (Cho & Sahni, 1981) and "open-shop with restricted preemptions" (de Werra & Solot, 1993). Cho and Sahni (1981) showed that finding a feasible schedule for the machine-preemption-only with the common release time and deadline of 1, i.e., $O|prmp, r_j = 1, d_j = 1|-$, is NP-hard for $m > 2$. Later, de Werra and Solot (1993) studied $O|prmp|-$ with restricted preemption and showed that obtaining a feasible schedule is NP-hard. de Werra and Erschler (1996) studied restricted preemption and modelled it as a variant of the colouring problem (the graph colouring problem concerns using the smallest number of distinct colours such that no two adjacent edges have the same colour) and proved that the decision problem is NP-complete.

Bräsel and Shakhlevich (1998) proved that the preemptive UET open-shop problem is equivalent to the preemptive identical parallel-machine problem with UET operations. They also discussed that while the number of dense schedules for the preemptive problem is uncountable, i.e., infinite, it is finite for the non-preemptive counterpart. Baptiste et al. (2009) proved a number of properties for the feasible and optimal preemptive schedules for the open-shop problem. For example, they showed that contrary to the results in Gonzalez and Sahni (1976), the maximum number of partial schedules ("slices") in an optimal solution of $O|prmp|C_{\max}$, $m \le n$, can be reduced to $k+m$, where k is the total number of operations (of all the jobs). de Werra, Hoffman, Mahadev, and Peled (1996) considered the decision problem of scheduling the preemptive open-shop within k time units, in which some operations are pre-assigned to machines at some point. They introduced a few polynomially solvable cases.

In the next section, we focus on the open-shop scheduling problem where the jobs or machines follow specific structures.

2.4. The structured open-shop

As discussed earlier, the open-shop scheduling problem is polynomially solvable only for a few variants and under very restricted conditions, see Sections 2.1 and 2.3. It follows that certain structures and properties of those variants are important for solving the problem. Therefore, there is a stream of research that aims to identify special structures for the input data so the problem is polynomially solvable. Five such important special structures have been studied in the literature, namely (1) "machine load", (2) "dominating machine", (3) "bottleneck machine", (4) "proportionate scheduling", and (5) "ordered scheduling". We discuss these structures in the following.

Machine load

By establishing a relationship between the maximum machine load and the maximum operation length, Fiala (1983) showed that an optimal schedule of length $l_{\max}$ can be constructed for $O||C_{\max}$ in $O(n^2m^3)$ time, if $l_{\max} \ge (16m' \log_2 m' + 5m')p_{\max}$, where $m' = 2^{\lceil \log_2 m \rceil}$ and $p_{\max}$ is the longest processing time among all the operations. Bárány and Fiala (1982) improved $l_{\max}$: If $l_{\max} \ge (16m \log_2 m + 26m)p_{\max}$, the optimal schedule can be constructed in $O(nm^3)$ time (for the details see Sevast'yanov, 1992). A series of improvements was later proposed. For example, if $l_{\max} \ge (\frac{16}{3} m \log_2 m'' + \frac{13}{9} m - \frac{4m}{9m''}(-1)^k)p_{\max}$, where $m'' = 2^k$ and $k = \log_2 m$, the optimal schedule is obtained in $O(nm^2 \log_2 m)$ time (Sevast'yanov, 1992), and if $l_{\max} \ge (m^2 - 1 + \frac{1}{m-1})p_{\max}$, the optimal schedule of length $l_{\max}$ can be obtained in $O(n^2m^2)$ time (Sevast'yanov, 1995). If all the processing times are selected from a bounded set of non-negative integers, the $O(nm)$ algorithm by Čepek, Vlach, and de Werra (1994) solves the restricted case considered by Fiala (1983). If $l_{\max} \ge 7p_{\max}$, the algorithm by Sevast'yanov (1998) (called the "non-strict vector summation") solves the three-machine open-shop problem in $O(n \log n)$ time.

Dominating machine

Machine i' dominates machine i, if the minimum processing time on machine i' is at least as large as the longest processing time on machine i, i.e., $\min_j\{p_{i'j}\} \ge \max_j\{p_{ij}\}$. Adiri and Aizikowitz (1989) investigated the three-machine problem in the presence of a dominating machine and $n \ge m$. They showed that that case can be solved in $O(n)$ time (so can its flow-shop counterpart). To solve the problem, one may disregard the dominated machine and solve the resulting two-machine problem by using the algorithm by Gonzalez and Sahni (1976). Then, the jobs on the dominated machine can be scheduled without conflicts. Sevast'yanov (1996) introduced two polynomially solvable cases of $O||C_{\max}$. He showed that for the open-shop with dominant machine i' of load $l_{i'} = l_{\max}$ and $\Delta = l_{\max} - l_i \ge (2m-4)p_{\max}, i \ne i'$, there exists an optimal schedule of length $l_{\max}$ that can be constructed in $O(nm^2)$ time. In addition, if $l_{\max} \ge (5.45m - 7)p_{\max}$ and $\Delta \ge (m-1)p_{\max}$, then a greedy algorithm always exists that delivers an optimal schedule in $O(n^2m^2)$ time. He also presented an approximation algorithm for the problem satisfying $l_{\max} \ge (m')p_{\max}$, where m' is a function of the number of machines. The algorithm has a running time of $O(nm^2)$ with an absolute performance guarantee of $(\lceil \frac{3m^2-2m}{l_{\max}+2mp_{\max}} \rceil - 1)p_{\max}$. Assuming zero-one operations (i.e., $p_{\max} = 1$) and the machines numbered in non-decreasing order of their loads, Kononov, Sevast'yanov, and Tchernykh (1999) showed that the optimal schedule for problem $O3||C_{\max}$ can be constructed in $O(n)$ time, if $l_1 = l_2 \ge l_3 + 2$. They also gave the conditions under which the optimal schedule can be obtained polynomially for some cases with $m \ge 4$. For the same problem, Tanaev, Gordon, and Shafransky (2012) also presented an $O(nm)$ algorithm. It should be noted that the problem with a dominating machine, but without the condition of $n \ge m$, is NP-hard.

Bottleneck machine

The bottleneck machine refers to the case where a job is processed on a subset of machines, rather than on all the machines. Here, a machine is a bottleneck if one of the operations of every job needs to be processed on that machine. Gonzalez and Sahni (1976) showed that the four-machine problem is NP-hard even for two-operation jobs, i.e., each job has exactly two operations. This NP-hardness result also holds if one of the machines is a bottleneck. For the three-machine and two-operation open-shop problem with a bottleneck machine, Drobouchevitch and Strusevich (1999) presented a linear-time solution algorithm, although the flow-shop counterpart of the problem is strongly NP-hard (see Herrmann & Lee, 1992). Recently, Drobouchevitch (2020) proposed an alternative linear-time algorithm for the same problem. Kyparisis and Koulamas (2000) investigated the m-machine case of the same problem and showed that it can be solved in $O(n + m \log m)$ time if the bottleneck machine i' is also the maximal machine, i.e., $p_{i'j} \ge p_{ij}, \forall j \in N, \forall i \in M \setminus \{i'\}$.

Proportionate scheduling

The proportionate open-shop scheduling problem is characterized by the fact that all the jobs have the same processing time on a given machine, i.e., $p_{ij} = p_i, \forall i \in M$. The three-machine problem is known to be at least ordinary NP-hard due to Liu and Bulfin (1987). Under the assumption that $p_1 \ge p_2 \ge \cdots \ge p_m$, Dror (1992) proposed (1) an optimal $O(mn)$ algorithm for $O|prop, n \ge m|C_{\max}$, where *prop* denotes the proportionate setting, (2) an optimal $O(m)$ algorithm for $O|prop, n = 2, m > 2|C_{\max}$, and (3) an NP-hardness proof for $O|prop, n \ge 3, n < m|C_{\max}$. In the three-machine setting, Koulamas and Kyparisis (2015) proved that if $p_1 = p_2$, the problem is polynomially solvable, and remains polynomially solvable even if $p_1 > p_2$ and $l_{\max} \le 2p_1 + p_2$, or if $p_1 > p_2$ and $l_{\max} \ge 3p_1 + p_3$. Under the general condition, they proposed an $O(n \log n)$-time $\frac{7}{6}$-approximation algorithm. Naderi, Zandieh, and Yazdani (2014) studied problem $O|prop|C_{\max}$ and proposed a polynomial algorithm to solve the case where $n \ge m$, and for $m > n$, presented an approximation algorithm with the worst-case performance bound of $2 - \frac{1}{n}$, and provided a heuristic algorithm by converting the problem into a simpler problem called the "machine batch fitting" problem. The complexity of the three-machine problem was investigated by Sevastyanov (2019). The study proposed a pseudo-polynomial time algorithm, and also new solvable cases.

Ordered scheduling

The ordered assumption is indeed a generalization of the proportionate case. We call two machines $i, i' \in M, i \ne i'$ ordered and denote it as $M_i > M_{i'}$, if $p_{ij} \ge p_{i'j}, \forall j \in N$. Analogously, jobs $j, j' \in N, j \ne j'$ are ordered $(j > j')$ if the processing time of j is larger

Table 4
Major abbreviations used throughout the paper sorted in alphabetical order.

Abbreviation	The complete term
ACO	Ant colony optimization
B&B	Branch-and-bound
BS	Beam search
GA	Genetic algorithm
PSO	Particle swarm optimization
SA	Simulated annealing
TS	Tabu search
VNS	Variable neighbourhood search

than that of j', i.e., if $p_{ij} \geq p_{ij'}, \forall i \in M$ (Khatami & Salehipour, 2020b; Khatami, Salehipour, & Hwang, 2019). An open-shop problem is ordered if it includes both ordered machines and ordered jobs.

Liu and Bulfin (1987) showed that the three-machine ordered open-shop problem, i.e., $O3|ord|C_{\max}$, where *ord* denotes the ordered setting, is *NP*-hard, implying that problems $O3|j > j'|C_{\max}$ (with ordered jobs only) and $O3|M_i > M_{i'}|C_{\max}$ (with ordered machines only) are *NP*-hard. They proposed an $O(n)$ algorithm for problem $O3|M_i > M_{i'}|C_{\max}$ with the additional constraint that the job with the longest processing time on the first machine is different from the job with the longest processing time on the second machine, i.e., the longest jobs on the machines are different. Kyparisis and Koulamas (1997) proposed an $O(n)$ algorithm for problem $O|M_1 > M_i, m \leq n|C_{\max}$ when the ith longest processing time on machine 1 is as large as the processing times of all the operations on machines i through m. They also generalized the results of Liu and Bulfin (1987) for the three-machine open-shop with ordered machines.

2.5. Solution methods

Various solution techniques, including exact, heuristic and meta-heuristic have been proposed for challenging and more general variants of the open-shop scheduling problem. In this section, we first review the characteristics of the well-known benchmark instances for the open-shop, and then we review the available solution techniques. Table 4 lists the abbreviations for major solution techniques that we review in this section.

Benchmark instances

Three well-known sets of benchmark instances are available for the open-shop scheduling problem. These three sets (and sometimes extended variants of them) have been utilized by many researchers for assessing the performance of the algorithms and the solution techniques. The benchmark instances are due to Taillard (1993), Brucker, Hurink, Jurisch, and Wöstmann (1997), and Guéret and Prins (1999).

Taillard (1993) introduced the first set of benchmark instances for the open-shop scheduling problem, in which an instance is characterized by the pair (n, m) and consists of six different sizes as follows: (4,4), (5,5), (7,7), (10,10), (15,15), and (20,20), where the processing times are randomly generated from the discrete uniform distribution in the range $U[1, 99]$. For each size, they generated ten instances that resulted in a total of 60 instances. This set of instances is considered easy because the trivial lower bound LB_0 (see below) is equal to the optimal makespan for 40 of the larger instances (Malapert et al., 2012). The trivial lower bound on the makespan can be calculated as the maximum of jobs' durations and machines' loads, denoted as LB_0, as follows:

$$LB_0 = \max\{\max_j\{\sum_i p_{ij}\}, \max_i\{\sum_j p_{ij}\}\}. \quad (1)$$

The first term in the lower bound represents the longest job duration and the second term defines the maximum machine load $l_{\max}$, i.e., $l_{\max} = \max\{l_i\}$, where $l_i = \sum_j p_{ij}$ represents the load of machine *i*. It should be noted that the first non-trivial lower bound for the open-shop scheduling problem is due to Guéret and Prins (1999). The lower bound is calculated by determining the optimal makespan for a relaxed version of the problem. We refer the interested reader to Guéret and Prins (1999) for the details.

Brucker et al. (1997) proposed a few measures to capture the hardness of an instance and applied them to Taillard (1993)'s instances. Since they observed that Taillard (1993)'s instances are easy to solve, they generated their own instances. They proposed a set of 52 challenging instances. Similar to the instances of Taillard (1993), they considered an equal number of jobs and machines to generate the instances. They proposed eight instances for $n = m \in \{3, 8\}$ and nine instances for $n = m \in \{4, \ldots, 7\}$. The third benchmark is that of Guéret and Prins (1999), in which there are ten instances for each size of $n = m \in \{3, \ldots, 10\}$, leading to 80 instances. It should be noted that the benchmarks proposed by Brucker et al. (1997), and Guéret and Prins (1999) are more challenging than those of Taillard (1993) because the advanced GA by Prins (2000), ACO by Blum (2005), and PSO by Sha and Hsu (2008) struggle to optimally solve all the instances of Brucker et al. (1997) and Guéret and Prins (1999). We refer the interested reader to Malapert et al. (2012) for the details on the performance of exact and heuristic methods in solving the instances of Taillard (1993), Brucker et al. (1997), and Guéret and Prins (1999).

Next, we review the available exact methods, followed by the heuristic and meta-heuristic algorithms that were proposed to solve the general open-shop scheduling problem.

2.5.1. Exact algorithms

The open-shop scheduling problem typically has a larger solution space compared with, e.g., the job-shop and the flow-shop problems, due to its unrestricted job processing order. The free job route also leads to a smaller gap between the optimal makespan and a lower bound (Guéret & Prins, 1999; Prins, 2000). The major exact solution methods to solve the open-shop scheduling problems include the B&B algorithm, the constraint programming and dynamic programming techniques.

Branch-and-bound based algorithms

Brucker et al. (1997) and Brucker, Hilbig, and Hurink (1999) proposed the first B&B algorithms for the open-shop problem. Brucker et al. (1997) used the ideas from Grabowski, Nowicki, and Zdrzałka (1986) for branching and from Carlier and Pinson (1989) for immediate selection of operations and developed a depth-first B&B based on a disjunctive graph formulation of the problem. They tested the algorithm on the instances of Taillard (1993). In Brucker et al. (1999), they first reduced an instance of the open-shop to a dedicated parallel-machine scheduling problem and then to a single-machine scheduling problem with positive and negative time-lags. Positive and negative time-lags are used as general timing constraints between the start times of the jobs. They concluded that such a transformation technique is not as efficient as solving the problem directly, i.e., without applying the transformation. Guéret and Prins (1998b) attempted to improve the performance of the B&B method by Brucker et al. (1997). Their proposed technique identifies certain forbidden intervals for the start and completion times of the operations, during which no operation can start or end in an optimal solution. For this reason, $n + m$ subset-sum problems are solved by an efficient dynamic program. Each subset-sum problem corresponds to either a job or a machine. They solved 40 instances of the benchmark of Taillard (1993) and obtained reductions in the number of backtracks for 31 instances. This is equivalent to more than 75% improvement. They also reported improved solutions for certain instances. Guéret, Jussien, and Prins (2000) also proposed an advanced backtracking scheme and combined it with the B&B by Brucker et al. (1997). They showed that it reduces the number of backtracks. Dorndorf, Pesch, and Phan-Huy (2001) designed several consistency tests that reduce the search space of their B&B algorithm. They used the two strategies of "top-down" and "bottom-up" to guide the search. These strategies only differ in the way in which the initial upper bound is chosen. The top-down strategy starts with an upper bound and attempts to improve it by applying the B&B algorithm. In the bottom-up approach, however, a lower bound is selected as a target upper bound and it is incremented until a feasible solution is found. They showed that the proposed algorithm outperforms those of Brucker et al. (1997) and Guéret et al. (2000).

The solution technique by Tamura, Taga, Kitagawa, and Banbara (2009) includes incorporating the constraint satisfaction problem with integer linear constraints into the Boolean satisfiability problem. The constraint satisfaction problem is defined by a set of variables and a set of constraints where each variable has a domain of possible values and each constraint specifies the allowed combinations of values over a subset of variables. Their method is similar to that of Crawford and Baker (1994) for the job-shop scheduling problem. They showed that the method is able to find the optimal solutions for all 192 instances of the three benchmarks of Taillard (1993), Brucker et al. (1997), and Guéret and Prins (1999), including the three instances of Brucker et al. (1997) that had not been optimally solved earlier.

Constraint programming based techniques

Laborie (2005) proposed a constraint programming method based on detection and resolution of the minimal critical set (MCS). The minimal critical set specifies the minimum requirement for a resource R that would be over-consumed if executed simultaneously. MSCs are chosen in such a way that the size of the search space is minimized. In the disjunctive scheduling context, MCSs are pairs of activities that conflict for the same unary resource. At each node, the branching consists of (1) selecting an MCS according to an estimation of the related reduction of the search space, (2) applying a simplification procedure on each MCS, and (3) branching on its possible precedence in the children nodes until no MCS remains. They managed to solve the 34 and then open instances of Guéret and Prins (1999), and Brucker et al. (1997), respectively.

The constraint programming method by Grimes, Hebrard, and Malapert (2009) sets the makespan equal to $\frac{LB+UB}{2}$ and iteratively solves a feasibility problem until the interval is empty. They optimally solved all the instances of the three benchmarks in reasonable times. The constraint programming algorithm by Malapert et al. (2012) combines randomization with the re-start techniques, and applies efficient propagation and scheduling heuristics. The algorithm not only generates all the known optimal solutions, it also outperforms all the previous methods in terms of the computational effort.

Dynamic programming techniques

Ozolins (2019) proposed a dynamic program for the open-shop scheduling problem. They proposed a dominance rule and utilized the ideas of forbidden intervals by Guéret and Prins (1998b) and symmetry breaking by Malapert et al. (2012) in their dynamic program. They showed that the algorithm solves instances from benchmarks of Taillard (1993) and Brucker et al. (1997) with up to 7 jobs and machines.

2.5.2. Heuristic and meta-heuristic algorithms

In this section we review major heuristic and meta-heuristic algorithms developed to tackle the open-shop scheduling problem. Those heuristics are mainly different in their mechanism for scheduling the operations, i.e., selecting the operation to be scheduled next. While some of the heuristic methods schedule one operation at a time, the others schedule a set of operations in each iteration of the algorithm. In general, the problem of inserting an additional job into a given schedule of n jobs such that the order of the n jobs in the given schedule does not change and the resulting makespan is as small as possible is strongly *NP*-hard (Vestjens, Wennink, & Woeginger, 2007).

The "insertion" algorithm of Bräsel et al. (1993) schedules one operation at a time. The algorithm is indeed a restricted B&B algorithm with $O(n^2m^2)$ time complexity that iteratively inserts an operation into a partially constructed schedule. The algorithm utilizes the BS procedure, a heuristic search exploring a graph through branching the most promising node, meaning that a number of solution paths in the branching tree is explored. They showed that their insertion algorithm performs better than the TS by Taillard (1993) because it obtains improved solutions for almost all the instances of Taillard (1993). Ramudhin and Marier (1996) generalized the shifting bottleneck procedure originally proposed for the job-shop scheduling problem (Adams, Balas, & Zawack, 1988). Their procedure consists of iteratively selecting a bottleneck job or machine, and re-optimizing its execution sequence. Another heuristic that works by scheduling a single operation is the algorithm of Guéret and Prins (1998a), which is a list scheduling method with $O(n^2m^2)$ time complexity. Given a priority (dispatching) rule, a list scheduling algorithm essentially determines the earliest time that a yet-to-be scheduled operation with the highest priority can be started. The process is iterated until all operations are scheduled. Two priority rules based on the residual work durations of the operations, similar to those of Prins and Carlier (1986), were used by Guéret and Prins (1998a) in order to prioritize the operations. Naderi, Ghomi, Aminnayeri, and Zandieh (2010)'s heuristic algorithms remove redundant solutions, which are generated as a result of permutation list encoding methods. For this reason, they proposed four rules. Indeed, these rules lead to four heuristics, all of which apply the insertion operator. They showed that their heuristics have superior performance to the existing algorithms of the longest total processing time (a generalization of the LAPT rule by Pinedo and Schrage (1982) for the two-machine problem), the method by Liaw (1998), and the generation of active and non-delay schedules based on the SPT rule.

The heuristic algorithm proposed by Bräsel et al. (1993) operates by iteratively scheduling a set of $\min\{n, m\}$ operations. For that, the algorithm finds the rank-minimal schedules by successively solving the weighted bipartite maximum cardinality matching problems. That results in a set of $\min\{n, m\}$ operations with rank 1, then rank 2 and so on. The heuristic of Guéret and Prins (1998a) includes two phases. The first phase partitions the operations into subsets by computing each subset as a matching in a weighted bipartite graph. For this reason, a weighted bipartite graph $G(X \cup Y, E, P)$ is constructed, in which each machine i and job j correspond to a vertex in X and a vertex in Y, respectively. Each operation O_{ij} is represented by an edge in E with the associated weight $p_{ij} \in P$. Next successive matchings are extracted from G until G is decomposed, i.e., all the vertices have a zero degree. Each matching specifies a subset of operations that can be performed simultaneously and therefore form a schedule slice. The resulting schedule slices are then concatenated in the order of generation to make a complete schedule (similar to Brucker et al., 1997). The second phase improves the schedule obtained in the first phase. The iterative improvement procedure of Liaw (1998) generates an initial solution by applying a heuristic that uses "the longest total remaining processing on the other machines" dispatching rule by Pinedo (1995). The improvement method of the procedure separates the sequencing and scheduling problems. The sequencing problem is solved via an iterative procedure that is based on Benders' decomposition, and the scheduling part is shown to be the dual of the longest path problem that can be efficiently solved with a label correcting algorithm. They showed that the procedure is able to solve most of their own randomly generated instances within 1% deviation from the trivial lower bound LB_0. The so-called rotation schedule by Bai and Tang (2011) is a heuristic that first schedules all the jobs on the first machine in the order 1 to n. Then, on any machine $2 \leq i \leq m$, the schedule starts with jobs j to n, and then jobs 1 to $j-1$ are added. The algorithm has several theoretical advantages. First, it is an asymptotically optimal method when the number of jobs n goes to infinity. Second, its worst case ratio is equal to the number of machines m.

The greedy algorithm by Strusevich (1998) for $Om||C_{\max}$ generates dense schedules and has an $O(m \log m + nm \min\{n, m\})$ running time. It has been conjectured that the algorithm guarantees the worst-case ratio of $2 - \frac{2}{(m+1)}$ for any m. It has also been shown that the algorithm obtains a schedule at most $\frac{3}{2}$ times of the optimal one and the bound is tight.

The major meta-heuristic methods proposed for the open-shop include the evolutionary algorithms. The first such method is the GA by Fang, Ross, and Corne (1993). Later, Fang, Ross, and Corne (1994) used a simple heuristic rule within GA, chosen from a pool of eight heuristic rules, as well as adaptively choosing the dispatching rules during the course of the algorithm, and showed that the adaptive strategy leads to new best solutions. Several strategies have been investigated to improve the performance of GA. For example, the GA by Louis and Xu (1996) uses the knowledge gained from solving the problem, by storing the past moves, and that by Khuri and Miryala (1999) incorporates the longest processing time (LPT) dispatching rule to generate quality solutions. Prins (2000) introduced advanced crossover operators and schedule generators that produce non-delay and active schedules. Other studies to improve the performance of GA include Puente, Díez, Varela, Vela, and Hidalgo (2003), in which the initial population is seeded by the probabilistic version of the dispatching rules by Liaw (1998), and Senthilkumar and Shahabudeen (2006), who used an operation-based representation and a scheduling algorithm that generates the active schedules. Recently, Rahmani Hosseinabadi, Vahidi, Saemi, Sangaiah, and Elhoseny (2019) investigated the impact of the crossover and mutation operators, and designed a competitive GA. GA has also been combined with other algorithms that lead to hybridized methods. Examples include TS (Liaw, 2000) and VNS (Zobolas, Tarantilis, & Ioannou, 2009) algorithms. Ahmadizar and Hosseinabadi Farahani (2012) presented a hybrid GA, in which a local optimization heuristic is applied in order to improve a pre-specified per cent of individuals selected from the population. Therefore, a fraction of the individuals undergo the local optimization heuristic (instead of all of them). Their hybrid GA performs significantly better than that of Prins (2000). However, there is no statistically significant difference between their proposed algorithm and the GA of Liaw (2000).

Other proposed evolutionary algorithms include the hybridized constructive algorithms of BS and ACO, called Beam-ACO (Blum, 2005). In each iteration, every ant performs a probabilistic BS, instead of using the traditional solution construction mechanism of ACO. To this end, adding a new component to a partial solution is performed by using a transition probability, and extending the partial solution is done by a reduced set of feasible solutions. The constructed solution is then improved by a local search. The algorithm outperforms the GAs by Prins (2000) and Liaw (2000). The discrete PSO of Sha and Hsu (2008), in which the algorithm is hybridized with four different decoding operators, has been shown to outperform the earlier meta-heuristics by Blum (2005); Liaw (2000); Prins (2000). Motivated by "the smaller the idle-time of a partial solution, the smaller the makespan" premise, Huang and Lin (2011) proposed a bee colony optimization algorithm that uses an idle time-based filtering scheme aiming to terminate the search for partial schedules with insufficient profitability. To this end, they define profitability of a partial schedule as reciprocal of its idle time, implying that the larger the idle time, the smaller the profitability. The filtering algorithm then fathoms further search on a partial schedule whose profitability is smaller than an acceptable threshold. The parallel GA by Ghosn, Drouby, and Harmanani (2016) solves the non-preemptive open-shop, and the swarm intelligence algorithm by Bouzidi, Riffi, and Barkatou (2019) solves problem $O||C_{\max}$. Pongchairerks and Kachitvichyanukul (2016) proposed a two-level PSO. The upper-level of the algorithm tunes the parameters of the lower-level process. They showed that their algorithm, which considers parameterized active schedules, is able to deliver slightly better solutions than, e.g., the Beam-ACO by Blum (2005), although they did not discuss the computation effort.

Two other widely applied meta-heuristics include TS and SA. Alcaide, Sicilia, and Vigo (1997) proposed a TS algorithm for problem $O||C_{\max}$. The initial solution is selected from the best solutions generated by using three simple list scheduling algorithms, i.e., the best solution delivered by the three algorithms is used as the initial solution. This solution is then improved by two critical path-based neighbourhood structures that are similar to those by Dell'Amico and Trubian (1993). They conducted numerical studies on instances with up to 25 machines and 250 jobs to assess the performance of the algorithm. Liaw (1999b) used the disjunctive graph model of Roy and Sussmann (1964) and proposed TS for the problem of finding the critical path on the graph (when the problem is modelled as a disjunctive graph, finding the critical path on the graph results in fining the optimal solution for the associated scheduling problem). The critical path is decomposed into a number of blocks, i.e., a sub-sequence of operations that belongs to the same job or that is processed on the same machine. The neighbourhood move includes swapping operations that belong to the same block. Testing the algorithm on random and benchmark instances, they obtained the optimal solutions for almost 97% of the randomly generated instances, and quality including optimal solutions, for the majority of the benchmark instances of Taillard (1993). Jussien and Lhomme (2002) proposed an algorithm based on TS and a filtering technique. Testing the algorithm on the three benchmarks, they obtained superior solutions to Alcaide et al. (1997), Liaw (1999b), and Prins (2000). In addition, the algorithm generates new best solutions for 25 instances of Guéret and Prins (1999), including six optimal solutions. Liaw (1999a) proposed SA that uses the neighbourhoods developed in his earlier work (Liaw, 1999b). Even though the algorithm is capable of finding quality solutions for a wide range of benchmark instances, it has a long computation time for large instances. Another SA is due to Harmanani and Ghosn (2016) that swaps, shifts, and rotates the neighbourhoods.

Colak and Agarwal (2005) proposed a neural network algorithm that uses ten heuristic rules. In this method, the open-shop problem is first converted into a neural network. Then a learning strategy is iteratively applied to improve the solutions. They obtained competitive solutions for the benchmark instances of Taillard (1993).

3. Non-classical resource settings in the open-shop scheduling problem

In Section 2, we introduced the classical open-shop scheduling problem and reviewed studies that consider the problem in the general settings. In this section we review studies that investigate

Table 5
Summary of results with machine unavailability.

Problem	Complexity and solution method	Reference
$O\|prmp, h_i\|C_{\max}$	Polynomial for undetermined holes; if holes are pre-determined, and if preemption is allowed in non-integer intervals, polynomial, and strongly *NP*-hard otherwise	Vairaktarakis and Sahni (1995)
$O2\|\beta\|C_{\max}$, where β is:		
$h(1,0), t_h = 0$	Polynomial in $O(n)$, the result holds for $h(0,1)$ as well	Lu and Posner (1993)
$h(1,0), t_h > 0, Re$	*NP*-hard, the result holds for $h(0,1)$ as well	Breit et al. (2001)
	Ordinary *NP*-hard, the result holds for $h(0,1)$ as well	Lorigeon et al. (2002)
$h(1,1), t_h > 0, Re$	PTAS	Kubzin et al. (2006)
$h(k,0), t_h > 0, Re$	PTAS	Kubzin et al. (2006)
$h(2,1), t_h > 0, Re$	Inapproximable, the result holds for $h(1,2)$ as well	Breit (2000)
$h(1,0), t_h > 0, N-Re$	$\frac{4}{3}$-approximation, the result holds for $h(0,1)$ as well	Breit et al. (2003)
$h(1,1), t_h > 0, N-Re$	2-approximation	Breit et al. (2003)
$h(2,0), t_h > 0, N-Re$	Inapproximable, the result holds for $h(0,2)$ as well	Breit et al. (2003)
$h(1,0), t_h \in [k_1,k_2], N-Re$	$\frac{3}{2}$-approximation, the result holds for $h(0,1)$ as well	Mosheiov et al. (2018)
$f(1,1), t_f > 0$	Optimal in $O(n)$	Kubzin and Strusevich (2006)

the open-shop scheduling problem in various settings and under different conditions on the machines (processing resources), e.g., machine availability, competing agents, and renewable and non-renewable resources.

3.1. Machine availability

The machine availability constraint models the situation in which the machines are not continuously available due to, e.g., maintenance or rest periods. A non-availability period is often called a “hole”. For the two-machine problem, the machine availability constraints are represented by $h(k_1, k_2)$, where $k_{1(2)}$ defines the number of holes on machine 1 (2) and $k_1 + k_2 \geq 1$ (Breit, Schmidt, and Strusevich, 2003, for the details see). In the following, we focus on the case where the machine unavailability durations are constant and known in advance. Vairaktarakis and Sahni (1995) studied the preemptive open-shop scheduling problem with an arbitrary number of holes and showed that when the start times of the holes are undetermined, i.e., the scheduler can decide the start times, the optimal solution can be found in polynomial time. Given pre-determined holes and preemption allowed in non-integer intervals, they also proposed a linear program. The problem, however, is strongly *NP*-hard if preemption is only possible within integer intervals.

The majority of the research with the machine availability constraint focuses on the non-preemptive open-shop. Under the non-preemptive condition, the three cases of “resumable”, “non-resumable”, and “semi-resumable” model the different impacts of the holes on the interrupted operations. The resumable setting (denoted as “Re” in Table 5) allows an interrupted operation (caused by a hole) to be resumed after the hole with no penalty. The non-resumable case (denoted as “N-Re” in Table 5) does not allow resumption, so the interrupted operation must be re-started. The semi-resumable refers to a situation in which the interrupted operation is partially re-started after a hole (Ma, Chu, & Zuo, 2010). We note that, to the best of our knowledge, there is no study on the open-shop scheduling problem in the semi-resumable setting.

Breit, Schmidt, and Strusevich (2001) studied the two-machine problem in the resumable setting. They showed that the problem with a single hole, which is not at the beginning of the schedule, i.e., $t_h > 0$ is *NP*-hard, and proposed a $\frac{4}{3}$-approximation algorithm. Later, Lorigeon, Billaut, and Bouquard (2002) proposed a pseudo-polynomial time dynamic program for the same problem and showed that the problem is ordinary *NP*-hard. They also showed that the worst-case performance ratio of any heuristic for the problem is 2. It should be noted that if the hole is at the beginning of the schedule ($t_h = 0$), the problem is solvable in $O(n)$ time due to Lu and Posner (1993). It is known that the two-machine open-shop with two holes on one machine and one hole on the other, i.e., $h(2,1)$ or $h(1,2)$ cannot be approximated unless $P = NP$ (Breit, 2000). Kubzin et al. (2006) studied two additional cases of a single hole on each machine and several holes on one machine. They proposed polynomial-time approximation schemes for each case.

Breit et al. (2003) studied two-machine problem in the non-resumable setting. They showed that the problem with two holes on either machines cannot be approximated in polynomial time unless $P = NP$. They provided a 2- and a $\frac{4}{3}$-approximation algorithms for cases with one hole on each machine, i.e., $h(1,1)$, and one hole on either machines, i.e. $h(1,0)$ or $h(0,1)$. Mosheiov, Sarig, Strusevich, and Mosheiff (2018) considered the two-machine problem in the non-resumable setting where a single hole is to be scheduled on either machines and it must be started in a given time window $[k_1, k_2]$. They proposed a $\frac{3}{2}$-approximation algorithm for the problem.

It is also possible that the durations of the holes are not constant, in which case there are “floating” holes, each of which, denoted by $f(k_1, k_2)$, is a time-dependent operation whose duration deteriorates as a function of its start time. Kubzin and Strusevich (2006) studied the two-machine problem with floating holes where there is one hole on each machine, i.e., $f(1,1)$. Using the algorithm by Lu and Posner (1993), they proposed an $O(n)$ time algorithm to optimally solve the problem. Table 5 summarizes the aforementioned studies with the machine availability constraint. In the table, PTAS denotes the polynomial-time approximation scheme.

3.2. Two competing agents

In the two-agent setting, there are two agents each aiming to minimize its own objective, which only depends on its own jobs. Zhao and Wang (2015) considered the two-machine open-shop scheduling problem with two competing agents and deteriorating processing times. They proved that minimizing the makespan of one agent, while the makespan of the other agent is bounded by a certain threshold (denoted as k and is a given parameter), is *NP*-hard. Jiang, Zhang, Bai, and Wu (2018) considered the two-machine problem with two competing agents to minimize the weighted sum of the makespan of the two agents. They proved that the problem of minimizing $C^A_{\max} + kC^B_{\max}$, with $k > 0$, where $C^{A(B)}_{\max}$ denotes the makespan of agent $A(B)$, is ordinary *NP*-hard. They also showed that the LAPT rule provides a 2-approximation algorithm when $k = 1$. Su and Hsiao (2015) studied the more general case where m machines with the machine availability and eligibility constraints exist, preemption is allowed, and the jobs have release times. They considered the same objective function as that of Zhao

Table 6
Summary of the results with two competing agents.

Problem	Complexity and solution method	Reference
$O2\|p_{ij}^{A(B)} = \alpha_{ij}^{A(B)}t, C_{\max}^B \leq k\|C_{\max}^A$	*NP*-hard	Zhao and Wang (2015)
$O2\|\|C_{\max}^A + kC_{\max}^B$	Ordinary *NP*-hard	Jiang et al. (2018)
$O2\|i = 1, 2\|C_{\max}^A + C_{\max}^B$	2-approximation	Jiang et al. (2018)
$O\|C_{\max}^B \leq k, r_j, prmp, avail., elig.\|C_{\max}^A$	LP, heuristic	Su and Hsiao (2015)

and Wang (2015). The eligibility constraint models the situation where the number of operations of a job can be fewer than the number of machines. Su and Hsiao (2015) presented an LP and proposed a dispatching rule-based heuristic for the problem. Table 6 summarizes the major studies in the context of two competing agents.

3.3. Renewable and non-renewable resources

The required resources for processing the operations can be classified as renewable and non-renewable. Additional non-machine resources, e.g., manpower and tools, can be denoted as $res_{\kappa\chi\delta}$, where κ, χ and δ denote the number of resources, the total amount of available resources per time unit, and the maximum resource requirement of the operations, respectively. In addition, the "." for each of those parameters indicates that the corresponding parameter can take any integer value, whereas a positive integer indicates that the value of the parameter is fixed, i.e., given. Also, *res* and *tres* indicate time-dependent and time-independent resources, respectively. The availability or consumption of a time-dependent resource may change over time.

Generally speaking, finding a feasible schedule for the resource constrained open-shop scheduling problem, even with one renewable resource over different time periods is *NP*-hard both with and without preemption (de Werra & Solot, 1993). The first studies on the open-shop scheduling problem with renewable resources were conducted by Blazewicz, Lenstra, and Kan (1983) and Blazewicz, Cellary, Slowinski, and Weglarz (1986), who presented the complexity results of some variants of the problem in the UET setting. Cochand, de Werra, and Slowinski (1989) and de Werra, Bewicz, and Kubiak (1991) considered non-renewable resources. They showed that problem $O|p_{ij} = \{0, 1\}, nr\ staircase|C_{\max}$ (where "*nr staircase*" denotes the non-renewable staircase, and a staircase pattern refers to the situation in which a non-renewable resource such as money may be available only at certain times and in fixed quantities, so its availability follows the staircase pattern) is strongly *NP*-hard, while problem $O|UET, nr\ staircase|C_{\max}$ is solvable in $O(mn)$ time. de Werra (1990) modelled the preemptive and non-preemptive open-shop scheduling problem as a graph, where the nodes represent the resources and the weighted edges denote the jobs. They called it the "chromatic scheduling problem". They proposed an algorithm that solves the preemptive case and showed that the algorithm polynomially solves the non-preemptive variant if the graph has certain properties. de Werra and Blazewicz (1992) and de Werra and Blazewicz (1993) extended the results of de Werra et al. (1991) and studied the preemptive open-shop with a renewable or a non-renewable resource. They showed that the problem is equivalent to the edge colouring problem in the bipartite multi-graph, and presented some cases that are polynomially solvable. Tautenhahn and Woeginger (1997) studied the UET open-shop problem in which the availability of a renewable resource changes over time, i.e., it is time-dependent. They showed that the problem to minimize the regular objective function $\sum f_j(C_j)$, i.e., $O|UET, tres_{\kappa\chi\delta}|\sum f_j(C_j)$, is polynomially solvable if the number of machines, number of resources, and resource demand of each operation are bounded. They also established the complexity status of some cases of the problem to minimize other objective functions. Jurisch and Kubiak (1997) considered $O2|res_{1..}|C_{\max}$ and $O2|res_{211}|C_{\max}$. For the former, in which each operation needs a number of a single renewable resource, they developed an $O(n^3)$ solution algorithm, that can also solve the preemptive case, i.e., $O2|res_{1..}, prmp|C_{\max}$ with no change in the makespan value. For the latter, in which each operation requires at most one unit of either resources, they showed that it is *NP*-hard, and its general version with an arbitrary number of resources, i.e., $O2|res.11|C_{\max}$, is strongly *NP*-hard.

Shabtay and Kaspi (2006) developed an $O(n \log n)$ algorithm for the two-machine open-shop with resource-dependent processing times, in which the availability of additional resources leads to shorter processing times, i.e., $p_{ij} = (w_{ij}/res_{ij})^k$, where w_{ij} is a positive value representing the workload of the operation, res_{ij} is the resource consumed by job j on machine i, and k is a positive constant. The total amount of the resource consumed is upper-bounded by R ($\sum_{i=1}^{2}\sum_{j=1}^{n} res_{ij} \leq R$), where R is a positive number. They showed that the general problem, i.e., $O|p_{ij} = (w_{ij}/res_{ij})^k, \sum_{i=1}^{2}\sum_{j=1}^{n} res_{ij} \leq R|C_{\max}$ is *NP*-hard when $m \geq 3$, and presented a fully polynomial-time approximation scheme (FPTAS) when preemption is allowed. Oulamara, Rebaine, and Serairi (2013) considered the two-machine open-shop in which every job, before its execution, undergoes a preparation phase. This phase requires a number of renewable resources that have capacity limits. Hence, the processing time of a job on each machine consists of two parts, preparation times, denoted by p'_{1j}, p'_{2j}, and execution times, denoted by p''_{1j}, p''_{2j}. It is evident that preparation always precedes execution. The problem is denoted as $O2|res^{prp}_{...}|C_{\max}$, where *prp* indicates that the resource consumption is related to the preparation phase. They showed that the problem is strongly *NP*-hard, and *NP*-hard for the non-preemptive and preemptive cases, respectively. They also proved that the problem remains *NP*-hard even either the preparation times or the processing times are constant, i.e., $p'_{1j}, p'_{2j} = p'$ or $p''_{1j}, p''_{2j} = p''$. However, if both the preparation and processing times are constant, the problem is polynomially solvable by a reduction to perfect matching in a specific bipartite graph.

Certain jobs cannot be processed simultaneously on different machines. This is referred to as the open-shop with conflict graphs (OSC) and the jobs that cannot be processed simultaneously are presented by adjacent vertices in the conflict graph. Tellache and Boudhar (2017) showed that OSC is equivalent to problem $O|res^{idnt}_{.11}|C_{\max}$, where *idnt* denotes that all the operations of a job have identical resource requirements. They proposed a heuristic algorithm for the general OSC problem. Also, they showed that the two-machine OSC problem is strongly *NP*-hard if $p_{1j} = p_{2j} = p_j$, $p_j \in \{1, 2, 3\}$, and the conflict graph is the complement of the bipartite graph. This implies that $O2|res^{idnt}_{.11}, p_{ij} \in \{1, 2, 3\}|C_{\max}$ is also strongly *NP*-hard. They then proved that the three-machine OSC problem is strongly *NP*-hard for any arbitrary conflict graph even with UET operations, which leads to the strong *NP*-hardness of $O3|res^{idnt}_{.11}, UET|C_{\max}$. They also identified some polynomially solvable cases: (1) the two-machine OSC problem with any arbitrary conflict graph is solvable in $O(n^{2.5})$ time if $p_{ij} \in \{0, 1, 2\}$, which results in the same complexity status for problem $O2|res^{idnt}_{.11}, p_{ij} \in \{0, k, 2k\}|C_{\max}$ for any fixed given in-

Table 7
Summary of the major studies with resource constraints.

Problem	Complexity and solution method	Reference
$O2\|res_{\cdots}, UET\|C_{\max}$	$O(n^{2.5})$	Blazewicz et al. (1983)
$O2\|res_{111}, chain, UET\|C_{\max}$	Strongly *NP*-hard	Blazewicz et al. (1986)
$O3\|res_{1..}, UET\|C_{\max}$	Strongly *NP*-hard	Blazewicz et al. (1986)
$O3\|res_{.11}, UET\|C_{\max}$	Strongly *NP*-hard	Blazewicz et al. (1986)
$O\|p_{ij} = \{0,1\}, nr\ staircase\|C_{\max}$	Strongly *NP*-hard	Cochand et al. (1989); de Werra et al. (1991)
$O\|UET, nr\ staircase\|C_{\max}$	$O(mn)$	Cochand et al. (1989); de Werra et al. (1991)
$O2\|res_{1..}; prmp\|C_{\max}$	$O(n^3)$	Jurisch and Kubiak (1997)
$O2\|res_{1..}\|C_{\max}$	$O(n^3)$	Jurisch and Kubiak (1997)
$O2\|res_{122}\|C_{\max}$	*NP*-hard	Jurisch and Kubiak (1997)
$O2\|res_{.11}\|C_{\max}$	Strongly *NP*-hard	Jurisch and Kubiak (1997)
$O2\|UET, tres_{.11}\|C_{\max}$	Strongly *NP*-hard	Tautenhahn and Woeginger (1997)
$O2\|UET, tres_{1..}\|C_{\max}$	Strongly *NP*-hard	Tautenhahn and Woeginger (1997)
$O\|UET, tres_{1.1}\|C_{\max}$	Strongly *NP*-hard	Tautenhahn and Woeginger (1997)
$O2\|p_{ij} = (w_{ij}/res_{ij})^k, \sum_{i=1}^{2}\sum_{j=1}^{n} res_{ij} \le R\|C_{\max}$	$O(n \log n)$	Shabtay and Kaspi (2006)
$O3\|p_{ij} = (w_{ij}/res_{ij})^k, \sum_{i=1}^{2}\sum_{j=1}^{n} res_{ij} \le R\|C_{\max}$	*NP*-hard	Shabtay and Kaspi (2006)
$O\|p_{ij} = (w_{ij}/res_{ij})^k, \sum_{i=1}^{2}\sum_{j=1}^{n} res_{ij} \le R, prmp\|C_{\max}$	FPTAS	Shabtay and Kaspi (2006)
$O2\|res^{prp}_{\cdots}\|C_{\max}$	Strongly *NP*-hard	Oulamara et al. (2013)
$O2\|res^{prp}_{\cdots}; prmp\|C_{\max}$	*NP*-hard	Oulamara et al. (2013)
$O2\|res^{idnt}_{.11}, p_{ij} \in \{1,2,3\}\|C_{\max}$	Strongly *NP*-hard	Tellache and Boudhar (2017)
$O3\|res^{idnt}_{.11}, UET\|C_{\max}$	Strongly *NP*-hard	Tellache and Boudhar (2017)

teger k, (2) the preemptive two-machine OSC problem is solvable in $O(n^3)$ time for any arbitrary conflict graph, and (3) the three-machine OSC problem with UET operations is solvable in $O(n^{2.5})$ time if the conflict graph is a complement of a triangle-free graph. A similar problem to OSC is the open-shop problem with agreement graphs, where the jobs that can be processed simultaneously are presented by adjacent vertices in the agreement graph. Tellache, Boudhar, and Yalaoui (2019) proved that the two-machine open-shop problem with proportionate processing times on certain types of agreement graphs is strongly NP-hard. On arbitrary graphs, they showed that the problem can be solved in $O(n^3)$ if all non-zero processing times on one of the machines are equal to 1. We summarize those reviewed studies in Table 7.

3.4. Transfer resource

The basic assumption for many scheduling problems is that as soon as a job is completed on a machine, it can be instantly processed by the next machine. In real-world practice, however, there might be a "time lag" or "delay" between the completion time of a job on the preceding machine and the start time of the job on the succeeding machine. The delay is typically incurred due to the transport time between the two machines that may need a set of transporters. The delay may also be caused by processing operations that are not in the form of machines, such as cooling procedures. In this section, we review different approaches that have been proposed to study such environments by grouping them into interstage delay and transport delay. The former studies concern delay due to technological reasons, while the latter studies deal with delay due to transporting jobs between machines. We then present the results for interstage resource, where the transporter moves the jobs between the processing machines.

3.4.1. Interstage delay

Delay can be due to technological reasons. For example, a cooling process may cause the delay. In the open-shop setting, such a delay has been characterized as either minimal or exact. The minimal delay is where the time lag is at least equal to the amount of the delay. We denote a minimum amount of delay by $\bar{L}$. The exact delay describes the situation where a succeeding operation should be started after its preceding operation, plus an exact amount of time. This case is commonly referred to as the coupled task scheduling problem (Khatami, Salehipour, & Cheng, 2020). We let L present an exact amount of delay. Also, the delay is "symmetric" if it has the same value from machine i to machine i' and from machine i' to machine i. For simplicity, for a two-machine problem with job-independent delay, we let $\tau = \bar{L}_{12}$ and $\sigma = \bar{L}_{21}$. If all the delays are identical, it is called the uniform delay.

Rayward-Smith and Rebaine (1992) investigated the non-preemptive open-shop scheduling problem with minimal delay. They proved that the two-machine problem is *NP*-hard even for the uniform delay. The problem with symmetric delay and an arbitrary number of machines is *NP*-hard, even for UET operations. However, the problem with uniform delay, an arbitrary number of machines and UET operations is solvable in polynomial time. It is also known that the non-preemptive two-machine problem with symmetric delay is strongly *NP*-hard even if the processing time of each job is identical on both machines (Dell'Amico & Vaessens, 1996). Also, the problem remains strongly *NP*-hard for minimal symmetric delay, even with UET jobs (Yu, Hoogeveen, & Lenstra, 2004).

Rebaine and Strusevich (1999) studied the two-machine problem with machine-dependent but job-independent delay, denoted by $O2|\tau,\sigma|C_{\max}$. They developed an $\frac{8}{5}$-approximation algorithm, which achieves the ratio of $\frac{3}{2}$ if $\tau = \sigma$. They showed that both bounds are tight. Brucker, Knust, Cheng, and Shakhlevich (2004) showed that the problem is solvable in constant time if all the operations have the same processing time, i.e., $p_{ij} = p$. By extending the algorithm by Pinedo and Schrage (1982), they also proposed an $O(n)$-time algorithm for $O2|\tau_j = \sigma_j, \max\{\tau_j\} \le p_{\min}, n \ge 6|C_{\max}$, where $p_{\min} > 0$ is the smallest processing time of all the operation. Later, Strusevich (1999) investigated the general job-dependent case of the problem, i.e., $O2|\tau_j = \sigma_j|C_{\max}$, and presented an $O(n \log n)$-time $\frac{3}{2}$-approximation algorithm and showed that the bound is tight. Munier-Kordon and Rebaine (2010) proposed an $O(n \log n)$-time algorithm for two solvable cases of the problem with minimal integral delay and UET operations: All distinct delays and only two distinct delays. They also presented two heuristics with an approximation ratio of $\frac{3}{2} - \frac{1}{2n}$ and an asymptotic worst-case ratio of $\frac{5}{4}$, respectively.

Ageev (2018) considered the problem with exact delay. They showed that the two-machine problem is *NP*-hard because the problem with exact delay is indeed a generalization of the no-wait environment. We note that the two-machine no-wait open-shop scheduling problem is *NP*-hard (Giaro, 2001). Ageev (2018) also proved that the existence of a $(1.5 - \varepsilon)$-approximation algorithm for the special case of $p_{1j} = p_{2j}$ implies $P = NP$ for any $\varepsilon > 0$.

Table 8
Summary of results concerning interstage delay.

Problem	Complexity and solution method	Reference
$O2\|UET, \tau_j = \sigma_j\|C_{\max}$	Strongly NP-hard	Yu et al. (2004)
$O2\|p_{1j} = p_{2j}, \tau_j = \sigma_j\|C_{\max}$	Strongly NP-hard	Dell'Amico and Vaessens (1996)
$O2\|\tau_j = \sigma_j\|C_{\max}$	$\frac{3}{2}$-approximation algorithm	Strusevich (1999)
$O2\|UET, \tau_j = \sigma_j\|C_{\max}$	$\frac{3}{2} - \frac{1}{2n}$ and $\frac{5}{4}$-approximation algorithms	Munier-Kordon and Rebaine (2010)
	$O(n\log n)$ if all delays are distinct, and pseudo-polynomial if $\bar{L}_j \in \{\bar{L}_1, \bar{L}_2\}$	Munier-Kordon and Rebaine (2010)
$O2\|\tau = \sigma\|C_{\max}$	NP-hard, remains NP-hard even if the processing times on one of the machines are all zero (Strusevich, 1999)	Rayward-Smith and Rebaine (1992)
	$\frac{3}{2}$-approximation algorithm	Rebaine and Strusevich (1999)
$O2\|\tau, \sigma\|C_{\max}$	$\frac{8}{5}$-approximation algorithm	Rebaine and Strusevich (1999)
$O2\|p_{ij} = p, \tau, \sigma\|C_{\max}$	$O(1)$	Brucker et al. (2004)
$O2\|\tau_j = \sigma_j, \max\{\tau_j\} \leq p_{\min}, n \geq 6\|C_{\max}$	$O(n)$	Rebaine and Strusevich (1999)
$O\|UET, \bar{L}_{ii'} = \bar{L}_{i'i}\|C_{\max}$	NP-hard	Rayward-Smith and Rebaine (1992)
$O\|UET, \bar{L}_{ii'} = \bar{L}_{i'i} = \bar{L}\|C_{\max}$	Polynomial	Rayward-Smith and Rebaine (1992)
$O2\|L_j\|C_{\max}$	$(1.5 - \varepsilon)$-approximation algorithm means $P = NP$	Ageev (2018)

When the delay takes only two values, they showed that there is no approximation algorithm with a ratio better than $(1.25 - \varepsilon)$ for any $\varepsilon > 0$, unless $P = NP$. Table 8 summarizes the major studies considering interstage delay.

3.4.2. Transport delay

Delays may also be caused by transporting jobs between machines. The routing open-shop scheduling problem (Averbakh, Berman, & Chernykh, 2005; 2006) includes the transport time, where the jobs are located at the nodes of an undirected transport network represented by graph $G = (V, E)$ with the set V of nodes and set E of edges, and the machines travel, typically at a unit speed, between the jobs. So, not only the processing times of the operations but also the travel times between the jobs should be considered. Starting at the depot the machines will return to the depot after finishing all the jobs. The problem is denoted as $RO\|\|C_{\max}$. The makespan is calculated as the time span between the start time of processing or moving of machines and the returning time of the last machine to the depot after performing all of its operations.

In a two-node network of two machines and n jobs, where more than one job can be located at a vertex, Averbakh et al. (2005) proposed a linear-time approximation algorithm with the worst-case performance ratio of $\frac{6}{5}$. Chernykh and Pyatkin (2017) presented certain specific results regarding the approximation algorithm of Averbakh et al. (2005). For example, they determined how the maximal ratio of the optimal makespan to a standard lower bound depends on the jobs' load distribution between the two nodes. They also presented an approximation algorithm with the same worst case as that by Averbakh et al. (2005); however, the worst case is completely specific to the load distribution. Later, Averbakh, Berman, and Chernykh (2006) showed that problem $RO2\|\|V\| = 2\|C_{\max}$ (in a two-node network) is NP-hard both for two machines, and for two jobs and m machines. By excluding the assumption concerning the number of nodes, they proposed a $(1 + \frac{\rho}{2})$-approximation and an $(\frac{(m+1)}{2} + \rho)$-approximation algorithm for the two-machine and m machine variants, respectively, where a ρ-approximate solution for TSP, $\rho \leq 2$, is given. Those approximation ratios were later improved by Yu, Liu, Wang, and Fan (2011) to $\max\{\lceil\frac{m}{2}\rceil, \frac{4}{3}\rho\} + \frac{1}{3}$ and $\max\{\frac{4\rho+3}{2\rho+3}, \frac{2\rho+2}{3}\}$. Kononov (2012) proposed an FPTAS for $RO2\|\|V\| = 2\|C_{\max}$, and also introduced conditions under which the problem is solvable in linear time. Pyatkin and Chernykh (2012) showed that $RO2\|prmp, \|V\| = 2\|C_{\max}$ (the preemption variant) is solvable in linear time, whereby that with an arbitrary number of machines is strongly NP-hard. The two-node routing open-shop with UET and unit travel times is conjectured to be polynomially solvable if $m = n$ (Golovachev & Pyatkin, 2019). van Bevern, Pyatkin, and Sevastyanov (2019) proposed the first algorithm with parameterized complexity for UET operations, and Chernykh and Lgotina (2020) and Chernykh and Krivonogova (2020) presented solvable case for the two-machine on a tree and three-machine problems, respectively, including a linear-time $\frac{4}{3}$ approximation algorithm for the three-machine two-node-network problem. A tree is a connected graph that has no cycle.

A number of studies addressed the triangular network, i.e., $\|V\| = 3$. For example, Chernykh and Kuzevanov (2013) presented an $\frac{11}{10}$-approximation algorithm for $RO2\|\|V\| = 3, prmp\|C_{\max}$, and Chernykh and Lgotina (2016) and Chernykh and Lgotina (2019) studied the non-preemptive variant with identical and unrelated travel speeds, and proposed $\frac{6}{5}$ and $\frac{5}{4}$-approximation algorithms, respectively, which run in linear time.

Several studies investigated the variant with an arbitrary number of machines. Chernykh, Kononov, and Sevastyanov (2013) use a $\frac{3}{2}$-approximation algorithm, originally proposed for TSP, and developed a ρ-approximation algorithm for $RO\|\|C_{\max}$, where $\rho = O(\sqrt{m})$, and a $\frac{13}{8}$-approximation algorithm for $RO2\|\|C_{\max}$. By reducing the original problem to the classical flow-shop scheduling problem, Yu and Zhang (2011) improved the results of Chernykh et al. (2013) and presented an $O(\log m(\log\log m)^{1+\varepsilon})$-approximation algorithm for $RO\|\|C_{\max}$. Later, Kononov (2015) showed that there exists an $O(\log m)$-approximation algorithm for $RO\|\|C_{\max}$. van Bevern and Pyatkin (2016) proposed a fixed-parameter algorithm for the problem with UET jobs and showed that it can be solved in $2^{|V||M|^2 \log |V||M|}.poly(|N|)$ time, where M and N are the machine and job sets, respectively.

In the classical parallel-machine scheduling models, machines can have identical, uniform, or unrelated speeds (Pinedo, 2016). Inspired by the classical parallel-machine models, Chernykh (2016) relaxed the assumption of unit speed. Each machine therefore has its own unrelated speed, leading to different travel times for the machines. The problem is denoted as $RO2\|Rtt\|C_{\max}$. He presented an approximation algorithm. Also, he proposed a linear-time algorithm for the case where the transport network is a tree and the depot is not pre-defined, i.e., it has to be chosen. In Yu et al. (2011)'s study, the machines do not return to the depot. The makespan is then the maximum completion time of all the jobs. They proposed an $(\frac{8-3\rho}{5-2\rho})$-approximation algorithm for the two-machine case, where $\rho \leq 2$ is the approximation factor for the shortest Hamiltonian path problem. They also presented a ρ'-approximation algorithm for the m-machine case, where

$$\rho' = \begin{cases} \max\{\lceil\frac{m}{2}\rceil, 2\} + \frac{1}{2}, & m \leq 6, \\ \lceil\frac{m}{2}\rceil + \frac{1}{3}, & m > 6. \end{cases} \quad (2)$$

Table 9
Summary of the results on transport delay.

Problem	Complexity and solution method	Reference
$RO2\|\|V\|=2\|C_{\max}$	NP-hard and $\frac{6}{5}$-approximation algorithm	Averbakh et al. (2006, 2006)
$RO\|\|C_{\max}$	Strongly NP-hard (when neither m nor n is bounded)	Averbakh et al. (2006)
$RO2\|\|V\|=2\|C_{\max}$	FPTAS	Kononov (2012)
$RO2\|prmp,\|V\|=2\|C_{\max}$	$O(n)$	Pyatkin and Chernykh (2012)
$RO\|prmp,\|V\|=2\|C_{\max}$	Strongly NP-hard	Pyatkin and Chernykh (2012)
$RO2\|\|V\|=3, prmp\|C_{\max}$	$\frac{11}{10}$-approximation algorithm	Chernykh and Kuzevanov (2013)
$RO2\|\|V\|=3\|C_{\max}$	$\frac{6}{5}$-approximation algorithm	Chernykh and Lgotina (2016)
$RO2\|\|C_{\max}$	$\frac{13}{8}$-approximation algorithm	Chernykh et al. (2013)
$OR2\|Rtt, tree, variable-depot\|C_{\max}$	$O(n)$	Chernykh (2016)
$RO\|\|V\|=3, n=1, Qtt\|C_{\max}$ (single job routing open-shop with uniform travel times)	NP-hard	Chernykh (2016)
$RO2\|\|V\|=3, Rtt\|C_{\max}$	$\frac{5}{4}$-approximation algorithm	Chernykh and Lgotina (2019)
$RO\|\|C_{\max}$	$O(\log m)$-approximation algorithm	Kononov (2015)

Table 10
Summary of the studies considering interstage transporters.

Problem	Complexity	Reference
$TO2\|\tau,\sigma,k=1,c\geq n\|C_{\max}$	2-approximation algorithm	Lee and Strusevich (2005)
$TO2\|\tau=\sigma,k=1,c\geq n\|C_{\max}$	$\frac{5}{3}$-approximation algorithm	Lee and Strusevich (2005)
$TO2\|k=1,c\geq n\|K$	$\frac{7}{5}$-approximation algorithm	Lushchakova et al. (2009)

It should be noted that the routing open-shop scheduling problem can also be seen as a variant of the open-shop with sequence-dependent families or batch setup times. That is, the travel times of the machines between the nodes can be modelled as the sequence-dependent setup time and, because a node may include several jobs, a family or a batch of jobs is visible. Whether the routing open-shop problem parameterized with number of batches is fixed-parameter tractable is an open question (Mnich & van Bevern, 2018). Table 9 summarizes the major studies considering transport delay.

3.4.3. Interstage resource

The interstage resource is different from the transport resource because in the former the transporter moves the jobs between the processing machines, so the machines are fixed. In the latter, however, the machines move to the jobs to process them, meaning that the jobs are fixed. Lee and Strusevich (2005) studied this problem with an uncapacitated interstage transporter for the two-machine problem. The transporter can carry the jobs between machines 1 and 2, i.e., from machine 1 to 2 and also from machine 2 to 1, where the transport times are denoted by τ and σ, respectively. The problem is then shown by $TO2|k=1,c\geq n|C_{\max}$, where k is the number of transporters and c is the capacity of the transporters. The transporter has indeed infinite capacity because $c\geq n$. They proposed an approximation algorithm with a ratio of 2, which can be improved to $\frac{5}{3}$ if the transport times are symmetric, i.e., $\tau=\sigma$. Lushchakova, Soper, and Strusevich (2009) generalized the problem: The transporter first carries the jobs to one of the machines at the beginning of the schedule and then collects all of them at the end of the schedule. Therefore, the objective function is to minimize the time when all the completed jobs are collected by the transporter, denoted by K, and the problem is denoted by $TO2|\nu=1,c\geq n|K$. They proposed a $\frac{7}{5}$-approximation algorithm and showed that the bound is tight. Table 10 summarizes the major studies considering interstage transporters.

3.5. No-wait, no-idle, and blocking open-shops

In this section we review studies that consider no-wait, no-idle, and blocking in the open-shop scheduling problem. No-wait means that the processing of a job/operation must immediately start after completion of the job on the preceding machine. In addition, if each machine processes the operations with no idle time, then the problem is denoted as no-wait no-idle. Blocking occurs when there is no (or limited) intermediate storage between the machines.

The two-machine preemptive no-wait open-shop scheduling problem to minimize the makespan, denoted as $O2|prmp,no-wait|C_{\max}$, has been shown to be strongly NP-hard (Hall & Sriskandarajah, 1996; Strusevich, 1991). The non-preemptive variant, i.e., $O2|no-wait|C_{\max}$, in which the jobs must visit both machines, is also known to be strongly NP-hard (Sahni & Cho, 1979). Gonzalez (1982) later showed that the problem with an arbitrary number of machines, i.e., $O|no-wait|C_{\max}$, is strongly NP-hard even if all the non-zero operations have identical positive processing times. They proved that for the two-machine case if all the jobs go through both machines and all the operations are identical, the problem is, however, trivial. Brucker et al. (1993) showed that the no-wait UET open-shop can be transformed to the m identical parallel-machine problem, and used the transformation to prove the NP-hardness of $O|UET,prec,no-wait|C_{\max}$. They showed that the following problems are polynomially solvable: $O|UET,r_j,no-wait|C_{\max}$, $O|UET,tree,no-wait|C_{\max}$, and $O2|UET,prec,no-wait|C_{\max}$. Sidney and Sriskandarajah (1999) showed that the algorithm by Gilmore and Gomory (1964), which optimally solves the two-machine no-wait flow-shop problem in $O(n\log n)$ time, has a tight bound of $\frac{3}{2}$ for the two-machine open-shop problem. Panwalkar and Koulamas (2014) studied the proportionate processing times in the two-machine problem. They showed that $O2|p_{ij}=p_j,no-wait|C_{\max}$ and $O2|p_{ij}=p_j+s_i,no-wait|C_{\max}$ can be solved in $O(n\log n)$ time, where s_i is a machine-specific setup time. In addition, they significantly improved the exhaustive enumeration efforts required for problems $O2|no-wait|C_{\max}$ and $O2|p_{ij}=\frac{p_j}{v_i},no-wait|C_{\max}$, where $v_i\geq 1$ is the speed of machine i.

In regard to producing schedules, the B&B algorithm by Liaw, Cheng, and Chen (2005) optimally solves small instances of the two-machine no-wait problem. Their heuristic algorithm that has an $O(n^3)$ running time is able to solve larger instances. Naderi and Zandieh (2014) proposed three mixed-integer linear programs for the no-wait problem. To overcome the limitation of the well-known permutation and rank matrix encodings for the open-shop scheduling problem, which likely generate infeasible solutions for the no-wait problem, they proposed a new encoding scheme. They also proposed the meta-heuristic algorithms of VNS and GA.

Recall that in the no-wait no-idle environment, each job is processed with no delay between its operations and each machine processes the operations with no idle time. The no-wait no-idle is also known as “compact” scheduling. Giaro, Rubale, and Malafiejski (1999) studied the no-wait no-idle open-shop with zero-one time operations. They showed that the problem is equivalent to the strongly NP-hard colouring problem in a bipartite graph. They also proved that there is no approximation algorithm for the problem with a ratio better than $\frac{6}{5}$, unless $P = NP$. Giaro (2001) showed that the decision problem as to whether a compact schedule exists for the two-machine open-shop is strongly NP-complete. They proved that the no-wait no-idle open-shop with an arbitrary number of machines is NP-hard when the scheduling graph is a path or cycle. Note that the scheduling graph is constructed as a simple bipartite graph where the jobs and machines represent the vertices of the two partitions, and the operations are represented by the edges connecting the jobs and machines. Also, the no-wait no-idle open-shop with two machines on a tree, i.e., the scheduling graph is a tree, is NP-hard, and that with an arbitrary number of machines is strongly NP-hard (Giaro, 2001). A linear-time 2-approximation algorithm has been presented for the latter problem (Giaro, 2001). The general no-wait no-idle open-shop scheduling problem with two machines is known to be strongly NP-hard (Billaut, Della Croce, Salassa, & Tkindt, 2019).

In the blocking open-shop scheduling problem, there is no intermediate storage between the machines. Yao, Soewandi, and Elmaghraby (2000) argued that the two-machine blocking open-shop is essentially the two-machine no-wait open-shop, because a schedule for the former can be converted to a schedule for the latter with the same makespan. They evaluated the performance of several heuristics for the two-machine blocking open-shop problem: The algorithm by Sidney and Sriskandarajah (1999), two heuristics for the matching problem, and a random search algorithm. They concluded that the random search heuristic outperforms the others. Meja, Caballero-Villalobos, and Montoya (2018) studied the open-shop setting in which there are no buffers between the machines, so blocking may occur. They modelled the problem as a Petri net, which is a directed bipartite graph widely used to model discrete-event systems, and proposed a graph search algorithm. The graph search algorithm systematically explores the edges of the graph in order to find the minimum path. Table 11 summarizes the major studies considering the no-wait, no-idle, and blocking settings.

4. Non-classical job settings in the open-shop scheduling problem

After reviewing the non-classical resource settings in Section 3, in this section we review the open-shop scheduling problem in the non-classical settings and with constraints on the jobs, e.g., the precedence constraint, and assumptions on the jobs' release and processing times. We also review the problems with job batching and rejection.

4.1. Precedence constraint

The precedence constraint stipulates that a job or an operation can be executed only if all of its predecessors have already completed their execution. The open-shop scheduling problem with UET operations and the precedence constraint has been studied in the literature. For example, Bräsel, Kluge, and Werner (1994) presented an $O(mn)$ algorithm for the open-shop problem with UET jobs (denoted as $O|UET, tree|C_{\max}$) where the precedence relations among the jobs follow a tree, i.e., if job j precedes job j', the last operation of job j must be completed before the first operation of job j'. They also presented a polynomial-time algorithm to solve the problem where the precedence constraints follow a general (arbitrary) form (denoted as $O2|UET, prec|C_{\max}$) (Bräsel, Kluge, & Werner, 1996). Coffman and Timkovsky (2002) studied the “ideal” schedules in which the objective includes simultaneously minimizing the maximum and total completion time (or flow time) of the jobs. For the two-machine case, they showed that the ideal schedule for $O2|prec, r_j, UET|C_{\max}, \sum C_j$ and $O2|no-wait, prec, r_j, UET|C_{\max}, \sum C_j$ can be obtained polynomially by an extension of Coffman Jr and Graham (1972)'s algorithm. Chen, Goebel, Lin, Su, and Zhang (2020) presented an $O(n^2)$-time $(2 - \frac{2}{m})$-approximation algorithm for the m-machine open-shop with UET jobs and a general precedence constraint, i.e., $Om|prec, UET|C_{\max}, m \geq 3$. Nonetheless, the complexity of the problem is still open even for $m = 3$ (see, e.g., Prot & Bellenguez-Morineau, 2018).

Shafransky and Strusevich (1998) considered the precedence constraint in the form of a given sequence of jobs' operations on only one of the machines (denoted as $GS(1)$ in Table 12). They showed that even though the preemptive case of the problem is solvable in $O(n)$ time for an arbitrary number of machines, the general case is strongly NP-hard for an arbitrary number of machines. They proved, however, that it is ordinary NP-hard if there are two machines, and proposed a pseudo-polynomial time dynamic program and used it to construct an FPTAS. They also designed a heuristic algorithm with a worst ratio of $\frac{5}{4}$. Table 12 summarizes the major studies on the open-shop scheduling problem with the precedence constraint.

4.2. Batch processing and setup time

There are a number of studies that consider processing the jobs in batches, setup time, and their combination, which we review in this section.

Strusevich (1993) studied the two-machine open-shop scheduling problem where each operation has three stages, namely setup, processing, and removal. The setup stage of an operation can only be started if the removal stage of the preceding operation is completed. The processing stages of a job cannot be performed simultaneously; however, the other stages can overlap. They proposed an algorithm based on the one by Gonzalez and Sahni (1976) that solves the problem in $O(n)$ time. In the two-machine open-shop problem considered by Glass, Shafransky, and Strusevich (2000), the setup stages of the operations are performed by a single server that is different from the processing machines. The problem is denoted as $O2, S||C_{\max}$ and they proved that it is strongly NP-hard. They showed that even the no-wait version of the problem is NP-hard. Mosheiov and Oron (2008) studied the m-machine open-shop with identical processing times and identical setup times, i.e., $p_{ij} = p, s_{ij} = s, \forall j \in N, \forall i \in M$, to minimize the makespan and the flow time. They proposed an $O(n)$-time heuristic for the former problem and a constant time algorithm based on the number of batches for the latter problem. Roshanaei, Esfehani, and Zandieh (2010) considered the general sequence-dependent setup time in the m-machine setting. They adapted two dispatching rules-based heuristics provided by Pinedo (2016) and Liaw (1998). They also proposed several heuristics, including a multi-neighbourhood search SA algorithm, and hybridization of SA and local search. Their multi-neighborhood search SA performs better than pure SA, VNS, and the GA of Senthilkumar and Shahabudeen (2006). The setup times and travel times between the machines were investigated in an open-shop setting proposed by Mejía and Yuraszeck (2020). The solution method includes a VNS meta-heuristic algorithm.

Table 11
Summary of the studies considering the no-wait, no-idle, and blocking settings.

Problem	Complexity and solution method	Reference
$O2\|no-wait\|C_{\max}$	Strongly *NP*-hard	Hall and Sriskandarajah (1996); Strusevich (1991)
$O2\|no-wait\|C_{\max}$	Strongly *NP*-hard when jobs must visit both machines	Sahni and Cho (1979)
	The algorithm by Gilmore and Gomory (1964) has a tight bound of $\frac{3}{2}$	Sidney and Sriskandarajah (1999)
	Heuristics	Yao et al. (2000)
	B&B	Liaw et al. (2005)
$O2\|no-wait\|C_{\max}$	Polynomial when all non-zero operations have equal length and jobs must visit both machines	Gonzalez (1982)
$O2\|p_{ij}=p_j, no-wait\|C_{\max}$	$O(n\log n)$, result holds for the case of $p_{ij}=p_j+s_i$ as well	Panwalkar and Koulamas (2014)
$O\|no-wait\|C_{\max}$	Strongly *NP*-hard even if all non-zero operations have equal length	Gonzalez (1982)
	MILP, VNS and GA	Naderi and Zandieh (2014)
$O\|UET, prec, no-wait\|C_{\max}$	*NP*-hard	Brucker et al. (1993)
$O\|UET, r_j, no-wait\|C_{\max}$	Polynomial	Brucker et al. (1993)
$O\|UET, tree, no-wait\|C_{\max}$	Polynomial	Brucker et al. (1993)
$O2\|UET, prec, no-wait\|C_{\max}$	Polynomial	Brucker et al. (1993)
$O2\|no-idle, no-wait\|C_{\max}$	Ordinary *NP*-hard if the scheduling graph is a tree, 2-approximation algorithm	Giaro (2001)
	Strongly *NP*-hard	Billaut et al. (2019)
$O\|p_{ij}\in\{0,1\}, no-idle, no-wait\|C_{\max}$	Strongly *NP*-hard, inapproximable within factor of $\frac{6}{5}$ unless $P=NP$	Giaro et al. (1999)
$O\|no-idle, no-wait\|C_{\max}$	Strongly *NP*-hard if the scheduling graph is a tree, 2-approximation algorithm, *NP*-hard when the scheduling graph is a path or cycle	Giaro (2001)
$O2, S\|no-wait\|C_{\max}$	*NP*-hard	Glass et al. (2000)

Table 12
Summary of results for problems with the precedence constraint.

Problem	Complexity and solution method	Reference
$O\|UET, tree\|C_{\max}$	$O(nm)$	Bräsel et al. (1994)
$O2\|UET, prec\|C_{\max}$	Polynomial	Bräsel et al. (1996)
$O2\|prec, r_j, UET\|C_{\max}\sum C_j$	Polynomial	Coffman and Timkovsky (2002)
$O2\|nowait, prec, r_j, UET\|C_{\max}\sum C_j$	Polynomial	Coffman and Timkovsky (2002)
$Om\|prec, UET\|C_{\max}, m\geq 3$	$O(n^2)$-time $(2-\frac{2}{m})$-approximation algorithm	Chen et al. (2020)
$O2\|GS(1)\|C_{\max}$	Ordinary *NP*-hard, $\frac{5}{4}$-approximation algorithm, FPTAS	Shafransky and Strusevich (1998)
$O\|GS(1)\|C_{\max}$	Strongly *NP*-hard	Shafransky and Strusevich (1998)
$O\|GS(1), prmp\|C_{\max}$	Polynomially solvable in $O(n)$	Shafransky and Strusevich (1998)

Glass, Potts, and Strusevich (2001) investigated batching and sequencing in the two-machine open-shop. Batching refers to forming batches of jobs to be processed together, while sequencing refers to determining the order in which the batches are processed. They proved that there exists an optimal schedule with at most three consistent batches, where the batches are consistent if they are identical on the two machines. They also showed that if the optimal schedule consists of a single batch, then the schedule can be easily obtained. In addition, the case with two batches has been shown to be ordinary *NP*-hard because it is pseudo-polynomially solvable, and the case with three batches has been conjectured not to be easier than the case with two batches. However, it is now known that the algorithm by Gribkovskaia, Lee, Strusevich, and de Werra (2006), which is based on the work by de Werra (1989) for the generic two-machine open-shop problem, delivers an optimal schedule in $O(n)$ time if the optimal schedule consists of three consistent batches. To conclude, the problem can be solved in linear time if one or three consistent batches are in the optimal schedule, while it is *NP*-hard if there are two consistent batches. We note that the schedules with more than three batches need not be considered because Glass et al. (2001) has showed that an optimal schedule with at most three consistent batches always exits. Potts, Strusevich, and Tautenhahn (2001) extended the work of Glass et al. (2001) and studied the two-machine open-shop where the processing time of a batch is the maximum of the processing times of the operations in the batch, denoted as "max-batch", and the number of jobs in a batch on the machines is bounded above by k_1 and k_2, respectively. The problem is denoted as $O2|max-batch, b_1=k_1, b_2=k_2|C_{\max}$, where b_i denotes the maximum number of jobs in a batch on machine *i*. They showed that if $k_1=k_2=k$, the problem is ordinary *NP*-hard, as it can be pseudo-polynomially solved by a dynamic program. Khormali, Mirzazadeh, and Faez (2012) considered the parallel-batching constraint and non-identical jobs: The jobs in a batch are processed simultaneously and they have different processing times. They proposed heuristics, including an SA and a GA. Batch scheduling with UET operations and machine-dependent setup times was investigated by Mor, Mosheiov, and Oron (2012). They showed that the equal allocation policy, i.e., assigning an identical number of jobs to the batches is optimal for the two- and three-machine problems. While the equal allocation policy may not necessarily be optimal for problems with more than three machines, the policy still delivers good quality schedules.

In manufacturing systems jobs might be grouped according to their processing similarities and characteristics, and are processed in groups. This modelling is conceptually very similar to batch processing and is called "group technology" (GT). Ben-Arieh and Dror are the first to expand GT to open-shop scheduling (Ben-Arieh & Dror, 1989; 1991). They proposed a two-phase algorithm in which the jobs are first grouped and then scheduled. In the two-machine open-shop with family setup times investigated by Kleinau (1993), the jobs are partitioned into groups and setup

Table 13
Summary of studies with batch processing and setup times.

Problem	Complexity and solution method	Reference
$O2\|batch\ (family)\ setup\|C_{\max}$	NP-hard	Kleinau (1993)
$O2\|batch\ (family)\ setup\|C_{\max}$	$\frac{6}{5}$-approximation algorithm	Billaut et al. (2008)
$O2\|sum-batch\|C_{\max}$	NP-hard, FPTAS	Glass et al. (2001)
$O2\|max-batch, b_1=1, b_2=2\|C_{\max}$	$O(n\log n)$	Potts et al. (2001)
$O2\|max-batch, b_1=1, b_2=k\|C_{\max}$	$O(n^{k(k-1)})$	Potts et al. (2001)
$O2\|max-batch, b_1=k, b_2=n\|C_{\max}$	NP-hard (for fixed k and $k\geq 1$)	Potts et al. (2001)
$O2\|max-batch, b_1=1, b_2=n\|C_{\max}$	$O(n^3\sum_{j=1}^{n} p_{1j})$	Potts et al. (2001)
$O2\|max-batch, b_1=k, b_2=n\|C_{\max}$	$O(n^4k^4(\sum_{j=1}^{n} p_{1j})^2)$	Potts et al. (2001)
$O2\|max-batch, b_1=n, b_2=n\|C_{\max}$	$O(n)$	Potts et al. (2001)
$O2\|s_f=0, p_{1j}=p_{2j}\|C_{\max}$	NP-hard	Blazewicz and Kovalyov (2002)
$O2, S\|\|C_{\max}$	Strongly NP-hard	Glass et al. (2000)
$O2\|G(2,2)\|C_{\max}$	NP-hard	Glass et al. (2001)
$O2\|G(3,3)\|C_{\max}$	$O(n)$	Esswein et al. (2005)

times do not occur between jobs of the same group. However, the setup time arises when a machine switches from processing a job of one group to a job of another group. The problem is denoted as $O2|batch\ setup|C_{\max}$ and was shown to be NP-hard. Kleinau (1993) presented several polynomially solvable cases. Later, Strusevich (2000) proposed a linear-time $\frac{5}{4}$-approximation algorithm for the same problem, and Billaut, Gribkovskaia, and Strusevich (2008) showed that to obtain the optimal solution for the problem, a group should not be split more than once. They applied this observation and designed a heuristic with a worst-case performance ratio of $\frac{6}{5}$ and showed that the bound is tight. Blazewicz and Kovalyov (2002) studied the two-machine with at least two groups such that processing of a group incurs a setup time s_f, which is neither sequence nor machine dependent. They showed that the problem to minimize the makespan is NP-hard even if there is no setup time, and $p_{1j}=p_{2j}, \forall j\in N$, i.e., $O2|s_f=0, p_{1j}=p_{2j}|C_{\max}$. The two-machine open-shop with two conflicting criteria of flexibility and makespan investigated by Esswein, Billaut, and Strusevich (2005) aims to find a schedule with the smallest makespan among the schedules with k consistent sequential groups on each machine. The problem is denoted as $O2|G(k_1,k_2)|C_{\max}$, where $G(k_1,k_2)$ implies that there are at most k_1 (k_2) groups on machine 1 (2). Problem $O2|G(2,2)|C_{\max}$ is NP-hard due to Glass et al. (2001); however, if there exists job j such that $p_{1j}+p_{2j}\geq \max\{l_1,l_2\}$, i.e., the total processing time of job j on both machines is greater than the maximum load of the machines, the linear-time algorithm by de Werra (1989) was shown to be optimal (Esswein et al., 2005). Otherwise, the algorithm produces an optimal schedule for problem $O2|G(3,3)|C_{\max}$. Table 13 summarizes the major studies considering batch processing and setup times.

4.3. Rejection

Rejection occurs when the scheduler is allowed to reject some jobs for execution, typically by incurring a penalty. Hoogeveen, Skutella, and Woeginger (2003) investigated the preemptive open-shop scheduling problem with job rejection. The objective function is to minimize the makespan of the accepted jobs plus the total rejection cost of the rejected jobs, dented as $Rjct$. Even the two-machine problem, i.e., $O2|prmp|C_{\max}+Rjct$, is ordinary NP-hard, but for problem $Om|prmp|C_{\max}+Rjct$, a pseudo-polynomial dynamic program and an FPTAS when the number of machines is fixed exist (Hoogeveen et al., 2003). When the number of machines is part of the input, i.e., problem $O|prmp|C_{\max}+Rjct$, Hoogeveen et al. (2003) showed that it is strongly NP-hard and gave a 1.58-approximation algorithm.

Shabtay, Gaspar, and Kaspi (2013) posed the question as to whether the non-preemptive variant of the problem, i.e., $O2||C_{\max}+Rjct$, is polynomially solvable. To answer the question, Zhang, Lu, and Yuan (2016) showed that it is indeed ordinary NP-hard, even for certain special cases. These special cases include identical processing times on the first (second) machine, i.e., $p_{1j}=p$ ($p_{2j}=p$), and identical rejection cost, i.e., $e_j=e$, where e_j is the rejection cost of job j, and $p>0$ and $e>0$ are constants. They proposed a pseudo-polynomial time dynamic program for the problem and an FPTAS algorithm with $O(\frac{n^6}{\varepsilon^2})$ time complexity. In addition, they showed that the 2-approximation algorithm by Shabtay and Gasper (2012) for the flow-shop setting is also valid for its open-shop counterpart.

Koulamas and Panwalkar (2015) studied the two-machine problem of selecting a subset of jobs with a given cardinality to minimize the makespan. The problem is denoted as $O2|k\ jobs|C_{\max}$. For a given $k=1,\dots,n$, the problem consists of selecting the best subset of k jobs from the set of n jobs. The problem is ordinary NP-hard because the similar problem to minimize the number of tardy jobs is ordinary NP-hard (Józefowska, Jurisch, & Kubiak, 1994). Koulamas and Panwalkar (2015) proposed an $O(n^2)$ algorithm to solve $O2|k\ jobs, M_1=M_{\max}|C_{\max}$, where machine 1 is the maximal. Table 14 summarizes the the major studies considering rejection.

4.4. Release time and on-line scheduling

A realistic assumption for scheduling problems is that all the jobs may not be available at time 0, but at different times known as the release times. The release time of job j is denoted by r_j. Typically, the problem with release times is more challenging than its counterpart without release times. For example, (Graham et al., 1979; Lawler, Lenstra, & Rinnooy Kan, 1981) have shown that the non-preemptive two-machine open-shop with release times is strongly NP-hard.

Cho and Sahni (1981) obtained feasible schedules for the preemptive open-shop scheduling problem with two distinct release times and identical due-dates for all the jobs, denoted as $O|r_j\in\{r_1,r_2\}, d_j=d, prmp|-$. Here, the common deadline d is the time by which all the operations of the jobs must be completed. They transformed the problem to an instance of the network flow problem with upper and lower bounds on the edges, and proposed an $O(n^3+m^4)$ solution algorithm, where $n\geq m$ and $m>2$. Zhan, Zhong, and Zhu (2011) provided another network flow formulation of the preemptive open-shop scheduling problem with release times and applied the maximum flow algorithm to solve it. Two distinct release times in the two-machine preemptive problem was investigated by Lu and Posner (1993): Some jobs have zero release times while the other jobs have a positive common release time. They developed an approach with an average-case complexity being polynomial in the number of jobs. Kubale (1997) studied the zero-one m-machine open-shop problem with integer release times and due-dates, i.e., $O|p_{ij}\in\{0,1\}, r_j, d_j\in\mathbb{Z}^+|C_{\max}$. They showed that the problem is *NP*-hard and proposed polynomial-time algorithms to solve two special cases: All the operations have UET, and at most $m+n$ operations have UET. Considering the feasibility problem of the preemptive open-shop with general release times and deadline, Sedeño-Noda, Alcaide, and González-Martín (2006) proposed an $O(\min\{n, m\}n^2m)$-time network flow algorithm and the first strongly polynomial combinatorial algorithm to solve the feasibility problem, and Pinedo (1995) (see also Pinedo, 2016) showed that the optimization version of the problem can be solved with a linear program utilizing the processing times matrix. Chen, Huang, and Tang (2008) showed that any dense schedule provides a $\frac{7}{4}$-approximation for the three-machine open-shop with release times, i.e., $O3|r_j|C_{\max}$ (as stated in Section 2, the dense schedules are conjectured to have a $2-\frac{1}{m}$ bound for the classical open-shop). They tightened the bound to $\frac{5}{3}$ for the two special cases where the processing times are machine-independent and where each job consists of at most two operations.

On-line scheduling differs from scheduling with release times in the sense that the data of the jobs (e.g., release times, processing times, etc.) are available only when the jobs arrive in the system. The quality of an on-line algorithm is assessed by comparing its solution to the optimal solution of the off-line counterpart, and is often referred to as the "competitive" ratio. An on-line algorithm is called ρ-competitive if the objective value obtained by the on-line algorithm is at most ρ times that of an optimal off-line solution, for any instance of the problem (Zhang & van de Velde, 2010a). The competitive ratio ρ of an on-line algorithm is defined as C_h/C^*, where C_h is the makespan obtained by the on-line algorithm and C^* is the optimal makespan of the off-line problem over all the instances. Bai and Tang (2013) proved that the dense schedules are asymptotically optimal for both the off-line and on-line open-shops with release times if the problem size is large enough. For both the off-line and on-line settings, they developed a heuristic that constructs the schedule based on combining the SPT dispatching rule and the dense schedules. Their experiments showed that such a dispatching rule-based heuristic obtains the optimal schedule when the number of jobs goes to infinity.

Chen and Woeginger (1995) investigated on-line scheduling of the two-machine open-shop in both preemptive and non-preemptive settings. For the preemptive case, they proposed a $\frac{4}{3}$-competitive algorithm and showed that no better ratio exists. For the non-preemptive setting, their algorithm has a worst-case ratio of 1.875. In addition, they showed that no on-line algorithm has a better ratio than $\frac{1}{2}(1+\sqrt{5}) \approx 1.618$. The two- and three-machine variants of the problem were investigated by Chen, Du, Han, and Wen (2001). In the preemptive three-machine setting, they provided an algorithm with a best possible competitive ratio of $\frac{27}{19}$. In the non-preemptive two-machine setting, they only considered permutation schedules, in which the sequences of the jobs on both machines are identical. They improved the results of Chen and Woeginger (1995) and proposed a permutation algorithm with a performance ratio of 1.848 and showed that the ratio of any such algorithm is never less than $\frac{(23-2\sqrt{13})}{9} \approx 1.754$. The study of Liu, Chu, Xu, and Zheng (2010) on-line scheduling of the preemptive two-machine open-shop assumes that the processing times are bounded (although still unknown before the jobs arrive), i.e., $1 \le p_{ij} \le k$ and $k \ge 1$ is a constant. They proposed an on-line algorithm with a competitive ratio of $\frac{5k-1}{4k}$ and showed that it is optimal, implying that no on-line algorithm with a smaller competitive ratio exists.

The on-line open-shop problem can be categorized into the clairvoyant and non-clairvoyant settings (Chen et al., 1998). In the clairvoyant setting, the processing time of a job is known upon its arrival, whereas in the non-clairvoyant setting, this is unknown until the job is fully processed. Chen et al. (1998) presented a $\frac{5}{4}$-approximation algorithm for the preemptive clairvoyant variant, and a greedy algorithm with a worst-case performance ratio of $\frac{3}{2}$ for the non-preemptive clairvoyant variant, and for both preemptive and non-preemptive non-clairvoyant variants. They claimed that generalization of their greedy algorithm to $m \ge 3$ yields a worst-case performance ratio of $2-\frac{1}{m}$ for all of the four aforementioned variants.

The on-line version of the two-machine open-shop problem with minimal delays, i.e., $O2|\bar{L}_j, on-line|C_{\max}$, was studied by Zhang and van de Velde (2010a). They proposed a greedy algorithm that produces non-delay schedules. They proved that the competitive ratio of the greedy algorithm is 2 and the bound is tight, and no on-line delay and non-delay algorithm performs any better. In other words, the greedy algorithm is the best possible choice for the problem. Zhang and van de Velde (2010a) also studied the "semi on-line" version of the problem, where the partial information $\max\{\bar{L}_j\} \le p_{\min}$ is available in advance. They showed that the competitive ratio of the greedy algorithm is $\frac{5}{3}$ and the bound is tight, and no on-line non-delay algorithm has superior performance. In addition, they showed that no on-line delay algorithm has a competitive ratio better than $\sqrt{2}$. Zhang and van de Velde (2010b) proposed the first polynomial-time approximation scheme (PTAS) for this class of problems if $\max\{\bar{L}_j\} \le kp_{\min}$ for any constant $k>0$. Table 15 shows the results for the cases with release times and on-line scheduling.

4.5. Start time

In this section we review studies that consider additional constraints on the start times of the operations. de Werra, Mahadev, and Peled (1993) studied the preemptive open-shop scheduling problem with the additional constraint that some operations must be processed simultaneously through requiring that the start times of such operations are the same. They showed that the decision problem is *NP*-complete. de Werra et al. (1993) and later de Werra and Erschler (1996) proposed polynomial solution algorithms for special variants of the problem. Middendorf (1998) studied the two-machine open-shop scheduling problem with "coordinated" start times, i.e., when one machine starts processing an operation, the other machine either has to be idle at that time or start processing another operation. He showed that if the constraint is imposed on both machines (instead of either of the machines), the problem is polynomially solvable; however, it is *NP*-hard if the constraint is imposed on either of the machines. In addition, he showed that the problem in the no-wait setting is *NP*-hard regardless of whether the constraint is imposed on either of the machines or on both machines. There are situations where a pair of jobs cannot be performed at the same time, for which the disjunctive constraint can be imposed to model the corresponding scheduling problem. We refer the interested reader to Hassan, Kacem, Martin, and Osman (2018) for a branch-and-cut algorithm for the *m*-machine open-shop with disjunctive constraints.

The execution of the jobs can also be overlapping. Here, the operations of a job can be processed simultaneously by more than one machine. That is called "concurrent scheduling". Wagneur and Sriskandarajah (1993) showed that, for any regular objective function, the permutation schedules are dominant in the concurrent setting. They further showed that the schedule is immaterial for the objective function of minimizing the makespan. Grinshpoun et al. (2014, 2017) introduced the partial concurrent open-shop scheduling problem, which models the setting in which only subsets of jobs' operations can be processed simultaneously. They showed that the problem is *NP*-hard even if one UET job exits. They proposed a heuristic algorithm for the general problem. The partial concurrent open-shop problem with integral process-

Table 14
Summary of studies with job rejection.

Problem	Complexity and solution method	Reference
$O2\|prmp\|C_{\max}+Rjct$	Ordinary *NP*-hard, FPTAS	Hoogeveen et al. (2003)
$O\|prmp\|C_{\max}+Rjct$	Strongly *NP*-hard, 1.58-approximation algorithm	Hoogeveen et al. (2003)
$O2\|\|C_{\max}+Rjct$	Ordinary *NP*-hard, FPTAS with $O(\frac{n^6}{\varepsilon^2})$	Zhang et al. (2016)
$O2\|k\ jobs\|C_{\max}$	Ordinary *NP*-hard	Józefowska et al. (1994)
$O2\|k\ jobs, M_1=M_{\max}\|C_{\max}$	$O(n^2)$	Koulamas and Panwalkar (2015)

Table 15
Summary of studies with release times and on-line scheduling.

Problem	Complexity and solution method	Reference
$O2\|r_j\|C_{\max}$	Strongly *NP*-hard	Graham et al. (1979) and Lawler et al. (1981)
$O3\|r_j\|C_{\max}$	$\frac{7}{4}$-approximation	Chen et al. (2008)
$O3\|p_{ij}=p_j, r_j\|C_{\max}$	$\frac{5}{3}$-approximation, result holds for $O3\|r_j\|C_{\max}$ when each job consists of at most 2 operations	Chen et al. (2008)
$O\|p_{ij}\in\{0,1\}, r_j, d_j\|C_{\max}$	*NP*-hard	Kubale (1997)
$O\|UET, r_j, d_j\|C_{\max}$	*NP*-hard, result holds when $p_{ij}\in\{0,1\}$ and the number of UET operations is at most $m+n$	Kubale (1997)
$O2\|r_j\in\{0,r\}, d_j=d, prmp\|C_{\max}$	Polynomial algorithm	Lu and Posner (1993)
$O\|r_j\in\{r_1,r_2\}, d_j=d, prmp\|-$	$O(n^3+m^4)$-algorithm, where $n\geq m$ and $m>2$	Cho and Sahni (1981)
$O\|r_j, d_j, prmp\|-$	$O(\min\{n,m\}n^2m)$-time network flow algorithm	Sedeño-Noda et al. (2006)
$O\|r_j, d_j, prmp\|C_{\max}$	LP	Pinedo (1995)
	Strongly polynomial algorithm	Sedeño-Noda et al. (2006)
$O\|r_j, prmp\|C_{\max}$	MILP	Zhan et al. (2011)
$O\|on-line\|C_{\max}$	1.875-competitive algorithm	Chen and Woeginger (1995)
$O\|on-line, prmp\|C_{\max}$	$\frac{4}{3}$-competitive algorithm	Chen and Woeginger (1995)
$O2\|on-line\|C_{\max}$	1.848-competitive permutation algorithm	Chen et al. (2001)
$O3\|on-line, prmp\|C_{\max}$	1.754-competitive algorithm	Chen et al. (2001)
$O\|on-line, clairvoyant\|C_{\max}$	$\frac{5}{4}$-approximation algorithm	Chen et al. (1998)
$O\|on-line, non-clairvoyant\|C_{\max}$	$\frac{3}{2}$-approximation algorithm, result holds for preemptive case as well, in both clairvoyant and non-clairvoyant settings	Chen et al. (1998)
$O2\|on-line, 1\leq p_{ij}\leq k, prmp\|C_{\max}$	$\frac{5k-1}{4k}$-competitive algorithm	Liu et al. (2010)
$O\|on-line, r_j\|C_{\max}$	Heuristic	Bai and Tang (2013)
$O2\|\bar{L}_j, on-line\|C_{\max}$	2-competitive algorithm, $\frac{5}{3}$-competitive for the semi on-line case with $\max\{\bar{L}_j\}\leq p_{\min}$	Zhang and van de Velde (2010a)
$O2\|\max\{\bar{L}_j\}\leq kp_{\min}, on-line\|C_{\max}$	PTAS	Zhang and van de Velde (2010b)

Table 16
Summary of studies with start time constraints.

Problem	Complexity and solution method	Reference
$O2\|coordinated\ machines\|C_{\max}$	Polynomial, becomes *NP*-hard if only one machine is coordinated	Middendorf (1998)
$O2\|no-wait, coordinated\ machines\|C_{\max}$	*NP*-hard, remains *NP*-hard even if only one machine is coordinated	Middendorf (1998)
$O\|disjunctive\ constraint\|C_{\max}$	ILP, branch-and-cut	Hassan et al. (2018)
$O\|conc\|C_{\max}$	Polynomial	Wagneur and Sriskandarajah (1993)
$O\|UET, n=1, p-conc\|C_{\max}$	*NP*-hard	Grinshpoun et al. (2014)
	Heuristic	Grinshpoun, Ilani, and Shufan (2017)
$O\|UET, p-conc, prmp\|C_{\max}$	*NP*-hard, polynomial if conflict graph is a perfect graph	Ilani et al. (2017)
$O\|UET, p-conc, prmp, res\|C_{\max}$	Optimal with two resources	Ilani et al. (2018)

ing times, UET operations, and preemption at integral time points is showed, by Ilani, Shufan, and Grinshpoun (2017), to be equivalent to the *NP*-hard set-colouring problem (also known as the graph multi-colouring problem, see Caramia & Dell'Olmo, 2001). They investigated the polynomially solvable cases and proposed a constructive heuristic algorithm for the case with uniform jobs, i.e., all the jobs have the same processing time and conflict graph. The preemptive variant of the original problem with limited resources was studied by Ilani, Grinshpoun, and Shufan (2018). They proposed an algorithm that is optimal for some special cases that consist of two resources.

Table 16 shows the results for cases with start time constraints.

4.6. Processing time

Various forms of the processing times for the jobs' operations have been considered in the literature, e.g., known and unknown processing times, as well as variable processing times. In the route-dependent open-shop scheduling problem, the processing times of a job's operations depend on the route on which the job passes through the machines. Adiri and Amit (1983) investigated the two-machine route-dependent open-shop scheduling problem, denoted as $O2|RD|C_{\max}$, where *RD* stands for route dependency. They showed that the problem is ordinary *NP*-hard and, for the case where one machine dominates the other, they proposed an $O(n)$ solution algorithm. Strusevich, Van De Waart, and Dekker (1999) also proposed a $\frac{3}{2}$-approximation algorithm for the problem that runs in $O(n^2)$ and showed that the bound is tight.

The majority of scheduling studies assume that the job processing or operation times are constant and known *a priori*. However, there are studies that consider scheduling with variable processing or operation times. For example, Kononov and Gawiejnowicz (2001) studied the two- and three-machine open-shop with both simple and general linear deteriorating jobs. Here, the processing time of a job depends on its start time. They showed that the two-machine problem with general linear deterioration, i.e.,

$O2|p_{ij} = \alpha_{ij} + \beta_{ij}t|C_{\max}$, is *NP*-hard, where t is the job's start time, and α_{ij} and β_{ij} are the basic processing time and the deterioration rate of job j on machine i. They also showed that the three-machine problem with simple linear deterioration, i.e., $p_{ij} = \beta_{ij}t$ is ordinary *NP*-hard even if the jobs have equal deterioration rates on one of the machines. Mosheiov (2002) and Li (2011) also studied simple linear deterioration. Mosheiov (2002) proposed an $O(n)$ algorithm for $O2|p_{ij} = \beta_{ij}t|C_{\max}$ and showed that the problem with three or more machines is *NP*-hard. Li (2011) studied a different variant of the two-machine open-shop with simple linear deterioration, where one of the machines is a non-bottleneck and has infinite capacity. The problem is denoted as $O2|NB, p_{ij} = \beta_{ij}t|C_{\max}$, where *NB* indicates one of the machines is a non-bottleneck. They proposed an $O(n^2 \sum_{j=1}^{n} \log(1+\beta_j)/\varepsilon)$-time FPTAS, where β_j is the deteriorating rate of job j on the machine with finite capacity.

The uncertainty of the processing times can be addressed by stochastic models, in which random variables are used to model the random attributes such as processing times, due-dates, machine breakdowns etc. The scheduler aims to determine the order to process the jobs on the machines so that the expected value of a performance criterion, e.g., makespan, is optimized. In this context, the priority rule used by the scheduler to generate a sequence (order) of the jobs for processing on the machines is called a "policy". A policy is classified into "static" and "dynamic", based on the amount of information available to the scheduler. If the list of the jobs is provided at the beginning of the planning horizon, the policy is referred to as static, whereas it is called dynamic if new information is available at any moment and the scheduler may then update the remaining schedule accordingly. Several studies have attempted to find the optimal dynamic policy for minimizing the expected makespan. Here, almost all studies consider the two-machine open-shop. Emmons (1973) was the first to study the two-machine open-shop with identical jobs, where the processing time of each job's operation is an independent and identically distributed (iid) random variable that follows the exponential distribution with parameters λ and μ, due to the different speeds of the machines. They considered the policy under which the scheduler is not allowed to preempt and always gives priority to the jobs that have not yet received processing on either machine, which is known as the "longest expected remaining processing time" (LERPT) first policy. By using LERPT, they obtained a closed form expression for the expected makespan. An important and popular policy, LERPT has attracted much research attention. For example, Pinedo and Ross (1982) showed that (1) LERPT minimizes the expected makespan for both the preemptive and non-preemptive two-machine open-shop where the processing times of the jobs on the machines are statistically independent and follow the exponential distribution with the rate μ_j, i.e., $O2|p_{ij} \sim \exp(\mu_j)|E(C_{\max})$, and different machine speeds, (2) LERPT minimizes the expected makespan when preemption is not allowed and the distribution of the processing times is "new better than used" (NBU), i.e., $O2|p_{ij} \sim \Phi_k|E(C_{\max})$, where $\Phi_k(t'+t)/\Phi_k(t) \geq \Phi_k(t), \forall t, t' \geq 0, k = 1, 2$, and (3) LERPT is the optimal policy when the job processing times on both machines are identical following the distribution Φ and preemption is not allowed. LERPT was also shown to minimize the expected makespan for the two-machine open-shop with identical jobs where the job arrivals and processing times follow the Poisson and exponential distributions, respectively (Chung & Mohanty, 1988). Righter (1997) showed that LERPT minimizes the expected makespan in certain preemptive and non-preemptive settings in the two-machine open-shop with the restriction that some jobs are processed on both machines and the remaining ones are processed only on one machine.

Some studies extended the aforementioned models. For example, Pinedo and Weber (1984) provided several bounds on the expected makespan for the two-machine open-shop under various distributions for the job processing times, and Frostig (1991) proved that for the two-machine open-shop where the processing times on a machine are iid random variables with the exponential distribution, the optimal static policy for minimizing the makespan includes assigning an equal number of jobs to be first processed by either machine. For the m-machine open-shop with random processing times, Koryakin (2003) showed that for a fixed m and an increasing number of jobs, if distribution Φ_{ij} satisfies several conditions, then the Sevast'yanov (1995)'s algorithm almost always constructs an optimal schedule. Alcaide, Rodriguez-Gonzalez, and Sicilia (2005) studied a stochastic open-shop subject to random machine breakdowns, where the remaining processing times of the preempted jobs are also random variables. They proposed a heuristic to minimize the expected makespan by solving a sequence of stochastic open-shop problems without breakdowns. At every point in time, the heuristic only takes into account all the available information up to that point.

The concept of "pliability" for jobs, introduced by Knust, Shakhlevich, Waldherr, and Weiß (2019), means that the total processing time of a job's operations is given, while the actual processing time of an operations is a decision variable to be decided by the scheduler. They presented three types of pliability: (1) unrestricted, denoted by *plbl*, under which the processing time of a job's operation must be less than or equal to the total processing time of the job, (2) restricted with a common lower bound, denoted by $plbl(\underline{p})$, under which the processing time of a job's operation must be at least greater than or equal to the given minimum length $\underline{p}$, e.g., the minimum length of an operation must be at least two time units, and (3) restricted, i.e., $plbl(\underline{p}_{ij}, \bar{p}_{ij})$, under which each operation has its own lower and upper bounds $\underline{p}_{ij}$ and $\bar{p}_{ij}$. They analyzed the complexity status of open-shop with unrestricted and restricted pliable jobs and obtained the results reported in Table 17.

5. Future research directions

In this section, we highlight the main open problems, as well as potential areas for future research on the open-shop scheduling problem, which we discussed in detail in the previous sections.

Related to the general structure of the open-shop scheduling problem, we highlight two themes. The first theme refers to the basic assumption that the jobs visit each machine only once. This assumption may not hold in certain real-world applications. For example, the manufacturing setting called the re-entrant shop has jobs visiting certain machines more than once. Even though this setting was first observed in the electronic industry (Graves, Meal, Stefek, & Zeghmi, 1983) and has been extensively studied in the flow-shop and job-shop settings ever since (see, e.g., Pan & Chen, 2005; Wang, Sethi, & van de Velde, 1997), we are not aware of any related study in the open-shop environment. The second theme focuses on the coupled task scheduling problem within the open-shop. Recently, the coupled task scheduling problem was introduced in the open-shop setting (Ageev, 2018) and, as such, it is still in the early stage of development. Given the recently emerging applications of the coupled task scheduling problem in the health care sector (see, e.g., Condotta and Shakhlevich (2014), Lamé, Jouini, and Stal-Le Cardinal (2016), and Khatami and Salehipour (2020a), and the references therein), we believe further research in this area is worthy.

We detailed the available solution methods for the classical open-shop scheduling problem, and discussed the state-of-the-art methods that can optimally solve all the instances in the three well-known benchmarks in a reasonable amount of time in Section 2.5. It is noted that two benchmarks of Brucker et al. (1997) and Guéret and Prins (1999) include a maximum number of ten jobs

Table 17
Summary of studies with processing time constraints.

Problem	Complexity and solution method	Reference
$O2\|RD\|C_{\max}$	NP-hard	Adiri and Amit (1983)
$O2\|RD,\, p_{1,min} \geq p_{2,max}\|C_{\max}$	$O(n)$	Adiri and Amit (1983)
$O2\|RD\|C_{\max}$	$\frac{3}{2}$-approximation algorithm	Strusevich et al. (1999)
$O2\|p_{ij} = \alpha_{ij} + \beta_{ij}t\|C_{\max}$ (linear deterioration)	NP-hard	Kononov and Gawiejnowicz (2001)
$O2\|p_{ij} = \beta_{ij}t\|C_{\max}$ (simple linear deterioration)	$O(n)$	Mosheiov (2002)
$O3\|p_{ij} = \beta_{ij}t,\, \beta_{3j} = \beta\|C_{\max}$	NP-hard	Kononov and Gawiejnowicz (2001)
$O\|p_{ij} = \beta_{ij}t\|C_{\max}$	NP-hard	Mosheiov (2002)
$O2\|NB,\, p_{ij} = \beta_{ij}t\|C_{\max}$	FPTAS	Li (2011)
$O2\|plbl\|C_{\max}$ $(n \leq m)$	$O(n)$	Knust et al. (2019)
$O2\|plbl(\underline{p})\|C_{\max}$ $(n \leq m)$	$O(n)$	Knust et al. (2019)
$O2\|plbl(\underline{p}_{ij}, \bar{p}_{ij})\|C_{\max}$ $(n \leq m)$	$O(n)$	Knust et al. (2019)
$O\|plbl\|C_{\max}$ $(n \leq m)$	$O(n)$	Knust et al. (2019)
$O\|plbl\|C_{\max}$ $(n > m)$	$O(1)$	Knust et al. (2019)
$O\|plbl(\underline{p})\|C_{\max}$ $(n > m)$	$O(1)$	Knust et al. (2019)
$O\|plbl(\underline{p}_{ij}, \bar{p}_{ij})\|C_{\max}$ $(n \leq m)$	Strongly NP-hard	Knust et al. (2019)

and machines only. We therefore believe designing more challenging instances is important. The new challenging instances may also necessitate the development of advanced solution methods.

With regard to the complexity of the problem, while complexity of the three-machine open-shop problem with two operations per job is still unknown (see Section 2.2), the conjecture by Drobouchevitch (2020) that the problem is polynomially solvable by their proposed algorithm for the three-machine problem with blocking is worth investigating. Regarding the approximation algorithms for the classical open-shop with an arbitrary number of machines ($O||C_{\max}$), albeit the greedy algorithm by Bárány and Fiala (1982) provides a worst-case performance ratio of 2, its gap with the inapproximability result of Williamson et al. (1997) is still open. Furthermore, the vexing conjecture that the optimal solution for $O||C_{\max}$ is at most $\frac{3}{2}$ larger than that of its preemption variant, i.e., $O|pmtn|C_{\max}$, remains unsettled, and in the words of Woeginger (2005) "settling this conjecture would mean a major breakthrough in the area". While the strong NP-hardness of $O||C_{\max}$ (m is an input) can be easily deduced from the seminal work of Williamson et al. (1997), the strong NP-hardness or pseudo-polynomial solvability of $Om||C_{\max}$ for any fixed $m \geq 3$ has remained open since 1970s. Indeed, the complexity of $Om||C_{\max},\ m \geq 3$ can also shed lights on the existence of an FPTAS for the same problem, which is still an open question. Finally in spite of all efforts, Chen and Strusevich (1993)'s conjecture for the worst case ratio of greedy algorithm of Bárány and Fiala (1982) for $Om||C_{\max}$, which is $2 - \frac{1}{m}$, remains unsettled for $m \geq 9$. Initiated by Sevast'yanov and Tchernykh (1998), another interesting line of research includes finding the tightest $\xi^* > 0$ such that for any instance I, $C_{\max} \in [LB_0(I), \xi^* LB_0(I)]$. Sevast'yanov and Tchernykh (1998) showed that for $O3||C_{\max}$, $\xi^* = \frac{4}{3}$. For $m > 3$, however, ξ^*, and hence, those intervals are unknown. In addition, the conjecture by Sevast'yanov and Tchernykh (1998) that for any instance of $Om||C_{\max}$ with an odd number of machines m, $C_{\max} \in [LB_0, (\frac{3}{2} - \frac{1}{2m})LB_0]$, and that the bound is tight are still open. Another interesting question, which is still open after almost three decades, is whether a given open-shop sequence is irreducible. The three conjectures by Andresen and Dhamala (2012) that we discussed in Section 2.2 may shed lights on resolving that problem.

Regarding the non-classical resource models, we suggest a number of potential research themes. To the best of our knowledge, there is no published research on the open-shop scheduling problem with machine availability in the semi-resumable setting (see Section 3.1). We also note that despite efforts made by Breit et al. (2001, 2003) to provide constant ratio approximation algorithms for two-machine problem under resumable and non-resumable settings, it is still unknown whether these problems admit PTAS or FPTAS. We do not find any study considering the open-shop scheduling problem with more than two agents (see Section 3.2) either. So there is a research gap concerning multi-agent open-shop scheduling with resource considerations. Conducting research to fill this gap is of both theoretical and practical interests because, in the multi-agent setting, each agent is interested in processing a subset of the jobs, while all the jobs share the same resources and the agents need to compete to optimize their own performance criteria (Agnetis, Billaut, Gawiejnowicz, Pacciarelli, & Soukhal, 2014). The research on renewable and non-renewable resources is very restricted, as we discussed in Section 3.3, and there are problems, particularly with $m > 3$, whose complexity are still open. We refer the interested reader to Tautenhahn and Woeginger (1997) for those open problems.

The available studies on interstage transporters (Section 3.4.1) are limited in many aspects. For example, only two machines were considered. Also, transporters are uncapacitated, whereas capacitated transporters have been considered for other shop scheduling settings. For example, Kise, Shioyama, and Ibaraki (1991) studied the two-machine flow-shop scheduling problem with transporters that only can transport one job at a time. Also, transporters are assumed to be homogeneous (have the same speed). That assumption can be relaxed. Indeed, the heterogeneous transporters were studied for other scheduling settings (Ahmadi-Javid & Hooshangi-Tabrizi, 2017). We also believe that the study of blocking open-shops, which has been dominated by the case of no buffer or no intermediate storage (Section 3.5), can be extended into more realistic cases with limited buffers between consecutive machines.

We identify a number of research avenues in the non-classical job settings. For example, the studies reported in Section 4.2 consider sequence-dependent setup times. We are not aware of any published research considering sequence-independent setup times for makespan minimization. Also, almost all the studies on batch processing deal with only two machines, showing a research gap for problems with more than two machines. The second avenue lies in the area of time-dependent processing times (see Section 4.6). While a few studies consider time-dependent processing times due to job deterioration, there is no study considering time-dependent processing times that follow the learning effect in which the processing time of a job is a decreasing function of its start time. While disruption management has been the subject of many studies in other shop scheduling models, we noticed that there is only little research conducted in the open-shop setting, e.g., there is only one paper devoted to machine breakdown, suggesting therefore a research gap for various machine breakdown settings.

As a concluding remark, most of the efforts for solving the problems reviewed in Sections 3 and 4 are devoted to analyzing the complexity status, and that mainly for problems with a few machines. There are no solution algorithms for large instances of those problems.

6. Concluding remarks

It has been more than 40 years since the open-shop scheduling problem to minimize the makespan was first introduced to the scheduling community. While the early years witnessed only a few studies on this topic, the majority of research has been developed in the last 30 years. In particular, there has been a considerable increase in research publications, both theoretical and practical, in the past few years, suggesting that open-shop scheduling is a thriving area of scheduling research that provides many opportunities for fruitful research. This is further acknowledged by the large number of papers published in the last decade which accounts for 30% of articles in the past 40 years. We believe that our work provides a comprehensive and timely survey of research on open-shop scheduling, which can be a valuable resource for understanding the development trajectory and guiding future research on the topic.

Acknowledgments

Mohammad Mahdi Ahmadian is the recipient of the UTS International Research Scholarship (IRS) and UTS Faculty of Science Scholarship. Mostafa Khatami is the recipient of the UTS International Research Scholarship (IRS) and the UTS President's Scholarship (UTSP). Amir Salehipour is the recipient of an Australian Research Council Discovery Early Career Researcher Award (project number DE170100234) funded by the Australian Government. T.C.E. Cheng was supported in part by The Hong Kong Polytechnic University under the Fung Yiu King - Wing Hang Bank Endowed Professorship in Business Administration.

References

Adak, Z., Akan, M. Ö. A., & Bulkan, S. (2020). Multiprocessor open shop problem: Literature review and future directions. *International Journal of Production Research, 40*, 547–569.

Adams, J., Balas, E., & Zawack, D. (1988). The shifting bottleneck procedure for job shop scheduling. *Management Science, 34*(3), 391–401.

Adiri, I., & Aizikowitz, N. (1989). Open-shop scheduling problems with dominated machines. *Naval Research Logistics (NRL), 36*(3), 273–281.

Adiri, I., & Amit, N. (1983). Route-dependent open-shop scheduling. *IIE Transactions, 15*(3), 231–234.

Adiri, I., & Amit, N. (1984). Openshop and flowshop scheduling to minimize sum of completion times. *Computers & Operations Research, 11*(3), 275–284.

Ageev, A. A. (2018). Inapproximately lower bounds for open shop problems with exact delays. In *Approximation and Online Algorithms* (pp. 45–55). Springer International Publishing AG.

Agnetis, A., Billaut, J.-C., Gawiejnowicz, S., Pacciarelli, D., & Soukhal, A. (2014). *Multiagent scheduling* (10, pp. 973–978). Berlin Heidelberg: Springer.

Ahmadi-Javid, A., & Hooshangi-Tabrizi, P. (2017). Integrating employee timetabling with scheduling of machines and transporters in a job-shop environment: A mathematical formulation and an anarchic society optimization algorithm. *Computers & Operations Research, 84*, 73–91.

Ahmadian, M. M., Salehipour, A., & Kovalyov, M. (2020). An efficient relax-and-solve heuristic for open-shop scheduling problem to minimize total weighted earliness-tardiness. *Available at SSRN 3601396*.

Ahmadizar, F., & Hosseinabadi Farahani, M. (2012). A novel hybrid genetic algorithm for the open shop scheduling problem. *The International Journal of Advanced Manufacturing Technology, 62*(5), 775–787.

Aksjonov, V. (1988). A polynomial-time algorithm of approximate solution of a scheduling problem. *Upravlyaemye Sistemy, 28*, 8–11(in Russian).

Alcaide, D., Rodriguez-Gonzalez, A., & Sicilia, J. (2005). A heuristic approach to minimize expected makespan in open shops subject to stochastic processing times and failures. *International Journal of Flexible Manufacturing Systems, 17*(3), 201–226.

Alcaide, D., Sicilia, J., & Vigo, D. (1997). A tabu search algorithm for the open shop problem. *Top, 5*(2), 283–296.

Anand, E., Panneerselvam, R., et al. (2015). Literature review of open shop scheduling problems. *Intelligent Information Management, 7*(01), 33.

Andresen, M., & Dhamala, T. N. (2012). New algorithms and complexity status of the reducibility problem of sequences in open shop scheduling minimizing the makespan. *Annals of Operations Research, 196*(1), 1–26.

Askin, R. G., Dror, M., & Vakharia, A. J. (1994). Printed circuit board family grouping and component allocation for a multimachine, open-shop assembly cell. *Naval Research Logistics (NRL), 41*(5), 587–608.

Averbakh, I., Berman, O., & Chernykh, I. (2005). A $\frac{6}{5}$-approximation algorithm for the two-machine routing open-shop problem on a two-node network. *European Journal of Operational Research, 166*(1), 3–24.

Averbakh, I., Berman, O., & Chernykh, I. (2006). The routing open-shop problem on a network: complexity and approximation. *European Journal of Operational Research, 173*(2), 531–539.

Azadeh, A., Farahani, M. H., Torabzadeh, S., & Baghersad, M. (2014). Scheduling prioritized patients in emergency department laboratories. *Computer Methods and Programs in Biomedicine, 117*(2), 61–70.

Bai, D., & Tang, L. (2011). Performance analysis of rotation schedule and improved strategy for open shop problem to minimise makespan. *International Journal of Systems Science, 42*(7), 1143–1153.

Bai, D., & Tang, L. (2013). Open shop scheduling problem to minimize makespan with release dates. *Applied Mathematical Modelling, 37*(4), 2008– 2015.

Bai, D., Zhang, Z.-H., & Zhang, Q. (2016). Flexible open shop scheduling problem to minimize makespan. *Computers & Operations Research, 67*, 207–215.

Baptiste, P., Carlier, J., Kononov, A., Queyranne, M., Sevast'yanov, S., & Sviridenko, M. (2009). Structural properties of preemptive schedules. *Discrete Analysis and Operations Research, 16*(1), 3–36.

Baptiste, P., Carlier, J., Kononov, A., Queyranne, M., Sevastyanov, S., & Sviridenko, M. (2011). Properties of optimal schedules in preemptive shop scheduling. *Discrete Applied Mathematics, 159*(5), 272–280.

Bárány, I., & Fiala, T. (1982). Nearly optimum solution of multimachine scheduling problems. *Szigma, 15*(3), 177–191.

Baron, O., Berman, O., Krass, D., & Wang, J. (2016). Strategic idleness and dynamic scheduling in an open-shop service network: Case study and analysis. *Manufacturing & Service Operations Management, 19*(1), 52–71.

Bartholdi, J. J., & McCroan, K. L. (1990). Scheduling interviews for a job fair. *Operations Research, 38*(6), 951–960.

Ben-Arieh, D., & Dror, M. (1989). Group scheduling using group technology concepts. *International Journal of Computer Integrated Manufacturing, 2*(4), 186–193.

Ben-Arieh, D., & Dror, M. (1991). Intelligent heuristic for fms scheduling using grouping. *Journal of Intelligent Manufacturing, 2*(6), 387–395.

van Bevern, R., & Pyatkin, A. V. (2016). Completing partial schedules for open shop with unit processing times and routing. In A. S. Kulikov, & G. J. Woeginger (Eds.), *Computer Science - Theory and Applications* (pp. 73–87). Cham: Springer International Publishing.

van Bevern, R., Pyatkin, A. V., & Sevastyanov, S. V. (2019). An algorithm with parameterized complexity of constructing the optimal schedule for the routing open shop problem with unit execution times. *Siberian Electronic Mathematical Reports, 16*(0), 42–84.

Billaut, J.-C., Della Croce, F., Salassa, F., & Tkindt, V. (2019). No-idle, no-wait: When shop scheduling meets dominoes, Eulerian paths and Hamiltonian paths. *Journal of Scheduling, 22*(1), 59–68.

Billaut, J.-C., Gribkovskaia, I., & Strusevich, V. (2008). An improved approximation algorithm for the two-machine open shop scheduling problem with family setup times. *IIE Transactions, 40*(4), 478–493.

Blazewicz, J., Cellary, W., Slowinski, R., & Weglarz, J. (1986). Scheduling under resource constraints: Deterministic models annals of operations research. *JC Baltzer AG Scientific Publishing Company*.

Blazewicz, J., & Kovalyov, M. Y. (2002). The complexity of two group scheduling problems. *Journal of Scheduling, 5*(6), 477–485.

Blazewicz, J., Lenstra, J. K., & Kan, A. R. (1983). Scheduling subject to resource constraints: classification and complexity. *Discrete Applied Mathematics, 5*(1), 11–24.

Blum, C. (2005). Beam-aco hybridizing ant colony optimization with beam search: An application to open shop scheduling. *Computers & Operations Research, 32*(6), 1565– 1591.

Bouzidi, A., Riffi, M. E., & Barkatou, M. (2019). Cat swarm optimization for solving the open shop scheduling problem. *Journal of Industrial Engineering International, 15*(2), 367–378.

Brandinu, G., & Trautmann, N. (2014). A mixed-integer linear programming approach to the optimization of event-bus schedules: A scheduling application in the tourism sector. *Journal of scheduling, 17*(6), 621–629.

Bräsel, H., Harborth, M., Tautenhahn, T., & Willenius, P. (1999a). On the hardness of the classical job shop problem. *Annals of Operations Research, 92*, 265–279.

Bräsel, H., Harborth, M., Tautenhahn, T., & Willenius, P. (1999b). On the set of solutions of the open shop problem. *Annals of Operations Research, 92*, 241–263.

Bräsel, H., & Kleinau, M. (1992). On the number of feasible schedules of the open-shop-problem-An application of special latin rectangles. *Optimization, 23*(3), 251–260.

Bräsel, H., & Kleinau, M. (1996). New steps in the amazing world of sequences and schedules. *Mathematical Methods of Operations Research, 43*(2), 195–214.

Bräsel, H., Kluge, D., & Werner, F. (1994). A polynomial algorithm for the $[n/m/0, t_{ij} = 1, tree/C_{\max}]$ open shop problem. *European Journal of Operational Research, 72*(1), 125–134.

Bräsel, H., Kluge, D., & Werner, F. (1996). Polynomial time algorithms for special open shop problems with precedence constraints and unit processing times. *RAIRO-Operations Research, 30*(1), 65–79.

Bräsel, H., & Shakhlevich, N. (1998). On the solution region for certain scheduling problems with preemption. *Annals of Operations Research, 83*, 1–22.

Bräsel, H., Tautenhahn, T., & Werner, F. (1993). Constructive heuristic algorithms for the open shop problem. *Computing, 51*(2), 95–110.
Breit, J. (2000). *Heuristische ablaufplanungsverfahren für flowshops und openshops mit beschränkt verfügbaren prozessoren*. Verlag nicht ermittelbar Ph.D. thesis..
Breit, J., Schmidt, G., & Strusevich, V. A. (2001). Two-machine open shop scheduling with an availability constraint. *Operations Research Letters, 29*(2), 65– 77.
Breit, J., Schmidt, G., & Strusevich, V. A. (2003). Non-preemptive two-machine open shop scheduling with non-availability constraints. *Mathematical Methods of Operations Research, 57*(2), 217–234.
Brucker, P., Hilbig, T., & Hurink, J. (1999). A branch and bound algorithm for a single-machine scheduling problem with positive and negative time-lags. *Discrete Applied Mathematics, 94*(1–3), 77–99.
Brucker, P., Hurink, J., Jurisch, B., & Wöstmann, B. (1997). A branch & bound algorithm for the open-shop problem. *Discrete Applied Mathematics, 76*(1–3), 43–59.
Brucker, P., Jurisch, B., & Jurisch, M. (1993). Open shop problems with unit time operations. *Zeitschrift für Operations Research, 37*(1), 59–73.
Brucker, P., Knust, S., Cheng, T. E., & Shakhlevich, N. V. (2004). Complexity results for flow-shop and open-shop scheduling problems with transportation delays. *Annals of Operations Research, 129*(1), 81–106.
Brucker, P., Sotskov, Y. N., & Werner, F. (2007). Complexity of shop-scheduling problems with fixed number of jobs: A survey. *Mathematical Methods of Operations Research, 65*(3), 461–481.
Cankaya, B., Wari, E., & Tokgoz, B. E. (2019). Practical approaches to chemical tanker scheduling in ports: A case study on the port of houston. *Maritime Economics & Logistics*, 1–17.
Caramia, M., & Dell'Olmo, P. (2001). Solving the minimum-weighted coloring problem. *Networks: An International Journal, 38*(2), 88–101.
Carlier, J., & Pinson, É. (1989). An algorithm for solving the job-shop problem. *Management Science, 35*(2), 164–176.
Čepek, O., Vlach, M., & de Werra, D. (1994). Nonpreemptive open shop with restricted processing times. *Zeitschrift für Operations Research, 39*(2), 227–241.
Chen, B., Du, D., Han, J., & Wen, J. (2001). On-line scheduling of small open shops. *Discrete Applied Mathematics, 110*(2), 133– 150.
Chen, B., & Strusevich, V. A. (1993). Approximation algorithms for three-machine open shop scheduling. *ORSA Journal on Computing, 5*(3), 321–326.
Chen, B., Vestjens, A. P., & Woeginger, G. J. (1998). On-line scheduling of two-machine open shops where jobs arrive over time. *Journal of Combinatorial Optimization, 1*(4), 355–365.
Chen, B., & Woeginger, G. J. (1995). *A study of on-line scheduling two-stage shops*. In D.-Z. Du, & P. M. Pardalos (Eds.) (pp. 97–107)). Boston, MA: Springer.
Chen, B., & Yu, W. (2001). How good is a dense shop schedule? *Acta Mathematicae Applicatae Sinica, 17*(1), 121–128.
Chen, R., Huang, W., Men, Z., & Tang, G. (2012). Open-shop dense schedules: Properties and worst-case performance ratio. *Journal of Scheduling, 15*(1), 3–11.
Chen, R., Huang, W., & Tang, G. (2008). Dense open-shop schedules with release times. *Theoretical Computer Science, 407*(1), 389– 399.
Chen, R.-j., & Yu, W.-c. (2003). Analysis of operation chain's properties of dense schedules for open-shop. *Journal of East China University of Science and Technology, 29*(5), 522–526.
Chen, X.-h., & Yu, W.-c. (2000). Upper-bound of performance ratio of dense schedules for open-shop. *Journal of East China University of Science and Technology, 26*(6), 670–673.
Chen, Y., Goebel, R., Lin, G., Su, B., & Zhang, A. (2020). Open-shop scheduling for unit jobs under precedence constraints. *Theoretical Computer Science, 803*, 144–151.
Cheng, T. C. E., & Shakhlevich, N. V. (2007). Two-machine open shop problem with controllable processing times. *Discrete Optimization, 4*(2), 175–184.
Chernykh, I. (2016). Routing open shop with unrelated travel times. In *International Conference on Discrete Optimization and Operations Research* (pp. 272–283). Springer.
Chernykh, I., Kononov, A., & Sevastyanov, S. (2013). Efficient approximation algorithms for the routing open shop problem. *Computers & Operations Research, 40*, 841–847.
Chernykh, I., & Krivonogova, O. (2020). On the optima localization for the three--machine routing open shop. In A. Kononov, M. Khachay, V. A. Kalyagin, & P. Pardalos (Eds.), *Mathematical Optimization Theory and Operations Research* (pp. 274–288). Cham: Springer International Publishing.
Chernykh, I., & Kuzevanov, M. (2013). Sufficient condition of polynomial solvability of two-machine routing open shop with preemption allowed. *Intellektualnye Sistemy, 17*(1–4), 552–556.
Chernykh, I., & Lgotina, E. (2016). The 2-machine routing open shop on a triangular transportation network. In *International Conference on Discrete Optimization and Operations Research* (pp. 284–297). Springer.
Chernykh, I., & Lgotina, E. (2019). How the difference in travel times affects the optima localization for the routing open shop. In *International Conference on Mathematical Optimization Theory and Operations Research* (pp. 187–201). Springer.
Chernykh, I., & Pyatkin, A. (2017). Refinement of the optima localization for the two-machine routing open shop. In *Proceedings of the 8th International Conference on Optimization and Applications (optima 2017). CEUR Workshop Proceedings: 1987* (pp. 131–138).
Chernykh, I. D., & Lgotina, E. V. (2020). Two-machine routing open shop on a tree: Instance reduction and efficiently solvable subclass. *Optimization Methods and Software*, 1–21.
Cho, Y., & Sahni, S. (1981). Preemptive scheduling of independent jobs with release and due times on open, flow and job shops.. *Operations Research, 29*(3), 511–522.
Chou, S.-Y., & Lin, S.-W. (2007). Museum visitor routing problem with the balancing of concurrent visitors. In *Complex Systems Concurrent Engineering* (pp. 345–353). Springer.
Chung, C.-S., & Mohanty, B. B. (1988). Minimizing expected makespan in a two-machine stochastic open shop with poisson arrival. *Journal of Mathematical Analysis and Applications, 133*(2), 498–508.
Cochand, M., de Werra, D., & Slowinski, R. (1989). Preemptive scheduling with staircase and piecewise linear resource availability. *Zeitschrift für Operations Research, 33*(5), 297–313.
Coffman, E. G., & Timkovsky, V. G. (2002). Ideal two-machine schedules of jobs with unit-execution-time operations. In *Proc. 8th Inter. Workshop on Proj. Manag. Sched*.
Coffman Jr, E. G., & Graham, R. L. (1972). Optimal scheduling for two-processor systems. *Acta Informatica, 1*(3), 200–213.
Colak, S., & Agarwal, A. (2005). Non-greedy heuristics and augmented neural networks for the open-shop scheduling problem. *Naval Research Logistics (NRL), 52*(7), 631–644.
Condotta, A., & Shakhlevich, N. V. (2014). Scheduling patient appointments via multilevel template: A case study in chemotherapy. *Operations Research for Health Care, 3*(3), 129– 144.
Costa, D. (1995). An evolutionary tabu search algorithm and the NHL scheduling problem. *INFOR: Information Systems and Operational Research, 33*(3), 161–178.
Crawford, J. M., & Baker, A. B. (1994). Experimental results on the application of satisfiability algorithms to scheduling problems. In *AAAI: 2* (pp. 1092–1097).
Czibula, O. G., Gu, H., & Zinder, Y. (2016). Timetabling of workplace training: a combination of mathematical programming and simulated annealing. In *Proceedings of the 11th International Conference on Practice and Theory of Automated Timetabling (PATAT16)* (pp. 95–107).
Daganzo, C. F. (1989). The crane scheduling problem. *Transportation Research Part B: Methodological, 23*(3), 159–175.
de Werra, D. (1989). Graph-theoretical models for preemptive scheduling. *Advances in Project Scheduling*, 171–185.
de Werra, D. (1990). Almost nonpreemptive schedules. *Annals of Operations Research, 26*(1), 243–256.
de Werra, D., Asratian, A. S., & Durand, S. (2002). Complexity of some special types of timetabling problems. *Journal of Scheduling, 5*(2), 171–183.
de Werra, D., & Blazewicz, J. (1992). Some preemptive open shop scheduling problems with a renewable or a nonrenewable resource. *Discrete Applied Mathematics, 35*(3), 205– 219.
de Werra, D., & Blazewicz, J. (1993). Addendum: Some preemptive open shop scheduling problems with a renewable or a nonrenewable resource. *Discrete Applied Mathematics, 43*(1), 103– 104.
de Werra, D., Bewicz, J., & Kubiak, W. (1991). A preemptive open shop scheduling problem with one resource. *Operations Research Letters, 10*(1), 9– 15.
de Werra, D., & Erschler, J. (1996). Open shop scheduling with some additional constraints. *Graphs and Combinatorics, 12*(1), 81–93.
de Werra, D., Hertz, A., Kobler, D., & Mahadev, N. V. (2000). Feasible edge colorings of trees with cardinality constraints. *Discrete Mathematics, 222*(1–3), 61–72.
de Werra, D., Hoffman, A. J., Mahadev, N. V., & Peled, U. N. (1996). Restrictions and preassignments in preemptive open shop scheduling. *Discrete Applied Mathematics, 68*(1–2), 169–188.
de Werra, D., Mahadev, N. V. R., & Peled, U. N. (1993). Edge-chromatic scheduling with simultaneity constraints. *SIAM Journal on Discrete Mathematics, 6*(4), 611–631.
de Werra, D., & Solot, P. (1993). Some graph-theoretical models for scheduling in automated production systems. *Networks, 23*(8), 651–660.
Dell'Amico, M., & Martello, S. (1996). Open shop, satellite communication and a theorem by egerváry (1931). *Operations Research Letters, 18*(5), 207–211.
Dell'Amico, M., & Trubian, M. (1993). Applying tabu search to the job-shop scheduling problem. *Annals of Operations Research, 41*(3), 231–252.
Dell'Amico, M., & Vaessens, R. J. M. (1996). *Flow and open shop scheduling on two machines with transportation times and machine-independent processing times in NP-Hard*. Università degli Studi di Modena. Dipartimento di economia politica.
Dorndorf, U., Pesch, E., & Phan-Huy, T. (2001). Solving the open shop scheduling problem. *Journal of Scheduling, 4*(3), 157–174.
Drobouchevitch, I. G. (2020). Three-machine open shop with a bottleneck machine revisited. *Journal of Scheduling*, 1–12.
Drobouchevitch, I. G., & Strusevich, V. A. (1999). A polynomial algorithm for the three-machine open shop with a bottleneck machine. *Annals of Operations Research, 92*, 185–210.
Dror, M. (1992). Openshop scheduling with machine dependent processing times. *Discrete Applied Mathematics, 39*(3), 197–205.
Egerváry, J. (1931). Matrixok kombinatorius tulajdonságairól (english translation by hw kuhn): On combinatorial properties of matrices. *Logistics Papers, George Washington University*, (1), 1–11.
Emmons, H. (1973). The two-machine job-shop with exponential processing times. In *Symposium on the Theory of Scheduling and its Applications* (pp. 278–293). Springer.
Ernst, A. T., Mills, R., & Welgama, P. (2003). Scheduling appointments at trade events for the australian tourist commission. *Interfaces, 33*(3), 12–23.
Esswein, C., Billaut, J.-C., & Strusevich, V. A. (2005). Two-machine shop scheduling: Compromise between flexibility and makespan value. *European Journal of Operational Research, 167*(3), 796–809.
Fang, H.-L., Ross, P., & Corne, D. (1993). *A promising genetic algorithm approach to*

job-shop scheduling, rescheduling, and open-shop scheduling problems. University of Edinburgh, Department of Artificial Intelligence.
Fang, H.-L., Ross, P., & Corne, D. (1994). A promising hybrid ga/heuristic approach for open-shop scheduling problems. In *Proceedings of the 11th European Conference on Artificial Intelligence* (pp. 590–594).
Fei, H., Meskens, N., Combes, C., & Chu, C. (2009). The endoscopy scheduling problem: A case study with two specialised operating rooms. *International Journal of Production Economics, 120*(2), 452–462.
Fiala, T. (1983). An algorithm for the open-shop problem. *Mathematics of Operations Research, 8*(1), 100–109.
Frostig, E. (1991). On the optimality of static policy in stochastic open shop. *Operations Research Letters, 10*(9), 509–512.
Gabow, H. N., & Kariv, O. (1982). Algorithms for edge coloring bipartite graphs and multigraphs. *SIAM Journal on Computing, 11*(1), 117–129.
Ghosn, S. B., Drouby, F., & Harmanani, H. M. (2016). A parallel genetic algorithm for the open-shop scheduling problem using deterministic and random moves. *International Journal of Artificial Intelligence, 14*(1), 130–144.
Giaro, K. (2001). Np-hardness of compact scheduling in simplified open and flow shops. *European Journal of Operational Research, 130*(1), 90– 98.
Giaro, K., Rubale, M., & Malafiejski, M. (1999). Compact scheduling in open shop with zero-one time operations. *INFOR: Information Systems and Operational Research, 37*(1), 37–47.
Gilmore, P. C., & Gomory, R. E. (1964). Sequencing a one state-variable machine: A solvable case of the traveling salesman problem. *Operations Research, 12*(5), 655–679.
Glass, C., Potts, C., & Strusevich, V. (2001). Scheduling batches with sequential job processing for two-machine flow and open shops. *INFORMS Journal on Computing, 13*(2), 120–137.
Glass, C. A., Shafransky, Y. M., & Strusevich, V. A. (2000). Scheduling for parallel dedicated machines with a single server. *Naval Research Logistics (NRL), 47*(4), 304–328.
Golovachev, M., & Pyatkin, A. V. (2019). Routing open shop with two nodes, unit processing times and equal number of jobs and machines. In *International Conference on Mathematical Optimization Theory and Operations Research* (pp. 264–276). Springer.
Gonzalez, T. (1979). A note on open shop preemptive schedules. *IEEE Transactions on Computers*, (10), 782–786.
Gonzalez, T. (1982). Unit execution time shop problems. *Mathematics of Operations Research, 7*(1), 57–66.
Gonzalez, T., & Sahni, S. (1976). Open shop scheduling to minimize finish time. *Journal of the ACM (JACM), 23*(4), 665–679.
Gonzalez, T. F. (2004). Open shop scheduling. *Chapman & Hall/CRC Computer and Information Science Series*. CRC Press.
Grabowski, J., Nowicki, E., & Zdrzałka, S. (1986). A block approach for single-machine scheduling with release dates and due dates. *European Journal of Operational Research, 26*(2), 278–285.
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 P. L. Hammer, E. L. Johnson, & B. H. Korte (Eds.), *Discrete Optimization II*. In *Annals of Discrete Mathematics: 5* (pp. 287– 326). Elsevier.
Graves, S. C., Meal, H. C., Stefek, D., & Zeghmi, A. H. (1983). Scheduling of re-entrant flow shops. *Journal of Operations Management, 3*(4), 197–207.
Gribkovskaia, I. V., Lee, C.-Y., Strusevich, V. A., & de Werra, D. (2006). Three is easy, two is hard: Open shop sum-batch scheduling problem refined. *Operations Research Letters, 34*(4), 459– 464.
Grimes, D., Hebrard, E., & Malapert, A. (2009). Closing the open shop: Contradicting conventional wisdom. In *International Conference on Principles and Practice of Constraint Programming* (pp. 400–408). Springer.
Grinshpoun, T., Ilani, H., & Shufan, E. (2014). Partially-concurrent open shop scheduling. In *Proceedings of the 10th International Conference of the Practice and Theory of Automated Timetabling (PATAT)* (pp. 188–201).
Grinshpoun, T., Ilani, H., & Shufan, E. (2017). The representation of partially–concurrent open shop problems. *Annals of Operations Research, 252*(2), 455–469.
Guéret, C., Jussien, N., & Prins, C. (2000). Using intelligent backtracking to improve branch-and-bound methods: An application to open-shop problems. *European Journal of Operational Research, 127*(2), 344–354.
Guéret, C., & Prins, C. (1998a). Classical and new heuristics for the open-shop problem: A computational evaluation. *European Journal of Operational Research, 107*(2), 306–314.
Guéret, C., & Prins, C. (1998b). Forbidden intervals for open-shop problems. *Technical Report*. Nantes, France: École des Mines de Nantes.
Guéret, C., & Prins, C. (1999). A new lower bound for the open shop problem. *Annals of Operations Research, 92*(0), 165–183.
Hall, N. G., & Sriskandarajah, C. (1996). A survey of machine scheduling problems with blocking and no-wait in process. *Operations Research, 44*(3), 510–525.
Harmanani, H. M., & Ghosn, S. B. (2016). An efficient method for the open-shop scheduling problem using simulated annealing. In S. Latifi (Ed.), *Information Technology: New Generations* (pp. 1183–1193). Cham: Springer International Publishing.
Hassan, M.-A., Kacem, I., Martin, S., & Osman, I. M. (2018). On the m-clique free interval subgraphs polytope: Polyhedral analysis and applications. *Journal of Combinatorial Optimization, 36*(3), 1074–1101.
Herrmann, J. W., & Lee, C.-Y. (1992). *Three-machine look-ahead scheduling problems*. Department of Industrial and Systems Engineering, University of Florida.
Hoogeveen, H., Skutella, M., & Woeginger, G. J. (2003). Preemptive scheduling with rejection. *Mathematical Programming, 94*(2-3), 361–374.
Huang, Y.-M., & Lin, J.-C. (2011). A new bee colony optimization algorithm with idle-time-based filtering scheme for open shop-scheduling problems. *Expert Systems with Applications, 38*(5), 5438– 5447.
Ilani, H., Grinshpoun, T., & Shufan, E. (2018). Partially concurrent open shop scheduling with preemption and limited resources. In *Proceedings of the 12th International Conference on the Practice and Theory of Automated Timetabling (PATAT)* (pp. 299– 311).
Ilani, H., Shufan, E., & Grinshpoun, T. (2017). Partially concurrent open shop scheduling with integral preemptions. *Annals of Operations Research, 259*(1), 157–171.
Impagliazzo, R., & Paturi, R. (2001). On the complexity of k-sat. *Journal of Computer and System Sciences, 62*(2), 367–375.
Inukai, T. (1979). An efficient ss/tdma time slot assignment algorithm. *IEEE Transactions on Communications, 27*(10), 1449–1455.
Jansen, K., Land, F., & Land, K. (2016). Bounding the running time of algorithms for scheduling and packing problems. *SIAM Journal on Discrete Mathematics, 30*(1), 343–366.
Jiang, F., Zhang, X., Bai, D., & Wu, C.-C. (2018). Competitive two-agent scheduling problems to minimize the weighted combination of makespans in a two-machine open shop. *Engineering Optimization, 50*(4), 684–697.
Józefowska, J., Jurisch, B., & Kubiak, W. (1994). Scheduling shops to minimize the weighted number of late jobs. *Operations Research Letters, 16*(5), 277–283.
Jurisch, B., & Kubiak, W. (1997). Two-machine open shops with renewable resources. *Operations Research, 45*(4), 544–552.
Jussien, N., & Lhomme, O. (2002). Local search with constraint propagation and conflict-based heuristics. *Artificial Intelligence, 139*(1), 21–45.
Khatami, M., & Salehipour, A. (2020a). A binary search algorithm for the general coupled task scheduling problem. *4OR*, 1–19.
Khatami, M., & Salehipour, A. (2020b). Optimal makespan for the coupled task ordered flow-shop. Available at SSRN 3524740,.
Khatami, M., Salehipour, A., & Cheng, T. C. E. (2020). Coupled task scheduling with exact delays: Literature review and models. *European Journal of Operational Research, 282*(1), 19– 39.
Khatami, M., Salehipour, A., & Hwang, F. J. (2019). Makespan minimization for the *m*-machine ordered flow shop scheduling problem. *Computers & Operations Research, 111*, 400–414.
Khormali, A., Mirzazadeh, A., & Faez, F. (2012). The openshop batch processing problem with non-identical processing times, using simulated annealing and genetic algorithms approaches. *The International Journal of Advanced Manufacturing Technology, 59*(9–12), 1157–1165.
Khuri, S., & Miryala, S. R. (1999). Genetic algorithms for solving open shop scheduling problems. In *Portuguese Conference on Artificial Intelligence* (pp. 357–368). Springer.
Kise, H., Shioyama, T., & Ibaraki, T. (1991). Automated two-machine flowshop scheduling: A solvable case. *IIE Transactions, 23*(1), 10–16.
Kleinau, U. (1993). Two-machine shop scheduling problems with batch processing. *Mathematical and Computer Modelling, 17*(6), 55–66.
Knust, S., Shakhlevich, N., Waldherr, S., & Weiß, C. (2019). Shop scheduling problems with pliable jobs. *Journal of Scheduling*, 1–27.
Kononov, A. (2012). On the routing open shop problem with two machines on a two-vertex network. *Journal of Applied and Industrial Mathematics, 6*(3), 318–331.
Kononov, A. (2015). $o(\log m)$-approximation for the routing open shop problem. *RAIRO-Operations Research, 49*(2), 383–391.
Kononov, A., & Gawiejnowicz, S. (2001). Np-hard cases in scheduling deteriorating jobs on dedicated machines. *Journal of the Operational Research Society, 52*(6), 708–717.
Kononov, A., Sevastyanov, S., & Sviridenko, M. (2012). A complete 4-parametric complexity classification of short shop scheduling problems. *Journal of Scheduling, 15*(4), 427–446.
Kononov, A., Sevast'yanov, S., & Tchernykh, I. (1999). When difference in machine loads leads to efficient scheduling in open shops. *Annals of Operations Research, 92*(0), 211–239.
Koryakin, R. A. (2003). On the stochastic open shop problem. In *International Symposium on Stochastic Algorithms* (pp. 117–124). Springer.
Koulamas, C., & Kyparisis, G. J. (2015). The three-machine proportionate open shop and mixed shop minimum makespan problems. *European Journal of Operational Research, 243*(1), 70–74.
Koulamas, C., & Panwalkar, S. S. (2015). Job selection in two-stage shops with ordered machines. *Computers & Industrial Engineering, 88*, 350–353.
Kubale, M. (1997). Open shop problem with zero-one time operations and integer release date/deadline intervals. *Discrete Applied Mathematics, 76*(1-3), 21–223.
Kubzin, M. A., & Strusevich, V. A. (2006). Planning machine maintenance in two-machine shop scheduling. *Operations Research, 54*(4), 789–800.
Kubzin, M. A., Strusevich, V. A., Breit, J., & Schmidt, G. (2006). Polynomial-time approximation schemes for two-machine open shop scheduling with nonavailability constraints. *Naval Research Logistics (NRL), 53*(1), 16–23.
Kyparisis, G. J., & Koulamas, C. (1997). Open shop scheduling with maximal machines. *Discrete Applied Mathematics, 78*(1-3), 175–187.
Kyparisis, G. J., & Koulamas, C. (2000). Flow shop and open shop scheduling with a critical machine and two operations per job. *European Journal of Operational Research, 127*(1), 120– 125.
Laborie, P. (2005). Complete mcs-based search: Application to resource constrained project scheduling. In *IJCAI* (pp. 181–186).
Lamé, G., Jouini, O., & Stal-Le Cardinal, J. (2016). Outpatient chemotherapy planning:

A literature review with insights from a case study. *IIE Transactions on Healthcare Systems Engineering, 6*(3), 127–139.
Lawler, E. L., & Labetoulle, J. (1978). On preemptive scheduling of unrelated parallel processors by linear programming. *Journal of the ACM (JACM), 25*(4), 612–619.
Lawler, E. L., Lenstra, J. K., Kan, A. H. R., & Shmoys, D. B. (1993). Sequencing and scheduling: Algorithms and complexity. *Handbooks in Operations Research and Management Science, 4*, 445–522.
Lawler, E. L., Lenstra, J. K., & Rinnooy Kan, A. (1981). Minimizing maximum lateness in a two-machine open shop. *Mathematics of Operations Research, 6*(1), 153–158.
Lee, C.-Y., & Strusevich, V. A. (2005). Two-machine shop scheduling with an uncapacitated interstage transporter. *IIE Transactions, 37*(8), 725–736.
Li, S. (2011). Scheduling proportionally deteriorating jobs in two-machine open shop with a non-bottleneck machine. *Asia-Pacific Journal of Operational Research, 28*(5), 623–631.
Liaw, C.-F. (1998). An iterative improvement approach for the nonpreemptive open shop scheduling problem. *European Journal of Operational Research, 111*(3), 509–517.
Liaw, C.-F. (1999a). Applying simulated annealing to the open shop scheduling problem. *IIE Transactions, 31*(5), 457–465.
Liaw, C.-F. (1999b). A tabu search algorithm for the open shop scheduling problem. *Computers & Operations Research, 26*(2), 109– 126.
Liaw, C.-F. (2000). A hybrid genetic algorithm for the open shop scheduling problem. *European Journal of Operational Research, 124*(1), 28– 42.
Liaw, C.-F., Cheng, C.-Y., & Chen, M. (2005). Scheduling two-machine no-wait open shops to minimize makespan. *Computers & Operations Research, 32*(4), 901– 917.
Liu, C. V., & Bulfin, R. L. (1988). Scheduling open shops with unit execution times to minimize functions of due dates. *Operations Research, 36*(4), 553.
Liu, C. Y., & Bulfin, R. L. (1987). Scheduling ordered open shops. *Computers & Operations Research, 14*(3), 257– 264.
Liu, M., Chu, C., Xu, Y., & Zheng, F. (2010). An optimal online algorithm for two-machine open shop preemptive scheduling with bounded processing times. *Optimization Letters, 4*(2), 227–237.
Lorigeon, T., Billaut, J.-C., & Bouquard, J.-L. (2002). A dynamic programming algorithm for scheduling jobs in a two-machine open shop with an availability constraint. *Journal of the Operational Research Society, 53*(11), 1239–1246.
Louis, S. J., & Xu, Z. (1996). Genetic algorithms for open shop scheduling and re-scheduling. In *Proc. of the 11th ISCA Int. Conf. on Computers and their Applications: 28* (pp. 99–102).
Lu, l., & Posner, M. C. (1993). An np-hard open shop scheduling problem with polynomial average time complexity. *Mathematics of Operations Research, 18*(1), 12–38.
Lushchakova, I. N., Soper, A. J., & Strusevich, V. A. (2009). Transporting jobs through a two-machine open shop. *Naval Research Logistics (NRL), 56*(1), 1–18.
Ma, Y., Chu, C., & Zuo, C. (2010). A survey of scheduling with deterministic machine availability constraints. *Computers & Industrial Engineering, 58*(2), 199–211.
Malapert, A., Cambazard, H., Guéret, C., Jussien, N., Langevin, A., & Rousseau, L.-M. (2012). An optimal constraint programming approach to the open-shop problem. *INFORMS Journal on Computing, 24*(2), 228–244.
Martello, S. (2010). Jenő egerváry: From the origins of the Hungarian algorithm to satellite communication. *Central European Journal of Operations Research, 18*(1), 47–58.
Mejía, G., & Yuraszeck, F. (2020). A self-tuning variable neighborhood search algorithm and an effective decoding scheme for open shop scheduling problems with travel/setup times. *European Journal of Operational Research, 285*, 484–496.
Meja, G., Caballero-Villalobos, J. P., & Montoya, C. (2018). Petri nets and deadlock--free scheduling of open shop manufacturing systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 48*(6), 1017–1028.
Middendorf, M. (1998). Shortest common superstrings and scheduling with coordinated starting times. *Theoretical Computer Science, 191*(1–2), 205–214.
Mnich, M., & van Bevern, R. (2018). Parameterized complexity of machine scheduling: 15 open problems. *Computers & Operations Research*.
Mor, B., Mosheiov, G., & Oron, D. (2012). The equal allocation policy in open shop batch scheduling. *Journal of the Operational Research Society, 63*(5), 646–652.
Mosheiov, G. (2002). Complexity analysis of job-shop scheduling with deteriorating jobs. *Discrete Applied Mathematics, 117*(1–3), 195–209.
Mosheiov, G., & Oron, D. (2008). Open-shop batch scheduling with identical jobs. *European Journal of Operational Research, 187*(3), 1282– 1292.
Mosheiov, G., Sarig, A., Strusevich, V. A., & Mosheiff, J. (2018). Two-machine flow shop and open shop scheduling problems with a single maintenance window. *European Journal of Operational Research, 271*(2), 388–400.
Munier-Kordon, A., & Rebaine, D. (2010). The two-machine open-shop problem with unit-time operations and time delays to minimize the makespan. *European Journal of Operational Research, 203*(1), 42–49.
Naderi, B., Ghomi, S. M. T. F., Aminnayeri, M., & Zandieh, M. (2010). A contribution and new heuristics for open shop scheduling. *Computers & Operations Research, 37*(1), 213– 221.
Naderi, B., & Zandieh, M. (2014). Modeling and scheduling no-wait open shop problems. *International Journal of Production Economics, 158*, 256–266.
Naderi, B., Zandieh, M., & Yazdani, M. (2014). Polynomial time approximation algorithms for proportionate open-shop scheduling. *International Transactions in Operational Research, 21*(6), 1031–1044.
Oulamara, A., Rebaine, D., & Serairi, M. (2013). Scheduling the two-machine open shop problem under resource constraints for setting the jobs. *Annals of Operations Research, 211*(1), 333–356.
Ozolins, A. (2019). Dynamic programming approach for solving the open shop problem. *Central European Journal of Operations Research*, 1–16.
Pan, J. C.-H., & Chen, J.-S. (2005). Mixed binary integer programming formulations for the reentrant job shop scheduling problem. *Computers & Operations Research, 32*(5), 1197–1212.
Panwalkar, S., & Koulamas, C. (2014). The two-machine no-wait general and proportionate open shop makespan problem. *European Journal of Operational Research, 238*(2), 471–475.
Pinedo, M. (1995). *Scheduling: Theory, Algorithms and Systems*. Englewood Cliffs, NJ: Prentice Hall.
Pinedo, M. (2016). *Scheduling*. Springer.
Pinedo, M., & Schrage, L. (1982). Stochastic shop scheduling: A survey. In *Deterministic and Stochastic Scheduling* (pp. 181–196). Springer.
Pinedo, M., & Weber, R. (1984). Inequalities and bounds in stochastic shop scheduling. *SIAM Journal on Applied Mathematics, 44*(4), 869–879.
Pinedo, M. L., & Ross, S. M. (1982). Minimizing expected makespan in stochastic open shops. *Advances in Applied Probability, 14*(4), 898–911.
Pongchairerks, P., & Kachitvichyanukul, V. (2016). A two-level particle swarm optimisation algorithm for open-shop scheduling problem. *International Journal of Computing Science and Mathematics, 7*(6), 575–585.
Potts, C., Strusevich, V., & Tautenhahn, T. (2001). Scheduling batches with simultaneous job processing for two-machine shop problems. *Journal of Scheduling, 4*(1), 25–51.
Prins, C. (1994). An overview of scheduling problems arising in satellite communications. *Journal of the Operational Research Society, 45*(6), 611–623.
Prins, C. (2000). Competitive genetic algorithms for the open-shop scheduling problem. *Mathematical Methods of Operations Research, 52*(3), 389–411.
Prins, C. (2010). *Open shop scheduling* (pp. 271–300)). John Wiley & Sons, Ltd.
Prins, C., & Carlier, J. (1986). Resource optimization in a TDMA/DSI system-the eutelsat approach. In *ICDSC-7; Proceedings of the Seventh International Conference on Digital Satellite Communications* (pp. 511–518).
Prot, D., & Bellenguez-Morineau, O. (2018). A survey on how the structure of precedence constraints may change the complexity class of scheduling problems. *Journal of Scheduling, 21*(1), 3–16.
Puente, J., Díez, H. R., Varela, R., Vela, C. R., & Hidalgo, L. P. (2003). Heuristic rules and genetic algorithms for open shop scheduling problem. In *Conference on Technology Transfer* (pp. 394–403). Springer.
Pyatkin, A., & Chernykh, I. (2012). The open shop problem with routing at a two-node network and allowed preemption. *Journal of Applied and Industrial Mathematics, 6*(3), 346–354.
Rahmani Hosseinabadi, A. A., Vahidi, J., Saemi, B., Sangaiah, A. K., & Elhoseny, M. (2019). Extended genetic algorithm for solving open-shop scheduling problem. *Soft Computing, 23*(13), 5099–5116.
Ramudhin, A., & Marier, P. (1996). The generalized shifting bottleneck procedure. *European Journal of Operational Research, 93*(1), 34–48.
Rayward-Smith, V. J., & Rebaine, D. (1992). Open shop scheduling with delays. *RAIRO-Theoretical Informatics and Applications, 26*(5), 439–447.
Rebaine, D., & Strusevich, V. A. (1999). Two-machine open shop scheduling with special transportation times. *Journal of the Operational Research Society, 50*(7), 756–764.
Righter, R. (1997). Stochastic scheduling for a two-machine open shop. *Journal of Applied Probability, 34*(3), 733–744.
Rinaldi, F., & Serafini, P. (2006). Scheduling school meetings. In *International Conference on the Practice and Theory of Automated Timetabling* (pp. 280–293). Springer.
Roshanaei, V., Esfehani, M. S., & Zandieh, M. (2010). Integrating non-preemptive open shops scheduling with sequence-dependent setup times using advanced metaheuristics. *Expert Systems with Applications, 37*(1), 259– 266.
Roy, B., & Sussmann, B. (1964). Les problemes dordonnancement avec contraintes disjonctives. *Note ds, 9*.
Sahni, S., & Cho, Y. (1979). Complexity of scheduling shops with no wait in process. *Mathematics of Operations Research, 4*(4), 448–457.
Schuurman, P., & Woeginger, G. J. (1999). Polynomial time approximation algorithms for machine scheduling: Ten open problems. *Journal of Scheduling, 2*(5), 203–213.
Sedeño-Noda, A., Alcaide, D., & González-Martín, C. (2006). Network flow approaches to pre-emptive open-shop scheduling problems with time-windows. *European Journal of Operational Research, 174*(3), 1501–1518.
Senthilkumar, P., & Shahabudeen, P. (2006). Ga based heuristic for the open job shop scheduling problem. *The International Journal of Advanced Manufacturing Technology, 30*(3-4), 297–301.
Sevastianov, S. V. (2005). An introduction to multi-parameter complexity analysis of discrete problems. *European Journal of Operational Research, 165*(2), 387–397.
Sevast'yanov, S. (1992). A polynomial-time open-shop problem with an arbitrary number of machines. *Cybernetics and Systems Analysis, 28*(6), 918–933.
Sevast'yanov, S. (1995). Vector summation in banach space and polynomial algorithms for flow shops and open shops. *Mathematics of Operations Research, 20*(1), 90–103.
Sevast'yanov, S. (1996). Efficient scheduling in open shops. In *Discrete Analysis and Operations Research* (pp. 235–255). Springer.
Sevast'yanov, S. (1998). Nonstrict vector summationin multi-operation scheduling. *Annals of Operations Research, 83*, 179–212.
Sevastyanov, S. V. (2019). Some positive news on the proportionate open shop problem. *Siberian Electronic Mathematical Reports, 16*(0), 406–426.
Sevast'yanov, S. V., & Tchernykh, I. D. (1998). Computer-aided way to prove theorems in scheduling. In *European Symposium on Algorithms* (pp. 502–513). Springer.

Sevast'yanov, S. V., & Woeginger, G. J. (1998). Makespan minimization in open shops: A polynomial time approximation scheme. *Mathematical Programming, 82*(1), 191–198.
Sha, D., & Hsu, C.-Y. (2008). A new particle swarm optimization for the open shop scheduling problem. *Computers & Operations Research, 35*(10), 3243– 3261.
Shabtay, D., Gaspar, N., & Kaspi, M. (2013). A survey on offline scheduling with rejection. *Journal of Scheduling, 16*(1), 3–28.
Shabtay, D., & Gasper, N. (2012). Two-machine flow-shop scheduling with rejection. *Computers & Operations Research, 39*(5), 1087– 1096.
Shabtay, D., & Kaspi, M. (2006). Minimizing the makespan in open-shop scheduling problems with a convex resource consumption function. *Naval Research Logistics (NRL), 53*(3), 204–216.
Shafransky, Y. M., & Strusevich, V. A. (1998). The open shop scheduling problem with a given sequence of jobs on one machine. *Naval Research Logistics, 45*, 705–731.
Shakhlevich, N., & Strusevich, V. (1993). Two machine open shop scheduling problem to minimize an arbitrary machine usage regular penalty function. *European Journal of Operational Research, 70*(3), 391–404.
Shakhlevich, N. V., Sotskov, Y. N., & Werner, F. (2000). Complexity of mixed shop scheduling problems: A survey. *European Journal of Operational Research, 120*(2), 343–351.
Shchepin, E. V. (2015). On the complexity of constructing multiprocessor little-preemptive schedules. *Proceedings of the Steklov Institute of Mathematics, 290*(1), 166–177.
Shchepin, E. V., & Vakhania, N. (2008). On the geometry, preemptions and complexity of multiprocessor and shop scheduling. *Annals of Operations Research, 159*(1), 183–213.
Shchepin, E. V., & Vakhania, N. (2011). A note on the proof of the complexity of the little-preemptive open-shop problem. *Annals of Operations Research, 191*(1), 251–253.
Shmoys, D. B., Stein, C., & Wein, J. (1994). Improved approximation algorithms for shop scheduling problems. *SIAM Journal on Computing, 23*(3), 617–632.
Sidney, J. B., & Sriskandarajah, C. (1999). A heuristic for the two-machine no-wait openshop scheduling problem. *Naval Research Logistics (NRL), 46*(2), 129–145.
Soper, A. J. (2015). A cyclical search for the two machine flow shop and open shop to minimise finishing time. *Journal of Scheduling, 18*(3), 311–314.
Strusevich, V. (2000). Group technology approach to the open shop scheduling problem with batch setup times. *Operations Research Letters, 26*(4), 181– 192.
Strusevich, V. A. (1991). *Complexity aspects of shop scheduling problems*. Erasmus Universiteit Rotterdam Ph.D. thesis..
Strusevich, V. A. (1993). Two machine open shop scheduling problem with setup, processing and removal times separated. *Computers & Operations Research, 20*(6), 597– 611.
Strusevich, V. A. (1998). A greedy open shop heuristic with job priorities. *Annals of Operations Research, 83*, 253–270.
Strusevich, V. A. (1999). A heuristic for the two-machine open-shop scheduling problem with transportation times. *Discrete Applied Mathematics, 93*(2–3), 287–304.
Strusevich, V. A., Van De Waart, A., & Dekker, R. (1999). A 3/2 algorithm for two-machine open shop with route-dependent processing times. *Journal of Heuristics, 5*(1), 5–28.
Su, L.-H., & Hsiao, M.-C. (2015). Two-agent scheduling in open shops subject to machine availability and eligibility constraints. *Journal of Industrial Engineering and Management, 8*(4), 1103–1124.
Taillard, E. (1993). Benchmarks for basic scheduling problems. *European Journal of Operational Research, 64*(2), 278– 285.
Tamura, N., Taga, A., Kitagawa, S., & Banbara, M. (2009). Compiling finite linear CSP into SAT. *Constraints, 14*(2), 254–272.
Tanaev, V., Gordon, W., & Shafransky, Y. M. (2012). *Scheduling theory. Single-stage systems*: 284. Springer Science & Business Media.
Tautenhahn, T. (1994). Scheduling unit-time open shops with deadlines. *Operations Research, 42*(1), 189.
Tautenhahn, T., & Woeginger, G. J. (1997). Unit-time scheduling problems with time dependent resources. *Computing, 58*(2), 97–111.
Tellache, N. E. H., & Boudhar, M. (2017). Open shop scheduling problems with conflict graphs. *Discrete Applied Mathematics, 227*, 103– 120.
Tellache, N. E. H., Boudhar, M., & Yalaoui, F. (2019). Two-machine open shop problem with agreement graph. *Theoretical Computer Science, 796*, 154–168.
Timkovsky, V. G. (2003). Identical parallel machines vs. unit-time shops and preemptions vs. chains in scheduling complexity. *European Journal of Operational Research, 149*(2), 355–376.
Vairaktarakis, G., & Sahni, S. (1995). Dual criteria preemptive open-shop problems with minimum makespan. *Naval Research Logistics (NRL), 42*(1), 103–121.
Van Den Akker, M., Hoogeveen, H., & Woeginger, G. J. (2003). The two-machine open shop problem: To fit or not to fit, that is the question. *Operations Research Letters, 31*(3), 219–224.
Vestjens, A. P. A., Wennink, M., & Woeginger, G. J. (2007). Complexity of the job insertion problem in multi-stage scheduling. *Operations Research Letters, 35*(6), 754–758.
Vincent, F. Y., Lin, S.-W., & Chou, S.-Y. (2010). The museum visitor routing problem. *Applied Mathematics and Computation, 216*(3), 719–729.
Wagneur, E., & Sriskandarajah, C. (1993). Openshops with jobs overlap. *European Journal of Operational Research, 71*(3), 366–378.
Wang, H., Chen, W., & Wang, J. (2020). Coupled task scheduling for heterogeneous multi-robot system of two robot types performing complex-schedule order fulfillment tasks. *Robotics and Autonomous Systems*, 103560.
Wang, M. Y., Sethi, S. P., & van de Velde, S. L. (1997). Minimizing makespan in a class of reentrant shops. *Operations Research, 45*(5), 702–712.
Wein, J. M. (1991). Algorithms for scheduling and network problems. *Technical Report*. Massachusetts Institute of Technology lab for computer science.
Williamson, D. P., Hall, L. A., Hoogeveen, J. A., Hurkens, C. A., Lenstra, J. K., Sevast'janov, S. V., & Shmoys, D. B. (1997). Short shop schedules. *Operations Research, 45*(2), 288–294.
Wirth, M., & Emde, S. (2018). Scheduling trucks on factory premises. *Computers & Industrial Engineering, 126*, 175–186.
Woeginger, G. J. (2005). Formulations, relaxations, approximations, and gaps in the world of scheduling. In *Multidisciplinary Scheduling: Theory and applications* (pp. 19–36). Springer.
Yao, M.-J., Soewandi, H., & Elmaghraby, S. E. (2000). Simple heuristics for the two machine openshop problem with blocking. *Journal of the Chinese Institute of Industrial Engineers, 17*(5), 537–547.
Yu, W., Hoogeveen, H., & Lenstra, J. K. (2004). Minimizing makespan in a two-machine flow shop with delays and unit-time operations is np-hard. *Journal of Scheduling, 7*(5), 333–348.
Yu, W., Liu, Z., Wang, L., & Fan, T. (2011). Routing open shop and flow shop scheduling problems. *European Journal of Operational Research, 213*(1), 24– 36.
Yu, W., & Zhang, G. (2011). Improved approximation algorithms for routing shop scheduling. In *International Symposium on Algorithms and Computation* (pp. 30–39). Springer.
Zhan, Y., Zhong, Y. G., & Zhu, H. T. (2011). Preemptive open-shop scheduling: Network flow based algorithm. In *Advanced Materials Research: 215* (pp. 111–114).
Zhang, L., Lu, L., & Yuan, J. (2016). Two-machine open-shop scheduling with rejection to minimize the makespan. *OR Spectrum, 38*(2), 519–529.
Zhang, X., & van de Velde, S. (2010a). On-line two-machine open shop scheduling with time lags. *European Journal of Operational Research, 204*(1), 14– 19.
Zhang, X., & van de Velde, S. (2010b). Polynomial-time approximation schemes for scheduling problems with time lags. *Journal of Scheduling, 13*(5), 553–559.
Zhao, L., Chien, C.-F., & Gen, M. (2018). A bi-objective genetic algorithm for intelligent rehabilitation scheduling considering therapy precedence constraints. *Journal of Intelligent Manufacturing, 29*(5), 973–988.
Zhao, X., & Wang, G. (2015). Two-machine shop scheduling with two agents and linear deteriorating jobs. In *The 27th Chinese Control and Decision Conference (2015 CCDC)* (pp. 1154–1159). IEEE.
Zobolas, G. I., Tarantilis, C. D., & Ioannou, G. (2009). Solving the open shop scheduling problem via a hybrid genetic-variable neighborhood search algorithm. *Cybernetics and Systems, 40*(4), 259–285.