RAIRO-Oper. Res. 54 (2020) 555–568
https://doi.org/10.1051/ro/2019116

RAIRO Operations Research
www.rairo-ro.org

SCHEDULING JOB SHOP PROBLEMS WITH OPERATORS WITH RESPECT TO THE MAXIMUM LATENESS

Imène Benkalai[1,*], Djamal Rebaine[1] and Pierre Baptiste[2]

Abstract. This paper deals with the problem of assigning operators to jobs, within a free assignment-changing mode, in a job-shop environment subject to a fixed processing sequence of the jobs. We seek an assignment of operators that minimizes the maximum lateness. Within this model, a job needs an operator during the entire duration of its processing. We show that the problem is $\mathcal{NP}$-hard when the number of operators is arbitrary and exhibit polynomial time algorithms for the cases involving one and two operators, respectively.

Mathematics Subject Classification. 90B35.

Received May 31, 2017. Accepted December 13, 2019.

1. Introduction

The largest part of the scheduling literature is dedicated to problems that consider only material resources and ignore human resources. However, the latter have grown to become an important area in scheduling theory.

Integrating human resources and their different characteristics is a relatively recent field of research in scheduling theory. In general, considering human resources when scheduling jobs makes the corresponding problems more difficult to solve.

The analysis of real-life production systems induces that numerous links exist between human and material resources in the aforementioned environments, especially on the operational level [8, 37]. Moreover, the human actor has always played a crucial role in economy. The reason is that, on the one hand, he is a consumer and a producer at the same time, and, on the other hand, he is the only one able to face unpredictable events despite the advances that have been achieved in production equipments [8]. In addition, it is getting more obvious that the classical scheduling models are of little help in the real world applications [29]. Indeed, to model more accurately a real-life situation, additional features are required among which are human operators. All of this to say that it is important to take into account the human component when making decisions related to the management of material resources in order to reduce the gap between theoretical research and practical applications, but also to design effective optimization tools.

As usual with the birth of a new area, studies in this field of scheduling have started with simplified assumptions. The few developed models had the objective of modelling more accurately the constraints encountered in

Keywords. Scheduling, job shop, operators, maximum lateness.

[1] Département d'Informatique et Mathématique, Université du Québec à Chicoutimi, Saguenay, QC G7H 2B1, Canada.

[2] Département de Mathématiques et Génie Industriel, École polytechnique de Montréal, Montréal, QC H3T 1J4, Canada.

[*]Corresponding author: imene.benkalai1@uqac.ca

practice, especially in industrial systems. Let us recall that in classical scheduling, it is implicitly assumed that the number of machines is equal to the number of operators, which makes it possible to dedicate an operator to one machine during all the scheduling process. This is rarely true in real systems.

In the present study we deal with job shop scheduling problems in which the number of operators is less than the number of machines[3]. We assume that the assignment of operators to machines is done in a free changing mode. In other words, the assignment of an operator to a given machine may be changed at any time. In addition, the processing sequence of jobs through the machines is assumed to be part of the known data. Let us observe that in our model we have two problems to solve: the assignment of the operators to the machines and the scheduling of the jobs on the machines. We have two ways of handling this: either we address these two problems simultaneously or we use a sequential approach. In this paper, we focus on the case where the sequence of jobs is fixed and the assignment of the operators is sought. Furthermore, the objective is to minimize the maximum lateness assuming that the operators have different work rates, *i.e.* an operator may be able to process jobs faster or slower than another operator. This models a situation that is often encountered in practice and, to the best of our knowledge, this problem has not yet been addressed in the literature.

This paper is organized as follows. Section 2 discusses the framework and notation, while a description of the problem under study is presented in Section 3. A review of related works is the subject of Section 4. Section 5 is devoted to the complexity status of the problem when the number of operators is arbitrary. Section 6 presents two polynomial-time algorithms for the case of two and one operators, respectively. Finally, concluding remarks are presented in Section 7.

2. Framework and notation

As we pointed out above, scheduling problems under resource constraints are in general more complex than the classical ones, for every job needs for its processing, in addition to the machines available in the scheduling environment, a given number of additional resources[4].

The classical scheme of notation presented in [29] was extended by Blazewicz *et al.* to specify the characteristics of scheduling problems with resource constraints [15]. Parameter $\beta_2 \in \{\varnothing,\ res\ \lambda\delta\rho\}$ is to describe these additional characteristics.

If $\beta_2 = \varnothing$, there are no resource constraints. Otherwise, if $\beta_2 = res\ \lambda\delta\rho$, the problem has specific constraints of some resources. Parameters $\lambda,\ \delta,\ \rho \in \{\cdot, k\}$ represent respectively the number of different resource types, their limited quantities and the needs in resources. The latter are arbitrary if equal to $\cdot$. Otherwise, they are equal to a given number k.

Once an assignment is fixed, operators can change their assignment in time according to either of the following modes [37]:

- *Free changing mode (f)*: an operator can interrupt the processing of a job at any time to handle another job. The interrupted job could later be resumed by this same operator of by another one.
- *Periodic changing mode (p)*: the processing of jobs is performed over a given number of fixed periods. An operator remains assigned to only one machine during the length p of a period. He can interrupt the processing of a job on a machine only at the end of a period to handle another job of another machine. The interrupted job could later be resumed by this same operator of by another one.
- *End of job changing mode (e)*: an operator cannot interrupt the processing a job until its completion on that machine. The operator can then either start processing another job on the same machine or start processing a job on another machine or simply remain idle.

We extend the above-mentioned notation to be able to describe the characteristics of our problem by adding the following symbols:

[3]The present study is an extension of [10].

[4]The resources are often available in an amount that is not enough to handle all the jobs simultaneously.

– S to indicate that the problem is subject to a fixed job sequence S.
– v_h to indicate that the operators have different performance levels.
– f to indicate the free assignment changing mode.

3. Problem description

The job shop scheduling problem we are addressing in this paper is described as follows. Given are $J = \{1, 2, \ldots, n\}$ a set of n jobs to be processed by a set $M = \{M_1, M_2, \ldots, M_m\}$ of m machines. Each job j comprises m operations O_{ij}, $i = 1, \ldots, m$, each of which has to be processed by machine M_i, $i = 1, \ldots, m$. In addition, every job has its own routing through the machines.

In our model, we assume that a job requires the presence of one of the k operators, $k < m$, during its entire processing. In other words, a machine will remain idle as long as there is no available operator to run it. Furthermore, the impact of sharing operators is modelled by waiting times that are due to the fact that the number of operators is less than the number of machines, the processing times are only affected by the work rates of operators. The following parameters p_{ij}, d_{ij} and C_{ij} represent respectively the processing time, the due date and the finish time of operation $\pi(i, j)$.

We assume that we have ℓ types of operators, k_h operators of each type, $h = 1, \ldots, \ell$, with $\sum_{h=1}^{\ell} k_h = k$. An operator of type h has a work rate v_h. The assignment change of these operators on the different machines is operated according to the free mode. In other words, an operator can interrupt the processing of a job at any time to process another job. The interrupted job may be resumed by the same operator or by another one.

The goal is to minimize the maximum lateness, $L_{\max} = \max_{j=1,\ldots,n} L_{mj}$ where $L_{ij} = C_{ij} - d_{ij}$. In a job shop environment, we note the given processing sequence of jobs $\pi = (\pi(i, j))$, $i = 1, \ldots, m$, $j = 1, \ldots, n$, where $\pi(i, j)$ denotes the operation of job j on machine M_i. The goal is to find an assignment of operators a^* such that

$$L_{\max}(\pi, a^*) = \min_{a \in A} L_{\max}(\pi, a),$$

where A denotes the set of feasible assignments, a^* an element of A, and $L_{\max}(\pi, a)$ the maximum lateness of jobs of the sequence π under assignment a. Following the notation presented in Section 2, we denote our problem as

$$J_m|res\ 1k1, S, f, v_h|L_{\max}.$$

Before proceeding any further, let us first describe the notation used here:

– Since operations can be preempted, an operation $\pi(i, j)$ may comprise o_{ij} sub-operations: $\pi(i, j)_b$, $b = 1, \ldots, o_{ij}$.
– When an operation is not preempted, $\pi(i, j) = \pi(i, j)_1$
– An operator h, $h = 1, \ldots, k$, handles n_h sub-operations: op_c, $c = 1, \ldots, n_h$.
– Δ_{ij} is the set of machines that precede machine M_i on the route of job j.
– $p_{h,op_c,\pi(i,j)_b}$ is the amount of processing time during which operator h runs its cth sub-operation, $1 \le c \le n_h$, which corresponds to an operation $\pi(i, j)_b$, $1 \le i \le m$, $1 \le j \le n$ and $1 \le b \le o_{ij}$.
– $av_{h,op_c,\pi(i,j)_b}$ is the time at which the cth sub-operation handled by operator h becomes available[5].
– $ts_{h,op_c,\pi(i,j)_b}$ is the time at which operator h in charge of sub-operation[6] $\pi(i, j)_b$ starts its processing.
– $C(\pi(i, j)_b, a)$ and $C(\pi(i, j), a)$ denote respectively the completion time of sub-operation $\pi(i, j)_b$ and operation $\pi(i, j)$ under assignment a. They are computed by the following recursive formulas.

[5]This operation corresponds to an operation $\pi(i, j)_b$, $1 \le i \le m$, $1 \le j \le n$ and $1 \le b \le o_{ij}$. A sub-operation becomes available when all its preceding sub-operations as well as all the operations of job j on machines in Δ_{ij} have been completed.

[6]Which is its cth sub-operation.

Given a solution (π, a), the completion times of operations are computed as follows.

$$ts_{h,op_0,\pi(i,j)_b} = av_{h,op_0,\pi(i,j)_b} = p_{h,op_0,\pi(i,j)_b} = -\infty;\ h = 1,\dots,k. \tag{3.1}$$

$$ts_{h,op_c,\pi(i,j)_b} = \max\left\{0, ts_{h,op_{c-1},\pi(i',j')_{b'}} + \frac{p_{h,op_{c-1},\pi(i',j')_{b'}}}{v_h}, av_{h,op_c,\pi(i,j)_b}\right\};\quad h = 1,\dots,k;\ c = 1,\dots,n_h. \tag{3.2}$$

$$av_{h,op_c,\pi(i,j)_b} = \max\left\{0, \max_{i'\in\Delta_{ij}} C(\pi(i',j),a), C(\pi(i,j)_{b-1},a)\right\};\quad h = 1,\dots,k;\ c = 1,\dots,n_h. \tag{3.3}$$

$$C(\pi(i,j)_b,a) = ts_{h,op_c,\pi(i,j)_b} + \frac{p_{h,op_c,\pi(i,j)_b}}{v_h};\quad i = 1,\dots,m;\ j = 1,\dots,n;\ b = 1,\dots,o_{ij}. \tag{3.4}$$

$$C(\pi(i,j),a) = C(\pi(i,j)_{o_{ij}},a);\ i = 1,\dots,m;\ j = 1,\dots,n. \tag{3.5}$$

Equation (3.2) compute the times at which operators start processing each of their sub-operations. These starting times occur when the concerned operation becomes available, upon the completion of the previous one, or at time 0 for their first operation.

Equation (3.3) denote the times at which sub-operations are available. A sub-operation $\pi(i,j)_b$ becomes available when all its preceding sub-operations as well as all the operations of job j on machines in Δ_{ij} have been completed.

Finally, equations (3.4) and (3.5) compute the completion times of sub-operations and operations. An operation is completed when all its sub-operations have been processed.

For the sake of clarity, we present the following example on an instance I of $J_3|res\ 121, S, f, v_h|L_{\max}$ with 3 jobs. The processing times p_{ij} and due dates d_j are summarized in Figure 1A. Figure 1B depicts the constraints imposed by the fixed job sequence π and the "route" of each job through the three machines. Let a be the following assignment:

– Operator 1 processes in this order the following operations: $\pi(1,2)$, $\pi(3,1)$, $\pi(3,2)$, 0.5 time units of $\pi(2,1)$, $\pi(2,2)$ and $\pi(1,1)$.
– Operator 2 processes in this order the following operations: $\pi(2,3)$, 1 time unit of $\pi(2,1)$, $\pi(1,3)$ and $\pi(3,3)$.

The Gantt diagram of solution (π, a) is illustrated in Figure 1C. We set the performance levels of the operators to $v_1 = 1$, $v_2 = 0.5$. Thus, if operator h, $h = 1, 2$ handles an operation $\pi(i,j)$, the the processing time of the latter will be $\frac{p_{ij}}{v_h}$. The corresponding value of the maximum lateness is

$$L_{\max}(\pi, a) = \max\{3.6, -0.5, 4.4\} = 4.4.$$

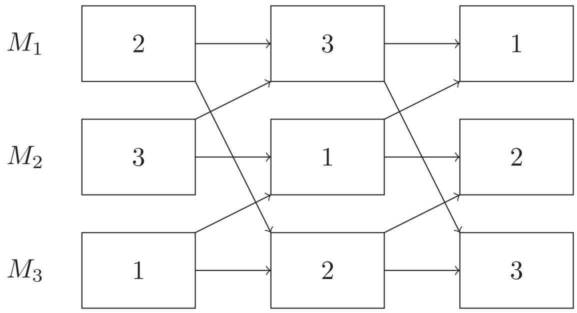
4. Related works

Job shop problems have been extensively studied in the scheduling literature [27]. The reason for that is twofold: first they capture many real-life situations. This makes them very useful on the practical side. Second, they are among the most difficult combinatorial optimization problems. So, their study may provide useful theoretical results for the scheduling theory. Regarding their complexity status, the classical job shop problem, denoted $J_m||C_{\max}$, is $\mathcal{NP}$-hard in the strong sense, even for $m = 2$ [29]. In what follows, we present a brief review of the literature dedicated to the job shop problem with operators.

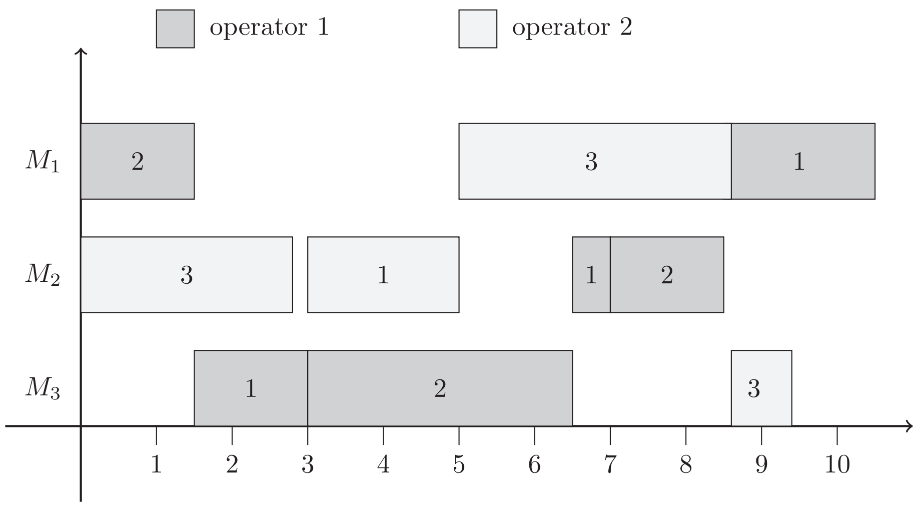
Scheduling problems with operators in general and job shop problems in particular being relatively recent and difficult to solve, we noted that only a few studies dealt with them. The difficulty of these problems

		$j=1$	$j=2$	$j=3$
p_{ij}	$i=1$	2	1.5	1.8
	$i=2$	1.5	1.5	1.4
	$i=3$	1.5	3.5	0.4
d_j		7	9	5

(A) Data associated with instance I.



(B) Precedence constraints generated by the processing order of each job on machines and the fixed sequence π.



Note that operator 2 has a performance level of $\frac{1}{2}$.

(C) The Gantt diagram and the makespan of solution (π, a).

Figure 1. An instance of the $J_m|res\ 1k1, S, f, v_h|L_{\max}$ problem. (A) Data associated with instance I. (B) Precedence constraints generated by the processing order of each job on machines and the fixed sequence π. (C) The Gantt diagram and the makespan of solution (π, a). Note that operator 2 has a performance level of $\frac{1}{2}$.

concerns mainly the modelling of the human resources' characteristics, such as the experience of operators, their qualifications, the variable workload that is assigned to them, etc.

In the classical scheduling models, it is implicitly assumed that the number of operators is the same as the number of machines. Therefore, an operator can operate exclusively the machine to which he is assigned. However, this assumption is rarely true in practice where the number of operators is generally smaller than the number of machines [9, 13].

In the studies dedicated to scheduling problems with operators, the authors defined different models based on various assumptions. These assumptions allowed to model with more or less precision the interactions that exist between human and material resources. Since the work of Vickson [35, 36], the majority of those studies dealt with processing times controllable by the amount of assigned resources. Solving these problems induced to assign a surplus of operators to the jobs in order to reduce the processing time of the latter. Unlike previous works, the model proposed by Cheurfa studied a cyclical flow shop with a number of operators that is less than the number of machines [17]. In his work, the author defines the concept of "sharing" operators and studies a decision problem rather than an optimization problem, in which operators have equivalent performance levels. The intervention of the latter is limited to assembling, controlling and disassembling operations [12].

Analyzing the literature allowed us to note that a fair part of the works that dealt with scheduling with operators was dedicated to identical parallel-machine environments. We may cite for example the works of Zouba [37]. In his Ph.D. thesis, the author studied the simultaneous handling of scheduling the jobs and assigning the operators in an identical parallel-machine environment, with the possibility for an operator to share his time between different machines. In that model, it mainly comes to study the impact of sharing the operators on the processing times of the jobs. In [7], this model was refined by assuming that the ratio between real and theoretical processing times is a linear function of the occupancy rates of jobs by the operator. Even if it does not allow to predict the exact processing times, this is able to forecast the global increase in the overall duration of a schedule [12].

Around the end of the 1990s and the beginning of the 2000s, many works dealt with the OWMM concept (One-Worker-Multiple-Machine), especially in just-in-time production systems. We may cite for example Cheng *et al.* [16] in which the authors studied a two-machine flow shop in order to minimize the makespan when the intervention of the operator is limited to setup operations at the beginning and at the end of jobs. Let us also cite the Ph.D. thesis of Baki where the author studies flow shop and open shop problems with one operator [6]. The latter is in charge of setup operations at the beginning of a sequence of jobs, hence the use of the batch concept consisting in grouping jobs and then scheduling them. Several objectives functions based on completion times and/or due dates were considered. For further details, see *e.g.* [6, 8].

Let us observe that most scheduling problems are $\mathcal{NP}$-hard. The addition of resource constraints may only make them more difficult to solve. As a result, many studies have dealt with special cases; for more details, see *e.g.* [6, 7].

Finally, let us mention the model of flow shop we proposed in [9, 13] with a number of operators smaller than the number of machines, and where jobs need an operator for the total duration of their processing. In that paper, heuristics and a lower bound were proposed. The relative importance of the human and material resources is also studied by comparing a simultaneous approach with a sequential approach. Initially allowing only a change of assignment at the end of a job, this model was extended to take into consideration the free changing mode in [12] and even adapted to minimize the maximum lateness [11].

Regarding the job shop, considering its inherent complexity, relatively few papers were dedicated to its study in the context of integrating human resources. These studies managed to provide high quality solution methods that prove to be quite competitive.

The problem of job shop with operators with a number of operator less than the number of machines was studied by Agnetis *et al.* in [1]. The authors provided complexity results for minimal cases and derived a set of effective solutions approaches that include a pseudo-polynomial dynamic programming approach, a fully polynomial time approximation scheme (FPTAS), an enumeration scheme based of a generalized disjunctive graph that was integrated in a branch-and-bound procedure and a dynamic programming based heuristic.

The dynamic programming algorithm outperformed by far the branch-and-bound and the designed heuristic showed robustness in providing results of fairly good quality.

Let us also mention the genetic algorithm of Mencia *et al.* [24]. The authors devised a schedule generation scheme inspired form that of Giffler and Thompson [20]. Their approach consists in assigning jobs to operators in order to build a feasible solution. Their scheme guaranties the production of an optimal solution. They used that scheme as a decoder for a rather classical genetic algorithm that used a job order crossover [14]. Since the research on this problem was in its early days, the only method they could compare their approach with was that of Agnetis *et al.* [1]. They claim that their genetic algorithm produced competitive results. In addition to that, since it was rather fast, it had a lot of room for improvement. This genetic algorithm was later enhanced in [26] by integrating a weak Lamarckian evolution and a narrowing-search-space technique.

Agnetis *et al.* also proposed two heuristics and a mixed integer programming formulation for a job shop with operators that was used to model a case of handicraft production [2]. Their study showed that they are able to get close-to-optimal solutions by exploiting in their heuristic algorithms the characteristics of the problem. Sierra *et al.* developed other schedule generation schemes that can be used in different settings such as a heuristic search, branching strategies or decoders in evolutionary algorithms. They extended their work on refining their previously developed scheme [24] by making it faster and more efficient. They used the resulting algorithm in an exact approach that included pruning by means of dominance rules [33]. Another study dealt with a car repair job shop with parallel operators and multiple constraints in [31]. After modelling the problem as a mixed linear program, the authors devised a simulated annealing algorithm to solve a case study with distinct cars. Cars require non-identical service time and the repair works have different cost characteristics. The objective is to minimize the total cost. The authors claim that their algorithm displayed high quality results and a high convergence rate.

Finally, let us mention the memetic algorithms of Mencia *et al.* [27]. They devised them with two different schedule builders and integrated a local search step by means of a tabu search procedure. Their neighborhood structure was based on reversing arcs in a disjunctive graph. Their algorithms were tested over an extensive set of instances and produced high quality results in short time.

Before closing this section, let us mention that few papers were published on minimizing the mean flow time; for more details, see *e.g.* [23,25,32]. Also, a new classification scheme was presented for integrated staff rostering and scheduling problems in [28]. It extends existing schemes for project and machine scheduling and provides some elementary reductions with some complexity results. Other papers deal with real-life problems. We may cite for example the work of Gourgand *et al.* [21]. They propose a mathematical model that aims at providing an aid tool which will plan medical exams at hospitals with the assignment of human and material resources. A set of tests is conducted on randomly generated instances and proposes exams plannings over a given horizon. Finally, we may precise that a fair part of the literature was dedicated to the study of robotic production cells, particularly in cyclic flow shop environments. It is observed in [4] that the addition of robots allows a substantial improvement of the production. A literature review on robotic production cells as well as its current challenges can be found in [3, 18, 19, 30].

5. The case with an arbitrary number of operators

In what follows, we derive a complexity result for the case when the number of operators k is arbitrary.

Our proof consists in showing that the following decision problem is $\mathcal{NP}$-complete for arbitrary k. $JSMS(m)$ is a job shop problem with ℓ jobs, $m \geq 2$ machines and an arbitrary number k of operators where the assignment of operators changes according to the free changing mode and S is a fixed job-sequence on the machines. We also assume that the operators have the same work rate, the jobs have all the same route, and the objective is to minimize the makespan which is the overall completion time of a schedule and is denoted $C_{\max}$. Given an integer α, does there exist an assignment $a \in A$ such that $C_{\max}(S, a) \leq \alpha$?

The reduction is constructed from the following parallel-machine decision problem known to be $\mathcal{NP}$-complete [34].

PM refers to an identical parallel machine problem with n jobs and a general precedence constraint graph, where p_h is the processing time of a job h, $1 \leq h \leq n$. The precedence constraints are between the jobs only, and preemption is allowed. Given an integer β, does there exist a schedule π such that $C_{\max}(\pi) \leq \beta$?

Let I be an instance of problem PM. We construct an instance I' of $JSMS(m)$ as follows. From the precedence graph G of I, we derive the precedence graph G' of I' as follows:

- Each job in G corresponds to an operation of a job in G'. The operations in G' have the same processing times as their corresponding jobs in G. In I', we consider jobs with missing operations, *i.e.* some jobs may not have to be processed on all the machines of the shop.
- We choose in G a job without any predecessor, then this job will be the first job on the first machine in G'. In what follows, we name this vertex the root.
- Suppose that a job has $d \geq 3$ successors, then we add $\left\lceil \frac{d}{2} \right\rceil$ dummy vertices each of which will have an inner degree equal to one and an outer degree of at most two. We iterate this procedure until every vertex has at most two successors, namely a left one and a right one. The dummy vertices correspond to operations that have processing times equal to zero.
- Suppose that a job has $d \geq 3$ predecessors, then we add $\left\lceil \frac{d}{2} \right\rceil$ dummy vertices each of which will have an outer degree equal to one and an inner degree of at most two. We iterate this procedure until every vertex has at most two predecessors. The dummy vertices correspond to operations that have processing times equal to zero. We add dummy vertices to impose that vertices in the precedence graph have inner and outer degrees of at most 2. This is needed as an operation in the flow shop problem has at most two successors and two predecessors.
- The number n of jobs of I' is the length of the path from the root to the last vertex that does not have a right successor following a prefix traversal in the right sub-graph; let F be that path. The right successor of each operation represents the operation of the next job on the same machine while its left successor represents the operation on the next machine.
- The number m of machine of I' is the length of the longest path from a vertex of F that picks systematically the next left successor until it finds none.
- Finally, we set $\alpha = \beta$.

It is obvious that this construction can be done in polynomial time. Now, we show that instance I is a yes-instance if, and only if, I' is a yes-instance.

Lemma 5.1. *If I is a yes-instance, then I' is a yes-instance.*

Proof. If I is a yes-instance, we have a schedule π such that $C_{\max}(\pi) \leq \beta$. We could add as many jobs of processing time equal zero as we want without increasing the makespan. The machines of I correspond to the operators of I'. Thus, we could build a solution to I' with $C_{\max}(S, a) \leq \alpha$ by adding the operations corresponding to the dummy vertices. This solution will also meet the precedence constraints. Thus, I' is a yes-instance. □

Lemma 5.2. *If I' is a yes-instance, then I is a yes-instance.*

Proof. If I' is a yes-instance, we have an assignment of operators a and a sequence S such that $C_{\max}(S, a) \leq \alpha$. Since the operators of I' correspond to the machines of I, from the solution of I',we could build a schedule π for I by skipping the operations that have a zero processing time such that $C_{\max}(\pi) \leq \beta$. Since the precedence constraints of I' comprise those of I, such a schedule will meet the precedence constraints of the latter. Thus, I is a yes-instance. □

Theorem 5.3. *Problem $J_m|res\ \ell \cdot 1, S, f|C_{\max}$ is $\mathcal{NP}$-hard if the number of operators is arbitrary.*

Proof. Problem $JSMS(m)$ is clearly in $\mathcal{NP}$. From Lemmas 5.1 and 5.2, instance I of PM has a solution if, and only if instance I' of $JSMS(m)$ has a solution. The result of the theorem is then established. □

For (every operation (i, j) without any successor)
$\{d'_{ij} \leftarrow d_{ij};\}$
Create two lists that are sorted in ascending order of modified due dates d'_{ij} and $b'_{ij} = d'_{ij} - p_{ij}$;
While $(\exists\ (i, j)\ /\ d_{ij}$ non modified and $d_{i'j'}$ modified $\forall (i', j') \in S(i, j))$
{
Choose (i, j);
Look for successors of that operation in the sorted lists;
Compute $\displaystyle\sum_{(i',j')\in S(i,j)} p_{i'j'}(d'_{i''j''}),\ \forall (i'', j'') \in S(i, j)$;
$d'_{ij} \leftarrow$

$$min\left\{d_{ij}, \min_{d'_{(i'',j'')}/(i'',j'')\in S(i,j)} \left\{d'_{(i'',j'')} - \frac{\sum_{(i',j')\in S(i,j)} p_{i'j'}(d'_{i''j''})}{v_1 + v_2}\right\}\right\};$$

For (every $(i', j') \in S(i, j)$)
$\{d'_{ij} \leftarrow min\{d'_{ij}, b'_{i'j'} = d'_{i'j'} - p_{i'j'}\};\}$
Insert d'_{ij} and $b'_{ij} = d'_{ij} - p_{ij}$ in the ordered lists;
}

FIGURE 2. Modification of the due dates.

Corollary 5.4. *Problem $J_m|res\ \ell \cdot 1, S, f|L_{\max}$ is $\mathcal{NP}$-hard if the number of operators is arbitrary.*

Proof. Obvious since $C_{\max}$ is a special case of $L_{\max}$ where all due dates are equal to 0. □

Corollary 5.5. *The problem $J_m|res\ \ell \cdot 1, S, v_h, f|L_{\max}$ is NP-hard if the number of operators is arbitrary.*

Proof. Obvious since $J_m|res\ \ell \cdot 1, S, f|L_{\max}$ is a special case of $J_m|res\ \ell \cdot 1, S, v_h, f|L_{\max}$ where all the work rates are equal. □

6. Well-solvable cases

In this section, we provide polynomial-time algorithms for two restricted cases *viz.* with two operators and one operator, respectively.

6.1. Case with $k = 2$ operators

Theorem 6.1. *The problem $J_m|res\ 2k_h1, S, v_h, f|L_{\max}$ is solvable in polynomial time.*

Proof. The proof is designed through the restriction method. Let us first observe that the processing constraints on operations that are generated by the routing order of jobs and by the fixed processing sequence S form a precedence graph. Thus, our problem can be seen as a two uniform parallel-machine problem with precedence constraints with preemption, denoted as $Q_2|prec, prmp|L_{\max}$, where the machines, the jobs and the precedence graph represent respectively the two operators, the $m \times n$ operations and the precedence relations between them. The possibility of preemption models the free changing mode. Since $Q_2|prec, prmp|L_{\max}$ is solvable in polynomial time [22], it then follows that $J_m|res\ 2k_h1, S, v_h, f|L_{\max}$ can also be solved in polynomial time. □

$\Delta \leftarrow 0$;
$u \leftarrow 2$;
$v \leftarrow 2$;
$T \leftarrow b_2$;
$P \leftarrow 0$;
While $(\Delta < t_{r+1} - t_r)$
{
  While $(T = b_{u-1} + \Delta)$
  {
    $u \leftarrow u - 1$;
    If $(u = 1)$
    $\{b_0 = -\infty;\}$
    $P \leftarrow P + \Delta$;
  }
  While $(T = b_{v+1})$
  {
    $v \leftarrow v + 1$;
    If $(v = z)$
    $\{b_{z+1} = \infty;\}$
  }
  $T_1 \leftarrow (\frac{(b_{u-1}+\Delta)(2-u+v_2)-(v-u+1)T)}{((2-u+v_2)-(v-u+1))}$;
  $T_2 \leftarrow b_{v+1}$;
  $T_3 \leftarrow T + \frac{(t_{r+1}-t_r-\Delta)(2-u+v_2)}{(v-u+1)}$;
  $T' \leftarrow min\{T_1, T_2, T_3\}$;
  $P \leftarrow P + (v - u + 1)(T' - T)$;
  $\Delta \leftarrow \frac{P}{(2-u+v_2)}$;
  $T \leftarrow T'$;
}

FIGURE 3. Construction of fixed intervals.

For the sake of completeness, we present below the procedure that we adapted from the algorithms of Lawler in [22] to solve our problem. The solution is built in three steps. First, we modify the due dates (Fig. 2) then, we build variable scheduling intervals (Fig. 3). Finally, a solution is provided by a priority scheduling algorithm (Fig. 4) in $O(m^2n^2)$. We define the following parameters:

- $d_{ij} = d_j - \sum_{q=i+1}^{m} p_{qj}$, the due date of the ith operation of job j.
- Without loss of generality, we set $v_1 = 1$ and $v_2 = \frac{\min\{v_1,v_2\}}{\max\{v_1,v_2\}}$.
- $p_{ij}(t)$ denotes the minimum time during which operation (i, j) has to be processed before time t for the due date to be met.
- $S(i, j)$ is the set of successors (whether immediate successors or not) of operation (i, j).
- $b_{ij}^{(r)} = d_{ij} - p_{ij}^{(r)}$, where $p_{ij}^{(r)}$ is the remaining time during which operation (i, j) needs to be processed at time t_r. It serves as a priority indicator. The smaller $b_{ij}^{(r)}$ is, the higher the priority of operation (i, j).

The z available operations at time t_r are re-indexed following the increasing order of $b_{ij}^{(r)}$.

Compute the inner degree of every operation (i,j);
Create a queue Q containing the operations with inner degree equal to 0;
Create a list A containing the operations available for processing*;
$r \leftarrow 0$;
While $(Q \neq \varnothing)$
{
    $r \leftarrow r+1$;
    $t_r \leftarrow 0$;
    $A \leftarrow A \cup Q$;
    $Q \leftarrow \varnothing$;
    While $(A \neq \varnothing)$
    {
        $t_{r+1} \leftarrow 0$;
        Run the algorithm in Figure 3 on the operations in A;
        If (the found values (Δ, u, v, T) violate $b_j + \Delta \leq d_j$, $1 \leq j \leq u-1$)
        { Run once more the algorithm in Figure 3 with

$$t_{r+1} = t_r + \min_y \{p_j^y / j = 1, \ldots, u-1\};$$

            $A \leftarrow A \setminus \{\text{operations completed in } [t_r, t_{r+1}]\}$;
            Update the inner degrees of operations;
            Update Q; $A \leftarrow A \cup Q$; $k \leftarrow k+1$;
        }
    }
}

N.B: for the construction of variable intervals, we proceed the same way except for the following cases :

- $T \leq d_j$, $j = u, u+1, \ldots, v$. If $T = d_j$, $j \geq u$, computation stops.
- If u is decremented, we check whether $b_{u-1} + \Delta > d_{u-1}$, if yes, computation stops.

* An operation is said to be available at time t if all of its predecessors have been entirely processed and there remains a positive amount of time during which this operation is to be processed.

FIGURE 4. Algorithm for solving $J_m|res\ 2k_h1, S, v_h, f|L_{\max}$.

Let $x_{ij}^{(r)}$ be the amount of operation (i,j) that is processed within interval $[t_r, t_{r+1}]$.

$$x_{ij}^{(r)} = \begin{cases} \Delta, & q < u, \\ \max\{0, T - b_q^{(r)}\}, & q \geq u, \end{cases}$$

with T, u, v and Δ determined by the algorithm in Figure 3, v being the index such that $b_v^{(r)} \leq T < b_{v+1}^{(r)}$, and q the index corresponding to operation (i,j) after the re-indexation.

While $(B \neq \varnothing)$
{
Choose (i,j) / $S(i,j) = \varnothing$ and $L_{ij} = \min_{(i,j)\in B}\{P(B) - d_{ij}\}$;
$B \leftarrow B \setminus \{(i,j)\}$;
}

FIGURE 5. Solving $J_m|res\ 111, S, f|L_{\max}$.

6.2. Case with one operator

We present here a much simpler algorithm for the case with one operator even though the time complexity remains the same as for the case of two operators.

Theorem 6.2. *The problem* $J_m|res\ 111, S, f|L_{\max}$ *is solvable in polynomial time.*

Proof. We use the same method as in the previous theorem. For the reduction, we use $1|prec, prmp, r_j|L_{\max}$ which is solvable in $O(m^2n^2)$ [5]. □

The algorithm presented in Figure 5 is an adaptation of the algorithm of Baker *et al.* [5] for the $1|prec, prmp, r_j|L_{\max}$ problem; it is in $O(m^2n^2)$. In our case, all the release dates are equal to 0 and, thus, all the operations belong to the same block B and can be processed without idle times. Let $P(B) = \sum_{(i,j)\in B} p_{ij}$, be te sum of all operations in B.

7. Conclusion

In the current economical environment, which is highly competitive, it has become more and more important for the good functioning of companies to have an efficient management of their resources with adequate decision-making systems. Within this context, it is known that classical scheduling models that do not consider operators are of little help in the real world applications [29]. It is necessary to develop models that are more realistic, particularly by taking into consideration human resources, so as to reach higher levels of efficiency in resource-management and, at the same time, narrow the gap between academic research and the real-world.

In this paper, we have studied the assignment of a number of operators that is less than the number of machines in a job shop environment with a given sequence of jobs and a free changing mode in order to minimize the maximum lateness. The operators we considered may have different performance levels. We provided polynomial-time algorithms for two restricted cases *viz.* with two and one operators, respectively. When the number k of operators is arbitrary, we proved the corresponding problem is $\mathcal{NP}$-hard, even if the operators have equivalent performance levels.

For an immediate research, it would be of interest to settle the complexity status of the problem with an arbitrary number of operators: either prove its strong $\mathcal{NP}$-completeness or exhibit a pseudo-polynomial-time algorithm for its resolution. It would also be interesting to develop heuristic methods for the general problem. At last but not least it would be of interest to consider other criteria and/or include additional features to our model such as sequence-dependent setup times and release dates.

References

[1] A. Agnetis, M. Flamini, G. Nicosia and A. Pacifici, A job shop problem with one additional resource type. *J. Scheduling* **14** (2010) 225–237.

[2] A. Agnetis, G. Murgia and S. Sbirilli, A job shop scheduling problem with human operators in handicraft production. *Int. J. Prod. Res.* **52** (2014) 3820–3831.

[3] M. Al-Salem and M. Kharbeche, Throughput optimization for the robotic cell problem with controllable processing times. *RAIRO: OR* **51** (2017) 805–818.

[4] Y.P. Aneja and H. Kamoun, Scheduling of parts and robot activities in a two machine robotic cell. *Comput. Oper. Res.* **26** (1999) 297–312.

[5] K.R. Baker, E.L. Lawler, J.K. Lenstra and A.H.G. Rinnooy Kan, Preemptive scheduling of a single machine to minimize maximum cost subject to release dates and precedence constraints. *Oper. Res.* **31** (1983) 381–386.

[6] M.F. Baki, *Some Problems in One-Operator Scheduling.* Ph.D. thesis, University of Waterloo (1999).

[7] P. Baptiste and A. Munier, Ordonnancement sur machines parallèles avec partage d'opérateurs. In: *10ème Congrès International de Génie Industriel-CIGI 2013* (2013).

[8] P. Baptiste, V. Giard, A. Hait and F. Soumis, Gestion de production et ressources humaines. Presses Internationales Polytechnique (2005).

[9] I. Benkalai, P. Baptiste and D. Rebaine, Ordonnancement d'ateliers de type flow shop avec contrainte d'opérateurs. In: *11ème Congrès International de Génie Industriel-CIGI 2015* (2015).

[10] I. Benkalai, P. Baptiste and D. Rebaine, Ordonnancement d'ateliers de type flow shop avec opérateurs en mode d'affectation libre. In: *17* ème *conférence ROADEF* (2016).

[11] I. Benkalai, P. Baptiste and D. Rebaine, Ordonnancement d'ateliers de type flow shop avec opérateurs en mode d'affectation libre. In: *17ème Conférence ROADEF de la Société Française de Recherche Opérationnelle et Aide à la Décision* (2016).

[12] I. Benkalai, D. Rebaine and P. Baptiste, Assigning operators in a flow shop environment. In: Information Systems, Logistics and Supply Chain (2016).

[13] I. Benkalai, D. Rebaine and P. Baptiste, Scheduling flow shops with operators. *Int. J. Prod. Res.* **57** (2019) 338–356.

[14] C. Bierwirth, A generalized permutation approach to jobshop scheduling with genetic algorithms. *Oper. Res. Spectr.* **17** (1995) 87–92.

[15] J. Blazewicz, K. Ecker, E. Pesch, G. Schmidt and J. Weglarz, Handbook on scheduling: From theory to applications. Springer (2007).

[16] T.C.E. Cheng, G. Wang and C. Sriskandarajah, One-operator-two-machine flowshop scheduling with setup and dismounting times. *Comput. Oper. Res.* **26** (1999) 715–730.

[17] M. Cheurfa, *Gestion des ressources humaines en production cyclique.* Ph.D. thesis, ENSM Saint-Étienne (2005).

[18] M. Dawande, H.N. Geismar, S.P. Sethi and C. Sriskandarajah, Sequencing and scheduling in robotic cells: Recent developments. *J. Scheduling* **8** (2005) 387–426.

[19] H.N. Geismar, M. Pinedo and C. Sriskandarajah, Robotic cells with parallel machines and multiple dual gripper robots: a comparative overview. *IIE Trans.* **40** (2008) 1211–1227.

[20] B. Giffler and G.L. Thompson, Algorithms for solving production scheduling problems. *Oper. Res.* **8** (1960) 487–503.

[21] N. Grangeon M. Gourgand and N. Klement, Activities planning and resources assignment on distinct places: a mathematical model. *RAIRO: OR* **49** (2015) 79–98.

[22] E.L. Lawler, Preemptive scheduling of precedence-constrained jobs on parellel machines. In: *Deterministic and Stochastic Scheduling, Proceedings of the NATO Advanced Study and Reasearch Institute on Theoretical Approaches to Scheduling Problems. D. Reidel publishing Co.* (1982) 101–123.

[23] C. Mencia, M.R. Sierra, M.A. Salido, J. Escamilla, and R. Varela, Solving the job shop scheduling problem with operators by depth-first heuristic search enhanced with global pruning rules. *Artif. Intell. Commun.* **28** (2015) 365–381.

[24] R. Mencia, M.R. Sierra, C. Mencia and R. Varela, Genetic algorithm for job shop with operatos, edited by J.M. Ferrandez, J.R. Alvarez Sanchez, F. de la Paz and F.J. Toledo. In: Vol. 6687 of *New challenges on bioinspired applications. IWINAC. Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg (2011).

[25] C. Mencia, M.R. Sierra and R. Varela, An efficient hybrid search algorithm for job shop scheduling with operators. *Int. J. Prod. Res.* **51** (2013) 5221–5237.

[26] R. Mencia, M.R. Sierra, C. Mencia and R. Varela, A genetic algorithm for job shop scheduling with operators enhanced by weal lamarckian evolution and search space narrowing. *Nat. Comput.* **13** (2014) 179–192.

[27] R. Mencia, M.R. Sierra, C. Mencia and R. Varela, Memetic algorithms for the job shop scheduling problem with operators. *Appl. Soft Comput.* **34** (2015) 94–105.

[28] M. Paul and S. Knust, A classification scheme for integrated staff rostering and scheduling problems. *RAIRO: OR* **49** (2015) 393–412.

[29] M. Pinedo, Scheduling: Theory, Algorithms and Systems. Prentice Hall (2002).

[30] S.P. Sethi, C. Sriskandarajah, G. Sorger, J. Blazewicz and W. Kubiak, Sequencing of parts and robot moves in a robotic cell. *Int. J. Flexible Manuf. Syst.* **4** (1992) 331–358.

[31] N. Shivasankaran, S.P. Kumar, G. Nallakumarasamy and K. VEnkatesh Raja, Repair shop job scheduling with parallel operators and multiple constraints using simulated annealing. *Int. J. Comput. Intell. Syst.* **6** (2013) 223–233.

[32] M.R. Sierra, C. Mencia and R. Varela, Optimally scheduling a job shop with operators and total flow time, edited by J.A. Lozana, J.A. Ganaz, J.A. Moreno. In: Vol. 7027 of *Advances in Artificial intelligence. CAEPIA. Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg (2011).

[33] M.R. Sierra, C. Mencia and R. Varela, New schedule generation schemes for the job shop problem with operators. *J. Intell. Manuf.* **26** (2015) 511–525.

[34] J.D. Ullman, Complexity of sequencing problems. In: Computer and Job/Shop Scheduling Theory. Wiley & Sons, Inc. New York (1976).

[35] R.G. Vickson, Choosing the job sequence and processing times to minimize total processing plus flow cost on a single machine. *Oper. Res.* **28** (1980) 1155–1167.

[36] R.G. Vickson, Two single machine sequencing problems involving controllable job processing times. *AIIE Trans.* **12** (1980) 258–262.

[37] M. Zouba, *Ordonnancement de machines parallèles identiques avec des contraintes de ressources humaines*. Ph.D. thesis, École Polytechnique Montréal (2009).