Assigning operators in a flow shop environment

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

[1] Université du Québec à Chicoutimi, Saguenay (Québec), Canada G7H-2B1.
{imene.benkalai1,drebaine}@uqac.ca
[2] École Polytechnique de Montréal, Montréal (Québec), Canada H3T-1J4.
pbaptiste@polymtl.ca

Abstract. This paper deals with the problem of assigning operators, within a free changing mode, in a flow-shop environment given an order for processing the set of tasks. We seek a schedule that minimizes the overall finish time, known as the makespan. Within this model, a task needs an operator during the entire duration of its processing. We show that the problem is NP-hard in the strong sense if the number of operators is greater than 3, and exhibit polynomial time algorithms for the cases involving one and two operators, respectively.

Key words: Complexity, Flow shop, Free changing mode, Operators.

1 Introduction

The integration of human resources as well as their different characteristics in scheduling problems is quite a recent field of research in scheduling theory. This is due to the fact that taking the human component into consideration when building a schedule makes the corresponding problem more difficult to solve.

Numerous links exist between human and material resources in production systems, especially on the operational level [21]. Moreover, the human actor has always played a crucial role in economy. The reason is that, on the one hand, he is consumer and producer at the same time, and, on the other hand, he is able to face unpredictable events despite the advances that have been achieved in production equipments [3]. Furthermore, the classical scheduling models are of little help in the real world applications [16]. All of this to say that it is important to take into account the human component when making decisions related to the management of resources in order to reduce the gap between theoretical research and practical applications.

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 modeling more accurately the constraints encountered in 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; this is rarely true in real systems. This paper deals with scheduling flow shops in which the number of operators is less than the number of machines. We assume that the assignment of operators to machines is done in a free changing mode. This means that the assignment of an operator to a machine may be changed at any time. We also assume the ordering of processing the jobs through the machines is known in advance.

This paper is organized as follows. A review of related works is the subject of Section 2. Section 3 presents a description of the problem under study. Section 4 is devoted to the NP-hardness of the problem, and the presentation of two polynomial-time algorithms for the case of two and one operators, respectively. Finally, we present in Section 5 a conclusion as well as perspectives for future work.

2 Problem description

In a flow shop problem, n tasks are to be scheduled on m machines in starting on machine 1, then on machine 2 and so on until machine m. In our model, we assume that a task requires the presence of one of the k operators during its entire processing. In other words, a machine will remain inactive as long as there is no available operator to run it. Thus, the processing times of tasks are not directly affected by the intervention of operators. The impact of sharing operators is rather modeled by waiting times that are due to the fact that the number of operators is less than the number of machines. The given data p_{ij} and C_{ij} represent respectively the processing time and the finish time of operation (i, j) *i.e.* the processing of job j on machine i.

We assume that we have ℓ types of operators, k_h, $h = 1, \dots, \ell$ operators of each type with $\sum_{h=1}^{\ell} k_h = k$. An operator of type ℓ has a work rate v_ℓ [1]. 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 task at any time to process another task.

The goal is to find a feasible schedule of tasks so as to minimize the makespan. Since we are dealing with the standard[2] flow shop problem with operators, we note the given processing ordering of tasks $\pi = (\pi_{ij})$, $i = 1, \dots, m$, $j = 1, \dots, n$. The goal is to find an assignment of operators a^* such that

$$C_{\max}(\pi, a^*) = \max_{a \in A} C_{\max}(\pi, a),$$

where A denotes the set of feasible assignments, a^* is in A, and $C_{max}(\pi, a)$ is the overall completion time of the ordering π under the assignment a.

The makespan of a solution (π,a) is computed as follows. Let $C(\pi(i, j), a)$ be the completion time of the task at the j^{th} position on machine i under the assignment a and t_{hqj} the time at which the operator h in charge of the operation

[1] The obtained results are identical for cases where operators have equal performance levels.

[2] Without permutation constraints.

at the j^{th} position on machine q becomes available. Therefore, $C_{\max}(\pi, a) = C(\pi(m, n), a)$, where $C(\pi(q, j), a)$ of the task $\pi(q, j)$ on machine q under the assignment a is computed by the following recursive formula.

$$
\begin{aligned}
C(\pi(1,1),a) &= t_{h11} + p_{1\pi(1,1)} && , \\
C(\pi(1,j),a) &= \textstyle\sum_{q=1}^{j-1}(p_{1\pi(1,q)}) + \sum_{f=2}^{j} t_{h1j} - C(\pi(1,j-1),a) + p_{1\pi(1,j)} && ;\ j = 2,\dots,n, \\
C(\pi(q,1),a) &= \max\{C(\pi(q-1,j'),a), t_{hqj}\} + p_{q\pi(q,1)} && ;\ q = 2,\dots,m, \\
C(\pi(q,j),a) &= \max\{C(\pi(q-1,j'),a), C(\pi(q,j-1),a), t_{hqj}\} + p_{q\pi(q,j)} && ; \\
& \quad q = 2,\dots,m; j = 2,\dots,n,
\end{aligned}
$$

where j' is the position at which operation (q, j) is scheduled on machine q, $q = 2, \dots, m$, $j = 1, \dots, n$.

3 Related works

Successively named k-level production system then flow shop [10], the basic problem, as studied in [12], consists in scheduling n tasks on m machines in order to minimize a given objective. The tasks go through the system by passing on each machine exactly once and following the same passage order (machine 1, then machine 2 and so on until machine m). Subsequently, many assumptions were added and/or relaxed to the original model, which lead to the design of many models, especially since the 80s [10].

The standard flow shop problem, denoted $F_m||C_{max}$, is NP-hard in the strong sense for $m \geq 3$ [9].

While we counted more than 1200 publications for the permutation[3] flow shop in 2006 [10], we notice that the standard flow shop has been little studied compared to the former. This is mainly due to the relative simplicity of the permutation flow shop when compared to its standard version. However, it was shown in [17] that for certain families of instances, the ratio between the best permutation solution and the best standard solution when minimizing the makespan is greater than $\frac{1}{2}\sqrt{m}$. Therefore, considering the standard version of the flow shop may help to achieve significant profits [4]. For further details, see e.g [13, 15].

In what follows, we present a brief review of the literature dedicated to the flow shop problem with operators.

Scheduling problems with operators in general and flow 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 concerns mainly the modeling of the human resources' characteristics, such as the experience of an operator, his qualifications, the variable workload that is assigned to him, etc.

[3] The permutation constraint enforces that the tasks are processed by the machines following the same order.
[4] Nevertheless, we recall that for $m \leq 3$, the two models are identical.

In the classical scheduling models, it is implicitly assumed that the number of operators is equal to 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 [5].

In each and every study that was 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 Vickson's works in 1980 [19, 20], the majority of studies that were undertaken in the field of scheduling with operators dealt with processing times that were controllable by the amount of assigned resources. Solving these problems induced to assign a surplus of operators to the tasks 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 [8]. In his work, the author defines the concept of "sharing" operators and studies a decision problem rather than an optimization problem, with operators with equivalent performance levels. The intervention of the latter is limited to assembling, control and disassembling operations.

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 [21]. In his PhD thesis, the author studied the simultaneous handling of scheduling the tasks and assigning the operators in an identical parallel-machine environment with a number of operators that is less than the number of machines and 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 tasks. In [4], that model was refined by assuming that the ratio between real and theoretical processing times is a linear function of the occupancy rates of tasks by the operator. Even if it doesn't allow to predict the exact processing times, this new model allows to forecast the global increase in the overall duration of a schedule.

Around the end of the 1990s and the beginning of the 2000s, many works dealt with the OWMM concept (One-Worker-Multiple-Machine) given its popularity, especially in just-in-time production systems. We may cite for example Cheng*et al.* [7] 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 tasks. Let us also cite the PhD thesis of Baki where the author studies flow shop and open shop problems with one operator [2]. The latter is in charge of setup operations at the beginning of a sequence of tasks, hence the use of the batch concept consisting in grouping tasks and then scheduling them. Several objectives functions based on completion times and/or due dates were considered. For further details on scheduling with human resources, see e.g. [2, 3].

Let us observe most scheduling problems are NP-hard. The addition of resource constraints may only make them more difficult to solve. As a result, many studies dealt with special cases such as restricted number of machines to $m = 2$. For more details, see e.g. [2, 4].

Before closing this section, let us mention the model of flow shop we proposed in [5] with a number of operators smaller than the number of machines, and where tasks need an operator for the total duration of their processing. In that paper, we designed heuristics, proposed a lower bound, and compared the relative importance of the human and material resources by comparing a simultaneous approach with a phase-by-phase approach.

4 Complexity results and polynomial-time algorithms

As with any combinatorial optimization problem, we first start the analysis with its complexity status. In what follows, we denote the problem under study by $F_m|res\ \ell k_h 1, S, v_h|C_{\max}$ in accordance with the notation proposed by Blazewicz *et al.* [6] to which we add the symbol S to indicate that the assignment is made under a given ordering of tasks S. In other words, the problem that we study is that of scheduling n tasks in an m-machine flow shop with ℓ types of operators, k_h, $h = 1, \ldots, \ell$ operators of each type with $\sum_{h=1}^{\ell} k_h = k$, and a given ordering of tasks. The goal is to minimize the makespan.

4.1 The k-operator case, $k \geq 3$

Theorem 1. *The problem $F_m|res\ \ell k_h 1, S, v_h|C_{\max}$ is NP-hard in the strong sense for $k \geq 3$ even for the case in which all the v_h are equal.*

Proof. The NP-hardness of the problem is showed via the restriction method. Let us first note that the ordering constraints imposed by the processing order of tasks on the machines as well as by S form a precedence graph. Therefore, the problem can be seen as a uniform-parallel-machine problem with precedence constraints with preemption, denoted as $Q_k|prec, prmp|C_{\max}$, where the machines, the tasks and the precedence graph represent respectively the k operators, the $m * n$ operations, and the precedence relations between these operations. This problem is NP-hard in the strong sense for $k \geq 3$. Indeed,

$$P_k|p_j = 1, prec, prmp|C_{max} \propto P_k|prec, prmp|C_{\max} \propto Q_k|prec, prmp|C_{\max}$$

and $P_k|p_j = 1, prec, prmp|C_{\max}$ is NP-hard in the strong sense [18]. The result of the theorem is thus established.

Let us note that for cases where v_h are all equal, we use $P_k|prec, prmp|C_{\max}$ for the complexity proof with a similar argument.

4.2 The two-operator case

Theorem 2. *The problem $F_m|res\ 2k_h1, S, v_h|C_{\max}$ is solvable in polynomial time.*

Proof. We use the same proof technique as we previously did. We use for the reduction the $Q_2|prec, prmp|C_{\max}$ problem, which is solvable in polynomial time. Indeed,

$$Q_2|prec, prmp|C_{\max} \propto Q_2|prec, prmp, r_j|L_{\max}$$

and $Q_2|prec, prmp, r_j|L_{\max}$ is solvable in $O(n^2)$ [14].

The *level* $L_s(T_{S(i,j)})$ of a task $T_{S(i,j)}$[5] at the time s is the minimum time that is necessary for processing the said task as well as all its successors from the time s forward. For our problem, $L_s(T_{S(i,j)}) = \left(\sum_{x=i}^{m}\sum_{y=j}^{n} p_{x,S(x,y)}\right) - p_{i,S(i,j)}$.

The algorithm in $O(m^2n^2)$ presented on Figure 1 is an adaptation of the algorithm of Horvath *et al.* [11]. It provides a solution with a minimum for $k = 2$ in the case where operators are identical but also when they have different performance levels. In practice, we would first have to assign values to the speeds of operators based on their performance levels before applying the algorithm.

$s \leftarrow 0$;
$h \leftarrow$ the number of free operators at time s;
$j \leftarrow$ the number of task at the highest level at time s;
Do
{
If($j \leq h$)
{ Assign the j fastest operators to the j tasks; }
Else
{ The j tasks are processed at the same rate by the h operators(*); }
If($\exists$ free operators)
{ Assign the tasks of next highest level; }
If(($\exists\ T$ that ends at t) or ($\exists\ T, T'\ /\ L_s(T) > L_s(T')$ and $L_t(T) = L_t(T')$))
{ $s \leftarrow t$; }
} **While**(Not all tasks have been processed)
To build a solution with the free assignment-changing mode, the j tasks that share the h operators during the steps (*) receive the same processing amount from the said operators;
Each shared interval is divided into j sub-intervals;
Each of the j tasks is scheduled in h sub-intervals, each time with a different operator;

Fig. 1. Algorithm for solving $F_m|res\ 2k_h1, S, v_h|C_{max}$.

[5] The task at the j^{th} position on the i^{th} machine in the given ordering S.

4.3 The one-operator case

Theorem 3. *The problem $F_m|res\ 111, S|C_{\max}$ is solvable in polynomial time.*

Proof. We use the same proof technique as we previously did. We use for the reduction the $1|prec, prmp|C_{\max}$ problem which is solvable in polynomial time. Indeed,

$$1|prec, prmp|C_{max} \propto 1|prec, prmp, r_j|L_{max}$$

and $1|prec, prmp, r_j|L_{\max}$ is solvable in $O(n^2)$ [1].

As we previously mentioned, the OWMM (One Worker Multiple Machine) concept is very popular, especially in just-in-time production systems and is often encountered in practice.

In the case of one operator, the sum of all the processing times $\sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}$ is an obvious lower bound. We notice that every *non delay*[6] has a makespan equal to this lower bound. Its optimality is hence established. Such a solution can be computed in $O(mn)$.

An optimal scheduling policy could be to process without interruption all the tasks of the first machine, then those of the second machine and so on until machine m.

5 Conclusion and future works

In the current economical environment, which is strongly competitive, it becomes important for the good functioning of companies to have an efficient management of their resources with adequate decision-making systems. In this context, and knowing that classical scheduling models that do not consider operators are of little help in the real world applications [16], it is necessary to develop models that are more realistic, particularly by taking into consideration human resources, so as to reach a higher level of efficiency in resource-management and at the same time reduce the gap between academical research and its real-world applications.

In this paper, we have studied the assignment of a number of operators that is less than the number of machines in a flow-shop environment with a given ordering of tasks and a free assignment-changing mode in order to minimize the makespan. We studied the complexity of our problem in the case of operators who possess different performance levels. This problem is NP-hard in the strong sense, except for the cases with one and two operators for which we provided a polynomial-time solution method. Thus, it would be interesting to develop heuristic methods for solving cases with $k (k \geq 3)$ operators and also to test the provided algorithms on instances driven from real-world situations. It would also be of interest to consider other objectives that are important in practice such as the mean flow time and the maximum lateness.

[6] A solution in which no machine is left idle when there are both an available operator and an available task.

References

1. 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. *Operations Research*, 31(2):381–386, 1983.
2. M.F. Baki. *Some Problems in One-Operator Scheduling*. PhD thesis, University of Waterloo, 1999.
3. P. Baptiste, V. Giard, A. Hait, and F. Soumis. *Gestion de production et ressources humaines*. Presses Internationales Polytechnique, 2005.
4. P. Baptiste and A. Munier. Ordonnancement sur machines parallèles avec partage d'opérateurs. In *10ème Congrès International de Génie Industriel-CIGI 2013*, 2013.
5. 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.
6. J. Blazewicz, K. Ecker, E. Pesch, G. Schmidt, and J. Weglarz. *Handbook on scheduling : From theory to applications*. Springer, 2007.
7. T.C.E. Cheng, G. Wang, and C. Sriskandarajah. One-operator-two-machine flowshop scheduling with setup and dismounting times. *Computers & Operations Research*, 26:715–730, 1999.
8. M. Cheurfa. *Gestion des ressources humaines en production cyclique*. PhD thesis, ENSM Saint-Étienne, 2005.
9. M.R. Garey, D.S. Johnson, and R. Sethi. The complexity of flowshop and jobshop scheduling. *Mathematics of Operations Research*, 1:117–129, 1976.
10. J.N.D. Gupta and E.F. Stafford Jr. Flowshop scheduling research after five decades. *European Journal of Operational Research*, 169:699–711, 2006.
11. E. C. Horvath, S. Lam, and R. Sethi. A level algorithm for preemptive scheduling. *J. Assoc. Copmut. Mach.*, 24(1):32–43, 1977.
12. S.M. Johnson. Optimal two and three-stage production schedules with setup times included. *Naval research logistics quarterly*, 1(1):61–68, 1954.
13. C. Koulamas. A new constructive heuristic for the flowshop scheduling problem. *European Journal of Operational Research*, 105:66–71, 1998.
14. 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*, pages 101–123. D. Reidel publishing Co., 1982.
15. C. J. Liao, L. M. Liao, and C. T. Tseng. A performance evaluation of permutation vs. non-permutation schedules in a flowshop. *International Journal of Production Research*, 44 (20):4297–4309, 2007.
16. M. Pinedo. *Scheduling : Theory, algorithms and systems*. Prentice Hall, 2002.
17. C.N. Potts, D.B. Shmoys, and D.P. Williamson. Permutation vs. non-permutation flowshop schedules. *Operations Research Letters*, 10:281–284, 1991.
18. J. D. Ullman. Complexity of sequencing problems. In *Computer and Job/shop scheduling theory*. Wiley & Sons, Inc. New York, 1976.
19. R.G. Vickson. Choosing the job sequence and processing times to minimize total processing plus flow cost on a single machine. *Operations Research*, 28(5):1155–1167, 1980.
20. R.G. Vickson. Two single machine sequencing problems involving controllable job processing times. *AIIE transactions*, 12(3):258–262, 1980.
21. M. Zouba. *Ordonnancement de machines parallèles identiques avec des contraintes de ressources humaines*. PhD thesis, École Polytechnique Montréal, 2009.