A linear programming approach for hybrid flexible flow shop with sequence-dependent setup times to minimise total tardiness

Farouk Yalaoui, Soukaina Oujana, Lionel Amodeo

Farouk Yalaoui, Soukaina Oujana, Lionel Amodeo. A linear programming approach for hybrid flexible flow shop with sequence-dependent setup times to minimise total tardiness. 17th IFAC Symposium on Information Control Problems in Manufacturing, Jun 2021, Budapest, Hungary. hal-03258772

HAL Id: hal-03258772
https://hal-utt.archives-ouvertes.fr/hal-03258772
Submitted on 11 Jun 2021

Proceedings of the 17th IFAC Symposium on
Information Control Problems in Manufacturing
Budapest, Hungary, June 7-9, 2021

A linear programming approach for hybrid flexible flow shop with sequence-dependent setup times to minimise total tardiness.

S. Oujana*. F. Yalaoui*L. Amodeo**

**Brodart packaging, 42 Rue de la Paix, 10000 Troyes*
(e-mail : Soukaina.oujana@utt.fr).
***Laboratory of Industrial Systems Optimization (LOSI), University of Technology of Troyes,*

12 rue Marie Curie, CS42060,10004 Troyes Cedex, France (e-mail :
farouk.yalaoui@utt.fr, lionel.amodeo@utt.fr)

Abstract: In this study, we consider a particular version of the hybrid flexible flow shop (HFFS) scheduling problem, inspired from a real-life application in a printing industry. The considered problem is a variation of the classical Flow shop problem, in which specific constraints are jointly considered, such as non-identical parallel machines, sequence-dependent setups on machines, machine eligibility, and precedence constraints among jobs, in order to minimize the total tardiness time. After a problem description, a mathematical model, in form of mixed integer linear programming (MILP) model, that incorporates these aspects is developed and evaluated using ILOG Cplex software.

Keywords: Hybrid flexible flow shop, Linear programming, Mathematical model, Optimization.

---

1. INTRODUCTION

A flow shop problem refers to a multi-stage process in which each stage has a single processor. The flow shop occurs for repeated production, when all jobs visit the stages in the same order with the same processing operations.

However, in real world applications, there are many variations of this traditional flow shop structure, one of them is the hybrid flow shop (HFS), in which stages may consist of several identical or non-identical processors in parallel (at least one stage with more than one processor). The purpose is to balance capacity and cycle time among stages, as well as reduce bottleneck issues (Zhang et al., 2021). Theoretically, the classical flow shop assume that all jobs need to visit all stages. However, in practice, each job might skip stages. HFS scheduling problem with stage skipping is also called hybrid flexible flow shop problem (HFFS). This ability of skipping stages, enhances the performance of the model for adapting to the real industrial environments. Applications of such industrial environments, is found in all kinds of real world applications, including food processing industry, ceramic tile manufacturing, the processing of wood, paper, textile, steel (Long et al., 2018a) and robotics (Batur et al., 2016), (Naderi et al. 2014).

This paper considers a hybrid flexible flow shop scheduling problem (HFFS), inspired from a real-life application in packaging industry. It is a combination of the classical flow shop scheduling problem and the parallel machines scheduling problem. Indeed, the purpose is to determine a schedule which optimizes a given objective function.

The problem under consideration is NP-hard, since even the HFS restricted to two processing stages, even in the case when one stage contains a single machine and the other one two machines, is NP-hard (Ruiz & Maroto, 2006).

Note that This research is motivated by a practical need for operations scheduling at a packaging company in France. Since the firm adopt a Make to order strategy, it is important to minimize the total delay, so that the process can operate effectively, and the initial delivery due date can be respected.

The process under study is characterized by several features, these include: unrelated parallel machines at each stage, sequence dependent setup times, machine eligibility, precedence constraints among jobs and job splitting,

By adding sequence-dependent setup times (ST_{sd}) to our problem, it becomes more functional. Because in practice when the execution of a job is finished on a machine, some setups are required for next jobs. ST_{sd} depends on the machine, the job that the machine was processing and the one to be processed An example of ST_{sd} in the studied industry, after printing with a black ink, deep cleaning must be performed if the job that comes next is printed with a white ink, while less cleaning is necessary if the next job is to print in dark grey. Setup times may involve operations that have to be performed on machines other than cleaning, such us adjustments to machines, fixing or a replacing a part of a machine.

Another important feature of our problem is that each job can be split into job sections, which can be processed simultaneously on different machines. The difference between job splitting and preemption is that job sections cannot be processed on different machines simultaneously if only preemption is allowed (Nait Tahar et al., 2006).

In summary, the main contributions of this study can be described as follows: considering stage skipping, sequence dependent setup times and all the above cited restrictions and characteristics, can satisfy the production requirements of several FHFS packaging plants; Also, a new formulation along with a mixed integer mathematical model for this problem that tackles highly realistic HFFS scheduling problem, is proposed. To the best of our knowledge, no works exists that jointly consider the set of realistic constraints included in the problem formulation of this paper.

The remainder of this paper is organized as follows. Section 2 gives a comprehensive review of the existing literature about the problem considered in this study. The production process and the scheduling problem are described in Section 3. Section 4, develops the mixed integer linear programming (MILP) mathematical model for solving the HFFS problem with total tardiness criterion. Numerical results are reported in Section 5. Finally, the last section concludes the paper and gives future research directions.

2. LITTERATURE REVIEW

The first studies on HFFS problem dates from the seventies through the work of Salvador (Salvador, 1973). This type of problem has received continuous interest from researchers.

Many works have been performed on the subject, various constraints have been taken into account and different criteria have been studied. Many solution algorithms have been presented. They can be divided into three categories: exact algorithms, heuristics and meta-heuristics.

(Kochhar & Morris, 1987) provide a local search approach to a realistic flexible flow line problem environment with setups, buffer capacities, blocking, starvation, breakdowns and downtimes. (Kurz & Askin, 2004) develop new or modified heuristics to solve a flexible flow line problem with identical parallel machines, non-anticipatory sequence-dependent setup times and the objective of makespan minimization. (Ruiz & Stützle, 2008) proposed two iterated greedy heuristics for a complex flowshop problem that results from the consideration of sequence dependent setups, considering two different optimization objectives, the minimization of both makespan and the total weighted tardiness. (Ruiz et al., 2008) propose a MIP model and some heuristics for a HFFS problem with sequence-dependent setup times, machine lags, release dates, machine eligibility and precedence relationships among jobs. The optimization criterion considered is the minimization of the makespan. (Bashir Naderi et al., 2010) study the HFFS with sequence dependent setups. The optimization criterion considered is makespan minimization. They propose two advanced algorithms that specifically deal with the flexible and setup characteristics of this problem. (B. Naderi et al., 2014) provide four different MILP models. Besides these later, they propose a novel hybrid particle swarm optimization algorithm equipped with an acceptance criterion and a local search heuristic. Long et al. 2018b, address a realistic hybrid flow shop scheduling problem with stage skipping and adjustable processing time in steelmaking. The optimization objectives include minimizing the makespan, the total waiting times and the deviation of the processing time. Khare and Agrawal 2019 study a HFS problem with sequence-dependent setup times to minimize total weighted earliness and tardiness. three metaheuristics were proposed, namely: hybrid squirrel search algorithm (HSSA), opposition-based whale optimization algorithm (OBWOA), and discrete grey wolf optimization (DGWO). Another scheduling problem from industry is presented recently by (Burcin Ozsoydan & Sağir, 2021), they provide a learning iterated greedy search metaheuristic to minimize the maximum completion time in a hybrid flexible flowshop problem with sequence dependent setup times.

A fair amount of the research that has been performed on HFFS scheduling has focused on a variety of objectives, ranging from minimizing the maximum completion time, the maximum tardiness and earliness and the total waiting time. Although many realistic considerations and constraints are addressed in several papers in literature, however very few papers consider such realistic constraints jointly.

3. PROBLEM STATEMENT

Our study is conducted in a packaging printing company based in France. Which produces, converts and prints flexible packaging. Some examples of products manufactured are: pouches, reels, sheets and labels. Each product goes through several processing operations depending on its product family or manufacturing operating range, such as printing, coating, perforation and winding. The task of planning and scheduling the work is very complex, a perfect knowledge of the problem is necessary to assist in these tasks.

For clarity of exposition, the studied configuration can be characterized as a make to order environment, its main characteristic is a great diversity of jobs to be performed (Oujana et al., 2021) (Silver et al., 2016), the average annual scheduled jobs count more than 400 different references. The number of orders to be scheduled varies according to overall economic conditions, it is usually over 20 and can easily reach

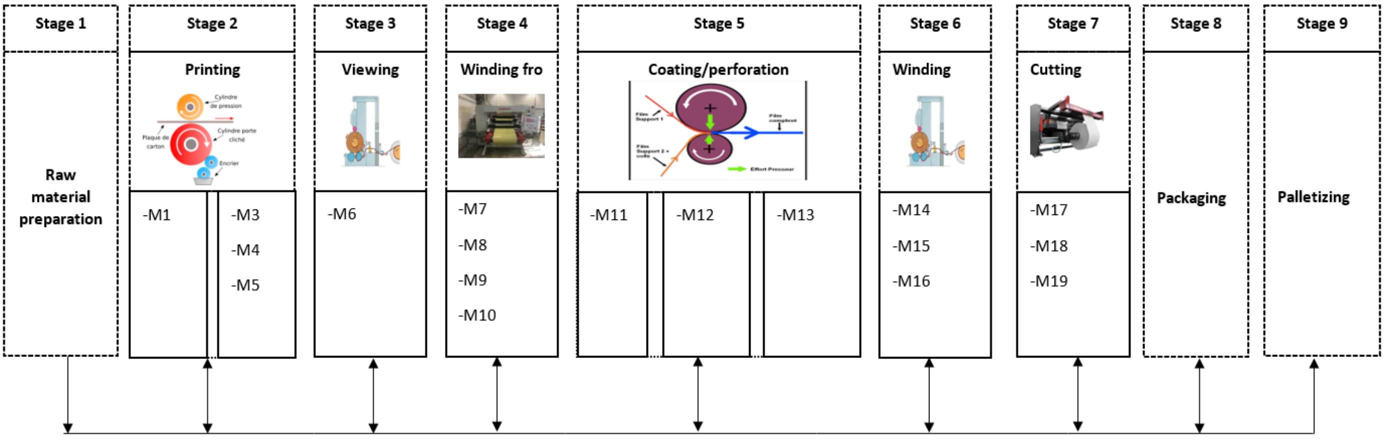

Fig. 1. The diagrammatic production system for a packaging printing company.

100 jobs per week. This allows to define the instances to be treated as medium and large ones.

In order to describe the terms and characteristics of production system, (Fig. 1) shows macro process chart of the process under study, which consist of 9 stages. The boxes in the map represent the processing steps and the number inside the box is the number of machines at each process. The machines are grouped by stage E (e=1,…,9). Each stage E is made up of a set of me machines (me ⩾1) that do the same function, of the same production capacity or of different production capacity (Oujana et al., 2021), called mixed parallel machines (identical and / or not identical). This process is characterized by flexibility, where stages might be skipped i.e., not all stages must be visited by all the jobs, and also sequence dependent setup times.

With the above features we can characterize the production problem as a flowshop with unrelated parallel machines at each stage, commonly known as a hybrid flowshop denoted using the classical notation $\mathrm{HFF_{19}|Prec, ST_{sd}, d_i|\sum_{j=1}^{n} t_i}$ (Graham et al., 1979). Since respect of the delivery deadline is important, we choose to use tardiness minimization as the scheduling criterion.

4. PROBLEM FORMULATION

In this section, we first describe the problem definition by introducing the notations that will be used throughout this paper. Recall that the HFFS problem considered in this paper has three main characteristics that are jointly considered: hybrid setting, where there are parallel machines at each stage, flexibility, where stages might be skipped and sequence dependent setups. We note here that the skipping is encoded in the process plan input.

The problem considered consists of scheduling N jobs (i = 1,…, N) on m machines (k = 1,…, m). It is assumed that all the jobs and the machines are available on time 0. The machines are grouped by stage E (e = 1,…, E). Each stage is made up of a set of parallel machines (identical and / or not identical). A job i consists of a number n of operations, each operation O_{ij} has an execution time on machine k. note that P_i denotes the execution time of job i.

$$P_i = \sum_{j \in Ji} P_{ij} \tag{1}$$

A setup time $\mathrm{st_{iji'j'k}}$ is necessary to pass from the execution of an operation O_{ij} to operation $O_{i'j'}$ on machine k. In our case the setup time of a job is a function of the preceding job on the machine and therefore the processing sequence of the jobs. $\mathrm{Sd_{st}}$ have significant implications on shop performance. Some important articles dealing with this type of configuration have been published (Szwarc & Gupta, 1987) (B. Naderi et al., 2009) (Ruiz & Maroto, 2006).

The assumptions considered for the studied configuration are as follows:

- ✓ The number of jobs, their processing times and the due dates are known beforehand.
- ✓ Pre-emption is not allowed.
- ✓ Each machine has a capacity and can only process one job at a time.
- ✓ Each job can only be processed on one machine at a time.
- ✓ The intermediate stock capacity (buffer) between the production stages is unlimited.
- ✓ transport times are not taken into account.

The notations used are as follows:

Indices:
i,i': index for jobs where i,i'∈ {1,…,N}.
j: Index for operations.
k: Index for machines where k ∈ {1,…,m}.

Parameters:
M: number of all material resources.
N: number of jobs to be scheduled.
J_i: set of operations of job i ∈N.
P_i: Processing time job i ∈N.
d_i: due date of job i ∈N.
$m_j \subset$ M: set of material resources that can perform the operation j ∈ji.
$\mathrm{st_{iji'j'k}}$: setup time to pass from the execution of an operation O_j to operation $O_{j'}$ on machine k.
$\mathscr{M}$: A very large number.
$\mathrm{m_{ij} \cap m_{i'j'}}$: Set of machines on which operations j of job i and j' of job i' can be processed.

Decision variables:

x_{ijk} = 1 if the operation O_{ij} is assigned to the material resource k.
0 otherwise.
$y_{iji'j'k}$ = 1 if the operation O_{ij} is processed before the operation $O_{i'j'}$ on the material resource k.
0 otherwise.
s_{ijk} = Starting time of the operation O_{ij} on machine k.
c_{ijk} = Completion time of the operation O_{ij} sur la machine k.
c_i = Completion time of job i.

5. THE MODEL

The model bellow is formulated as a mixed integer linear programming model.

5.1 Objective function

The objective function should be defined considering the production targets of the problem. The flexible manufacturing plants dealing with Make to order environment usually seek the effective utilization of their manufacturing resources, taking setups into consideration, so that delivery deadlines could be respected. Hence, in our case, the mathematical model aims at minimizing the sum of the total tardiness of all jobs, which is computed through constraint (2).

Then the tardiness of a job i is given by $t_i = \max(c_i - d_i)$
The objective is given as below:

$$\text{Minimize } Z = \sum_{i=1}^{n} t_i \tag{2}$$

5.2 Constraints

$$\sum_{k=1}^{me} \mathrm{x_{ijk}} = 1\,, \quad \forall\, \mathrm{i} \in \mathrm{N}\,, \mathrm{j} \in \mathrm{Ji}, \mathrm{k} \in \mathrm{m_j} \tag{3}$$

$$\mathrm{c_{ijk}} \geq \mathrm{s_{ijk}} + \mathrm{p_{ijk}} - \mathcal{M}(1 - \mathrm{x_{ijk}})\,, \quad \forall\, \mathrm{i} \in \mathrm{N}, \mathrm{j} \in \mathrm{Ji}, \mathrm{k} \in \mathrm{m}_j \tag{4}$$

$$\mathrm{s_{ijk}} + \mathrm{c_{ijk}} \le \mathrm{M}(\mathrm{x_{ijk}}), \qquad \forall\, \mathrm{i} \in \mathrm{N}, \mathrm{j} \in \mathrm{Ji}, \mathrm{k} \in \mathrm{m}_j \tag{5}$$

$$\mathrm{c_{ijk}} \ge \mathrm{s_{ijk}}\ , \qquad \forall\, \mathrm{i} \in \mathrm{N}, \mathrm{j} \in \mathrm{Ji}, \mathrm{k} \in \mathrm{m}_j \tag{6}$$

$$\textstyle\sum_{\mathrm{i}\in\mathrm{j}} \sum_{\mathrm{j,j'}\in\mathrm{oi}} \sum_{\mathrm{k}\in\mathrm{mi}} \mathrm{y_{iji'j'k}} = 1\ , \quad \forall\ \mathrm{i}, i' \in \mathrm{N},\ \mathrm{j,j'} \in \mathrm{J_i}\ ,\ \mathrm{J_{i'}}\ ,\ ,\ \mathrm{k} \in \mathrm{m_{ij}} \cap \mathrm{m}_{i'\mathrm{j'}} \tag{7}$$

$$\mathrm{s_{ijk}} \ge \mathrm{c_{i'j'k}} + \mathrm{st_{iji'j'k}} - \mathcal{M}(1 - \mathrm{y}_{iji'j'k}), \quad \forall\ \mathrm{i}, i' \in \mathrm{N},\ \mathrm{j,j'} \in \mathrm{J_i}\ ,\ \mathrm{J_{i'}} \tag{8}$$

$$\mathrm{s_{i'j'k}} \ge \mathrm{c_{ijk}} + \mathrm{st_{iji'j'k}} - \mathcal{M}\left(\mathrm{y}_{iji'j'k}\right), \quad \forall\ \mathrm{i}, i' \in \mathrm{N},\ \mathrm{j,j'} \in \mathrm{J_i}\ ,\ \mathrm{J_{i'}}\ ,\ \mathrm{k} \in \mathrm{m_{ij}} \cap \mathrm{m}_{i'\mathrm{j'}} \tag{9}$$

$$\mathrm{c_{ijk}} = \textstyle\sum_{k=1}^{\mathrm{m_{ij}}} p_{ijk} + S_{ijk} + st_{iji'j'k} \quad \forall\ \mathrm{i} \in \mathrm{N}, \mathrm{j} \in \mathrm{J_i}\ , \mathrm{k} \in \mathrm{m}_j \tag{10}$$

$$\mathrm{c_i} = \textstyle\sum_{k=1}^{\mathrm{m_{ij}}} \mathrm{c_{ijk}}\ , \qquad \forall\ \mathrm{i} \in \mathrm{N}, \mathrm{j} \in \mathrm{Ji}, \mathrm{k} \in \mathrm{m_{ij}} \tag{11}$$

$$\mathrm{t_i} \ge \mathrm{c_i} - \mathrm{d_i}\ , \qquad \forall\ \mathrm{i} \in \mathrm{N} \tag{12}$$

$$\mathrm{t_i} \ge 0\ , \qquad \forall\ \mathrm{i} \in \mathrm{N} \tag{13}$$

$$\mathrm{x_{ijk}} \in \{0,1\}, \qquad \forall\ \mathrm{i} \in \mathrm{N}, \mathrm{j} \in \mathrm{o_i},\ \mathrm{k} \in \mathrm{m_{ij}} \tag{14}$$

$$\mathrm{y_{iji'j'k}} \in \{0,1\}, \quad \forall\ \mathrm{i,i'} \in \mathrm{N}, \mathrm{j} \in J_{\mathrm{i}}, \mathrm{j'} \in \mathrm{J_{i'}},\ \mathrm{k} \in \mathrm{m_{ij}} \cap \mathrm{m_{i'j'}} \tag{15}$$

$$\mathrm{s_{ijk}} \ge 0\ , \qquad \forall\ \mathrm{i} \in \mathrm{N}, \mathrm{j} \in \mathrm{J_i},\ \mathrm{k} \in \mathrm{m_{ij}} \tag{16}$$

$$\mathrm{c}_{ijk} \ge 0\ , \qquad \forall\ \mathrm{i} \in \mathrm{N}, \mathrm{j} \in \mathrm{J_i},\ \mathrm{k} \in \mathrm{m_{ij}} \tag{17}$$

In the above formulation, (3) specifies the assignment constraint that is defined for every operation j of job i. This restriction determines that each job i must be processed in just one material resource k. Hence, x_{ijk} is non-zero if task i is allocated to processing unit *k*; otherwise, the variable is set to zero. Constraint (4) ensure that a job's completion time is at least equal to or greater than the sum of its start time and its processing time. Constraint (5) Ensures that the end date of each job on machines that are not processing the job becomes 0. Constraint set (6) controls the completion of jobs at stages that the job might skip. Constraint set (7) is precedence constraint set and enforces each operation of each job can only be started after its precedent operation has been finished. Constraints (8) and (9) jointly are used to sequence any pair of tasks (i,i′) assigned to a same processing unit *k*. they assure that two jobs cannot be processed at the same time on a machine, which ensure the nonoverlap of the operations assigned to the same machine. Constraint set (10) represents that the completion time of any operation is the sum of its starting time, setup and processing time. Constraint set (11) calculate the completion time of a job which is the sum of the completion time of all the operations belonging to its processing route. Constraint (12) ensures that the tardiness of a job is greater than or equal to the difference between its completion time and due date, this constraint provide us with the value of the individual tardiness of each job. Constraint (13) guarantees that only positive values for tardiness are considered. Finally, Constraint sets (14), (15), (16) and (17) define the domains of the decision variables.

5. COMPUTATIONAL RESULTS

This section describes the computational tests which are used to evaluate the performance of the presented MILP model. Two testing problems of Hybrid flexible flow shop scheduling problem, are tested. These computational experiments are done to assess the effectiveness of the proposed MILP model. Due to computational time constraints and limited computational resources for most of the real-world problems, a small CPU time is considered as a performance measure. The objective is to reach optimal results with minimum computational time. In order to be useful in real conditions, it is necessary that the scheduling models give good solutions in reasonable time. In our case, calculations for each instance are done in a specified CPU runtime limit ($\mathrm{CPU_{bound}} = 120$ minutes). Finally, the computational results are compared with those of state-of-the-art approaches, and also with the current performance of the studied workshop.

5.1. Literature benchmark

This section compares the experimental results obtained by applying the proposed MILP of $\mathrm{HFF_{19}|Prec, ST_{sd}, d_i|} \sum_{\mathrm{j=1}}^{\mathrm{n}} \mathrm{t_i}$ to two benchmark test instances:

- Shao et al. 2017, who addressed block-FSP-$\sum_{i=1}^{n} t_i$ who used the same benchmark set used (Ronconi & Henriques, 2009). This benchmark set comes from 120 benchmark instances of Taillard (1993) for permutation flow-shop scheduling problem (PFSP), ranging from 20 jobs/5 machines to 500 jobs/20 machines.
- Ruiz and Stützle 2008, who dealt with ST_{sd} -FSP-WT, they used Ruiz's benchmark (Ruiz, 2004) and adapted by including weights and due dates , these latter were constructed using a protocol inspired from that presented by (Hasija & Rajendran, 2004).

The testing problems are determined by size of the problem (n×o×m) in which index n denotes number of jobs, o denotes the maximum number of operations for all jobs and m denotes the machine number.

the MILP model presented in Section "Problem formulation" is coded using ILOG Cplex 12.10 software. and run on a DELL personal computer with an Intel® Core™ i5-8250U @ 1.6 1.8 GHz CPU, 8 GB RAM, and Window 10 operating system. Seven sizes of problems are tested, divided into two categories: small size with 5, 10 and 15 jobs, and medium size with 20, 25, 30 and 35 jobs. The tardiness of each instance was calculated, and the CPU time was obtained by implementing the proposed MILP model on the above cited instances.

Table 1 summarises the results obtained for the different instances considered in the literature (n×m).

Table 1. Data label

n	m	$\sum_{i=1}^{n} t_i$	CPU time
Shao's instances			
5	20	0	10s
10	20	0	50s
15	20	0	2 min
Ruiz's instances			
20	19	120	4 min
25	19	196	12 min
30	19	--	>20 min
35	19	--	>30 min

5.2. Industrial case application

The proposed model was tested using real data retrieved from Sage X3 which is an enterprise resource planning (ERP) system that seek to integrate all business processes and functions allowing real data of production feedback. We used one year analysed real data of production to create instances of 10,20,30,40 and 50 jobs. A real total tardiness is computed for each instance. The exact details of production data cannot be disclosed due to confidentiality reasons.

The comparison consists of the current performance obtained by the workshop for one week, and tested for 5 random weeks with a total tardiness, respectively, negative for the instances with 20 jobs and less. The results obtained, reported in Table 2, show that this completely eliminate the tardiness for instances with 20 jobs and less.

For all the analyzed instances, on average, the MILP made an improvement of 189.51 hours on the average total tardiness per job compared to the results given by the method used currently in the workshop operation planning

Table 2. Implementation results of the mathematical model on real instances

Instances BRO n×o×m	N° jobs	N° Ops	N° mch	t_i	CPU time
5×12×19	5	12	19	0	2 s
10×15×19	7	15	19	0	8 s
15×19×19	9	19	19	0	1 m
20×24×19	10	24	19	0	3 m
25×35×19	11	30	19	-	>2h
30×40×19	16	40	19	-	>2h
35×52×19	20	52	19	-	>2h

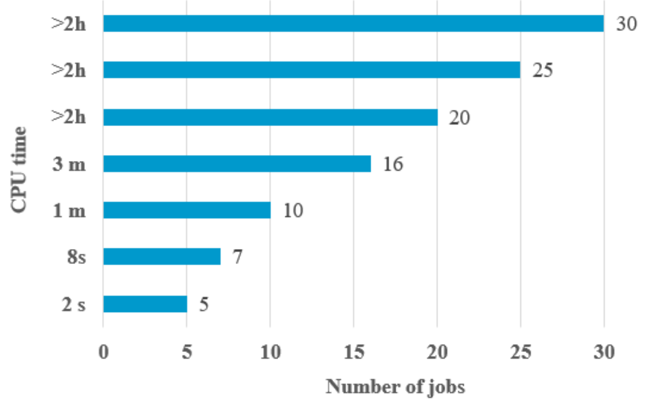

Fig. 2. Computational time of instances according to the number of jobs.

The above figure shows that the model finds an optimal solution within 3 minutes or less for instances that do not exceed 20 jobs, while, no solution found for instances with 25 jobs and more after two hours of execution.

The Computational results of this mathematical model on real instances of the studied company and adapted instances of literature show the limits of an exact resolution for this scheduling problem

6. CONCLUSIONS

In this paper, a flexible hybrid flow shop scheduling problem inspired by a real application of a packaging industry has been studied. This problem can be described with different realistic constraints jointly considered, such as non-identical parallel machines, sequence-dependent setups on machines, and precedence constraints among jobs. We proposed a new method based on linear programming techniques for this scheduling problem involving a set of realistic constraints included in the problem formulation of this paper, with the objective of minimizing the total tardiness.

This MILP model is solved using ILOG Cplex software. The efficiency of this model in improving the total delay has been demonstrated for instances with 20 jobs and less in both the industrial case and experimental instances. The model eliminates the tardiness within 3 minutes. Due to computational time constraints, this exact resolution cannot be considered for the real industrial application with instances greater than 20 jobs.

Future research will consist in considering other resolution methods by developing heuristics and metaheuristics for big-size problems.

ACKNOWLEDGEMENTS

This research was supported by Brodart packaging (France). The authors would like to express their sincere thanks to David Brodart who proofread the article.

REFERENCES

Batur, G. D., Erol, S., & Karasan, O. E. (2016). Robot move sequence determining and multiple part-type scheduling in hybrid flexible flow shop robotic cells. *Computers & Industrial Engineering*, *100*, 72–87. https://doi.org/10.1016/j.cie.2016.08.006

Burcin Ozsoydan, F., & Sağir, M. (2021). Iterated greedy algorithms enhanced by hyper-heuristic based learning for hybrid flexible flowshop scheduling problem with sequence dependent setup times: A case study at a manufacturing plant. *Computers & Operations Research*, *125*, 105044. https://doi.org/10.1016/j.cor.2020.105044

Graham, R. L., Lawler, E. L., Lenstra, J. K., & Kan, A. H. G. R. (1979). Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey. In P. L. Hammer, E. L. Johnson, & B. H. Korte (Eds.), *Annals of Discrete Mathematics* (Vol. 5, pp. 287–326). Elsevier. https://doi.org/10.1016/S0167-5060(08)70356-X

Hasija, S., & Rajendran, C. (2004). Scheduling in flowshops to minimize total tardiness of jobs. *International Journal of Production Research*, *42*(11), 2289–2301. https://doi.org/10.1080/00207540310001657595

Khare, A., & Agrawal, S. (2019). Scheduling hybrid flowshop with sequence-dependent setup times and due windows to minimize total weighted earliness and tardiness. *Computers and Industrial Engineering*, *135*, 780–792. Scopus. https://doi.org/10.1016/j.cie.2019.06.057

Kochhar, S., & Morris, R. J. T. (1987). Heuristic methods for flexible flow line scheduling. *Journal of Manufacturing Systems*, *6*(4), 299–314. Scopus. https://doi.org/10.1016/0278-6125(87)90006-9

Kurz, M. E., & Askin, R. G. (2004). Scheduling flexible flow lines with sequence-dependent setup times. *European Journal of Operational Research*, *159*(1), 66–82. Scopus. https://doi.org/10.1016/S0377-2217(03)00401-6

Long, J., Zheng, Z., Gao, X., & Pardalos, P. M. (2018a). Scheduling a realistic hybrid flow shop with stage skipping and adjustable processing time in steel plants. *Applied Soft Computing*, *64*, 536–549. https://doi.org/10.1016/j.asoc.2017.12.044

Long, J., Zheng, Z., Gao, X., & Pardalos, P. M. (2018b). Scheduling a realistic hybrid flow shop with stage skipping and adjustable processing time in steel plants. *Applied Soft Computing*, *64*, 536–549. https://doi.org/10.1016/j.asoc.2017.12.044

Naderi, B., Gohari, S., & Yazdani, M. (2014). Hybrid flexible flowshop problems: Models and solution methods. *Applied Mathematical Modelling*, *38*(24), 5767–5780. https://doi.org/10.1016/j.apm.2014.04.012

Naderi, B., Zandieh, M., Khaleghi Ghoshe Balagh, A., & Roshanaei, V. (2009). An improved simulated annealing for hybrid flowshops with sequence-dependent setup and transportation times to minimize total completion time and total tardiness. *Expert Systems with Applications*, *36*(6), 9625–9633. Scopus. https://doi.org/10.1016/j.eswa.2008.09.063

Naderi, Bashir, Ruiz, R., & Zandieh, M. (2010). Algorithms for a realistic variant of flowshop scheduling. *Computers & Operations Research*, *37*, 236–246. https://doi.org/10.1016/j.cor.2009.04.017

Nait Tahar, D., Yalaoui, F., Chu, C., & Amodeo, L. (2006). A linear programming approach for identical parallel machine scheduling with job splitting and sequence-dependent setup times. *International Journal of Production Economics*, *99*(1), 63–73. https://doi.org/10.1016/j.ijpe.2004.12.007

Oujana, S., Amodeo, L., & Yalaoui, F. (2021). Logical Workflow Analysis based on Multiple Criteria Decision Analysis: Industrial Application for a Make to Order Environment: *Proceedings of the 10th International Conference on Operations Research and Enterprise Systems*, 401–407. https://doi.org/10.5220/0010248604010407

Ronconi, D. P., & Henriques, L. R. S. (2009). Some heuristic algorithms for total tardiness minimization in a flowshop with blocking. *Omega*, *37*(2), 272–281. https://doi.org/10.1016/j.omega.2007.01.003

Ruiz, R. (2004). A comprehensive review and evaluation of permutation flowshop heuristics. *European Journal of Operational Research*, *40*. https://doi.org/10.1016/S0377-2217(04)00255-3

Ruiz, R., & Maroto, C. (2006). A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility. *European Journal of Operational Research*, *169*(3), 781–800. https://doi.org/10.1016/j.ejor.2004.06.038

Ruiz, R., Şerifoğlu, F. S., & Urlings, T. (2008). Modeling realistic hybrid flexible flowshop scheduling problems. *Computers & Operations Research*, *35*(4), 1151–1175. https://doi.org/10.1016/j.cor.2006.07.014

Ruiz, R., & Stützle, T. (2008). An Iterated Greedy heuristic for the sequence dependent setup times flowshop problem with makespan and weighted tardiness objectives. *European Journal of Operational Research*, *187*, 1143–1159. https://doi.org/10.1016/j.ejor.2006.07.029

Salvador, M. S. (1973). A Solution to a Special Class of Flow Shop Scheduling Problems. In S. E. Elmaghraby (Ed.), *Symposium on the Theory of Scheduling and Its Applications* (pp. 83–91). Springer. https://doi.org/10.1007/978-3-642-80784-8_7

Shao, Z., Pi, D., & Shao, W. (2017). Self-adaptive discrete invasive weed optimization for the blocking flow-shop scheduling problem to minimize total tardiness. *Computers & Industrial Engineering*, *111*, 331–351. https://doi.org/10.1016/j.cie.2017.07.037

Silver, E. A., Pyke, D. F., & Thomas, D. J. (2016). *Inventory and Production Management in Supply Chains*. CRC Press.

Szwarc, W., & Gupta, J. N. D. (1987). A flow-shop problem with sequence-dependent additive setup times. *Naval Research Logistics (NRL)*, *34*(5), 619–627. Scopus. https://doi.org/10.1002/1520-6750(198710)34:5<619::AID-NAV3220340503>3.0.CO;2-B

Zhang, C., Tan, J., Peng, K., Gao, L., Shen, W., & Lian, K. (2021). A discrete whale swarm algorithm for hybrid flow-shop scheduling problem with limited buffers. *Robotics and Computer-Integrated Manufacturing*, *68*, 102081. https://doi.org/10.1016/j.rcim.2020.102081