



# New efficiency-stability criterion for flow shop rescheduling problem with mixed blocking constraints

Ayoub Tighazoui, Christophe Sauvey, Nathalie Sauer

## ► To cite this version:

Ayoub Tighazoui, Christophe Sauvey, Nathalie Sauer. New efficiency-stability criterion for flow shop rescheduling problem with mixed blocking constraints. 17th IFAC Symposium on Information Control Problems in Manufacturing (INCOM 2021), Jun 2021, Budapest, Hungary. pp.1181-1186, 10.1016/j.ifacol.2021.08.139 . hal-03423730

**HAL Id: hal-03423730**

**<https://hal.science/hal-03423730>**

Submitted on 6 Feb 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

# New efficiency-stability criterion for flow shop rescheduling problem with mixed blocking constraints

A. Tighazoui\* C. Sauvey\* N. Sauer\*

\* Université de Lorraine, LGIPM, F-57000 Metz, France

e-mail: {ayoub.tighazoui, christophe.sauvey, nathalie.sauer}@univ-lorraine.fr

**Abstract:** New jobs arrival, jobs cancelation or machine breakdowns, are unexpected events that may occur during production run in real environments, thus disrupting the established schedule. Hence, rescheduling process is necessary for efficiently revising the initial schedule, preferably with little movements. In the literature, many scheduling problems consider unlimited buffer spaces between successive machines in flow shop environments. However, in real life systems, the capacity between machines may be limited or no-store existing. To fill the void, this paper considers a flow shop rescheduling problem with mixed blocking constraints, under disruptions caused by new jobs arrival. A new performance measure is investigated, considering simultaneously the schedule efficiency measured by the total weighted waiting time, and the schedule stability evaluated by the total weighted completion time deviation. This combination could be very helpful for real life industrial systems. A mixed integer linear programming model (MILP) is designed, as well as an iterative methodology based on the predictive reactive rescheduling strategy. At last, two heuristic methods are implemented and discussed, in terms of solution quality and computing time.

Copyright © 2021 The Authors. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0>)

**Keywords:** Flow shop rescheduling, waiting time, completion time deviation, mixed blocking constraints, stability.

## 1. INTRODUCTION AND LITERATURE REVIEW

Technological development and customers' practices change, have a direct impact on production management process. Henceforth, a customer can modify its order at any time. This changed situation influences the actual production scheduling (Ivanov *et al.*, 2016; Moeuf *et al.*, 2018). In fact, during job execution, many unexpected events can appear and disrupt the already established schedule, such as: new orders arrival, orders cancelation, machine breakdowns, lack of raw materials, etc. (Katragjini *et al.*, 2013). Therefore, production firms are forced to quickly react for efficiently revising the already established schedule, in preference with less movements. Hence, rescheduling process is nécessaire (Vieira *et al.*, 2003).

In the literature, several papers have already established literature reviews for this area, for instance Ouelhadj and Petrovic (2009) and Uhlmann and Frazzon (2018). As well, several interesting papers dealing with flow shop rescheduling system. Rahmani and Ramezani (2016) studied a flexible flow shop rescheduling problem disrupted by new jobs arrival. The authors considered the weighted tardiness as an efficiency criterion and the absolute deviation of completion times as a stability measure. They adopted a predictive reactive rescheduling strategy and designed a variable neighborhood search algorithm to solve the problem. For other related papers, the reader may refer to Katragjini *et al.* (2015), Qin *et al.* (2018), Peng *et al.* (2019), Tighazoui *et al.* (2020).

On the other hand, flow shop rescheduling problems have taken interest on studying different criteria for measuring the

performance of the schedule. In terms of schedule *efficiency*, the Makespan (Katragjini *et al.*, 2015), the total tardiness (Li *et al.*, 2015b), and the total flowtime (Liu, F *et al.*, 2017), etc, are criteria that have already been considered in the flow shop rescheduling literature. However, in dynamic environments, the impact of jobs deviation is also assessed, referred to as *stability* criteria. Akkan (2015) considered the sum of absolute starting time deviations as a stability measure, which are the sum of the absolute difference between the starting times of jobs before and after the rescheduling. Moreover, we have found other related papers dealing with stability criteria in rescheduling problems, for example: total deviation penalty and starting time deviation (Rangsaritratamee *et al.*, 2004), the amount of jobs that change the machine after the rescheduling (He *et al.*, 2019), etc. In this paper, a new performance measure is investigated. Firstly, in terms of schedule efficiency, the *Total Weighted Waiting Time (TWWT)* is considered. The waiting time of the job represents the total period during which the job has waited before the execution (Jurčičin and Šebo, 2015). Guo *et al.*, 2013 is among the first works that considered the waiting time on a rescheduling problem, the authors developed a mathematical model for the single machine rescheduling problem for minimizing the sum of the waiting times of rework jobs and original loads with release time. They proved that the problem is NP-hard and proposed a dynamic heuristic insert algorithm of polynomial-time to solve this kind of problem. In our approach, we considered the *TWWT* as a criterion, in which the waiting time can be regarded as the waiting period of a product in front of a workstation, and the weight as the customer priority. Rinnooy Kan (1976) classified this problem as NP-hard for a single

machine scheduling problem. Secondly, in terms of schedule stability, the *Total Weighted Completion Time Deviation (TWCTD)* is considered. This criterion consists in assessing the deviation between jobs' completion times in the original schedule and revised one, associating for each job a weight. Thus, it will be more difficult to modify the important orders.

Many flow shop scheduling problems consider unlimited buffer spaces between successive machines. However, in the real-life environments, the capacity between successive machines is limited or no-store existing, only few papers in the rescheduling literature considered this issue (Kecman et al., 2013; Tao and Liu, 2019). In the static scheduling problems, several blocking constraints are studied, such as *RSb* (Release when Starting blocking) in the case when a machine is blocked by a job, until it starts on the following machine (Wang et al., 2006), *RCb* or *RCb\** (Release when Completing blocking) characterizes the case when a machine is blocked by a job, until it is finished on the following machine (*RCb\**) (Trabelsi et al., 2011) or starts on the further next one (*RCb*). Mixed blocking constraints where successive machines can be subject to different types of blocking constraints have also been studied by Trabelsi et al. (2012).

To fill the void in existing rescheduling literature, this paper deals with different types of constraints mixed in a production rescheduling system. The system disruption is caused by the arrival of new jobs. A MILP model is designed for this problem, and an online iterative methodology, based on a predictive reactive strategy, which first consists in generating in the first step an initial schedule for minimizing the *TWWT*. After each job occurrence, we update the schedule, but this time for simultaneously minimizing the schedule efficiency combined with the stability, which is measured by the *TWCTD*, we associate then a coefficient for each part of the objective function (i.e.  $\alpha \cdot \text{efficiency} + (1-\alpha) \cdot \text{stability}$ ). To the best of our knowledge, there is not any rescheduling system studied in the literature, on which this kind of problem has been discussed. It is contributing to operations research area by:

- Considering a flow shop rescheduling problem, with mixed blocking constraints.
- Studying a new performance measure inspired from real industrial systems, in which the *TWWT* is regarded as an efficiency criterion, and the *TWCTD* is regarded as a stability criterion.
- Developing for this problem a MILP model and an iterative methodology, based on predictive reactive strategy for dealing with the online part.
- Implementing two heuristic methods and evaluating them in terms of solution quality and computing time.

The rest of the paper is organized as follows. In section 2, the problem and the proposed methodology are described. In section 3, the MILP is formulated and its numerical results are presented. In section 4, heuristic methods are developed and evaluated. Finally, we conclude and propose some perspectives in section 5.

## 2. PROBLEM DESCRIPTION AND METHODOLOGY

### 2.1 Problem description

A flow shop rescheduling problem is studied, with mixed blocking constraints, under disruptions caused by new jobs arrival. To each job  $j$ , is associated a release date  $r_j$ , a weight  $w_j$ , and a processing time of job  $j$  in the machine  $m$ ,  $p_{jm}$ . When a job  $j$  starts its execution on the machine  $m$ , it is processed to completion time  $CT_{jm}$  without interruption, preemption is not considered.

**Efficiency measure:** The waiting time  $W_j$  of job  $j$  is defined as the total duration that the job  $j$  is waiting in the system (Jurčišin and Šebo, 2015). It characterizes the sum of waiting periods before the starting times of the job on each machine. Then, the waiting time  $W_j$  of a job  $j$  is defined as  $W_j = CT_{jM} - r_j - \sum_{m=1}^M p_{jm}$ , where  $CT_{jM}$  is the completion time of the job  $j$  in the last machine  $M$ . Assuming that  $N = \{1, 2, \dots, n\}$  is the set of jobs that are already available on  $t = 0$ , the objective function before the disruption is  $\sum_{j=1}^n w_j W_j$ , denoted by *TWWT*.

**Stability measure:** Before the disruption, let  $CT_{ojm}$  be the original completion time of the job  $j$ . After disruption occurrence, the schedule may change, and the job  $j$  really finishes at  $CT_{jm}$ . Therefore, the deviation between  $CT_{ojm}$  and  $CT_{jm}$  is used to assess the system stability. Furthermore, a weight  $w_j$  is associated to each job  $j$ , to penalize more largely the important orders. Assuming that  $F = \{1, 2, \dots, M\}$  is the set of the machines, the stability measure is  $\frac{1}{M} \sum_{j=1}^n \sum_{m=1}^M w_j (CT_{jm} - CT_{ojm})$ , denoted by *TWCTD*. The expression is divided by  $M$ , in order to normalize it with respect to the efficiency criterion.

Both parts of the objective function are balanced with  $\alpha$ , the efficiency-stability coefficient. Ultimately, we define the new objective function, which simultaneously considers efficiency and stability, as:

$$\alpha \sum_{j=1}^{n'} w_j W_j + (1 - \alpha) \frac{1}{M} \sum_{j=1}^{n'} \sum_{m=1}^M w_j (CT_{jm} - CT_{ojm})$$

It is considered that  $N' = \{1, 2, \dots, n'\}$  is the set of jobs after the disruption. It contains the yet unexecuted jobs and the new ones. Only the set  $N'$  of jobs are considered in the stability criterion, since the new jobs did not exist before.

**Mixed blocking constraints:** In the classical flow shop problem, the capacity between successive machines is considered to be unlimited. In this case, there is no blocking situation, referred to as *Without blocking Wb*. In a classical blocking constraint *RSb* a machine remains blocked by a job until this one starts on the following machine. In the case of specific constraint *RCb\** a machine is blocked by a job, until this one is finished on the following machine. In the particular case of the constraint *RCb* a machine is blocked by a job, until this one is finished and it has left the following machine, this date corresponds to the starting time of the job on the third machine in the process. In this paper, we study the case when the described constraints are mixed in one production system. Thus, we have introduced a vector  $V$  representing a sequence of constraints. For example,  $V = (Wb, RSb, RCb^*, RCb)$ , where  $V_m$  is the blocking constraint between the machine  $m$  and  $m+1$ .

## 2.2 Proposed predictive-reactive methodology

The proposed predictive reactive strategy consists in solving, in the first step, the initial problem consisting in minimizing the *TWWT*. When a disruption occurs, a new schedule is generated, and this time the objective is the minimization of *TWWT* combined with *TWCTD*. The online methodology consists of going through the simulation horizon step by step. When a new job arrives, it is combined with the uninitiated jobs, and a new schedule is established. The finite time horizon  $[0, T]$  is discretised into periods  $\Delta t$ , representing the time period length. The job occurrence may be possible only at these times. We have assumed for this study that  $\Delta t = 1$  unit of time, the same that we have already used for the jobs data  $p_{jm}$  and  $r_j$ . When a job arrives at time  $t$ , this date is its release date. Thereby, the variable  $\theta(t)$  is introduced, which is equal to one if a job occurs and zero otherwise.

$$\theta(t) = \begin{cases} 1 & \text{If a job is occurred} \\ 0 & \text{Otherwise} \end{cases} \quad (1)$$

At each step, the value of  $\theta(t)$  indicates if a job has occurred or not. We assume in this study that only one job occurs when  $\theta(t) = 1$ .

Fig. 1 presents the flowchart which summarizes the proposed predictive reactive strategy. The methodology goes through the time horizon and checks whether a job occurs through the state of  $\theta(t)$ . Accordingly, it updates the problem data and constraints, as well as the objective function. Then, the new problem is solved.

## 3. MILP FORMULATION

The implemented mathematical model is divided in two parts, before and after the disruptions. For each part, a subsection is dedicated. The third subsection shows the numerical results obtained through the model resolution.

### 3.1 Mathematical model before the disruption

#### Parameters:

$N$ : set of jobs  $\{1, 2, \dots, n\}$

$K$ : set of positions  $\{1, 2, \dots, n\}$

$F$ : set of machines  $\{1, 2, \dots, M\}$

$j$ : index of job,  $j = 1, 2, \dots, n$

$k$ : index of position,  $k = 1, 2, \dots, n$

$m$ : index of machine,  $m = 1, 2, \dots, M$

$w_j$ : weight of job  $j$

$r_j$ : release date of job  $j$

$p_{jm}$ : processing time of job  $j$  on the machine  $m$

$bigM$ : big value,  $bigM = \sum_{j=1}^n \sum_{m=1}^M p_{jm}$

$$B_{hm} = \begin{cases} 1 & \text{If there is a blocking constraint of type } h \\ & \text{between the machine } m \text{ and } m+1 \\ 0 & \text{Otherwise} \end{cases}$$

With:

$h=1$  if there is no blocking constraint between the machine  $m$  and  $m+1$ .

$h=2$  if there is a blocking of type *RSb* between the machine  $m$  and  $m+1$ .

$h=3$  if there is a blocking of type *RCb\** between the machine  $m$  and  $m+1$ .

$h=4$  if there is a blocking of type *RCb* between the machine  $m$  and  $m+1$ .

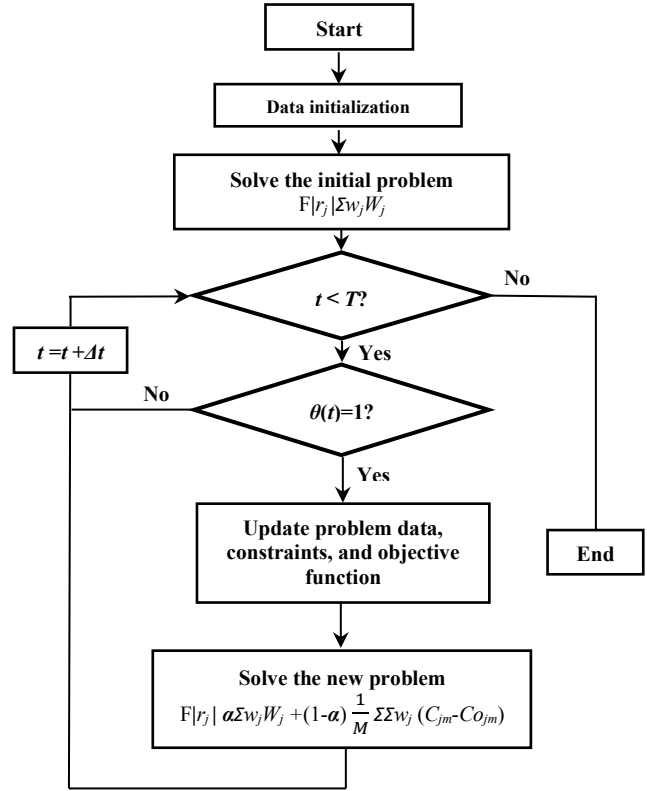


Fig. 1 The proposed predictive-reactive strategy

#### Decision variables:

$$X_{jk} = \begin{cases} 1 & \text{If the job } j \text{ is assigned to } k^{\text{th}} \text{ position} \\ 0 & \text{Otherwise} \end{cases}$$

$S_{km}$ : Starting time of the job in  $k^{\text{th}}$  position on the machine  $m$ .

$C_{km}$ : Completion time of the job in  $k^{\text{th}}$  position on the machine  $m$ .

$CT_{jm}$ : Completion time of job  $j$  on the machine  $m$ .

$W_j$ : Waiting time of job  $j$ .

**Objective function:**  $\text{Min } \sum_{j=1}^n w_j W_j \text{ s.t.}$

$$\sum_{k=1}^n X_{jk} = 1 \quad \forall j \in N \quad (2)$$

$$\sum_{j=1}^n X_{jk} = 1 \quad \forall k \in K \quad (3)$$

$$C_{km} = S_{km} + \sum_{j=1}^n p_{jm} X_{jk} \quad \forall k \in K, \forall m \in F \quad (4)$$

$$S_{km} \geq C_{k-1,m} \quad \forall k \in K, \forall m \in \{2, \dots, M\} \quad (5)$$

$$S_{km} \geq C_{k-1,m-1} \cdot B_{1m} + S_{k-1,m+1} \cdot B_{2m} + C_{k-1,m+1} \cdot B_{3m} + S_{k-1,m+2} \cdot B_{4m} \quad (6)$$

$$\forall k \in \{2, \dots, n\}, \forall m \in \{1, \dots, M-2\} \quad (7)$$

$$S_{kM-1} \geq C_{k-1,M-1} \cdot B_{1M-1} + S_{k-1,M} \cdot B_{2M-1} + C_{k-1,M} \cdot B_{3M-1} \quad \forall k \in \{2, \dots, n\} \quad (8)$$

$$S_{kM} \geq C_{k-1,M} \quad \forall k \in \{2, \dots, n\} \quad (9)$$

$$S_{km} \geq \sum_{j=1}^n r_j X_{jk} \quad \forall k \in K, \forall m \in F \quad (10)$$

$$CT_{jm} \geq C_{km} - bigM(1 - X_{jk}) \quad \forall j \in N, k \in K, \forall m \in F \quad (11)$$

$$CT_{jm} \leq C_{km} + bigM(1 - X_{jk}) \quad \forall j \in N, k \in K, \forall m \in F \quad (12)$$

$$W_j = CT_{jm} - r_j - \sum_{m=1}^M p_{jm} \quad \forall j \in N$$

$$X_{jk} \in \{0,1\}, \quad \forall j \in N, \forall k \in K \quad (13)$$

$$S_{km}, C_{km}, CT_{jm}, W_j \geq 0 \quad \forall j \in N, k \in K, \forall m \in F \quad (14)$$

Eq(2) each job is affected to one position. Eq(3) each position is occupied by one job. Eq(4) for all machines, the completion time of  $k^{th}$  position is equal to the starting time of  $k^{th}$  position plus the assigned processing time. Eq(5) the starting time of  $k^{th}$  position is greater than or equal to its completion time in the previous machine. Eq(6) the starting time of a position depends on different types of blocking constraints, represented by the parameter  $B_{hm}$ . Eq(7) the starting time of a position in the penultimate machine where there is no  $RCb$  constraint depends on the two following machines. Eq(8) the starting time in the last machine which can only be without blocking. Eq(9) for all machines, the starting time of  $k^{th}$  position is greater than or equal to its assigned release date. Eq(10) and (11) for all machines, the completion time of job  $j$  is greater than or equal to the completion time of its assigned position, where  $bigM$  must be sufficiently large. Eq(12) the waiting time of job  $j$  depends on the completion time in the last machine, release date, and processing time. Eq(13) constraints the variable  $X_{jk}$  to be a binary decision variable. Eq(14) are non-negativity constraints, making all decision variables greater than or equal to zero.

### 3.2 Mathematical model after the disruptions

In this second case, all the jobs that have already begun their execution before  $t_d$  (disruption date) are omitted from the list.

**The new parameters are as follows:**

$n_j$ : the number of arrived jobs

$nt$ : the number of unexecuted jobs at time  $t_d$

$n' = nt + n_j$

$N'$ : set of jobs  $\{1, 2, \dots, n'\}$

$K'$ : set of positions  $\{1, 2, \dots, n'\}$

$j$ : index of job  $j=1, 2, \dots, n'$

$k$ : index of position  $k=1, 2, \dots, n'$

$CTo_{jm}$ : original completion time of job  $j$  on the machine  $m$  (obtained by the resolution of the initial problem).

$\alpha$ : efficiency-stability coefficient.

The decision variables  $X_{jk}$ ,  $CT_{jm}$ ,  $S_{km}$ ,  $C_{km}$  and  $W_j$  are also used for this new model. Thus, the new objective function is as follows:

$$\text{Min } \alpha \sum_{j=1}^{n'} w_j W_j + (1 - \alpha) \frac{1}{M} \sum_{j=1}^n \sum_{m=1}^M w_j (CT_{jm} - CTo_{jm})$$

The constraints (2) until (14) are also used for this model.

$$CT_{jm} \geq CTo_{jm} \quad \forall j \in N, m \in F \quad (15)$$

The constraint (15) signifies that the completion time of a job will be greater than its original completion time. In fact, when a job appears for the first time, only the efficiency criterion is considered. Thus, the job is placed in its ideal position. Hence, we assumed that it will not be possible to move this job back after the rescheduling.

### 3.3 Numerical results

The proposed MILP model has been written on FICO Xpress IVE and the simulations have been performed on a Core i5 2.40GHz laptop. In this sub section, the numerical results of the MILP resolution for a flow shop problem composed by 5 machines are presented. Ten different instances are generated randomly, following the specifications given in Table 1.

**Table 1 Characteristics of the parameter values**

Parameters	Values
$w_j$	$\sim U(1,5)$
$p_{jm}$	$\sim U(1,4) (ut)$
$T$	48 (ut)
$\theta(t)$	$\sim B(p_\theta)$
$V$	( $Wb, RSb, RCb^*, RCb$ )

In the following, the maximum, minimum, average and the standard deviation of the computing time ( $CT$ ) for the sets of tested instances are presented in seconds. The instances contain 5 and 7, respectively, initial jobs. Over the time horizon  $T$ , the schedule is disrupted by the new jobs arrival. The value of  $\theta(t)$  follows the Bernoulli distribution with the appearance frequency parameter  $p_\theta$ , varied with different values. In this first study, it is assumed that the value of  $\alpha$  is fixed at 0.8. The aim here is to determine which problem size the MILP can solve.  $\alpha$  will be varied for different values in the next study. Table 2 shows the obtained results of computing time for different values of  $p_\theta$ .

**Table 2 Computing time study**

Initial jobs	$p_\theta(TNAJ)$	Min CT	Max CT	Avg CT	Std dev CT
5	0.2 (10 jobs)	1.7	2.5	2.1	0.2
	0.3 (14 jobs)	3.1	4.4	3.7	0.5
	0.4 (19 jobs)	7.9	16	11	2.5
	0.5 (24 jobs)	111	567	253	211
7	0.2 (10 jobs)	10	26	17	7.1
	0.3 (14 jobs)	24	128	45	32
	0.4 (19 jobs)	44	3966	1339	1885
	0.5 (24 jobs)	> 8h	> 8h	> 8h	-

TNAJ: The Total Number of Appearing Jobs

Furthermore, we have also taken interest to the Maximum Duration of an Iteration ( $MDI$ ), reported in Table 3. This parameter gives an idea about the necessary computing time of the MILP software between the job appearance and the getting of a solution.

**Table 3 Maximum duration of iteration**

Initial jobs	$p_\theta(TNAJ)$	Min MDI	Max MDI	Avg MDI	Std dev MDI
5	0.2 (10 jobs)	0.2	0.4	0.2	0.06
	0.3 (14 jobs)	0.3	0.6	0.4	0.08
	0.4 (19 jobs)	0.8	2.2	1.6	0.5
	0.5 (24 jobs)	21	316	106	141
7	0.2 (10 jobs)	2.4	5.1	3.8	1.04
	0.3 (14 jobs)	3.4	69	13	19
	0.4 (19 jobs)	6.5	3264	1083	1607
	0.5 (24 jobs)	> 8h	> 8h	> 8h	-

TNAJ: Total Number of Appearing Jobs

As can be seen in Tables 2 and 3, the  $CT$  and  $MDI$  depend on the set of initial jobs and on the appearance frequency of jobs. If these ones are bigger, the average and the standard deviation of  $CT$  and  $MDI$  increase, which makes difficult to estimate the computing time necessary to solve the proposed problem. In the presented results, a break appears when the number of jobs turns around 30 jobs (7 initial jobs and 24 jobs appearing).

Indeed, when a job with a long processing time occupies the machine and other new jobs still appear, a lot of unexecuted jobs can be accumulated, since the machine is busy. This phenomenon increases the number of jobs concerned by the rescheduling in some particular step and decelerates the MILP resolution. Thus, the larger part to reschedule, the longer computing time becomes. Therefore, we present heuristic methods in the following section, in order to accelerate the getting of a solution on the rescheduling part.

#### 4. HEURISTIC METHODS

##### 4.1 Description of the heuristic methods

When the disruption occurs at time  $t_d$ , all jobs initiated before this date must keep the same position. These jobs constitute a fixed part. On the other hand, the rest of jobs combined with the new arrived job, constitute the part to reschedule. The heuristic methods divide the part to reschedule in two parts: a new fixed part, and a part to reschedule which contains only  $Nr$  jobs, assuming that  $Nr$  is the number of jobs that can be solved in a reasonable time.

In the first step, this problem is solved considering only  $Nr$  jobs. If the new added job is placed on the first position (left position), it means that it is possible that this job could be placed earlier in the optimal solution. Hence, in a second step, the part to reschedule is shifted (see Fig.2) and the problem is solved again with the new part of the schedule. This second step is repeated as long as the new added job is in the first position, or if the release date  $t_d$  is reached by the shifting process. We denote by *Step* the number of jobs to shift.

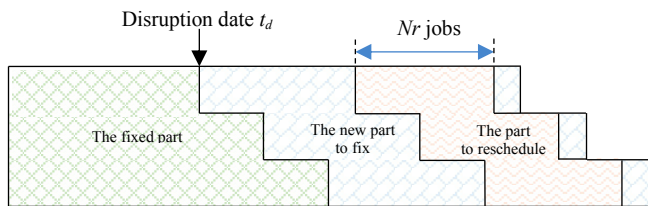


Fig. 2 Proposed method with shifted  $Nr$  jobs

Two starting positions for choosing the  $Nr$  jobs to reschedule are possible, either at the end or at the beginning of the schedule. Accordingly, Heuristic 1 (H1) refers to the method which brows the positions from the end of the sequence to the beginning (as described on Fig.2), and Heuristic 2 (H2) the reverse, from the beginning to the end of the schedule.

##### 4.2 Evaluation of the heuristic methods

The proposed heuristics are compared with the MILP method, for different values of  $p_\theta$  and  $\alpha$ , both in terms of solution quality and computing time.  $\alpha$  is integrated in the algorithm when solving the problem after the disruption as shown in Fig.1. However, with the high disruptions' probabilities ( $p_\theta = 0.8$  and  $p_\theta = 1$ ), the MILP software is not always able to provide a solution in a reasonable time. Hence, the solutions provided by *WSPT* are used to be compared with the two heuristic solutions. The number of improvements established by each heuristic before providing the final solution have also

been calculated. It is assumed in this subsection that  $Nr = 5$  and *Step* = 4. Ten different instances are tested, and the averages are presented in Table 4.

When  $p_\theta = 0.2$ , both heuristics and the exact method provide the same results in terms of solution quality and computing time. In this case, as the disruptions is low, both heuristics do not need to make improvements for providing a solution.

When  $p_\theta = 0.5$ , the system is subjected to the medium disruptions. As can be observed in Table 4, H2 provides the same solutions as the exact method. H1 provides solutions close to the optimal solutions. Both heuristics provide solutions in a shorter time. In fact, H2 acts from the left (from the beginning of the sequence to the end). The beginning of the sequence contains jobs with high weight, their optimization has a great impact on the solution efficiency. Hence, H2 is more effective than H1.

The MILP model fails to solve the problem in a reasonable time with high disruptions probability ( $p_\theta = 0.8$  and  $p_\theta = 1$ ). Thus, *WSPT* is used as an approximate method for obtaining the original completion times  $CTO_{jms}$ , in order to compare the heuristics between them, it quickly provides a solution. However, as *WSPT* is an approximate method for this problem, H2 provides in most of cases a better solution than *WSPT* and H1.

We also observed that, when  $\alpha$  decreases, H1 makes less movements and takes less time for providing the solution, contrarily to H2. Indeed, H1 makes an improvement only if the new job is positioned in the left position (the first position in  $Nr$  jobs). Therefore, when  $\alpha$  decreases, the stability is considered. So, the first positions in the left are already occupied by the old jobs so as not to disrupt the established schedule.

On the other hand, the number of improvements increases with  $p_\theta$  as there are more disruptions in the system.

#### 5. CONCLUSION AND PERSPECTIVES

This work studied a new performance measure in a flow shop rescheduling problem with mixed blocking constraints. The proposed criterion is a linear combination of the *TWWT* as the schedule efficiency, and the *TWCTD* as the schedule stability. A sequence of different blocking constraints is considered between the machines. This association has never been studied in the research literature before, and constitutes a new contribution in the operational research field. A MILP model is developed for describing this system, as well as the predictive-reactive strategy. Numerical results show that the MILP resolution is limited in terms of the number of jobs. Accordingly, two heuristics have been developed to deal with this issue. The evaluation of the heuristics allows us to conclude that:

- The smaller the part to reschedule, the faster the computing time. However, the quality of the solution reduces.
- As the stability makes it difficult to move the old jobs, the efficiency-stability coefficient impacts the computing time of the heuristics.
- Heuristic 2 is more efficient than Heuristic 1. In most of cases, it provides a better solution than *WSPT* algorithm.

In further works, we intend to develop heuristics acting on blocking constraints, as well as other metaheuristics.



**Table 4** Heuristics evaluation with different values of  $p_\theta$  and  $\alpha$ 

		$p_\theta = 0,2$ (10 jobs)			$p_\theta = 0,5$ (24 jobs)			$p_\theta = 0,8$ (39 jobs)			$p_\theta = 1$ (48 jobs)		
		H1	H2	Exact	H1	H2	Exact	H1	H2	WSPT	H1	H2	WSPT
$\alpha = 1$	Time (s)	1.59	1.45	1.47	44.39	22.3	169.10	510.6	291.45	0.27	2935,42	1052,41	0,69
	Solution	<b>512.11</b>	<b>512.11</b>	<b>512.11</b>	2038.5	<b>2029</b>	<b>2029</b>	5586.63	<b>5500</b>	6159.67	8167,33	<b>7887,00</b>	8925,25
	N° Imprv	0	0	-	16.5	7	-	59.13	30.5	-	110,00	52,50	-
$\alpha = 0,75$	Time (s)	1.14	1.16	1.30	41.01	24.77	134.08	459.33	377.07	0.25	1764,60	1627,68	0,54
	Solution	<b>427.09</b>	<b>427.09</b>	<b>427.09</b>	<b>1658.78</b>	<b>1658.78</b>	<b>1658.78</b>	4807.06	<b>4717.53</b>	5014.17	7074,75	<b>6957,12</b>	7530,08
	N° Imprv	0	0	-	14.67	8	-	49.88	37.75	-	82,33	69,00	-
$\alpha = 0,5$	Time (s)	1.14	1.17	1.34	39.27	27.26	379.25	293.77	459.81	0.25	818,05	2320,45	0,81
	Solution	<b>299.49</b>	<b>299.49</b>	<b>299.49</b>	1366.77	<b>1310.87</b>	<b>1310.87</b>	4361.36	<b>3910.8</b>	3930.84	6823,43	<b>5894,07</b>	5974,17
	N° Imprv	0	0	-	12	9	-	31.63	46.25	-	43,67	91,00	-

H1: Heuristic 1, H2: Heuristic 2, WSPT: Weighted Shortest Processing Time, N° Imprv: Number of improvements. Solution:  $\alpha$  TWWT+ (1- $\alpha$ ) TWCTD

## REFERENCES

- Akkan, C. (2015). Improving schedule stability in single-machine rescheduling for new operation insertion. *Computers & Operations Research*, 64, 198-209.
- Guo, Y., Wang, Q., & Huang, M. (2013). Rescheduling with release time to minimize sum of waiting time considering waiting constraint of original loads. *Acta Automatica Sinica*, 39(12), 2100-2110.
- Ivanov, D., Dolgui, A., Sokolov, B., Werner, F., & Ivanova, M. (2016). A dynamic model and an algorithm for short-term supply chain scheduling in the smart factory industry 4.0. *International Journal of Production Research*, 54(2), 386-402.
- Jurčišin, R., & Šebo, J. (2015). Basic production scheduling concept software application in a deterministic mechanical production environment. *Acta Simulatio*, 1(4), 1-4.
- Kan, A. R. (1976). Problem formulation. In *Machine Scheduling Problems*. Springer, Boston, MA, 5-29, ISBN: 978-90-247-1848-1.
- Katrąjini, K., Vallada, E., & Ruiz, R. (2013). Flow shop rescheduling under different types of disruption. *International Journal of Production Research*, 51(3), 780-797.
- Katrąjini, K., Vallada, E., & Ruiz, R. (2015). Rescheduling flowshops under simultaneous disruptions. *International Conference on Industrial Engineering and Systems Management (IESM)* (pp. 84-91). IEEE.
- Kecman, P., Corman, F., D'Ariano, A., & Goverde, R. M. (2013). Rescheduling models for railway traffic management in large-scale networks. *Public Transport*, 5(1-2), 95-123.
- Liu, F., Wang, S., Hong, Y., & Yue, X. (2017). On the robust and stable flowshop scheduling under stochastic and dynamic disruptions. *IEEE Transactions on Engineering Management*, 64(4), 539-553.
- Moeuf, A., Pellerin, R., Lamouri, S., Tamayo-Giraldo, S., & Barbaray, R. (2018). The industrial management of SMEs in the era of Industry 4.0. *International Journal of Production Research*, 56(3), 1118-1136.
- Ouelhadj, D., & Petrovic, S. (2009). A survey of dynamic scheduling in manufacturing systems. *Journal of scheduling*, 12(4), 417.
- Peng, K., Pan, Q. K., Gao, L., Li, X., Das, S., & Zhang, B. (2019). A multi-start variable neighbourhood descent algorithm for hybrid flowshop rescheduling. *Swarm and Evolutionary Computation*, 45, 92-112.
- Qin, W., Zhang, J., & Song, D. (2018). An improved ant colony algorithm for dynamic hybrid flow shop scheduling with uncertain processing time. *Journal of Intelligent Manufacturing*, 29(4), 891-904.
- Rahmani, D., & Ramezani, R. (2016). A stable reactive approach in dynamic flexible flow shop scheduling with unexpected disruptions: A case study. *Computers & Industrial Engineering*, 98, 360-372.
- Rangsaritsamee, R., Ferrell Jr, W. G., & Kurz, M. B. (2004). Dynamic rescheduling that simultaneously considers efficiency and stability. *Computers & Industrial Engineering*, 46(1), 1-15.
- Tao, Z., & Liu, X. (2019). Dynamic Scheduling of Dual-Resource Constrained Blocking Job Shop. In *International Conference on Intelligent Robotics and Applications* (pp. 447-456). Springer, Cham.
- Tighazoui, A., Sauvey, C., & Sauer, N. (2020). New efficiency-stability criterion in a rescheduling problem with dynamic jobs weights. *7th International Conference on Control, Decision and Information Technologies (CoDIT)* (Vol. 1, pp. 475-480). IEEE.
- Trabelsi, W., Sauvey, C., & Sauer, N. (2011). Complexity and Mathematical Model for Flowshop Problem Subject to Different Types of Blocking Constraint. *IFAC Proceedings Volumes*, 44(1), 8183-8188.
- Trabelsi, W., Sauvey, C., & Sauer, N. (2012). Heuristics and metaheuristics for mixed blocking constraints flowshop scheduling problems. *Computers & Operations Research*, 39(11), 2520-2527.
- Uhlmann, I. R., & Frazzon, E. M. (2018). Production rescheduling review: Opportunities for industrial integration and practical applications. *Journal of manufacturing systems*, 49, 186-193.
- Vieira, G. E., Herrmann, J. W., & Lin, E. (2003). Rescheduling manufacturing systems: a framework of strategies, policies, and methods. *Journal of scheduling*, 6(1), 39-62.
- Wang, L., Zhang, L., & Zheng, D. Z. (2006). An effective hybrid genetic algorithm for flow shop scheduling with limited buffers. *Computers & Operations Research*, 33(10), 2960-2971.