



**HAL**  
open science

# Rescheduling of a Single Machine Problem Thanks to Time Dependent Variable Weights

Ayoub Tighazoui, Christophe Sauvey, Nathalie Sauer, Bertrand Rose

► **To cite this version:**

Ayoub Tighazoui, Christophe Sauvey, Nathalie Sauer, Bertrand Rose. Rescheduling of a Single Machine Problem Thanks to Time Dependent Variable Weights. 2023 9th International Conference on Control, Decision and Information Technologies (CoDIT), 03-06 July 2023, Rome, Italy, Jul 2023, Rome, Italy. 10.1109/CoDIT58514.2023.10284229 . hal-04303930

**HAL Id: hal-04303930**

**<https://hal.science/hal-04303930>**

Submitted on 23 Nov 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.

# Rescheduling of a single machine problem thanks to time dependent variable weights

A. Tighazoui<sup>1\*</sup>, C. Sauvey<sup>2</sup>, N. Sauer<sup>2</sup>, B. Rose<sup>1</sup>  
{ayoub.tighazoui, bertrand.rose}@unistra.fr  
{christophe.sauvey, nathalie.sauer}@univ-lorraine.fr

**Abstract**— Rescheduling process consists in ordering in the time all production orders in response to disruptions for optimizing a criterion. When the disruption is due to orders arrival, the heavy weight orders fit into the schedule forcing the low weight ones to wait. Therefore, the low weight orders may not run. For dealing with this issue, in this paper, we propose to reschedule with time dependent variable weights. Indeed, the longer the task remains in the system, the more its weight increases. A single machine rescheduling problem is treated with orders arriving over time. In the predictive phase, a MILP model is implemented to solve a scheduling problem with the Total Weighted Waiting Time (*TWWT*) as an objective. In the reactive phase, when a new order arrives, a MILP is regenerated solving the new problem with the *TWWT* and the stability as an objective. Experimental results discuss the impact of increasing the jobs weight and its relationship with the system performance and the resolution time.

**Keywords**— Rescheduling; time dependent variable weights; single machine; predictive-reactive strategy; waiting time; stability.

## I. INTRODUCTION AND LITERATURE REVIEW

Using new technological tools, the customers have the possibility to create, at any time, an order. These practices change have an impact on the companies' organization. Indeed, the companies must quickly react for revising the schedule in response to the new orders arrival. Hence, rescheduling process is necessary. Rescheduling is defined as a revision of an established schedule in response to new orders arrivals or any process disruption [1,2]. In rescheduling literature, the authors have been interested on different types of disruptions, such as the arrival of new orders [3,4], resources unavailability [5,6,7], job disappearance [8], raw materials lack [9,10].

For measuring the performance of the schedule, many efficiency criteria are often considered in rescheduling problem. such as the Makespan [11,12,13], tardiness [14,15], total flow time [16], and maximum lateness [17]. However, other authors have studied weighted criteria like the total weighted completion times [18,19], the total weighted tardiness [20], or the total weighted waiting time [21,22,23]. The integration of the weight into the performance measure consists of prioritizing the important orders. At each time a new heavy weight order arrives, it is fit into the schedule

forcing the low weight ones to wait. Thus, the low weight orders wait too much before being executed or they will never be executed.

Most of cited works were not interested on this issue although it causes customer dissatisfaction. Differently from existing works, in this paper, we consider the Total Weighted Waiting Time (*TWWT*) for measuring the efficiency of the schedule and we propose to reschedule with time dependent variable weights. Indeed, the longer the task remains in the system, the more its weight increases. The waiting time of an order is the period during which this order has waited in front of workstation before it is carried out [24]. In industrial workshop, the waiting time measures waiting of the task in front of workstation, and the weight can represent the customer's priority. In hospital environment, the waiting time can be regarded as the delay between the patient's arrival and his actual treatment, and the weight is the emergency level. Thus, so as not to the low weight patients become constantly shifted, it is necessary to increase their weights as function of their effective waiting time. In rescheduling literature, Guo et al. [25] is among the first authors who studied the waiting time on scheduling systems. They are interested on rescheduling problem on a single resource for minimizing the total of the waiting times of the tasks, considering the original loads and release time.

Efficiency criterion measures the performance of the schedule. However, in dynamic cases, other kind of criteria can be assumed such as the stability criterion. This latter assesses the orders deviation [26]. Several authors have been interested in the stability criterion [27,28,29]. For example, Pfeiffer et al. [30] use two penalties for assessing the instability of the schedule. In first, they use the jobs starting times deviation, which is the deviation between jobs starting time in the planned schedule and the new one. In second, they use the deviation of the starting time from the actual date. In our systems, as we use a weighted criterion, we regard the Total Weighted Completion Time Deviation (*TWCTD*) as a stability criterion. This latter assesses the deviation of the weighted completion times between the planned schedule and the re-planned one (after the disruption). Indeed, with a heavy weight, it will be difficult to deviate the order, which prevents delaying a rush order.

Two rescheduling strategies are existing in the literature [31]. The first is completely reactive, referred to as dynamic scheduling, it consists of dynamically constructing the schedule using dispatching rules when a new order arrives. It uses the actual information and dispatching rules to decide the order of the job in the list [32]. The second is predictive-

<sup>1</sup>Université de Strasbourg, ICUBE CNRS 7357 Strasbourg, France

<sup>2</sup>Université de Lorraine, LGIPM F-57000 Metz, France

\* Corresponding author

reactive strategy. This latter consists of creating a first schedule and regenerating, at each time a disruption appears, a new schedule [1]. Several authors have already used this approach for rescheduling. For instance, Bahroun et al. [33] use a predictive-reactive approach to reschedule the tasks in a single resource rescheduling problem. Thanks to tentative information, the authors construct a feasible schedule. Then, in the reactive phase, they built new schedules at each time a disruption occurs.

In this paper, we investigate a single resource rescheduling problem, the disruption is due to arrival of new orders. To reschedule the orders in the resource, a predictive-reactive strategy is implemented. In the predictive step, an initial schedule is constructed for optimizing  $TWWT$ . In the reactive step, when a new order arrives, the data of the problem is updated, and a new schedule is generated, simultaneously optimizing the efficiency criterion  $TWWT$  and the stability criterion  $TWCTD$ .

For describing this problem, a MILP model is formulated. The novelty of this model is that it introduces a new concept in the rescheduling problems which is the integration of the dynamic weights. While the orders are not executed, their weights increase. This behavior allows to consider the low weight earliest orders thanks to increasing their priorities as function of time. In the scheduling literature, very few works are interested on this issue [34]. The paper contributes on operational research by:

- Introducing a single machine rescheduling problem with time dependent variable weights. This helps the low weight orders to be planned.

- Implementing a MILP model and a predictive-reactive strategy for rescheduling the orders with the objective of optimizing the efficiency and the stability of the schedule.

- Discussing the impact of increasing the jobs weight and its relationship with the system performance.

The rest of the article is organized as follows. In chapter II, we describe the problem. In chapter III, we formulate the MILP model. In chapter IV, we discuss the experimental results. In the final chapter, we conclude and provide some perspectives.

## II. PROBLEM DESCRIPTION

We consider  $n$  orders that are available to be scheduled in a single resource. For each order  $j$ , we define a release date  $r_j$ , a weight  $w_j$ , and a duration  $p_j$ . The waiting time  $W_j$  of the order  $j$  is the difference between its release date  $r_j$  and its starting time  $S_j$ ,  $W_j = S_j - r_j$ . When the execution of the order  $j$  starts, it will be proceeded without preemption to completion time  $C_j$ .  $C_j = S_j + p_j$ . Thus, the waiting time is also defined as  $W_j = C_j - p_j - r_j$ .

The predictive-reactive strategy consists, in the predictive step, of generating an initial schedule which optimizes the  $TWWT$ , i.e.,  $\sum_{j \in N} w_j W_j$ , defined as the efficiency criterion. During the production, a new order may arrive. This latter will be combined with the not yet executed orders to form a set  $N$ . In the reactive step, we generate a new schedule for optimizing the  $TWWT$  combined with

$TWCTD$  as a stability criterion, i.e.  $\sum_{j \in N} w_j (C_j - C_{0j})$ , with  $C_{0j}$  the original completion time of order  $j$  when it is scheduled for the first time. As the objective of the reactive step is to optimize the efficiency and the stability, we introduced a coefficient  $\beta$  as a weight of each part of the objective function, i.e.,

$$\beta \sum_{j \in N} w_j W_j + (1 - \beta) \sum_{j \in N} w_j (C_j - C_{0j}).$$

### Time dependent variable weights

The proposed weight equation is given by  $w_{jt} = w_j (t - r_j + 1)^\mu$ , the weight becomes a function of time  $w_{jt}$ , release date  $r_j$  and the index  $\mu$ . When the order waits for a long time, the difference between the actual date  $t$  and release  $r_j$  date becomes large. Consequently, the value of  $w_{jt}$  increases. We have added one to the difference so as not to obtain zero weight when  $t = r_j$ . The index  $\mu$  regulates the weight effect, we assume that its value is between 0 and 1 to get relatively small weight increasing values. When  $\mu = 0$ , the orders' weights are static and does not depend on the time. When  $\mu = 1$ , the weight augmentation becomes too large. If  $\mu$  exceeds 1, the weight value will increase excessively. Thus,  $\mu \in [0,1]$ . The objective functions of the predictive and reactive phases are respectively described by Equation (1) and Equation (2):

$$f_1 = \sum_{j \in N} w_{jt} W_j \quad (1)$$

$$f_2 = \beta \sum_{j \in N} w_{jt} W_j + (1 - \beta) \sum_{j \in N} w_{jt} (C_j - C_{0j}) \quad (2)$$

Algorithm 1 describes the predictive-reactive strategy adopted in our approach.

#### Algorithm 1:

Initialize data

Solve the initial problem  $1|r_j|\sum_{j \in N} w_{jt} W_j$

While ( $t \leq T$ ) do

If ( $\gamma(t) = 1$ ) then

Update problem data and constraints

Solve the new problem

$1|r_j|\beta \sum_{j \in N} w_{jt} W_j + (1 - \beta) \sum_{j \in N} w_{jt} (C_j - C_{0j})$

End-if

End-do

The algorithm browses iteratively the horizon time  $T$ . At each time  $t$ , it checks the state of  $\gamma(t)$ . This latter is a binary variable, it gives a value 1 if a new order arrives and 0 otherwise.

$$\gamma(t) = \begin{cases} 1 & \text{if a new order arrives} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

If  $\gamma(t) = 1$ , the algorithm updates the problem data and constraints. Then, it solves the new rescheduling problem.

## III. MILP FORMULATION

In this section, we formulate a MILP model for describing the problem of  $TWWT$  minimization with time dependent variable weights. The MILP formulation consists of assigning orders to positions. It is provided by [22] for a single machine rescheduling problem, we then adapted it to match time

dependent variable weights. Thus, in predictive phase, the parameters, the decision variables, and the constraints are given by:

Parameters:

- $N$ : set of orders  $\{1, 2, \dots, n\}$   
 $K$ : set of positions  $\{1, 2, \dots, n\}$   
 $j$ : index of order,  $j=1, 2, \dots, n$   
 $k$ : index of position,  $k=1, 2, \dots, n$   
 $w_{j0}$ : weight of order  $j$  at time 0  
 $r_j$ : release date of order  $j$   
 $p_j$ : duration of order  $j$   
 $\mu$ : weight index  
 $bigM$ : large value,  $bigM = \sum_{j \in N} p_j + \max r_j$

Decision variables:

$$x_{jk} = \begin{cases} 1, & \text{if order } j \text{ is assigned to position } k \\ 0, & \text{otherwise} \end{cases}$$

- $CP_k$ : completion time of position  $k$ .  
 $SP_k$ : starting time position  $k$ .  
 $C_j$ : completion time of order  $j$ .  
 $W_j$ : waiting time of order  $j$ .

Objective function:  $\min \sum_{j \in N} w_{j0} W_j$

s.t:

$$\sum_{j \in N} x_{jk} = 1, \forall k \in K \quad (4)$$

$$\sum_{k \in N} x_{jk} = 1, \forall j \in N \quad (5)$$

$$SP_k \geq \sum_{j=1}^n r_j x_{jk}, \forall k \in K \quad (6)$$

$$SP_{k+1} \geq SP_k + \sum_{j=1}^n p_j x_{jk}, \forall k \in 1..n-1 \quad (7)$$

$$CP_k = SP_k + \sum_{j=1}^n p_j x_{jk}, \forall k \in K \quad (8)$$

$$C_j \geq CP_k - BigM(1 - x_{jk}), \forall j \in N \forall k \in K \quad (9)$$

$$C_j \leq CP_k + BigM(1 - x_{jk}), \forall j \in N \forall k \in K \quad (10)$$

$$W_j = C_j - p_j - r_j, \forall j \in N \quad (11)$$

$$W_j \text{ and } C_j \geq 0, \forall j \in N \quad (12)$$

$$SP_k \text{ and } CP_k \geq 0, \forall k \in K \quad (13)$$

$$x_{jk} \in \{0,1\}, \forall j \in N, \forall k \in K \quad (14)$$

Equations' description:

Equations (4) and (5) grant a unique match between order and position. Equation (6) ensures starting time of position  $k$  to be greater than or equal to its release date. Equation (7) stands for precedence between the positions. Equation (8) calculates the position completion time. Equations (9) and (10) calculate the order completion time. Equation (11) calculates orders waiting time. Equations (12) and (13) are non-negativity constraints. Equation (14) fixes  $x_{jk}$  domain of variation.

After the arrival of a new order, we combine the new order with the not yet executed orders to form a set  $N'$ .

Updated parameters:

- $t$ : disruption date  
 $N'$ : set of updated orders  $\{1, 2, \dots, n'\}$   
 $K'$ : set of updated positions  $\{1, 2, \dots, n'\}$   
 $j$ : index of order,  $j=1, 2, \dots, n'$   
 $k$ : index of position,  $k=1, 2, \dots, n'$   
 $Co_j$ : original completion time of order  $j$ , calculated when  $j$  is scheduled for the first time.

$x_{o_{jk}}$ : variable assigning order  $j$  to position  $k$  in the last schedule.

$w_{jt}$ : weight of order  $j$  at time  $t$

$\beta$ : efficiency-stability weight.

Updated constraints:

After the disruption occurrence, we use the constraints described in the equations (4) until (14) with the updated parameters. As well, the constraint (15), which allows saving the in-course sequence:

$$\text{if } SP_k < t, x_{jk} = x_{o_{jk}}, \forall j \in N, \forall k \in K \quad (15)$$

Equation (15) makes the order in the same position if its execution is started.

Decision variables:

We use the same decision variable as the one used before. The new objective consists in optimizing the efficiency and stability:

$$\text{Minimize } \beta \sum_{j \in N'} w_{jt} W_j + (1 - \beta) \sum_{j \in N'} w_{jt} (C_j - Co_j)$$

#### IV. EXPERIMENTAL RESULTS

The solver FICO Xpress IVE is used for programming the MILP model. The latter has been performed accordingly to Algorithm 1 on Core i7 2.90 GHz laptop. In Section A, a description of the instances' generation.

##### A. Instances generation

**Table 1** Parameter values generation

Parameters	Values
$T$	48 <i>ut</i>
$w_{j0}$	$\sim U(1,5)$ <i>ut</i>
$p_j$	$\sim U(1,4)$ <i>ut</i>
$r_j$	$\sim U(0,2)$ <i>ut</i>

The instances used in this study have been randomly generated accordingly to Table 1. They can be adapted for real applications. We have assumed that 1 unit of time (*ut*) is equivalent to 10 minutes. Thus, the simulation has been carried out over 8h, representing the factory opening time.  $T = 8h = 480 \text{ min} = 48 \text{ ut}$ .  $w_{j0}$  is the weight value of order  $j$  at  $t=0$ , it varies from 1 to 5.  $w_{j0}$  values were generated following a discrete uniform distribution. It is assumed that the weight represents 5 customer priorities. The duration  $p_j$  is regarded as the product manufacturing duration, it varies following a discrete uniform distribution between 1 to 4 *ut*. At time  $t=0$ , we generate  $r_j$  following a discrete uniform distribution with a parameter between 0 to 2 *ut*. The value of  $\gamma(t)$  is also randomly generated using the Bernoulli distribution, giving at each date  $t$ , a value 1 with a probability  $\psi$  and 0 with a probability  $1 - \psi$ . In the experiment,  $\psi$  is varied according to different values.

##### B. Example of calculation

In Table 2, values parameters for 5 orders are given.

**Table 2 Orders' information**

	Order 1	Order 2	Order 3	Order 4	Order 5
$p_j$	1	4	3	2	2
$r_j$	0	0	1	0	1
$w_{j0}$	1	5	1	3	2

The result of *TWWT* minimization provided by Xpress IVE is  $f_1 = \sum_{j=1}^5 w_{j0} W_j = 36$  with the sequence 4-2-5-1-3. Table 3 presents the solution details,  $S_j$  and  $C_j$  are respectively the starting and completion time of order  $j$ .

**Table 3 Solution of initial problem**

	Order 4	Order 2	Order 5	Order 1	Order 3
$S_j$	0	2	6	8	9
$C_j$	2	6	8	9	12

We can conclude that *WSPT* consisting in scheduling by ascending order of  $p_j/w_{j0}$  is optimal for this problem.

Now, we assume that 2 orders arrive when the sequence is running, order 6 at time 2 and order 7 at time 3. It is assumed that  $\mu = 0.5$ . The new orders' information after the arrival of order 6 and order 7 are respectively given in Tables 4 and 5.

**Table 4 Orders' information after the arrival of order 6**

	Order 1	Order 2	Order 3	Order 4	Order 5	Order 6
$p_j$	1	4	3	2	2	1
$r_j$	0	0	1	0	1	2
$w_{j2}$	1.73	8.66	1.41	5.19	2.82	5

**Table 5 Orders' information after the arrival of order 7**

	Order 1	Order 2	Order 3	Order 4	Order 5	Order 6	Order 7
$p_j$	1	4	3	2	2	1	1
$r_j$	0	0	1	0	1	2	3
$w_{j3}$	2	10	1.73	12	3.46	7.07	1

In this example, we assumed that  $\beta = 1$ , the stability criterion is not considered. The results of  $f_2$  minimization after the arrival of order 6 and order 7 are respectively given in Tables 6 and 7.

**Table 6 Solution responding to order 6 arrival**

	Order 4	Order 6	Order 2	Order 5	Order 1	Order 3
$S_j$	0	2	3	7	9	10
$C_j$	2	3	7	9	10	13

In Table 6, Order 6 arrives at time  $t = 2$ . Order 4 has already begun its execution; it has kept the same position. The rest of jobs have been combined with Order 6 to form  $N'$ . The solution gives  $f_2 = 59.56$ .

**Table 7 Solution responding to order 7 arrival**

	Order 4	Order 6	Order 2	Order 5	Order 1	Order 7	Order 3
$S_j$	0	2	3	7	9	10	11
$C_j$	2	3	7	9	10	11	14

In Table 7, Order 7 arrives at time  $t = 3$ . Order 4 and Order 6 have already begun their execution, they kept the same position. The rest of jobs have been combined with Order 7 to form  $N'$ . The solution gives  $f_2 = 81.87$ . We can remark that *WSPT* rule is respected in these solutions when  $\beta = 1$ .

The weights increase helps Order 1 to be planned before Order 7 since  $p_1/w_{13} < p_7/w_{73}$ , instead of what could happen in the static weight case when  $p_7/w_{73} = p_1/w_{10}$ .

### C. Impact of the weight variation on the flowtime

To evaluate the impact of the weight variation on the orders' flowtime within the system, we assess the Mean Flowtime (*MF*) and Standard Deviation of the Flowtime (*SDF*), varying  $\mu$  from 0 to 1. The flowtime  $F_j$  of order  $j$  measures the time difference between the release date  $r_j$  and the completion time  $C_j$ , how long the order stays in the system,  $F_j = C_j - r_j$ . Thus, the mean flow time calculates the average of the flowtime depending on the number of the orders,  $MF = \frac{1}{n'} \sum_{j \in N'} (C_j - r_j)$ . More, the standard deviation of the flowtime measures the dispersion of flowtime values around *MF*.

The study is established for 10 different instances. Each instance starts with 5 initial orders. Then, new orders arrive with a probability  $\psi = 0.8$ . The 10 instances are tested for different values of  $\beta$  to measure the impact of  $\beta$  on *MF*. In our simulation, the variation of  $\beta$  starts from 0.5 since under this value *TWWT* becomes negligible, that is not realistic. Thus,  $\beta \in \{0.5; 1\}$  and  $\mu \in \{0; 1\}$ . Table 8 presents the averages of *MF* and *SDF* for the tested instances.

**Table 8 Variation of MF and SDF in function of  $\mu$  and  $\beta$** 

$\mu$	$\beta = 0.5$		$\beta = 0.75$		$\beta = 1$	
	<i>MF</i>	<i>SDF</i>	<i>MF</i>	<i>SDF</i>	<i>MF</i>	<i>SDF</i>
0	8	4.49	8	4.49	8	4.83
0.1	8	4.49	8	4.49	8	4.83
0.2	8	4.49	8	4.49	8	4.49
0.3	8	4.49	8	4.49	8	4.49
0.4	8	4.49	8	4.49	8	4.49
0.5	8	4.49	8	4.49	8	4.49
0.6	8.3	4.21	8	4.49	8	4.49
0.7	8.3	3.74	8	4.49	8	4.49
0.8	8.4	3.65	8	4.49	8	4.49
0.9	8.4	3.65	8.3	3.74	8	4.49
1	8.4	3.65	8.3	3.74	8	4.49

The increase of  $\mu$  increases *MF* value. However, *SDF* decreases. *SDF* is the dispersion around the mean flowtime, its value decreases since the values of  $F_j$  approach each other. We can conclude that the increase of  $\mu$  helps the low weight orders to be planned.

Moreover, the augmentation of  $\beta$  makes it difficult to decrease *SDF* since the new orders are, systematically, scheduled in the last position as the stability of the schedule becomes important. In this case, a high value of  $\mu$  is needed to help the low weight orders to be planned. In conclusion, for helping the low weight orders to be planned, we need a high value of the index  $\mu$  when the efficiency-stability weight  $\beta$  is large.



#### D. Impact of the weight variation on the objectives

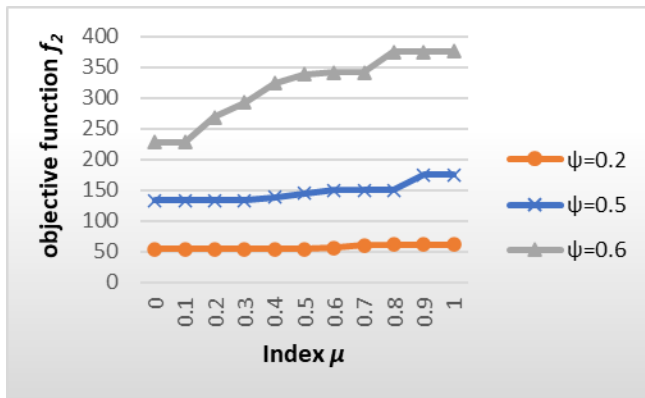
In this subsection, we study the impact of the index  $\mu$  on both objectives. Thus, for different values of  $\mu$ , we calculate  $TWWT$ ,  $TWCTD$  and  $f_2$ . These values are calculated with the initial order weight to ensure the comparability. We also calculate the deviation rate from the static weight case. The tests are performed for the same instances as those presented in Section C. Fixing the value of  $\beta$  on 0.5, different values of  $\psi$  are tested. Table 9 presents the variation of the objectives in function of  $\mu$  and  $\psi$ .

**Table 9 Variation of the objectives in function of  $\mu$  and  $\psi$**

$\mu$	$\psi=0.2$				$\psi=0.5$				$\psi=0.6$			
	a	b	c	D	a	b	c	d	a	b	c	d
0	93	15	54	0	236	31	133.5	0	384	72	228	0
0.1	93	15	54	0	236	31	133.5	0	384	72	228	0
0.2	93	15	54	0	236	31	133.5	0	475	63	269	15
0.3	93	15	54	0	236	31	133.5	0	529	58	293.5	22
0.4	93	15	54	0	248	29	138.5	4	614	34	324	30
0.5	93	15	54	0	270	20	145	8	652	26	339	33
0.6	105	9	57	5	288	13	150.5	11	670	15	342.5	33
0.7	114	7	60.5	11	285	16	150.5	11	670	15	342.5	33
0.8	118	5	61.5	12	285	16	150.5	11	742	8	375	39
0.9	118	5	61.5	12	345	6	175.5	24	742	8	375	39
1	118	5	61.5	12	345	6	175.5	24	745	7	376	39

a:  $TWWT$ , b:  $TWCTD$ , c:  $f_2$ , d: deviation rate from static weight (%)

The value of the stability criterion,  $TWCTD$  decreases when  $\mu$  increases. The proposed formula helps the system stability since it forces the orders to keep the same positions. As the stability becomes more important, the schedule efficiency reduces, which explains the increase of  $TWWT$  with  $\mu$ , especially when the schedule is more disrupted.  $f_2$  increases also with  $\mu$ , especially when  $\psi = 0.6$ . Fig.1 depicts this behavior.



**Fig. 1 Variation of  $f_2$  in function of  $\mu$  and  $\psi$**

The decision maker must choose a value of the index  $\mu$  in function of its priority. We have performed other tests with  $\psi$  exceeding 0.6, but resolution is not possible within a reasonable time.

#### E. Impact of $\psi$ on the resolution time

$\psi$  is the probability of the order arrival in a period  $t$ . The increase of  $\psi$  increases the chance of the order arrival. The aim of this study is to evaluate which problem size the MILP can solve and how fast in function of  $\psi$  evolution. The study is conducted for 10 instances per problem type. Thus, the average of Resolution Time ( $RT$ ) is calculated. The instances contain 5 and 7 prime orders. Over the horizon time  $T = 48$   $ut$ , other orders occur progressively in function of  $\psi$ . Different values of the index  $\mu$  are tested  $\{0; 0.5; 1\}$ .  $\beta$  is fixed on 0.5. The results are presented in Table 10.

**Table 10 Impact of  $\psi$  in the resolution time**

Prime orders	$\Psi$ (TNA $R$ )	Resolution time in seconds		
		$\mu = 0$	$\mu = 0.5$	$\mu = 1$
5 orders	0.2 (10 orders)	0.07	0.08	0.09
	0.5 (24 orders)	0.4	0.39	0.41
	0.6 (28 orders)	233.7	235.2	232.8
	0.7 (33 orders)	1110.6	1112.3	1202.6
7 orders	0.2 (10 orders)	0.11	0.12	0.11
	0.5 (24 orders)	0.64	0.66	0.67
	0.6 (28 orders)	330.8	333.2	337.6
	0.7 (33 orders)	-	-	-

TNA $R$ : Total Number of Arrival Orders

Resolution time depends on both prime orders and  $\psi$ . When these ones increase, resolution time increases also. In this study, we observe a break when  $\psi = 0.7$ , which is averagely equivalent to 33 arriving orders when  $T = 48$   $ut$ .

- With 5 prime orders and  $\psi = 0.7$ , the MILP makes averagely up to 20 min to solve the problem.
- With 7 prime orders and  $\psi = 0.7$ , the MILP is not able to solve the problem. The program ran for 12h, we have then interrupted the simulation.

Varying  $\mu$ , the resolution time remains stable. Hence, time dependent variable weights has no impact on the resolution time. As a conclusion, the MILP can averagely solve a problem up to 5 prime orders subjected to disruptions occurring with a probability of  $\psi = 0.7$ . It is equivalent to 38 orders when  $T = 48$   $ut$ .

## V. CONCLUSION

This study investigates a rescheduling problem with time dependent variable weights. The aim is to propose a new concept for helping low weight orders to be treated when the optimization concerns a weighted criterion. In fact, when the disruption is due to orders arrival, the heavy weight orders fit into the schedule, forcing the low weight ones to wait. The proposed concept consists in increasing the weights of the orders as a function of time. Indeed, the longer the task remains in the system, the more its weight increases. The study is conducted on a single machine rescheduling problem. Predictive-reactive algorithm regenerates, at each period, a MILP model for solving the problem of the total weighted waiting time minimization. Instability of the schedule is also assessed by the weighted completion time deviation. Three important experimentations are performed.

Firstly, for measuring the efficiency of the proposed concept, we have measured the impact of the weight variation on the orders flowtime. Numerical results show that the increase of the weight as a function of time decreases the standard deviation of the flowtime. In other words, the order's flowtimes approximate to each other. Thus, the low weight orders have more chance to be planned.

Secondly, we measured the impact of the weight variation on the objective function. The experimentation shows that the increase of the weight values matches the system

stability since this concept ensures that the orders keep the same positions, that is the principle of the schedule stability. However, schedule efficiency degrades. Ultimately, we studied the impact of instances size on the resolution time. The MILP can solve a problem with up to 38 orders. However, the time dependent variable weights, has no impact on the resolution time of the MILP.

This research work can be very helpful, not only for the researchers working on operational research, but also for industrial managers or operating rooms planners, for instance. The concept of time dependent variable weights can be applied on any scheduling system taking into account the tasks priority. However, our study is established on a single machine rescheduling and the problem resolution is performed with a MILP formulation. Hence, our further works could focus on other machine environments, closer to real systems, such as flexible job shop or hybrid flowshop. Moreover, to browse more orders in a reasonable time, we will design metaheuristic methods.

#### REFERENCES

- [1] 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.
- [2] Mourtzis, D., & Vlachou, E. (2018). A cloud-based cyber-physical system for adaptive shop-floor scheduling and condition-based maintenance. *Journal of manufacturing systems*, 47, 179-198.
- [3] Renner, E., Salassa, F., & T'kindt, V. (2022). Single machine rescheduling for new orders with maximum lateness minimization. *Computers & Operations Research*, 144, 105815.
- [4] An, Y., Chen, X., Gao, K., Zhang, L., Li, Y., & Zhao, Z. (2023). A hybrid multi-objective evolutionary algorithm for solving an adaptive flexible job-shop rescheduling problem with real-time order acceptance and condition-based preventive maintenance. *Expert Systems with Applications*, 212, 118711.
- [5] Yang, Y., Ren, F., & Zhang, M. (2022). A Hybrid Multiagent-Based Rescheduling Mechanism for Open and Stochastic Environments Concerning the Execution Stage. In *Australasian Joint Conference on Artificial Intelligence* (pp. 556-569). Springer, Cham.
- [6] An, Y., Chen, X., Gao, K., Li, Y., & Zhang, L. (2022). Multiobjective Flexible Job-Shop Rescheduling With New Job Insertion and Machine Preventive Maintenance. *IEEE Transactions on Cybernetics*.
- [7] Moon, K., Lee, M., & Lee, K. (2022). Rescheduling Problem for Heavy Cargo Logistics with Transporters. In *IFIP International Conference on Advances in Production Management Systems* (pp. 485-493). Springer, Cham.
- [8] Mor, B.; Mosheiov, G. (2021). Single machine scheduling to maximize the weighted number of on-time jobs with job-rejection. *Oper. Res.* 1–13.
- [9] Wu, T., Ma, X., Yang, L., & Zhao, Y. (2019). Proactive maintenance scheduling in consideration of imperfect repairs and production wait time. *Journal of Manufacturing Systems*, 53, 183-194.
- [10] Mohan, J., Lanka, K., & Rao, A. N. (2019). A review of dynamic job shop scheduling techniques. *Procedia Manufacturing*, 30, 34-39.
- [11] Poongothai, V., Godhandaraman, P., & Jayashri, P. (2020). Parallel machines with rescheduling, fixed due date, machine breakdown to reduce makespan and tardiness. In *AIP Conference Proceedings* (Vol. 2277, No. 1, p. 200004). AIP Publishing LLC.
- [12] Gao, K., Yang, F., Li, J., Sang, H., & Luo, J. (2020). Improved jaya algorithm for flexible job shop rescheduling problem. *IEEE Access*, 8, 86915-86922.
- [13] Valledor, P., Gomez, A., Puente, J., & Fernandez, I. (2022). Solving Rescheduling Problems in Dynamic Permutation Flow Shop Environments with Multiple Objectives Using the Hybrid Dynamic Non-Dominated Sorting Genetic II Algorithm. *Mathematics*, 10(14), 2395.
- [14] Sun, R., & Liu, X. (2022). Approximation scheme for single-machine rescheduling with job delay and rejection. In *International Conference on Algorithmic Applications in Management* (pp. 35-45). Springer, Cham.
- [15] Brik, B., Bettayeb, B., Sahnoun, M. H., & Louis, A. (2019). Accuracy and localization-aware rescheduling for flexible flow shops in industry 4.0. In *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)* (pp. 1929-1934). IEEE.
- [16] Li, X., & Gao, L. (2020). A Hybrid Intelligent Algorithm and Rescheduling Technique for Dynamic JSP. In *Effective Methods for Integrated Process Planning and Scheduling* (pp. 345-375). Springer, Berlin, Heidelberg.
- [17] Nicosia, G., Pacifici, A., Pferschy, U., Resch, J., & Righini, G. (2021). Optimally rescheduling jobs with a LIFO buffer. *arXiv preprint arXiv:2106.15901*.
- [18] Luo, W., Luo, T., Goebel, R., & Lin, G. (2018). Rescheduling due to machine disruption to minimize the total weighted completion time. *Journal of Scheduling*, 21(5), 565-578.
- [19] Curry, J., & Peters\*, B. (2005). Rescheduling parallel machines with stepwise increasing tardiness and machine assignment stability objectives. *International Journal of Production Research*, 43(15), 3231-3246.
- [20] Yang, B. (2007). Single machine rescheduling with new jobs arrivals and processing time compression. *The International Journal of Advanced Manufacturing Technology*, 34(3), 378-384.
- [21] Tighazoui, A., Sauvey, C., & Sauer, N. (2021). New efficiency-stability criterion for flow shop rescheduling problem with mixed blocking constraints. *IFAC-PapersOnLine*, 54(1), 1181-1186.
- [22] Tighazoui, A., Sauvey, C., & Sauer, N. (2021). Minimizing the Total Weighted Waiting Times and Instability in a Rescheduling Problem with Dynamic Jobs Weight. *Applied Sciences*, 11(15), 7040.
- [23] Tighazoui, A., Sauvey, C., & Sauer, N. (2021). Predictive-reactive strategy for identical parallel machine rescheduling. *Computers & Operations Research*, 134, 105372.
- [24] Jurčičin, R., & Šebo, J. (2015). Basic production scheduling concept software application in a deterministic mechanical production environment. *Acta Simulatio*, 1(4), 1-4.
- [25] 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.
- [26] Mejía, G., Montoya, C., Bolívar, S., & Rossit, D. A. (2022). Job shop rescheduling with rework and reconditioning in Industry 4.0: an event-driven approach. *The International Journal of Advanced Manufacturing Technology*, 119(5), 3729-3745.
- [27] Masuchun, R., & Ferrell, W. G. (2004, July). Dynamic rescheduling with stability. In *2004 5th Asian Control Conference (IEEE Cat. No. 04EX904)* (Vol. 3, pp. 1876-1884). IEEE.
- [28] Cui, W.; Lu, Z.; Li, C.; Han, X. (2018). A proactive approach to solve integrated production scheduling and maintenance planning problem in flow shops. *Comput. Ind. Eng.* 115, 342–353.
- [29] Akkan, C. (2015). Improving schedule stability in single-machine rescheduling for new operation insertion. *Comput. Oper. Res.* 64, 198–209.
- [30] Pfeiffer, A., Kádár, B., & Monostori, L. (2007). Stability-oriented evaluation of rescheduling strategies, by using simulation. *Computers in Industry*, 58(7), 630-643.
- [31] Herrmann, J.W. (2006). Rescheduling Strategies, Policies, and Methods. In *Handbook of Production Scheduling*; Springer: Boston, MA, USA, pp. 135–148.
- [32] El-Bouri, (2012). A cooperative dispatching approach for minimizing mean tardiness in a dynamic flowshop. *Comput. Oper. Res.* 39, 1305–1314.
- [33] Bahroun, Z.; Shamayleh, A.; As'ad, R.; Zakaria, R. (2021). Flexible decision support tool for dynamic single machine scheduling problems. *J. Ind. Prod. Eng.* 38, 213–238.
- [34] Zhang, B., Pan, Q. K., Meng, L. L., Zhang, X. L., & Jiang, X. C. (2022). A decomposition-based multi-objective evolutionary algorithm for hybrid flowshop rescheduling problem with consistent sublots. *International Journal of Production Research*, 1-26.