



**HAL**  
open science

## Predictive-reactive strategy for identical parallel machine rescheduling

Ayoub Tighazoui, Christophe Sauvey, Nathalie Sauer

► **To cite this version:**

Ayoub Tighazoui, Christophe Sauvey, Nathalie Sauer. Predictive-reactive strategy for identical parallel machine rescheduling. *Computers and Operations Research*, 2021, 134, pp.105372. 10.1016/j.cor.2021.105372 . hal-03254513

**HAL Id: hal-03254513**

**<https://hal.science/hal-03254513>**

Submitted on 14 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.

# Predictive-reactive strategy for identical parallel machine rescheduling

Ayoub Tighazoui, Christophe Sauvey, Nathalie Sauer

Université de Lorraine, LGIPM, F-57000 Metz, France  
{ayoub.tighazoui; christophe.sauvey; nathalie.sauer}@univ-lorraine.fr

## Abstract

Through new technologies development, customers can create or cancel an order in real time, which disrupts the established production schedule. This reality forces many companies to quickly react for dealing with this situation through rescheduling processes. When efficiency usually measures the performance of a scheduling system, in dynamic environments, stability measures the impact of jobs deviation. Differently from previous works, this paper investigates a new approach to measure simultaneously efficiency by total weighted waiting times, and stability by weighted completion time deviation. This mix could be a very helpful and significant criterion in industrial and health care environments. This paper considers a parallel machine rescheduling problem with jobs arriving over time. Based on predictive-reactive strategy, a mixed integer linear programming model is developed, as well as an online iterative methodology to solve this rescheduling problem. Finally, numerical results are performed for studying the impact of efficiency and stability coefficient, as well as the computing time to solve the described problem.

**Keywords:** Rescheduling, Parallel Machine, Waiting time, Stability, Completion time deviation, Disruption.

## 1. Introduction

Due to the fourth revolution experiencing, today's industry is developing new technologies and solutions for the production process. In addition, the customers' practices are also changed. Henceforth, a customer can, at any time, create or cancel an order in real time. Indeed, these changed situations have a direct impact on work organization, especially production scheduling (Ivanov *et al.* 2016; Uhlmann, and Frazzon 2018; Da Silva *et al.*, 2019). In fact, production planning is subject to unexpected events that disrupt the established schedule, such as: new jobs arrival, cancelation of jobs, machine failures, quality problems, shortage of raw materials, etc. (Li *et al.*, 1993; Sabuncuoglu and Karabuk 1999; Katragjini *et al.*, 2013). Hence, the decision makers are forced to quickly react to deal with these disruptions. Therefore, rescheduling is necessary to revise the initial schedule in a cost-effective manner (Vieira *et al.*, 2003).

Rescheduling is the process of updating an established schedule in response to disruptions (Vieira *et al.*, 2003). In the rescheduling literature, the classical efficiency criteria are often used to measure the performance of a schedule, including: the Makespan, the total completion time, the maximum lateness, etc. Differently from previous researches, this work considers a new efficiency criterion, which is the *Total Weighted Waiting Times (TWWT)*. On the scheduling literature, the waiting time is the period that the job has waited before its execution in a system (Jurčišin and Šebo 2015). Indeed, the *TWWT* could be a very helpful and significant criterion in industrial and health care environments. For instance, in production systems, it can represent

41 the waiting time of a job in front of a workstation, considering the importance of each job as  
42 the weight. In hospital systems, the waiting time can represent the delay between a patient's  
43 arrival and his actual treatment, considering the emergency level as the weight. [Guo et al.,  
44 \(2013\)](#) is among the first works that considered the waiting time on rescheduling problems. The  
45 authors formulated a single machine rescheduling problem to minimize the sum of waiting  
46 times of rework jobs and the original loads with release time. They proved that the problem is  
47 NP-hard and designed a dynamic insert heuristic algorithm of polynomial-time. To fill the void  
48 in existing literature, in our study, we considered the *TWWT* as an efficiency scheduling  
49 measure, which has been defined for the first time by [Kan \(1976\)](#), and classified as NP-hard for  
50 a single machine. To the best of our knowledge, there is not any rescheduling problem in the  
51 literature which discussed this criterion.

52 The efficiency usually measures the performance of a scheduling system, but, in dynamic  
53 environments, the impact of jobs deviation is also studied in the literature, known as stability  
54 measure ([Wu et al., 1993](#); [Pfeiffer et al., 2007](#); [Zhang et al., 2013](#)). This criterion measures the  
55 impact of disruptions induced by moving jobs during a rescheduling event ([Rangaritratsamee  
56 et al., 2004](#)). In real life systems, when the sequence is changed, this matter may generate  
57 additional costs, such as reallocation costs, raw-materials reordering costs, etc ([Rahmani and  
58 Ramezani 2016](#)). [Ahmadi et al., \(2016\)](#) studied a multi-objective jobshop rescheduling  
59 problem, in which the disruptions were caused by random machine breakdown. The authors  
60 considered the makespan as the efficiency measure, and calculated the stability with the average  
61 difference between the completion times of the predicted schedule and the realized completion  
62 time. In fact, when a disruption occurs, initial completion times of jobs may change and are  
63 replaced by real completion times obtained after disruption. Two evolutionary algorithms are  
64 applied to solve this problem, with the objective simultaneously combines makespan  
65 improvement and stability. Indeed, this paper presents a new approach to measure the schedule  
66 stability, which consists in calculating the difference between the completion times of the job  
67 in the initial schedule and in the new one, associating for each job a weight. We denote by the  
68 *Total Weighted Completion Time Deviation (TWCTD)* this stability criterion. In fact, when a  
69 job has a high weight (i.e. more important), it is difficult to disrupt it. This is the case of an  
70 urgent patient in a hospital, for example.

71 According to ([Vieira et al., 2003](#); [Herrmann 2006](#)), there are two basic rescheduling  
72 strategies. The first is the *dynamic scheduling strategy* which consists in dispatching jobs when  
73 it is necessary and using the available information at the moment of dispatching. This strategy  
74 uses dispatching rules or other heuristics to prioritize jobs waiting to be processed at a  
75 workstation ([El-Bouri 2012](#); [Nurre and Sharkey 2018](#); [Bektur, and Saraç 2019](#); [Riahi et al.,  
76 2019](#)). The second is the *Predictive-reactive scheduling strategy* that consists in generating an  
77 initial schedule in a first step, and to update the schedule in response to a disruption ([Vieira et  
78 al., 2003](#); [Duenas and Petrovic 2008](#)). In our approach, we firstly generate an initial schedule  
79 with the objective of minimizing the *TWWT*. After a disruption, we generate a second schedule  
80 to minimize a multi-criterion combining simultaneously efficiency and stability, with the  
81 stability measured with the *Total Weighted Completion Time Deviation*.

82 The analysis of scheduling performance often requires more than one aspect (Loukil et al.,  
83 2005). Therefore, in the rescheduling literature, several papers deal with multi-objective  
84 function. For instance, Wang, D et al., 2015 considered a rescheduling problem disrupted by  
85 machine breakdown and deterioration effect. The authors aim to minimize a multi objective  
86 function, considering initial operational cost assessed by the total completion time and resource  
87 consumption, as well as the rescheduling cost assessed by match-up time and additional  
88 resource cost. In this paper, we established a relationship between both parts of the objective  
89 function, by associating a coefficient for each measure (i.e.  $\alpha$  efficiency +  $(1 - \alpha)$  stability). We  
90 called  $\alpha$  the *efficiency-stability coefficient*, and studied the influence of this coefficient on the  
91 system.

92 According to Alagöz and Azizoğlu (2003), in the flexible manufacturing systems, changes  
93 in jobs assignments to machines may generate some additional costs, like setup and tooling  
94 costs. Thus, the authors considered the number of jobs processed on different machines after  
95 the disruption as the stability measure in parallel machines rescheduling problem. In the  
96 operating rooms scheduling case, this matter can also generate additional costs related to the  
97 preparation phase before the operation. In our study, we have used the lexicographic  
98 optimization to reduce the number of jobs processed on different machines after disruption.  
99 This method is investigated in many papers dealing with the multi objective problems. It always  
100 provides a Pareto optimal solution (Chang 2015). It consists in obtaining an arbitrarily small  
101 improvement for the most important criterion through any loss in other less important criteria  
102 (Zykina 2004). In our work, we have assumed the bi-objective function considering  
103 simultaneously the efficiency and the stability as the priority function. From the optimal value  
104 obtained for this objective function, we minimize then the number of jobs processed on different  
105 machines, taking as constraint the optimal value of the priority function.

106 This work addresses a rescheduling problem on identical parallel machines, when the  
107 disruption is caused by the arrival of new jobs. A mixed integer linear programming model is  
108 developed to describe the problem, and an online iterative methodology to solve it. The  
109 proposed methodology provides, for each step, the optimal solution. The presented  
110 rescheduling problem has not been studied in the research literature before. The main  
111 contributions of this work are as follows:

- 112 – We implement a mathematical model based on sequence formulation, after we have  
113 improved it to match the parallel machines problem. Besides, we develop an online  
114 iterative methodology adapting this model to our rescheduling problem, and we  
115 discuss its limitation in function of the number of jobs.
- 116 – We consider for the first time ‘*TWWT*’ as a criterion in a parallel machine rescheduling  
117 problem, which is a very significant criterion in several environments.
- 118 – We adopt a new approach in parallel machine rescheduling problem by simultaneously  
119 considering the efficiency defined by the *TWWT*, and the stability as the *Total*  
120 *Weighted Completion Time Deviation*.
- 121 – We establish a lexicographic optimization, by minimizing the number of jobs which  
122 are processed on different machines after disruption.

123 The rest of the paper is organized as follows: we present in section 2 the literature review.  
124 Our research motivation is given in section 3. In section 4 the problem description, as well as  
125 the strategy that we propose to tackle it. In section 5, the mathematical model is presented. We  
126 then analyze the numerical results in section 6. We conclude and propose some perspectives in  
127 section 7.

## 128 2. Literature review

129

130 In this section, we focus on the different issues studied in the rescheduling. The rescheduling  
131 literature is very rich. [Vieira et al. \(2003\)](#), [Li and Ierapetritou \(2008\)](#), [Ouelhadj and Petrovic \(2009\)](#), and recently [Uhlmann, and Frazzon \(2018\)](#) provided interesting reviews of the state-of-  
132 the-art of developing research on rescheduling at those times. In the rescheduling literature, the  
133 classical efficiency criteria are often used to measure the performance of a schedule, including  
134 the makespan ([Yuan and Mu 2007](#); [Zakaria and Petrovic 2012](#); [Tang et al., 2016](#); [Li et al. 2017](#);  
135 [Wu et al., 2018](#)), the total completion time ([Lu et al., 2003](#); [Yang 2007](#)), the total weighted  
136 completion time ([Hall and Potts 2010](#); [Luo et al., 2018](#)), the average flowtime, the average  
137 tardiness and the percentage tardy ([Kunnathur et al, 2004](#)), the total weighted tardiness ([Mason  
138 et al., 2004](#); [Kaplan and Rabadi 2015](#)), the total flow time ([Ozlen and Azizoğlu 2011](#)), and the  
139 maximum lateness ([Hall and Potts 2004](#)). As well, the rescheduling literature has taken interest  
140 in studying different types of machine environments such as: single machine ([Unal et al., 1997](#);  
141 [Hall and Potts 2004](#); [Hall et al., 2007](#)), flowshop ([Yan-hai et al., 2005](#); [Katragjini et al., 2013](#)),  
142 jobshop ([Bierwirth and Mattfeld 1999](#); [Mason et al., 2004](#); [Salido et al., 2017](#); [Yan et al., 2018](#);  
143 [Larsen and Pranzo 2019](#)), or openshop ([Liu and Zhou 2013b](#)). Moreover, there are also several  
144 papers dealing with parallel machines rescheduling problems. [Yin et al., \(2016\)](#) studied a  
145 rescheduling problem on identical parallel machines, and considered the total completion time  
146 as the original scheduling objective and the maximum time deviation or the total virtual  
147 tardiness with respect to the original schedule as the deviation cost associated with a disruption  
148 caused by machine breakdowns. The complexity of the described problem is studied and a  
149 pseudo-polynomial time solution algorithm is proposed. [Hamzadayi and Yildiz \(2016\)](#) treated  
150 a dynamic identical parallel machines scheduling problem, in which sequence dependent setup  
151 operations are performed by a single server. An event driven rescheduling strategy is proposed  
152 to control the dynamic system, with the objective of minimizing the makespan. The authors  
153 indicated that the simulated annealing based complete rescheduling approach produces the best  
154 performance. [Alagöz and Azizoglu \(2003\)](#) addressed a rescheduling problem in parallel  
155 machine environments under machine eligibility constraints. The authors considered the total  
156 flow time as efficiency measure, and the number of jobs processed on different machines after  
157 disruption as stability measure. Several heuristics are proposed to provide a set of approximate  
158 efficient schedules. For other related papers, dealing with parallel machine rescheduling, the  
159 reader may refer to [Church and Uzsoy \(1992\)](#); [Vieira et al., \(2000\)](#); [Azizoglu and Alagöz  
160 \(2005\)](#); [Curry and Peters \(2005\)](#); [Duenas and Petrovic \(2008\)](#); [Arnaout \(2010\)](#); [ShangGuan et  
161 al., \(2012\)](#); [Liu and Zhou \(2013a\)](#), and recently [Wang et al., \(2018\)](#); [Ferrer et al.,\(2018\)](#);  
162 [Kovalyov et al., \(2019\)](#); [Cunha et al., \(2020\)](#).

164 The waiting time is the period during which a job is waiting before its execution in a system  
165 (Jurčišin and Šebo 2015). In the scheduling literature, this criterion is used to measure the  
166 schedule efficiency. Guo *et al.*, (2016) treated a rescheduling problem with the objective of  
167 minimizing the maximum waiting time, inspired from quartz manufacturing industry, where  
168 waiting times represent the waiting for some materials before the reheating step in oven, where  
169 both the rework stream and the regular production stream meet. Thus, the authors studied and  
170 analyzed the complexity of a rescheduling problem on a single machine, with the objective of  
171 deciding where to insert the rework jobs in an initial sequence of regular jobs to minimize  
172 maximum waiting time. Guo and Xie (2017) formulated a two mixed integer programming  
173 models for a single machine rescheduling problem with the objective of minimizing the total  
174 waiting time. The studied problem came also from quartz glass factory, considering that the  
175 waiting time is the waiting for materials before the welding step, and its minimization saves the  
176 energy consumption. However, in dynamic environments, stability criteria are also studied.  
177 Several papers in the literature deal with this criterion. Akkan (2015) considered a multi  
178 objective rescheduling problem that simultaneously measures the scheduling efficiency with  
179 the maximum tardiness, and the stability with the sum of absolute starting time deviations,  
180 which are the sum of absolute difference between starting times of jobs before and after  
181 rescheduling. The author demonstrated that his algorithm improves the stability of schedules  
182 with no degradation in the maximum tardiness. Gao *et al.* (2018) studied a rescheduling  
183 problem on a flexible jobshop, with newly arrived priority jobs which are inserted in the existing  
184 schedule. The authors considered a bi-objective rescheduling criterion. The first is the  
185 efficiency, and the second is the stability which is defined as the percentage of machines-  
186 changed operations of the existing jobs during rescheduling. Pfeiffer *et al.*, (2007) proposed a  
187 schedule stability measure calculated with two penalties. The first is the starting time deviation  
188 penalty, which is the difference between starting times of jobs at the new and previous  
189 rescheduling points. The second is the penalty related to the deviation of job starting time from  
190 the current time. Moreover, we found other related works dealing with stability measure in  
191 rescheduling problems, such as starting time deviation and total deviation penalty  
192 (Rangsaritratamee *et al.*, 2004), starting time deviation (Cui *et al.*, 2018), the amount of  
193 operations and starting times operations which have been altered (Li *et al.*, 2015, Peng *et al.*  
194 2018), the absolute positional disruption, the positional disruption and absolute completion time  
195 disruption (Hoogeveen *et al.*, 2012), the total sequence disruption (Yuan *et al.*, 2007), the  
196 maximum time disruption (delivery times of jobs to customers changes) (Liu and Ro 2014). For  
197 considering the both aspects, efficiency combining with stability, the most of the treated  
198 mathematical models use a multi objective function, such as (Wu *et al.*, 1993;  
199 Rangsaritratamee *et al.*, 2004; Qi *et al.*, 2006; Wang *et al.*, 2017; Peng *et al.*, 2019).

200 In the rescheduling literature, we found also several papers dealing with operating rooms  
201 rescheduling, modeling the problem as a machine scheduling problem. For instance, Stuart and  
202 Kozan (2012) addressed a rescheduling problem of a day-to-day running of a day surgery unit,  
203 modeling the problem as a single machine scheduling problem, with sequence dependent  
204 processing time and due date. They proposed a branch and bound algorithm to solve the  
205 problem. Wang, B *et al.*, (2015) discussed a surgery scheduling problem for single surgical  
206 suite, where each patient should be treated in same order by three stages. The system is

207 disrupted by the arrival of random emergency surgeries. The authors handled the system as a  
208 no-wait permutation flowshop scheduling problem and developed a predictive reactive  
209 approach to tackle it. For other related papers, the reader may refer to [Fei et al., \(2010\)](#), [Farrokhi](#)  
210 [et al., \(2014\)](#), [M’Hallah and Al-Roomi \(2014\)](#), [Addis et al., \(2016\)](#), [Zhong and Shi \(2018\)](#).

211 The *Predictive-reactive scheduling strategy* consists in generating an initial schedule in a  
212 first step, and to update it in response to a disruption ([Vieira et al., 2003](#)). In the literature,  
213 several researchers have adopted predictive-reactive approach. For example, [Yang and Geunes](#)  
214 [\(2008\)](#) considered a predictive–reactive scheduling on a single machine with uncertain future  
215 jobs. The authors determined the amount of planned idle time for uncertain jobs and their  
216 positions in the predictive schedule. Then, the schedule reacts when a disruption occurs, through  
217 including the new jobs in the idle times. [Gürel et al. \(2010\)](#) developed an anticipative  
218 scheduling approach based on a predictive reactive strategy, on parallel non-identical machines  
219 with controllable processing times. Firstly, an initial schedule is designed to minimize the total  
220 manufacturing cost of the jobs. When a machine breakdown occurs, a reactive schedule is  
221 generated and the remaining schedule is repaired.

### 222 3. Motivation

223

224 Our research motivation is inspired from real cases which can be encountered, either in  
225 hospital environment or in industrial environment. In what follows, we present an illustration  
226 of our problem in the context of hospital systems and in the context of industrial systems.

227 Among the hospital departments, the Operating Theater (*OT*) represents the largest hospital  
228 cost category by 33% ([Macario et al., 1995](#)). Hence, operating rooms scheduling has been  
229 identified as an interesting potential area by hospital decision makers. Moreover, among the  
230 heard complaints in the health care services, the long waiting lists are the most received  
231 ([Cardoen et al., 2010](#)). Therefore, it is relevant to react on the minimization of the patient  
232 waiting time, to reduce hospital costs and improve patient’s satisfaction. In practice, there are  
233 two types of operations, the elective operations that are known in advance, and the emergency  
234 operations caused by the arrival of an urgent patient. These ones are unknown in advance, and  
235 represent a disruption for the already established schedule. To cope with this uncertainty,  
236 [Wullink et al. \(2007\)](#) proved that the approach consisting to share *OT* between elective and  
237 emergency operations is better than the approach of dedicating one or several *OTs* for  
238 emergency operations. That justifies our approach of modelling the system as a parallel machine  
239 rescheduling problem. In addition, according to surgery priorities, the patients can be divided  
240 to five majors’ groups ([Abedini et al., 2016](#)). Thus, we have considered for each job a weight  
241 representing its emergency level. The aim of our study is to model this practical example as a  
242 parallel machine rescheduling problem, consisting in the predictive (elective) step to minimize  
243 the patient weighted waiting times, and in the reactive step (i.e. when the emergency operations  
244 occur), the same criterion combined with the stability, in order not to deviate too much from  
245 the initial schedule.

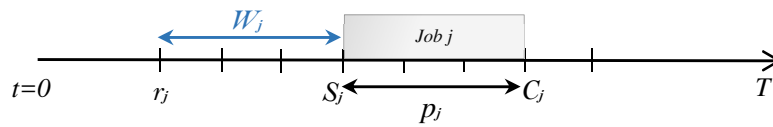
246 This described system can also be an illustration of industrial cases, as described by [Guo et](#)  
247 [al. \(2016\)](#), where the waiting time is the waiting period for some materials before the reheating

248 step in oven, with disruptions caused by the arrival of new orders. The weight can also be  
 249 considered, representing the customer priority.

#### 250 4. Problem description

251 We study a parallel machine rescheduling problem, under disruptions caused by the arrival  
 252 of new jobs. For each job  $j$ , we associate a processing time  $p_j$ , a release date  $r_j$  and a weight  $w_j$ .  
 253 When a job  $j$  starts at time  $S_j$ , it is processed to completion time  $C_j$  without interruption (i.e.  $C_j$   
 254 =  $S_j + p_j$ ). The waiting time  $W_j$  of job  $j$  is the period during which the job has waited, from its  
 255 entry into the system  $r_j$ , until its execution at the starting time. Then, it is defined as  $W_j = S_j -$   
 256  $r_j = C_j - r_j - p_j$  (see Fig.1).

257



258

259

Fig. 1 The waiting time of job  $j$

260 In the studied system, we supposed that we have a set of  $n$  jobs,  $N = \{1, 2, \dots, n\}$ , already  
 261 available to be scheduled in the set of parallel machines  $M = \{1, 2, \dots, nm\}$ , all the jobs  
 262 information  $(p_j, r_j, w_j)$  is available at the beginning. An initial schedule should be determined  
 263 with the objective of minimizing the  $TWWT$ , i.e.  $\sum_{j \in N} w_j W_j$ . Indeed, a mathematical model is

264 developed to solve the initial problem, that can be represented in standard notation by  
 265  $P | r_j | \sum w_j W_j$ . During execution of the jobs, the initial schedule can be disrupted by the  
 266 arrival of new jobs. Consequently, a new set of jobs is generated,  $N' = \{1, 2, \dots, n'\}$ , which  
 267 contains the unexecuted jobs (i.e. the jobs that haven't started) as well as new arrival jobs. The  
 268 methodology is to omit the already executed jobs, and to combine the unexecuted jobs with  
 269 new arrival ones to create an updated schedule. Therefore, a new reactive schedule is generated  
 270 to update the existing one, simultaneously considering efficiency measured by original  
 271 objective  $\sum_{j \in N'} w_j W_j$  and stability measured by the deviation from the original schedule, as

272 described hereafter.

273 *Stability measure:* If no disruption occurs, the job  $j$  will finish processing at completion time  
 274 of original schedule  $Co_j$ . Else, if a disruption occurs and a new job is inserted inside the old  
 275 schedule, some jobs will change their positions. Hence, a new schedule will be established.  
 276 Consequently, the job  $j$  will really finish in the real completion time  $C_j$ . Then, the difference  
 277 between  $Co_j$  and  $C_j$  can be used to assess the deviation cost. In addition, we used the weight  $w_j$   
 278 for each job, which makes it difficult to deviate important jobs. Hence, the stability objective is  
 279 defined as  $\sum_{j \in N} w_j (C_j - Co_j)$ . We called it the *Total Weighted Completion Times Deviation*

280 ( $TWCTD$ ). In this approach, we only calculate the stability criterion for the jobs existing in the  
 281 previous schedule ( $j \in N$ ). However, once the new jobs are scheduled in their first sequence,



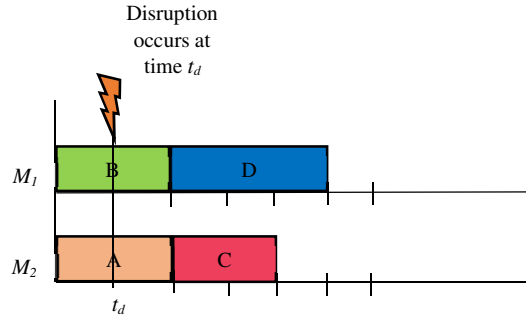
282 they will be concerned by the stability criterion in the ulterior schedules. When  
 283  $w_j (C_j - Co_j) > 0$ , a deviation has occurred and an instability penalty is generated.

284 Along with that, we established a relationship between the both parts of objective functions,  
 285 by integrating  $\alpha$  as an efficiency-stability coefficient. Thus,  $\alpha$  is associated to efficiency part  
 286 and  $(1 - \alpha)$  to stability, where  $\alpha$  is a real number between zero and one. Finally, we defined the  
 287 new objective function that simultaneously measures efficiency and stability as  
 288  $\alpha \sum_{j \in N'} w_j W_j + (1 - \alpha) \sum_{j \in N} w_j (C_j - Co_j)$ . Then, the dynamic rescheduling problem can be

289 represented in standard notation by:  $P | r_j | \alpha \sum w_j W_j + (1 - \alpha) \sum w_i (C_j - Co_j)$ .

290 In the rescheduling process, the disruptions occur while the machine executes jobs.  
 291 Therefore, we noted by  $t_d$  the current time when a disruption occurs (see Fig.2).

292



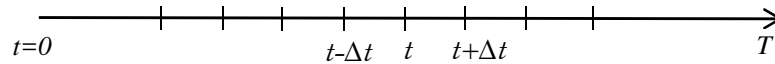
293 **Fig. 2 Disruption occurred at time  $t_d$**

294 Indeed, all executed jobs before  $t_d$  must keep the same position. Thus, these jobs are omitted,  
 295 and the new set  $N'$  will be scheduled, then a new sequence will be established, it begins, for  
 296 each machine, from the completion time of the last executed job.

### 297 **Online iterative methodology**

298 Firstly, we have discretized the finite time horizon  $[0, T]$ , into periods  $\Delta t$  (see Fig.3). Indeed,  
 299  $\Delta t$  is the time period length which represents the time step. The occurrence of disruptions may  
 300 be possible only at these times. In fact, when a job occurs at time  $t$ , this date will be its release  
 301 date. In order to simplify the calculations, we assumed that  $\Delta t = 1$  unit of time, the same time  
 302 unit that we used for the jobs data  $p_j$ ,  $r_j$ , and  $w_j$ .

303



304

305 **Fig. 3 Time discretization**

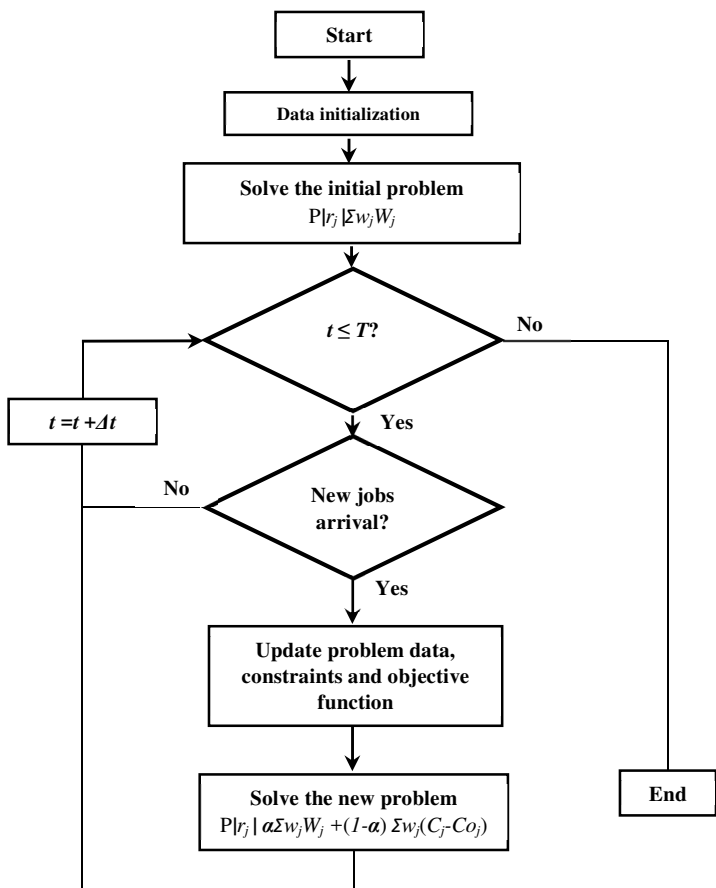
306 We then defined  $\theta(t)$  a binary variable, equals to 1 if a job (or several jobs) occurs at time  $t$ ,  
 307 and 0 otherwise. The generation of its values is explained in the numerical results section. The  
 308 algorithm below describes the online iterative methodology.

```

309 Solve the initial problem
310 LIST = {J1, ..., Jn}
311 Memorize solution
312 For (t in 1 to T)
313   If (θ(t)=1)
314     - Add new job(s) to LIST.
315     - Omit the already executed jobs from LIST (comparing their memorized
316       starting time with t)
317     - Reschedule the new problem with jobs in LIST
318     - Memorize solution
319   End If
320 End For

```

321 The following flowchart describes the predictive-reactive strategy that we used to tackle the  
322 problem (see Fig.4).



323 Fig. 4 The flowchart of predictive-reactive strategy

324 **5. Mathematical model**

325 A mixed integer linear programming model is implemented in this section, it is based on  
326 predictive-reactive strategy, Thus, in the first subsection, we present the mathematical model  
327 before disruption. In the second subsection, we present the mathematical model after disruption.

328 **5.1 Mathematical model before disruption**

329 The proposed model is based on sequence position formulation (assigning a job to a  
 330 position). In the scheduling literature, several papers used this formulation. For instance, Eren  
 331 (2009), Baker and Keller (2010), Kooli and Serairi (2014) and Guo and Xie (2017) in single  
 332 machine scheduling problems. As well, other authors, like Abdel-Jabbar *et al.* (2014) and  
 333 Beezão *et al.* (2017), used this formulation for parallel machine scheduling problems. However,  
 334 most of this works consider the Makespan as a criterion. Indeed, our problem consists in  
 335 minimizing the *TWWT*. Hence, supplementary data and variables are considered, such as  
 336 (weights, release dates, waiting time). In fact, we were inspired by Beezão *et al.* (2017) model  
 337 and have adapted it to our variables and constraints. The parameters, variables and constraints  
 338 of the model are given below:

339 Parameters:

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

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

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

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

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

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

346  $w_j$ : weight of job  $j$

347  $r_j$ : release date of job  $j$

348  $p_j$ : processing time of job  $j$

349  $bigM$ : big value,  $bigM = \sum_{j=1}^n p_j + \max r_j$

350 Decision variables:

351 
$$x_{jkm} = \begin{cases} 1 & \text{if the job } j \text{ is assigned to machine } m \text{ in } k^{\text{th}} \text{ position} \\ 0 & \text{Otherwise} \end{cases}$$

352  $CP_{km}$ : Completion time of job in  $k^{\text{th}}$  position in machine  $m$ .

353  $SP_{km}$ : Starting time of job in  $k^{\text{th}}$  position in machine  $m$ .

354  $C_j$ : Completion time of job  $j$ .

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

356 Objective function:  $Min \sum_{j \in N} w_j W_j$

357  $s.t:$

$$\sum_{m=1}^{nm} \sum_{k=1}^n x_{jkm} = 1 \quad \forall j \in N \quad (1)$$

359

$$360 \quad \sum_{j=1}^n x_{jkm} \leq 1 \quad \forall k \in K, \forall m \in M \quad (2)$$

$$361 \quad \sum_{j=1}^n x_{j(k+1)m} \leq \sum_{j=1}^n x_{jkm} \quad \forall k = 1..n-1, \forall m \in M \quad (3)$$

$$362 \quad SP_{km} \geq \sum_{j=1}^n r_j x_{jkm} \quad \forall k \in K, \forall m \in M \quad (4)$$

$$363 \quad SP_{(k+1)m} \geq SP_{km} + \sum_{j=1}^n p_j x_{jkm} \quad \forall k = 1..n-1, \forall m \in M \quad (5)$$

$$364 \quad CP_{km} = SP_{km} + \sum_{j=1}^n p_j x_{jkm} \quad \forall k \in K, \forall m \in M \quad (6)$$

$$365 \quad C_j \geq CP_{km} - bigM (1 - x_{jkm}) \quad \forall j \in N, \forall k \in K, \forall m \in M \quad (7)$$

$$366 \quad C_j \leq CP_{km} + bigM (1 - x_{jkm}) \quad \forall j \in N, \forall k \in K, \forall m \in M \quad (8)$$

$$367 \quad W_j = C_j - r_j - p_j \quad \forall j \in N \quad (9)$$

$$368 \quad CP_{km}, SP_{km}, C_j, W_j \geq 0 \quad \forall j \in N, \forall k \in K, \forall m \in M \quad (10)$$

$$369 \quad x_{jkm} \in \{0,1\} \quad \forall j \in N, \forall k \in K, \forall m \in M \quad (11)$$

370 Constraint (1) specifies that each job is affected to one position in a machine. Constraint (2)  
 371 consists in having only one or no job per position in a machine. Constraint (3) specifies that, if  
 372 the position  $k$  is not occupied, the position  $k+1$  is not occupied too. Constraint (4) consists in  
 373 making, for all machines, the starting time of  $k^{\text{th}}$  position greater than or equal to assigned  
 374 release date. Constraint (5) consists in making, for all machines, the starting time of  $(k+1)^{\text{th}}$   
 375 position greater than or equal to the previous position completion time. Constraint (6) specifies  
 376 that, for all machines, the completion time of  $k^{\text{th}}$  position is equal to starting time of  $k^{\text{th}}$  position  
 377 plus the assigned processing time. Constraint (7) defines, for all machines, the completion time  
 378 of job  $j$ , which is greater than or equal to completion time of assigned position, where  $bigM$   
 379 must be sufficiently large. For that reason, we defined for our model a value of  $bigM$  according  
 380 to the total completion times and maximum value of release dates to take into account jobs  
 381 availability. Thus,  $bigM = \sum_{j=1}^n p_j + \max r_j$ . Constraint (8) is a cut that allows reducing the  
 382 computing time. Constraint (9) defines the waiting time of job  $j$ , according to its completion  
 383 time, release date, and processing time, Constraint (10) consists in making all variables greater  
 384 than or equal to zero. Constraint (11) consists in having a binary decision variable.

## 385 5.2 Mathematical model after disruption

386 This second model is used after a disruption caused by the arrival of new jobs. We denoted  
387 by  $nt$  the number of the unexecuted jobs at time  $t$ , and  $n_j$  the number of new arrived jobs. Thus,  
388 the new jobs have the same entry parameters as the first ones. For each job, we have  $w_j$ ,  $p_j$ , and  
389  $r_j$  which correspond to the time when the job occurs. So, the new parameters are as follows:

390 New parameters:

391  $n' = nt + n_j$

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

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

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

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

396  $Co_j$ : original completion time of job  $j$  (obtained by the resolution of the initial problem).

397  $\alpha$ : efficiency-stability coefficient.

398  $xo_{jkm}$ : variable assigning jobs to positions in the initial problem

399 We notice that the parameters  $xo_{jkm}$ , and  $Co_j$  are not decision variables. They are memorized  
400 when the initial model is solved. we recall that we only calculate the stability criterion for the  
401 jobs existing in the previous schedule ( $j \in N$ ). The decision variables  $x_{jkm}$ ,  $CP_{km}$ ,  $SP_{km}$ ,  $C_j$ ,  $W_j$  are  
402 the same for this new model. However, the new objective function is as follows:

$$403 \quad \text{Min } \alpha \sum_{j \in N'} w_j W_j + (1 - \alpha) \sum_{j \in N} w_j (C_j - Co_j)$$

404 The constraints (1), (2), (3), (4), (5), (6), (7), (8), (9), (10) and (11) are also used for this new  
405 model.

406 In addition, when new jobs are added, the completion time of a job will be greater than its  
407 original completion time, because the original completion time have been obtained through  
408 minimization of the principal criterion “*TWWT*” only, since it is not concerned by the stability.  
409 Indeed, the job is placed in its ideal position, and it will not be moved forward after the  
410 rescheduling. Thus, in all rescheduling steps, we assumed that:

$$411 \quad C_j \geq Co_j \quad \forall j \in N \quad (12)$$

## 412 6. Numerical results

413 The proposed mixed integer linear programming model has been designed on FICO Xpress  
414 IVE and has been performed on a Core i5 2.40GHz laptop. The programed model is composed  
415 of two parts. The first contains the predictive approach, and consists in solving the initial  
416 problem normally, with the objective of minimizing the *TWWT*. The second contains the  
417 reactive approach, and consists in rescheduling at each time the jobs occur, accordingly to the  
418 online iterative methodology previously described. The system omits the already executed jobs,

419 and combines the still unexecuted jobs with the new ones to create an updated schedule. The  
 420 model implementation has been developed accordingly to the chart presented on Fig.4. This  
 421 proposed methodology has been tested on the instances described hereafter, and it allowed  
 422 giving, at each step, the optimal solution.

423 In this section, we present the numerical results obtained through the model application, on  
 424 two parallel machines. The data used in this section are as follows:

425 **Table 1 parameter values**

Parameters	Values
$w_j$	$\sim U(1,5)$
$p_j$	$\sim N(2.5,0.5) (ut)$
$T$	$24 (ut)$

426

427 The data that we have chosen can be adapted for real cases, either in a hospital or in an  
 428 industrial environment. The simulation will be over a 6-hours' time horizon. The weight values  
 429 can represent the 5-emergency levels or 5 priority levels of customers. We assumed also that  $\Delta t$   
 430  $=1$  unit of time ( $ut$ ), and  $1 ut$  is equivalent to 15 minutes. The processing times can represent  
 431 the durations of operations or a product manufacturing time, and follow a normal distribution  
 432 with mean 2.5 and standard deviation 0.5, obtaining operations of  $1 ut$  (15 minutes) to  $4 ut$  (60  
 433 minutes), with more probability to have a processing time of  $2 ut$  or  $3 ut$ . The values of the  
 434 variable  $\theta(t)$  are randomly generated with the Bernoulli distribution which gives, at each time  
 435  $t$ , the value 1 with probability  $p_\theta$ , and 0 with probability  $1-p_\theta$ . At first, we considered  $p_\theta = 0.8$   
 436 (high appearance probability), then we tested also for  $p_\theta = 0.5$  (medium appearance probability)  
 437 and  $p_\theta = 0.2$  (small appearance probability).

438 We established three main studies. Firstly, we examined the influence of efficiency-stability  
 439 coefficient  $\alpha$  on the optimal schedules. Secondly, we used the lexicographic method to  
 440 minimize the number of altered jobs. Finally, we tested the computing time of the model  
 441 resolution with different instances, to show its limitations according to the problems' size.

## 442 **6.1 Influence of efficiency-stability coefficient on the optimal schedules**

443 In this study, we vary the value of  $\alpha$  from 0.5 to 1 to discuss its impact on the optimal  
 444 schedule. We have assumed that the efficiency criterion is the principal one, while the stability  
 445 criterion is the secondary. Thus, it is not necessary to test the values of  $\alpha$  inferior to 0.5.

446 We first calculate the initial objective obtained by solving the offline problem. After each  
 447 disruption, we update the problem, then we calculate the new objective function. The number  
 448 of altered jobs is also calculated in this study, which represents the number of jobs which  
 449 changed of machine after rescheduling. For that, we included the supplementary parameters:  
 450  ${}^i A_{jm}$  to assign jobs to machines in iteration  $i$ , and  $Dif_j$  that represents the deviation of a job from  
 451 its machine.

$$452 \quad {}^i A_{jm} = \begin{cases} 1 & \text{if } \sum_{k=1}^n x_{jkm} = 1 \\ 0 & \text{Otherwise} \end{cases} \quad \forall j, \forall m, \forall i \quad (13)$$

453

454

$$Dif_j = \begin{cases} 1 & \text{if } \sum_{m=1}^{nm} |{}^i A_{jm} - {}^{i-1} A_{jm}| \neq 0 \\ 0 & \text{Otherwise} \end{cases} \quad \forall j$$

455

$$(14)$$

456

457 Therefore, the number of altered jobs is given by:  $\sum_{j \in N} Dif_j$ . We note that  $A_{jm}$  and  $Dif_j$  are  
 458 not considered as decision variables in this study, but they will be considered as such in the  
 459 lexicographic optimization sub section.

460 The study was conducted and analyzed for 25 different instances, generated accordingly to  
 461 the parameters presented in Table 1. We present, in the following example, the data and the  
 462 results obtained on a particular instance with two parallel identical machines.

463 **Example**

464 In this example, the processing times  $p_j$ , release dates  $r_j$  and the weights  $w_j$  for 5 jobs are  
 465 given in the Table 2.

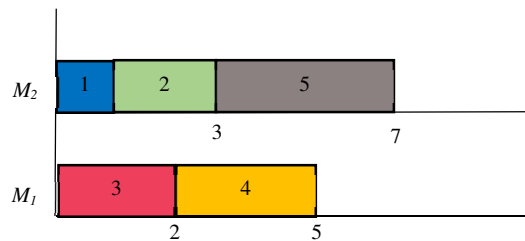
466

Table 2. Initial problem data

job	1	2	3	4	5
$r_j$	0	1	0	1	2
$p_j$	1	2	2	3	4
$w_j$	5	1	4	2	1

467

468 The problem is solved with the objective to minimize the total weighted waiting times. So,  
 469 the obtained optimal value is  $\sum_{j=1}^5 w_j W_j = 3$ . The optimal solution is given in Fig.5, we noted  
 470 it “the first schedule”.



471

472

Fig. 5 The optimal solution for the initial problem

473 In the reactive approach, we assumed that the disruptions occur by the arrival of new jobs.  
 474 In this example, a total of 20 jobs are treated online. The obtained results are given in the Table  
 475 3.

476 In the Table 3, we assumed that a disruption occurs at each step with appearance probability  
 477  $p_{\theta} = 0.8$  (20 disruptions appeared in the total). At each disruption, we reschedule the problem,  
 478 and we extract four information: the value of the efficiency criterion “ $TWWT$ ”, the value of the  
 479 stability criterion “ $TWCTD$ ”, the number of jobs which are executed on a different machine  
 480 after the disruption “ $N^{\circ}$  of altered jobs”, as well as the value of the objective function that  
 481 combines  $TWWT$  and  $TWCTD$  thanks to  $\alpha$ . We have also numbered the schedules with their

482 step number. Different values of  $\alpha$  are tested, to discuss the impact of this parameter on the  
 483 obtained solutions.

484 **Table 3. Influence of  $\alpha$  on the optimal schedules**

- 485 (1) *TWWT*: Total Weighted Waiting Times  $\sum_{j \in N'} w_j W_j$   
 486 (2) *TWCTD*: Total Weighted Completion Time Deviation  $\sum_{j \in N} w_j (C_j - Co_j)$   
 487 (3)  $N^\circ$  of altered jobs  
 488 (4) Objective function  
 489

$\alpha$	(1)	(2)	(3)	(4)	(1)	(2)	(3)	(4)	(1)	(2)	(3)	(4)	(1)	(2)	(3)	(4)
	2 <sup>nd</sup> Schedule				3 <sup>rd</sup> Schedule				4 <sup>th</sup> Schedule				5 <sup>th</sup> Schedule			
1	5	2	2	5	8	5	1	8	13	6	0	13	16	9	2	16
0,9	5	2	2	4,7	8	2	2	7,4	13	3	0	12	16	3	1	14,7
0,8	5	2	2	4,4	8	2	2	6,8	13	3	0	11	16	3	1	13,4
0,7	5	2	2	4,1	8	2	2	6,2	13	3	0	10	16	3	1	12,1
0,6	5	2	2	3,8	8	2	2	5,6	13	3	0	9	16	3	1	10,8
0,5	5	2	2	3,5	8	2	2	5	13	3	0	8	16	3	1	9,5
	6 <sup>th</sup> Schedule				7 <sup>th</sup> Schedule				8 <sup>th</sup> Schedule				9 <sup>th</sup> Schedule			
1	18	10	1	18	23	11	0	23	27	13	0	27	29	13	0	29
0.9	19	3	0	17,4	25	4	0	22,9	28	4	1	25,6	29	5	2	26,6
0.8	19	3	0	15,8	25	4	0	20,8	28	4	0	23,2	29	5	0	24,2
0.7	19	3	0	14,2	25	4	0	18,7	28	4	0	20,8	29	5	0	21,8
0.6	19	3	0	12,6	25	4	0	16,6	28	4	0	18,4	29	5	0	19,4
0.5	19	3	0	11	25	4	0	14,5	28	4	1	16	29	5	2	17
	10 <sup>th</sup> Schedule				11 <sup>th</sup> Schedule				12 <sup>th</sup> Schedule				13 <sup>th</sup> Schedule			
1	30	14	1	30	32	16	0	32	36	16	0	36	38	16	0	38
0.9	30	5	0	27,5	32	7	0	29,5	36	8	1	33,2	38	10	1	35,2
0.8	30	5	0	25	32	7	0	27	36	8	1	30,4	38	10	1	32,4
0.7	30	5	0	22,5	32	7	0	24,5	36	8	1	27,6	38	10	1	29,6
0.6	30	5	0	20	33	5	0	21,8	39	5	0	25,4	43	5	0	27,8
0.5	30	5	0	17,5	33	5	0	19	39	5	0	22	43	5	0	24
	14 <sup>th</sup> Schedule				15 <sup>th</sup> Schedule				16 <sup>th</sup> Schedule				17 <sup>th</sup> Schedule			
1	41	16	0	41	42	18	0	42	44	18	2	44	46	20	0	46
0.9	41	10	0	37,9	42	11	0	38,9	44	11	2	40,7	46	13	0	42,7
0.8	41	10	0	34,8	42	11	0	35,8	44	11	2	37,4	46	13	2	39,4
0.7	41	10	0	31,7	42	11	0	2	44	11	2	34,1	46	13	2	36,1
0.6	45	5	0	29	47	7	0	31	49	7	2	32,2	54	9	0	36
0.5	45	5	0	25	47	7	2	27	49	7	2	28	54	9	0	31,5
	18 <sup>th</sup> Schedule				19 <sup>th</sup> Schedule				20 <sup>th</sup> Schedule				21 <sup>th</sup> Schedule			
1	50	20	0	50	52	21	0	52	55	24	1	55	57	26	0	57
0.9	49	16	0	45,7	50	17	2	46,7	52	19	2	48,7	53	20	2	49,7
0.8	49	16	2	42,4	50	17	0	43,4	52	19	0	45,4	53	20	2	46,4
0.7	49	16	2	39,1	50	17	0	40,1	52	19	0	42,1	53	20	2	43,1
0.6	58	9	0	38,4	60	10	2	40	63	13	2	43	65	15	2	45
0.5	58	9	1	33,5	60	10	2	35	63	13	1	38	66	14	0	40

490

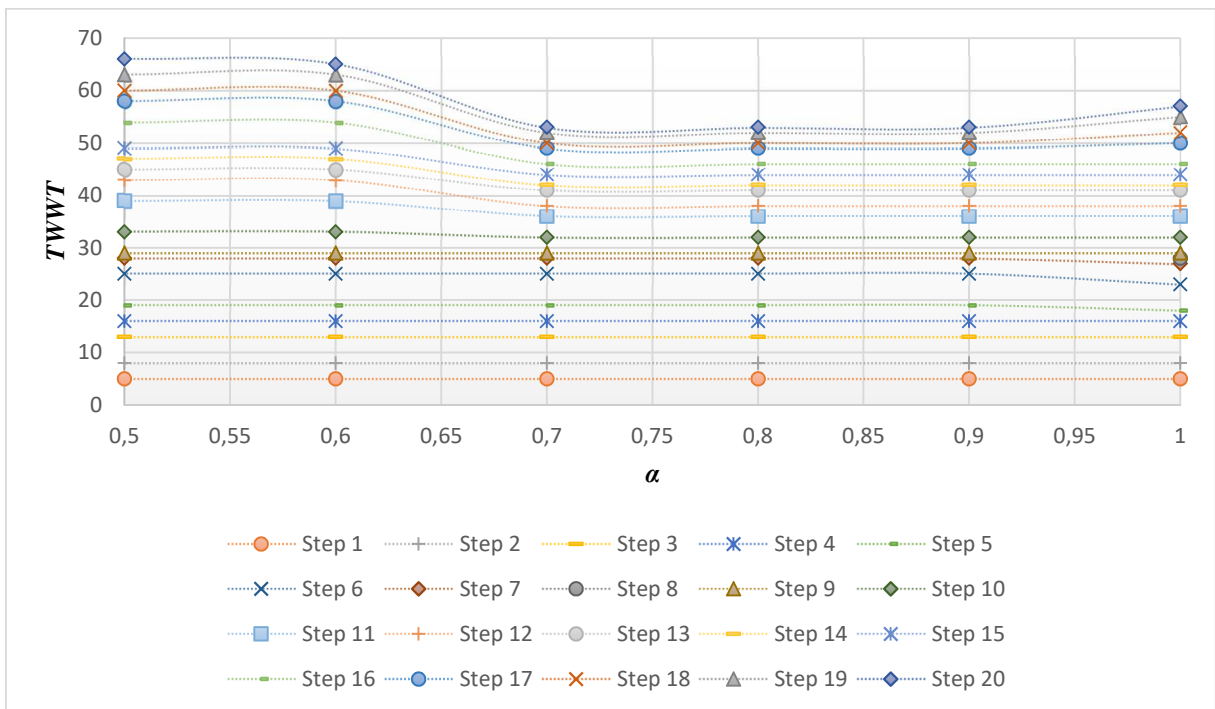
491 For  $\alpha = 1$ , the objective function only considers the first criterion, *TWWT*. We noted that,  
 492 when the new jobs arrive, they have been scheduled according to their weights and their  
 493 processing times. Generally, when  $\alpha = 1$ , and if all jobs are available, the *wSPT-rule* is optimal  
 494 for this problem.

495 Otherwise, for  $\alpha$  values between 0.9 and 0.7, the stability is also considered. In this case, we  
 496 have noted two observations:



- Short term: The optimal schedule has the same value of  $TWWT$  as when  $\alpha = 1$ , but the value of the second criterion  $TWCTD$  is smaller. Indeed, the schedule is more stable. In this case, during the model execution, if there are two solutions which give the same value of  $TWWT$ , the system chooses the one that minimizes the stability.
- Long term: The model gives a better value of both criteria ( $TWWT$  and  $TWCTD$ ) compared to  $\alpha = 1$ . This is a result of the proactive effect induced by the stability criterion. Indeed, when we have two similar solutions in terms of efficiency, the system prioritizes the one that minimizes  $\sum_{j \in N} w_j(C_j - Co_j)$ . This decision impacts the efficiency criterion in the next steps.

For  $\alpha = 0.6$  and  $0.5$ , the  $TWWT$  increases, because we give more importance to the stability criterion. Figure 6 shows, for each step, the variation of  $TWWT$  according to  $\alpha$ .



508

509

Fig. 6  $TWWT$  in function of  $\alpha$  with  $p_{\theta}=0.8$

In the short term, the value of  $TWWT$  is stable in most of cases. In the long term, the value of  $TWWT$  is smaller when  $\alpha = 0.7, 0.8$  and  $0.9$  compared to others values as result of proactive effect induced by the stability criterion. To clearly illustrate this phenomenon, we present the Gantt diagram (Table 5) of the optimal sequences given by model resolution for  $\alpha = 0.8$ , compared to those obtained with  $\alpha = 1$ , on the eleven first modified schedules (first eleven steps) for the instance below (Table 4).

516

517

518

Table 4. Problem data

Job $j$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$r_j$	0	1	0	1	2	2	3	4	6	7	9	10	11	12	13	14
$p_j$	1	2	2	3	4	3	4	1	1	4	1	2	1	3	2	4
$w_j$	5	1	3	2	1	1	1	5	3	1	5	3	4	1	1	1

519

520

521  
522  
523

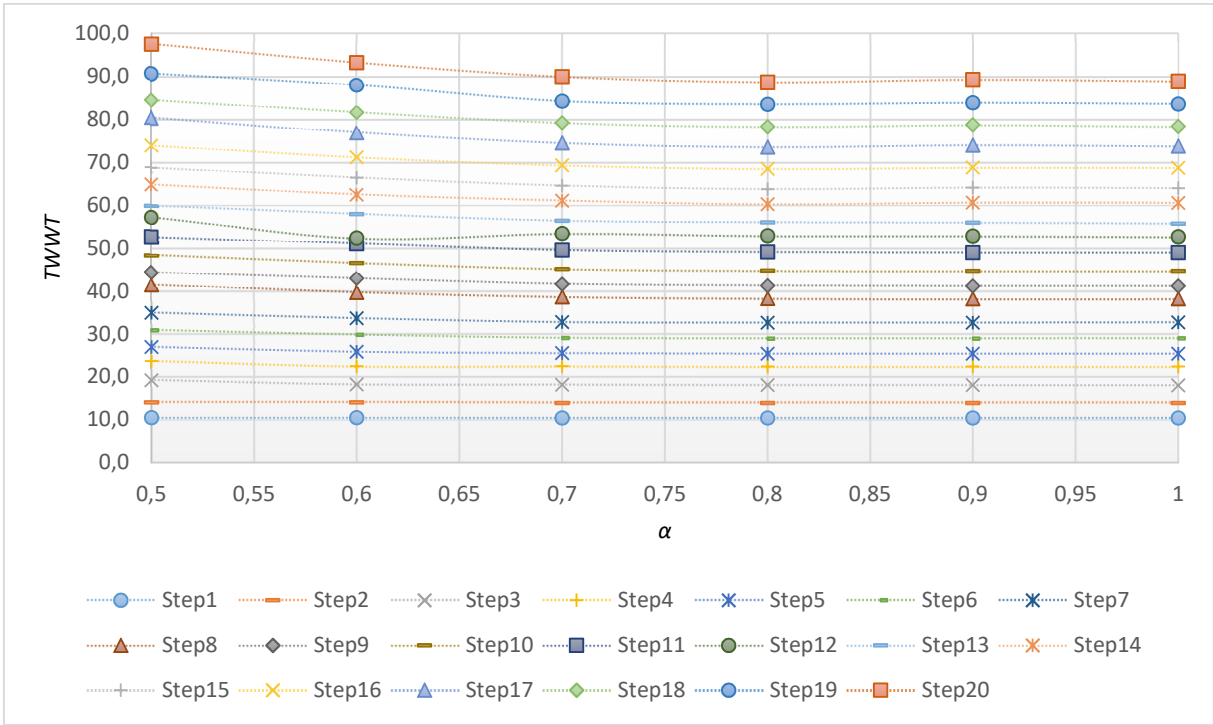
Table 5. Gantt diagram of the optimal sequences

Initial schedule	1   2   5		3   4							
Step	$\alpha = 0.8$				<i>TWWT</i>	$\alpha = 1$				<i>TWWT</i>
1	1   2   5	3   4   6			6	1   2   6	3   4   5			6
2	1   2   6	3   4   5	7		9	1   2   6	3   4   5	7		9
3	1   2   6	3   4   8	5	7	15	1   2   6	3   4   8	5	7	15
4	1   2   6	3   4   8	5	7	16	1   2   6	3   4   8	5	7	16
5	1   2   6	3   4   8	5	7	19	1   2   6	3   4   8	5	7	19
6	1   2   6	3   4   8	5	7	25	1   2   6	3   4   8	5	7	25
7	1   2   6	3   4   8	5	7	28	1   2   6	3   4   8	5	7	28
8	1   2   6	3   4   8	5	7	29	1   2   6	3   4   8	5	7	29
9	1   2   6	3   4   8	5	7	30	1   2   6	3   4   8	5	7	30
10	1   2   6	3   4   8	5	7	32	1   2   6	3   4   8	5	7	32
11	1   2   6	3   4   8	5	7	34	1   2   6	3   4   8	5	7	35

524

525 From the first step until step 10, in both cases, the model provides the same value of the  
 526 principal criterion *TWWT*. However, when  $\alpha = 0.8$ , the model provides more a stable schedule  
 527 compared to  $\alpha = 1$ . This difference has impacted the schedule in step 11. For instance, in the  
 528 step 9, with  $\alpha = 1$ , the system places the job 14 before the job 7. Whereas, if we consider the  
 529 stability criterion (*TWCTD*), the system proposes the opposite to have a stable schedule.  
 530 Consequently, this choice impacts the value of *TWWT* in the step 11. Thus, the difference of  
 531 one unit of time between the schedule when  $\alpha = 1$  and  $\alpha = 0.8$ .

532 To confirm this behavior, we present in Figure 7, the average of *TWWT* in function of *alpha*,  
 533 on the 25 instances generated.



534

535

**Fig. 7 The average of TWWT in function of  $\alpha$  with  $p_\theta=0.8$**

536

Among the 25 instances that we have tested, for 8 of them, the proactive effect appears. In Fig 7, we observe that the average of TWWT is stable in the short term. However, in the long term, it is increasing when  $\alpha$  goes down to 0.7.

537

538

539

In the case of  $p_\theta = 0.8$  (high appearance probability), and when alpha equals 1, the system does not consider the stability. So, if we have various solutions that gives the same value of TWWT, the system chooses arbitrarily one of them, as observed in the step 5 on Table 5, between the jobs 7 and 10. Consequently, this arbitrary choice impacts the next schedules. In the example of Table 5, the difference is shown in the step 11. Therefore, as result of this behaviour, we have concluded that the stability criterion can generate a proactive effect, when  $\alpha$  is between 0.7 and 0.9, providing better results compared to a mono-criterion considering only efficiency. This behaviour, that we called “proactive effect” appears when we have a various choice in some steps, i.e. when the set of jobs that have to be executed is bigger.

540

541

542

543

544

545

546

547

To confirm this conclusion, we have tested other sets of instances, with  $p_\theta = 0.5$  (medium appearance probability) and  $p_\theta = 0.2$  (small appearance probability) on the same time horizon  $T = 24 ut$ . The average of TWWT in function of  $\alpha$  is respectively presented in figures 8 and 9.

548

549

550

551

552

553

554

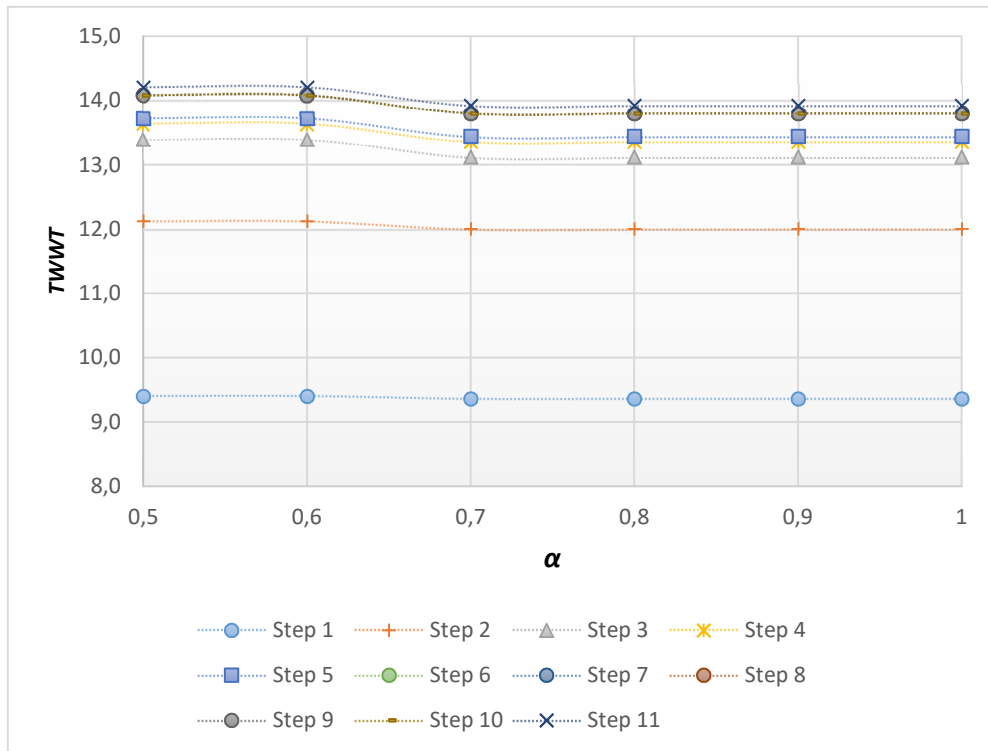


Fig. 8 The average of  $TWWT$  in function of  $\alpha$  with  $p_\theta=0.5$

555

556 In figure 8, the average of  $TWWT$  increases when  $\alpha$  becomes inferior to 0.7. Moreover, we  
 557 have observed the proactive effect in only one out of the 25 instances we have tested. In fact,  
 558 when  $p_\theta = 0.5$ , the jobs appear with a medium frequency. Then, the set of jobs that have to be  
 559 executed at each step is small compared to  $p_\theta = 0.8$ . Hence, the system does not have various  
 560 choices at each step. Thus, the proactive effect does not appear as often as for  $p_\theta = 0.8$ .

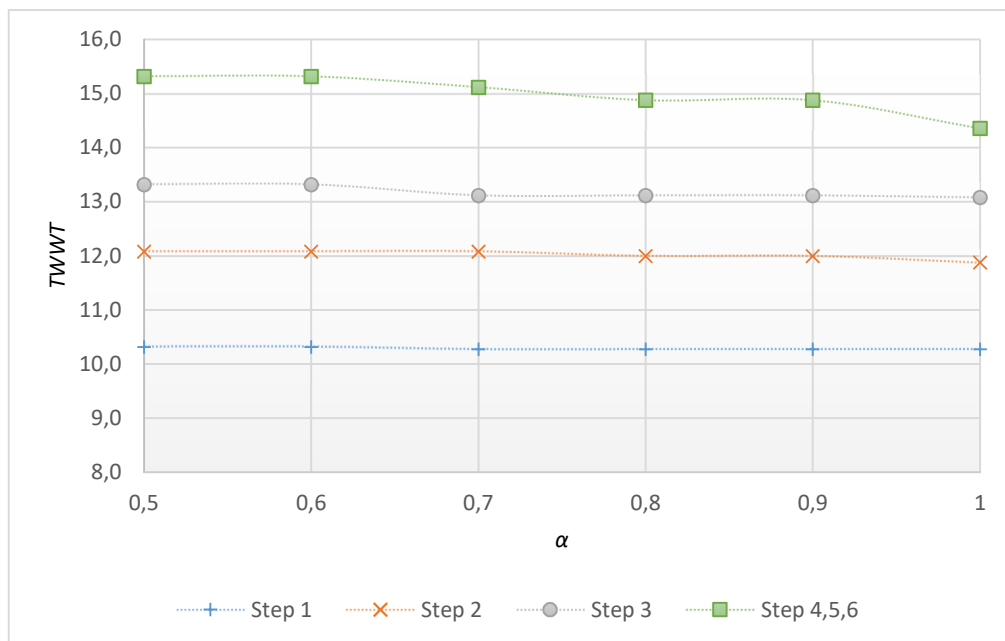
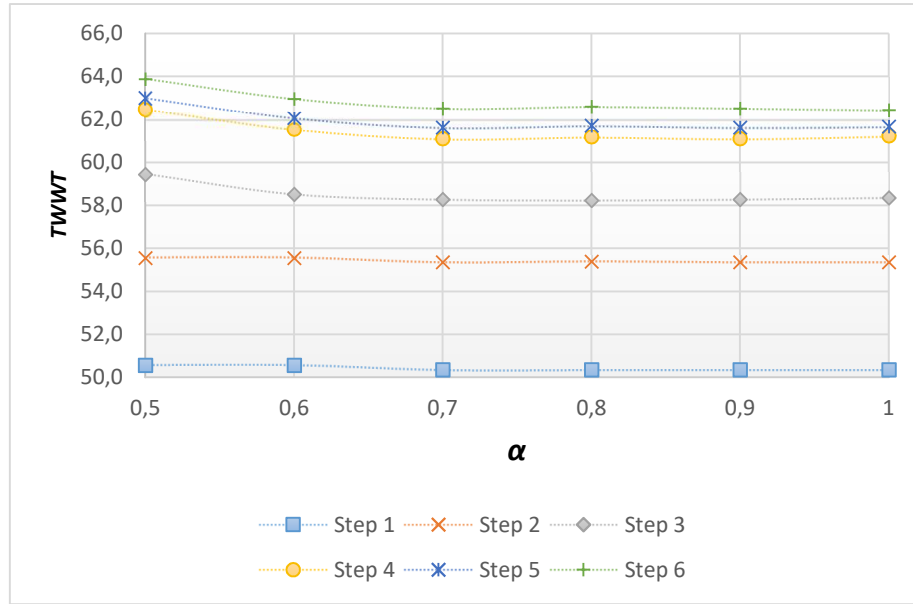


Fig. 9 The average of  $TWWT$  in function of  $\alpha$  with  $p_\theta=0.2$

561

562 In figure 9, the average of  $TWWT$  decreases with  $\alpha$ . Indeed, when  $p_\theta = 0.2$ , the jobs appear  
 563 with a small frequency. Then, the number of disruptions is very small, and we have not observed  
 564 the proactive effect. However, in the last steps (4,5,6), the average of  $TWWT$  is identical,

565 because the disruptions occur after the execution of existing jobs. Thus, the jobs do not wait,  
 566 and  $TWWT$  does not increase. Therefore, we have also tested this case by increasing the number  
 567 of initial jobs to seven jobs. The results are shown in Figure 10.



568

569

Fig. 10 The average of  $TWWT$  in function of  $\alpha$  for 7 initial jobs and  $p_{\theta}=0.2$

570 With 7 initial jobs, we observed the proactive effect in 2 instances out of the 25 tested. The  
 571 increase of the number of initial jobs allows to increase the set of jobs that have to be executed  
 572 at each time, and the probability to observe a proactive effect. This is what we have observed.

## 573 6.2 Lexicographic optimization for minimizing the number of altered jobs

574 The lexicographic optimization consists in optimizing the first objective function (of higher  
 575 priority). Then, among the possible alternative optima, we optimize the second objective and  
 576 so on (Mavrotas 2009). In this part, we performed as follows: we first optimize the multi  
 577 objective function that simultaneously measures efficiency and stability criteria,  
 578  $f_1 = \alpha \sum_{j \in N'} w_j W_j + (1-\alpha) \sum_{j \in N} w_j (C_j - Co_j)$ , obtaining the min  $f_1 = f_1^*$ . Then, we secondly

579 optimize the number of altered jobs  $f_2 = \sum_{j \in N} Dif_j$  by adding the constraint  $f_1 \leq f_1^*$ , in order to

580 keep the optimal solution of the first optimization. To do this, we put the constraints  $A_{jm}$  and  
 581  $Dif_j$  into a linear form:

582

$$583 \quad A_{jm} = \sum_{k=1}^{n'} x_{jkm} \quad \forall j \quad \forall m \quad (15)$$

$$584 \quad Dif_j \geq {}^1 A_{jm} - {}^0 A_{jm} \quad \forall j \quad \forall m \quad (16)$$

585 Constraint (15) specifies that each job is affected to machine  $m$ . Constraint (16) consists in  
 586 having the binary variables  $Dif_j$  positive when the job changes of machine during an iteration,  
 587 and equal to zero otherwise.

588 In the solution presented in Table 3, the number of altered jobs is not minimized. Therefore,  
589 at each iteration, the model gives a different value of the number of altered jobs. In the following  
590 table, we present the lexicographic optimization of the same example. The aim is to minimize  
591  $f_2$  with  $f_1 \leq f_1^*$  as a constraint. However, if the model cannot reduce the number of altered jobs  
592 to zero, we relaxed the constraint in order to obtain a better solution for  $f_2$ . Therefore, the  
593 constraint becomes  $f_1 \leq f_1^*(1+\varepsilon)$ , with  $\varepsilon$  a relaxation coefficient. In this study, we vary the value  
594 of this coefficient until we reduce  $f_2$  to zero. For that purpose, we fixed the value of  $\alpha$  at 0.8, the  
595 value for which we observed the most the proactive effect in the last sub section. The results  
596 we have obtained are given in the Table 6.

597 **Table 6. Optimal solution of number of altered jobs, with  $f_1 \leq f_1^*(1+\varepsilon)$ .**

- 598 (1) *TWWT*: Total Weighted Waiting Times  $\sum_{j \in N} w_j W_j$   
599 (2) *TWCTD*: Total Weighted Completion Time Deviation  $\sum_{j \in N} w_j (C_j - Co_j)$   
600 (3)  $f_1^*(1+\varepsilon)$   
601 (4)  $f_2$ :  $\sum_{j \in N} Dif_j$   
602

$\varepsilon$	(1)	(2)	(3)	(4)	(1)	(2)	(3)	(4)	(1)	(2)	(3)	(4)	(1)	(2)	(3)	(4)
	2 <sup>nd</sup> Schedule				3 <sup>rd</sup> Schedule				4 <sup>th</sup> Schedule				5 <sup>th</sup> Schedule			
0	5	2	4,4	0	8	2	6,8	0	13	3	11	0	16	3	13,4	0
	6 <sup>th</sup> Schedule				7 <sup>th</sup> Schedule				8 <sup>th</sup> Schedule				9 <sup>th</sup> Schedule			
0	19	3	15,8	0	25	4	20,8	0	28	4	23,2	0	29	5	24,2	0
	10 <sup>th</sup> Schedule				11 <sup>th</sup> Schedule				12 <sup>th</sup> Schedule				13 <sup>th</sup> Schedule			
0	30	5	25	0	32	7	27	0	36	8	30,4	1	42	7	35	0
0.01	-	-	-	-	-	-	-	-	36	8	30,7	1	-	-	-	-
0.02	-	-	-	-	-	-	-	-	36	8	31,0	1	-	-	-	-
0.03	-	-	-	-	-	-	-	-	36	8	31,3	1	-	-	-	-
0.04	-	-	-	-	-	-	-	-	36	8	31,6	1	-	-	-	-
0.05	-	-	-	-	-	-	-	-	38	7	31,9	0	-	-	-	-
	14 <sup>th</sup> Schedule				15 <sup>th</sup> Schedule				16 <sup>th</sup> Schedule				17 <sup>th</sup> Schedule			
0	44	7	0	0	46	9	38,6	0	48	9	40,2	0	53	11	44,6	0
	18 <sup>th</sup> Schedule				19 <sup>th</sup> Schedule				20 <sup>th</sup> Schedule				21 <sup>th</sup> Schedule			
0	57	11	47,8	0	59	12	49,4	0	62	15	52,6	1	64	17	54,6	0
0.01	-	-	-	-	-	-	-	-	62	15	53,1	1	-	-	-	-
0.02	-	-	-	-	-	-	-	-	63	16	53,6	0	-	-	-	-

603 As can be seen in Table 6, from the 2<sup>nd</sup> schedule until the 11<sup>th</sup>, the model reduced the value  
604 of  $f_2$  to zero, with  $\varepsilon = 0$ . For the 12<sup>th</sup> schedule, we have increased the value of  $\varepsilon$  up to 0.05 to  
605 obtain  $f_2 = 0$ , and  $f_1^*$  becomes  $f_1^*(1+\varepsilon)$ . For the 13<sup>th</sup> schedule, we proceed in the same way,  
606 considering the last solution obtained in the previous schedule.

607 Among the 20 schedules, we varied  $\varepsilon$  in only two of them. It means that, in the majority of  
608 cases, we can minimize  $f_2$  without impacting  $f_1^*$ . However, when we reduce  $f_2$  to zero by varying  
609  $\varepsilon$ , the value of  $f_1^*(1+\varepsilon)$  increases as well as the values of *TWWT* and *TWCTD*. This causes an  
610 increase in  $f_1^*$  in the following iterations, as in the 13<sup>th</sup> schedule for example, in which  $f_1^* = 35$   
611 compared to its value in Table 3, where  $f_1^* = 32.4$ .

### 612 6.3 Computing time study

613 In this study, we tested on 10 different instances the Maximum Duration of Iteration (*MDI*)  
614 and the Computing Time (*CT*) for the execution of all iterations. The sets of tested instances  
615 contain respectively 5 and 7 initial jobs. They are disrupted by the arrival of new jobs over a  
616 horizon of  $T = 24$  ut and  $\alpha = 0.8$ . We varied the appearance probability  $p_\theta$  with different values  
617 (0.2; 0.5; 0.8; 1) to analyze its impact on the *MDI* and *CT*. Then, we calculated the minimum,

618 maximum, average and standard deviation of *MDI* and *CT*. The obtained results are respectively  
 619 presented in Table 7 and Table 8.

620 **Table 7 Maximum Duration of Iteration**

Initial jobs	$p_\theta$	Min <i>MDI</i> (s)	Max <i>MDI</i> (s)	Avg <i>MDI</i> (s)	Std dev <i>MDI</i> (s)
5	0.2	0.12	0.26	0.17	0.05
	0.5	0.13	0.38	0.21	0.07
	0.8	0.21	1.14	0.44	0.31
	1	0.24	112	19.25	38.4
7	0.2	1.37	2.88	1.97	0.51
	0.5	1.58	3.66	2.03	0.69
	0.8	4.37	542	64.1	167
	1	403	6324	3606	2069

621

622 When  $p_\theta$  increases, the set of jobs that have to be executed at each iteration increases too  
 623 much, as well as the average of *MDI*. As well, the average *MDI* increases also when the set of  
 624 initial jobs is bigger.

625 In the rescheduling system, we have to establish a planning after each disruption. This one  
 626 should be made as soon as possible, and at latest before the appearance of another disruption.  
 627 In our study, according to the time discretization, the disruption can be possible at each  $\Delta t$ .  
 628 Thus, if the *MDI* exceeds  $\Delta t$ , it is unacceptable. In our case, based on the assumption that  $\Delta t$  is  
 629 equivalent to 15 minutes, and according to Table 7, the proposed model is effective in solving  
 630 the problems of 7 initial jobs subject to disruptions which appear at each period by a probability  
 631 with up to 0.8. It is equivalent to 27 jobs in the total on average.

632 **Table 8. Computing time**

Initial jobs	$p_\theta$	Min <i>CT</i> (s)	Max <i>CT</i> (s)	Avg <i>CT</i> (s)	Std dev <i>CT</i> (s)
5	0.2	0.27	0.51	0.35	0.07
	0.5	0.42	0.69	0.51	0.07
	0.8	1.01	2.31	1.57	0.4
	1	2.39	460	65	142
7	0.2	3.98	6.12	4.96	0.68
	0.5	5.71	9.86	6.52	1.2
	0.8	16.48	8012	829	2523
	1	1231	20945	10622	6933

633

634 We can remark that the computing time depends on the set of initial jobs and the appearance  
 635 frequency of jobs. If these ones are bigger, the average and the standard deviation of *CT*  
 636 increase, which makes difficult to estimate the computing time for solving the proposed model.

637 As mentioned previously, the set of jobs that have to be rescheduled at each step contains  
 638 the unexecuted jobs combined with the new ones. Thus, when a job with a long processing time  
 639 occupies the machine, if disruptions still appears during this period, the set of jobs that have to  
 640 be rescheduled increases, because it will contain a lot of unexecuted jobs that are waiting since  
 641 the machine is busy. This matter appears when the disruptions coincide with a long processing

642 time's job. This explains the big values of the average and standard deviation of  $CT$  when  $p_\theta =$   
643  $0.8$  and  $p_\theta = 1$ , observed in Table 8.

## 644 **7. Conclusion and perspectives**

645 In this paper, we have proposed a new multi criterion objective function for a parallel  
646 machine rescheduling problem. The proposed measure combines simultaneously efficiency  
647 represented by total weighted waiting times, and stability as the weighted completion time  
648 deviation. This multi criterion has never been studied in the literature before, and it can be a  
649 very helpful and significant criterion in industrial and hospital environments. We also measured  
650 the number of jobs processed on different machines after disruption. To tackle this problem, we  
651 adopted a predictive-reactive strategy which consists in updating at each time the schedules, in  
652 response to a disruption caused by the arrival of new jobs. Consequently, we developed a mixed  
653 integer linear programming model, as well as an iterative methodology to solve the rescheduling  
654 problem. The resolution of the problem provides, for each step, an optimal solution. The  
655 numerical results we have presented in this paper allow us to observe that:

- 656 – The stability criterion can generate a proactive effect, when  $\alpha$  is between  $0.7$  and  $0.9$ ,  
657 providing better results compared to a mono-criterion considering only efficiency.
- 658 – Using the lexicographic optimization, in the majority of cases, we can minimize the  
659 number of jobs which are processed on different machines after disruption without  
660 impacting the principal objective function.
- 661 – The computing time and maximum duration of iteration depend on the set of initial  
662 jobs and on the appearance frequency of the new arrival jobs. When these ones are  
663 bigger,  $CT$  and  $MDI$  increase.

664 These research works can help decision makers to provide, at each step, an optimal schedule  
665 in response to a disruption caused by the arrival of new jobs in a parallel machines scheduling  
666 problem. Although the proposed model provides optimal solutions, it is limited in terms of  
667 number of jobs. Therefore, in our future works, we intend to propose heuristics that allow to  
668 browse more jobs in a short time. We also intend to take interest in flowshop rescheduling  
669 problem.

670



## References

- 671  
672
- 673 Abdel-Jabbar, M. A. H., Kacem, I., & Martin, S. (2014). Unrelated parallel machines with precedence constraints:  
674 application to cloud computing. 3rd International Conference on Cloud Networking (pp. 438-442). IEEE.
- 675 Abedini, A., Ye, H., & Li, W. (2016). Operating room planning under surgery type and priority constraints.  
676 *Procedia Manufacturing*, 5, 15-25.
- 677 Addis, B., Carello, G., Grosso, A., & Tànfani, E. (2016). Operating room scheduling and rescheduling: a rolling  
678 horizon approach. *Flexible Services and Manufacturing Journal*, 28(1-2), 206-232.
- 679 Ahmadi, E., Zandieh, M., Farrokh, M., & Emami, S. M. (2016). A multi objective optimization approach for  
680 flexible job shop scheduling problem under random machine breakdown by evolutionary algorithms.  
681 *Computers & Operations Research*, 73, 56-66.
- 682 Akkan, C. (2015). Improving schedule stability in single-machine rescheduling for new operation insertion.  
683 *Computers & Operations Research*, 64, 198-209.
- 684 Alagöz, O., & Azizoglu, M. (2003). Rescheduling of identical parallel machines under machine eligibility  
685 constraints. *European Journal of Operational Research*, 149(3), 523-532.
- 686 Arnaout, J. P. (2010). Rescheduling of Parallel Machines with Setup Times using Simulation. In IIE Annual  
687 Conference. Proceedings (p. 1). Institute of Industrial and Systems Engineers (IISE).
- 688 Azizoglu, M., & Alagöz, O. (2005). Parallel-machine rescheduling with machine disruptions. *IIE Transactions*,  
689 37(12), 1113-1118.
- 690 Baker, K. R., & Keller, B. (2010). Solving the single-machine sequencing problem using integer programming.  
691 *Computers & Industrial Engineering*, 59(4), 730-735.
- 692 Beezão, A. C., Cordeau, J. F., Laporte, G., & Yanasse, H. H. (2017). Scheduling identical parallel machines with  
693 tooling constraints. *European Journal of Operational Research*, 257(3), 834-844.
- 694 Bektur, G., & Saraç, T. (2019). A mathematical model and heuristic algorithms for an unrelated parallel machine  
695 scheduling problem with sequence-dependent setup times, machine eligibility restrictions and a common  
696 server. *Computers & Operations Research*, 103, 46-63.
- 697 Bierwirth, C., & Mattfeld, D. C. (1999). Production scheduling and rescheduling with genetic algorithms.  
698 *Evolutionary computation*, 7(1), 1-17.
- 699 Cardoen, B., Demeulemeester, E., Beliën, J. (2010). Operating room planning and scheduling: A literature review.  
700 *European journal of operational research*, 201(3), 921-932.
- 701 Chang, K.-H. (2015). Multiobjective Optimization and Advanced Topics. *Design Theory and Methods Using*  
702 *CAD/CAE*, 325–406. doi:10.1016/b978-0-12-398512-5.00005-0
- 703 Church, L. K., & Uzsoy, R. (1992). Analysis of periodic and event-driven rescheduling policies in dynamic shops.  
704 *International Journal of Computer Integrated Manufacturing*, 5(3), 153-163.
- 705 Cui, W., Lu, Z., Li, C., & Han, X. (2018). A proactive approach to solve integrated production scheduling and  
706 maintenance planning problem in flow shops. *Computers & Industrial Engineering*, 115, 342-353.
- 707 Cunha, V., Santos, I., Pessoa, L., & Hamacher, S. (2020). An ILS heuristic for the ship scheduling problem:  
708 application in the oil industry. *International Transactions in Operational Research*, 27(1), 197-218.
- 709 Curry, J., & Peters\*, B. (2005). Rescheduling parallel machines with stepwise increasing tardiness and machine  
710 assignment stability objectives. *International Journal of Production Research*, 43(15), 3231-3246.
- 711 Da Silva, N. C. O., Scarpin, C. T., Pécora Jr, J. E., & Ruiz, A. (2019). Online single machine scheduling with setup  
712 times depending on the jobs sequence. *Computers & Industrial Engineering*, 129, 251-258.

- 713 Duenas, A., & Petrovic, D. (2008). An approach to predictive-reactive scheduling of parallel machines subject to  
714 disruptions. *Annals of Operations Research*, 159(1), 65-82.
- 715 El-Bouri, A. (2012). A cooperative dispatching approach for minimizing mean tardiness in a dynamic flowshop.  
716 *Computers & Operations Research*, 39(7), 1305-1314.
- 717 Eren, T. (2009). Minimizing the total weighted completion time on a single machine scheduling with release dates  
718 and a learning effect. *Applied Mathematics and Computation*, 208(2), 355-358.
- 719 Farrokhi, V., Kacem, I., & Pokoradi, L. (2014). Ranking the solution techniques for reactive scheduling problem  
720 in operating room. In 2014 International Conference on Control, Decision and Information Technologies  
721 (CoDIT) (pp. 001-006). IEEE.
- 722 Fei, H., Meskens, N., & Chu, C. (2010). A planning and scheduling problem for an operating theatre using an open  
723 scheduling strategy. *Computers & Industrial Engineering*, 58(2), 221-230.
- 724 Ferrer, S., Nicolò, G., Salido, M. A., Giret, A., & Barber, F. (2018, August). Dynamic rescheduling in energy-  
725 aware unrelated parallel machine problems. In IFIP International Conference on Advances in Production  
726 Management Systems (pp. 232-240). Springer, Cham.
- 727 Gao, K., Yang, F., Zhou, M., Pan, Q., & Suganthan, P. N. (2018). Flexible job-shop rescheduling for new job  
728 insertion by using discrete Jaya algorithm. *IEEE transactions on cybernetics*, (99), 1-12.
- 729 Guo, Y., Wang, Q., & Huang, M. (2013). Rescheduling with release time to minimize sum of waiting time  
730 considering waiting constraint of original loads. *Acta Automatica Sinica*, 39(12), 2100-2110.
- 731 Guo, Y., Huang, M., Wang, Q., & Leon, V. J. (2016). Single-machine rework rescheduling to minimize maximum  
732 waiting-times with fixed sequence of jobs and ready times. *Computers & Industrial Engineering*, 91, 262-273.
- 733 Guo, Yandong & XIE, Xiaodong. (2017). Two mixed integer programming formulations on single machine to  
734 reschedule repaired jobs for minimizing the total waiting-time. Chinese Automation Congress (CAC). IEEE.  
735 pp. 2129-2133.
- 736 Gürel, S., Körpeoğlu, E., & Aktürk, M. S. (2010). An anticipative scheduling approach with controllable  
737 processing times. *Computers & Operations Research*, 37(6), 1002-1013.
- 738 Hall, N. G., & Potts, C. N. (2004). Rescheduling for new orders. *Operations Research*, 52(3), 440-453.
- 739 Hall, N. G., & Potts, C. N. (2010). Rescheduling for job unavailability. *Operations Research*, 58(3), 746-755.
- 740 Hall, N. G., Liu, Z., & Potts, C. N. (2007). Rescheduling for multiple new orders. *INFORMS Journal on*  
741 *Computing*, 19(4), 633-645.
- 742 Hamzadayi, A., & Yildiz, G. (2016). Event driven strategy based complete rescheduling approaches for dynamic  
743 m identical parallel machines scheduling problem with a common server. *Computers & Industrial Engineering*,  
744 91, 66-84.
- 745 Herrmann, J. W. (2006). Rescheduling strategies, policies, and methods. *Handbook of production scheduling*, pp  
746 135-148. Springer, Boston, MA.
- 747 Hoogeveen, H., Lenté, C., & T'kindt, V. (2012). Rescheduling for new orders on a single machine with setup  
748 times. *European Journal of Operational Research*, 223(1), 40-46.
- 749 Ivanov, D., Dolgui, A., Sokolov, B., Werner, F., & Ivanova, M. (2016). A dynamic model and an algorithm for  
750 short-term supply chain scheduling in the smart factory industry 4.0. *International Journal of Production*  
751 *Research*, 54(2), 386-402.
- 752 Jurčičin, R., & Šebo, J. (2015). Basic production scheduling concept software application in a deterministic  
753 mechanical production environment. *Acta Simulatio*, 1(4), 1-4.
- 754 Kan, A. R. (1976). Problem formulation. In *Machine Scheduling Problems*. Springer, Boston, MA, 5-29, ISBN:  
755 978-90-247-1848-1.

- 756 Kaplan, S., & Rabadi, G. (2015). Minimising the total weighted tardiness and instability for the parallel machine  
757 re-scheduling problem with deadlines and ready times. *International Journal of Planning and Scheduling*, 2(2),  
758 87-109.
- 759 Katragjini, K., Vallada, E., & Ruiz, R. (2013). Flow shop rescheduling under different types of disruption.  
760 *International Journal of Production Research*, 51(3), 780-797.
- 761 Kooli, A., & Serairi, M. (2014). A mixed integer programming approach for the single machine problem with  
762 unequal release dates. *Computers & operations research*, 51, 323-330.
- 763 Kovalyov, M. Y., Kress, D., Meiswinkel, S., & Pesch, E. (2019). A parallel machine schedule updating game with  
764 compensations and clients averse to uncertain loss. *Computers & Operations Research*, 103, 148-157.
- 765 Kunnathur, A. S., Sundararaghavan, P. S., & Sampath, S. (2004). Dynamic rescheduling using a simulation-based  
766 expert system. *Journal of manufacturing technology management*, 15(2), 199-212.
- 767 Larsen, R., & Pranzo, M. (2019). A framework for dynamic rescheduling problems. *International Journal of*  
768 *Production Research*, 57(1), 16-33.
- 769 Li, J. Q., Pan, Q. K., & Mao, K. (2015). A discrete teaching-learning-based optimisation algorithm for realistic  
770 flowshop rescheduling problems. *Engineering Applications of Artificial Intelligence*, 37, 279-292.
- 771 Li, R. K., Shyu, Y. T., & Adiga, S. (1993). A heuristic rescheduling algorithm for computer-based production  
772 scheduling systems. *The International Journal of Production Research*, 31(8), 1815-1826.
- 773 Li, X., Peng, Z., Du, B., Guo, J., Xu, W., & Zhuang, K. (2017). Hybrid artificial bee colony algorithm with a  
774 rescheduling strategy for solving flexible job shop scheduling problems. *Computers & Industrial Engineering*,  
775 113, 10-26.
- 776 Li, Z., & Ierapetritou, M. (2008). Process scheduling under uncertainty: Review and challenges. *Computers &*  
777 *Chemical Engineering*, 32(4-5), 715-727.
- 778 Liu, L., & Zhou, H. (2013a). On the identical parallel-machine rescheduling with job rework disruption. *Computers*  
779 *& Industrial Engineering*, 66(1), 186-198.
- 780 Liu, L., & Zhou, H. (2013b). Open shop rescheduling under singular machine disruption. *Computer Integrated*  
781 *Manufacturing Systems*, (10), 12.
- 782 Liu, Z., & Ro, Y. K. (2014). Rescheduling for machine disruption to minimize makespan and maximum lateness.  
783 *Journal of Scheduling*, 17(4), 339-352.
- 784 Loukil, T., Teghem, J., & Tuyttens, D. (2005). Solving multi-objective production scheduling problems using  
785 metaheuristics. *European journal of operational research*, 161(1), 42-61.
- 786 Lu, X., Sitters, R. A., & Stougie, L. (2003). A class of on-line scheduling algorithms to minimize total completion  
787 time. *Operations Research Letters*, 31(3), 232-236.
- 788 Luo, W., Luo, T., Goebel, R., & Lin, G. (2018). Rescheduling due to machine disruption to minimize the total  
789 weighted completion time. *Journal of Scheduling*, 21(5), 565-578.
- 790 M'Hallah, R., & Al-Roomi, A. H. (2014). The planning and scheduling of operating rooms: A simulation approach.  
791 *Computers & Industrial Engineering*, 78, 235-248.
- 792 Macario, A., Vitez, T., Dunn, B., McDonald, T. (1995). Where are the costs in perioperative care?: Analysis of  
793 hospital costs and charges for inpatient surgical care. *Anesthesiology: The Journal of the American Society of*  
794 *Anesthesiologists*, 83(6), 1138-1144.
- 795 Mason, S. J., Jin, S., & Wessels, C. M. (2004). Rescheduling strategies for minimizing total weighted tardiness in  
796 complex job shops. *International Journal of Production Research*, 42(3), 613-628.
- 797 Mavrotas, G. (2009). Effective implementation of the  $\epsilon$ -constraint method in multi-objective mathematical  
798 programming problems. *Applied mathematics and computation*, 213(2), 455-465.

- 799 Nurre, S. G., & Sharkey, T. C. (2018). Online scheduling problems with flexible release dates: Applications to  
800 infrastructure restoration. *Computers & Operations Research*, 92, 1-16.
- 801 Ouelhadj, D., & Petrovic, S. (2009). A survey of dynamic scheduling in manufacturing systems. *Journal of*  
802 *scheduling*, 12(4), 417.
- 803 Ozlen, M., & Azizoğlu, M. (2011). Rescheduling unrelated parallel machines with total flow time and total  
804 disruption cost criteria. *Journal of the Operational Research Society*, 62(1), 152-164.
- 805 Peng, K., Pan, Q. K., Gao, L., Zhang, B., & Pang, X. (2018). An Improved Artificial Bee Colony algorithm for  
806 real-world hybrid flowshop rescheduling in Steelmaking-refining-Continuous Casting process. *Computers &*  
807 *Industrial Engineering*, 122, 235-250.
- 808 Peng, K., Pan, Q. K., Gao, L., Li, X., Das, S., & Zhang, B. (2019). A multi-start variable neighbourhood descent  
809 algorithm for hybrid flowshop rescheduling. *Swarm and Evolutionary Computation*, 45, 92-112.
- 810 Pfeiffer, A., Kádár, B., & Monostori, L. (2007). Stability-oriented evaluation of rescheduling strategies, by using  
811 simulation. *Computers in Industry*, 58(7), 630-643.
- 812 Qi, X., Bard, J. F., & Yu, G. (2006). Disruption management for machine scheduling: the case of SPT schedules.  
813 *International Journal of Production Economics*, 103(1), 166-184.
- 814 Rahmani, D., & Ramezani, R. (2016). A stable reactive approach in dynamic flexible flow shop scheduling with  
815 unexpected disruptions: A case study. *Computers & Industrial Engineering*, 98, 360-372.
- 816 Rangsaritratamee, R., Ferrell Jr, W. G., & Kurz, M. B. (2004). Dynamic rescheduling that simultaneously  
817 considers efficiency and stability. *Computers & Industrial Engineering*, 46(1), 1-15.
- 818 Riahi, V., Newton, M. H., Polash, M. M. A., & Sattar, A. (2019). Tailoring customer order scheduling search  
819 algorithms. *Computers & Operations Research*, 108, 155-165.
- 820 Sabuncuoglu, I., & Karabuk, S. (1999). Rescheduling frequency in an FMS with uncertain processing times and  
821 unreliable machines. *Journal of Manufacturing Systems*, 18(4), 268-283.
- 822 Salido, M. A., Escamilla, J., Barber, F., & Giret, A. (2017). Rescheduling in job-shop problems for sustainable  
823 manufacturing systems. *Journal of cleaner production*, 162, S121-S132.
- 824 ShangGuan, C. X., Li, J. T., & Shi, R. F. (2012). Rescheduling of parallel machines under machine failures. In  
825 *Advanced Materials Research (Vol. 443, pp. 724-730)*. Trans Tech Publications Ltd.
- 826 Stuart, K., & Kozan, E. (2012). Reactive scheduling model for the operating theatre. *Flexible Services and*  
827 *Manufacturing Journal*, 24(4), 400-421.
- 828 Tang, D., Dai, M., Salido, M. A., & Giret, A. (2016). Energy-efficient dynamic scheduling for a flexible flow shop  
829 using an improved particle swarm optimization. *Computers in Industry*, 81, 82-95.
- 830 Uhlmann, I. R., & Frazzon, E. M. (2018). Production rescheduling review: Opportunities for industrial integration  
831 and practical applications. *Journal of manufacturing systems*, 49, 186-193.
- 832 Unal, A. T., Uzsoy, R., & Kiran, A. S. (1997). Rescheduling on a single machine with part-type dependent setup  
833 times and deadlines. *Annals of Operations Research*, 70, 93-113.
- 834 Vieira, G. E., Herrmann, J. W., & Lin, E. (2003). Rescheduling manufacturing systems: a framework of strategies,  
835 policies, and methods. *Journal of scheduling*, 6(1), 39-62
- 836 Vieira, G. E., Herrmann, J. W., & Lin, E. (2000). Predicting the performance of rescheduling strategies for parallel  
837 machine systems. *Journal of manufacturing Systems*, 19(4), 256-266.
- 838 Wang, B., Han, X., Zhang, X., & Zhang, S. (2015). Predictive-reactive scheduling for single surgical suite subject  
839 to random emergency surgery. *Journal of Combinatorial Optimization*, 30(4), 949-966.

- 840 Wang, D. J., Liu, F., Wang, Y. Z., & Jin, Y. (2015). A knowledge-based evolutionary proactive scheduling  
841 approach in the presence of machine breakdown and deterioration effect. *Knowledge-Based Systems*, 90, 70-  
842 80.
- 843 Wang, D. J., Liu, F., & Jin, Y. (2017). A multi-objective evolutionary algorithm guided by directed search for  
844 dynamic scheduling. *Computers & Operations Research*, 79, 279-290.
- 845 Wang, D., Yin, Y., & Cheng, T. C. E. (2018). Parallel-machine rescheduling with job unavailability and rejection.  
846 *Omega*, 81, 246-260.
- 847 Wullink, G., Van Houdenhoven, M., Hans, E. W., van Oostrum, J. M., van der Lans, M., & Kazemier, G. (2007).  
848 Closing emergency operating rooms improves efficiency. *Journal of Medical Systems*, 31(6), 543-546.
- 849 Wu, S. D., Storer, R. H., & Pei-Chann, C. (1993). One-machine rescheduling heuristics with efficiency and  
850 stability as criteria. *Computers & Operations Research*, 20(1), 1-14.
- 851 Wu, Z., Sun, S., & Xiao, S. (2018). Risk measure of job shop scheduling with random machine breakdowns.  
852 *Computers & Operations Research*, 99, 1-12.
- 853 Yan-hai, H., Jun-qi, Y., Fei-fan, Y., & Jun-he, Y. (2005). Flow shop rescheduling problem under rush orders.  
854 *Journal of Zhejiang University-SCIENCE A*, 6(10), 1040-1046.
- 855 Yan, P., Liu, S. Q., Sun, T., & Ma, K. (2018). A dynamic scheduling approach for optimizing the material handling  
856 operations in a robotic cell. *Computers & Operations Research*, 99, 166-177.
- 857 Yang, B. (2007). Single machine rescheduling with new jobs arrivals and processing time compression. *The*  
858 *International Journal of Advanced Manufacturing Technology*, 34(3-4), 378-384.
- 859 Yang, B., & Geunes, J. (2008). Predictive–reactive scheduling on a single resource with uncertain future jobs.  
860 *European Journal of Operational Research*, 189(3), 1267-1283.
- 861 Yin, Y., Cheng, T. C. E., & Wang, D. J. (2016). Rescheduling on identical parallel machines with machine  
862 disruptions to minimize total completion time. *European Journal of Operational Research*, 252(3), 737-749.
- 863 Yuan, J., & Mu, Y. (2007). Rescheduling with release dates to minimize makespan under a limit on the maximum  
864 sequence disruption. *European Journal of Operational Research*, 182(2), 936-944.
- 865 Yuan, J., Mu, Y., Lu, L., & Li, W. (2007). Rescheduling with release dates to minimize total sequence disruption  
866 under a limit on the makespan. *Asia-Pacific Journal of Operational Research*, 24(06), 789-796.
- 867 Zakaria, Z., & Petrovic, S. (2012). Genetic algorithms for match-up rescheduling of the flexible manufacturing  
868 systems. *Computers & Industrial Engineering*, 62(2), 670-686.
- 869 Zhang, L., Gao, L., & Li, X. (2013). A hybrid genetic algorithm and tabu search for a multi-objective dynamic job  
870 shop scheduling problem. *International Journal of Production Research*, 51(12), 3516-3531.
- 871 Zhong, W., & Shi, Y. (2018). Two-stage no-wait hybrid flowshop scheduling with inter-stage flexibility. *Journal*  
872 *of Combinatorial Optimization*, 35(1), 108-125.
- 873 Zykina, A. V. (2004). A lexicographic optimization algorithm. *Automation and Remote Control*, 65(3), 363-368.
- 874