An Unrelated Parallel Machines Rescheduling Problem: An Industrial Case Study

Alice Berthier(✉), Alice Yalaoui, Hicham Chehade, Farouk Yalaoui, Lionel Amodeo, and Christian Bouillot

Computer Laboratory and Digital Society (LIST3N), UTT, Université de Technologie de Troyes, 12 rue Marie Curie, CS42060, 10004 Troyes Cedex, France
alice.berthier@utt.fr

Abstract. This study tackles an unrelated parallel machines rescheduling problem. Sequence and machine dependent setup times and limited resources are taken into consideration. The study focuses on the objective of proposing an efficient and stable rescheduling solution. The resolution approach is explained and illustrated. Different indicators to optimize the rescheduling planning are tested and results are analyzed. The problem is inspired from a concrete case of textile industry.

Keywords: Rescheduling · Unrelated parallel machines · Genetic algorithm

1 Introduction

The purpose of this paper is to propose a method to adapt the production scheduling when disruption occurs. The stake is to have a limited impact on the workshop organization and the productivity performances.

The workshop considered in this study is composed of unrelated parallel machines with machine and sequence dependent setup times. Two types of limited resources are considered: number of operators available that limits the number of parallel machines able to run at the same time and number of adjusters that limits the number of setup. Each adjuster can do only one setup at a time. The objective is to reschedule a known number of jobs after a disruption.

An initial production planning is provided by a scheduling algorithm already implemented with c_{max} minimization. However, perfect production conditions are very unrealistic, disruptions can occur and the initial planning may no longer be up to date. The different disruptions that can occur in this problem are:

- Arrival of a new job
- Deletion of a job
- Machine breakdown

Published by Springer Nature Switzerland AG 2021
A. Dolgui et al. (Eds.): APMS 2021, IFIP AICT 630, pp. 81–91, 2021.
https://doi.org/10.1007/978-3-030-85874-2_9

– Lack of human resources (operator or adjuster)

The rescheduling objective is to find the best possible planning to finish all the jobs as soon as possible by keeping stability in the planning initially provided and integrating work in progress information. This is why this study is focused on the objective of maximizing performance ($min\ c_{max}$) while maintaining stability.

This problem stems from a real case encountered in textile industry, facing the industry 4.0 revolution. The development of online-business requires more and more flexibility and reactivity, specially with the COVID 19 crisis context as e-business have greatly increased (+100% in 2020). The adaptation needed is reflected in the entire process of the clothing manufacturing industry from the knitting of the fabric to the assembling stage.

The rest of the study is organized as follows. Section 2 gives a literature review on this kind of problem. Section 3 provides contribution of this study with the exposition of resolution method. The next section gives results obtained. A conclusion end up the study in Sect. 5.

2 State of the Art

In the literature, only two works tackle problems on the same type of system studied in this paper. Work of Berthier et al. [4] deals with a dynamic layout problem in the same industrial environment. The importance of flexibility in such workshop may be encountered at a tactical level. In [5], the authors propose a complete study of the scheduling problem (MILP and AG) to deal with c_{max} minimization in such systems.

In fact, often, real-world scheduling problems are dynamic systems and they need to respond to exogenous events [6]. Different rescheduling approaches are proposed in the literature.

The first one is to use a standard scheduling method with the new data after disruption. This can rich high quality solution on the performance objective. However, solution stability is not guarantee [10]. On real life production, getting a totally different schedule is very unfavorable to a good workshop organization and management.

The second one is to use a proactive scheduling. This is generated by inserting idle time between the pre-scheduling activities, enabling the disruptions to be smoothed out through the system in order to maintain the schedule quality [1]. Stochastic approaches are an other way to do it [13].

The last one is reactive scheduling, commonly referred to as rescheduling. It is a procedure to modify the existing schedule during processing to adapt to changes in a production or operational environment. Kim [8] recently studies a rescheduling problem of unrelated parallel machines with job-dependent setup times under forecasted machine breakdown.

To briefly review some authors that tackle similar problems in literature: [16] consider an hybrid flowshop with random disturbance and develop and implement a heuristic on an expert system software. In [11] incoming workflows to be executed on a large-scale distributed system are modeled as directed graphs, where tasks may fail their computations. Heuristics for the problem have been implemented in a specific application simulator. In [14] a steel making continuous caster process is considered with uncertain tasks and a Lagrangian Decomposition method is developed to solve it. [17] consider a flexible job shop with partial and total rescheduling to deal with rush orders, job cancellations and machine breakdowns. Finally, [3] tackles a dynamic job shop with new orders, rush orders, order cancellations, due date changes and machine breakdowns. Rescheduling is event driven and it is carried out considering different criteria in lexicographic order.

On a majority of studies, contrary to scheduling problems, the complexity comes from the combination of two conflicting objectives. Rescheduling problems are taken into consideration: performance and stability measurements. Multiple indicators in the literature are proposed. The definition and use of appropriate performance metrics or quality indicators is crucial. Currently, there are many proposed metrics [7] that can be classified into unary, which assign each non-dominated set a number that reflects a certain quality aspect, and binary, which assign a number to a pair of Pareto approximations. However, each industry has its own characteristics that involves specific indicators. For example, stability can be evaluate by the number of jobs processed on different machines in the original and new schedules [2]. Other approaches defined stability in terms of deviation of job starting times between the original and revised schedules and the difference of job sequences [9,12].

3 Resolution Method

The contribution of this study is to explore multiple evaluation metrics of the rescheduling solution. First, instances are randomly generated and schedule with the genetic algorithm developed by Berthier et al. [5]. This algorithm is based on makespan minimization. The entire production has to be finished as soon as possible without any priority under the time horizon. Then, disturbances are generated. The new problem is to reschedule with two conflicting objectives: keep a good performance but also guarantee stability. The flowchart of the resolution approach is given in Fig. 1.

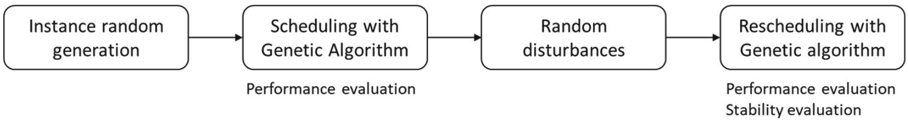

Fig. 1. Flowchart of the resolution approach

The stability in rescheduling is more complex to evaluate than c_{max} evaluation. This study focuses on the measurement of the number of machine assignment differences between the initial scheduling and the rescheduling planning for each job. To illustrate this, an example is given in the following. This example does not take human resources limitation into consideration. Table 1 gives the processing time p_{mi} of each machine $m = (1...M)$ for each job $i = (1...N)$ per unite and the quantity of each job is given in the last row of the table. Table 2 gives the setup times between jobs $i = (0...N)$ and $j = (1...N)$ on machine 1, 2 and 3. Figure 2 shows the result of the example instance scheduling. The c_{max} value reached is 29. Now, supposing that M2 becomes unavailable from time 7 to time 37. If jobs 4 and 5 are shifted after the disruption, the c_{max} value is increasing up to 54 but stability measurement is equal to 0: no job is machine changed (Fig. 3). But if machine job assignment is change as in Fig. 4, the stability indicator is degraded to 2. However, the performance measure c_{max} is improve to 42.

This stability indicator is explored in different combinations and analyzed in order to get the most pertinent and efficient rescheduling planning to the company. These indicator can be compared to a limit parameter. If the limit is crossed, the objective function is penalized. This allows a tolerance and plays up on the performance objective.

Table 1. Processing times p_{mi}

p_{mi}	1	2	3	4	5	6
M1	6	3	5	4	7	5
M2	4	4	9	3	5	4
M3	5	2	7	4	6	6
Quantity	3	5	2	4	2	3

Table 2. Setup times s_{mij}

s_{1ij}	1	2	3	4	5	6	s_{2ij}	1	2	3	4	5	6	s_{3ij}	1	2	3	4	5	6
0	1	3	2	2	4	1	0	3	4	2	1	3	3	0	1	2	3	4	5	2
1	0	1	2	4	3	1	1	0	2	3	3	4	4	1	0	3	5	1	1	3
2	1	0	4	4	1	4	2	2	0	1	1	4	2	2	4	0	3	1	1	1
3	2	4	0	3	3	3	3	4	3	0	4	3	4	3	2	1	0	2	2	1
4	3	2	4	0	1	2	4	4	4	2	0	1	1	4	1	2	3	0	2	2
5	4	4	3	3	0	4	5	1	2	4	3	0	4	5	1	1	1	2	0	1
6	4	1	1	2	1	0	6	2	3	2	3	2	0	6	1	2	2	1	3	0

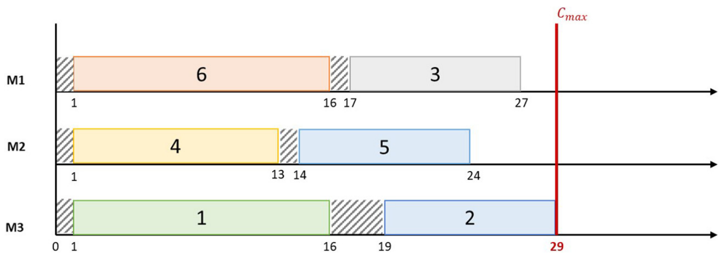

Fig. 2. Initial scheduling of the example

Fig. 3. Rescheduling example after machine unavailability with stability equal to 0

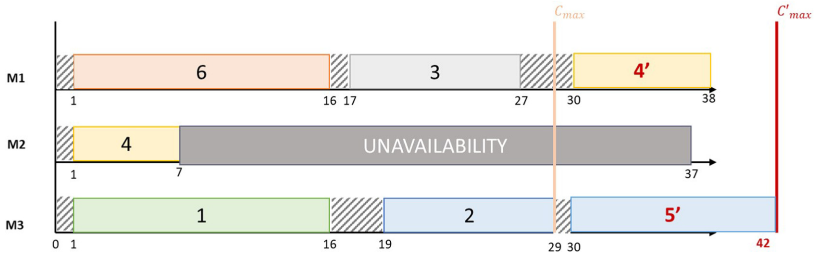

Fig. 4. Rescheduling example after machine unavailability with stability equal to 2

3.1 Instances Generator

In order to have several instances of characteristics close to the real data, an instance generator has been developed. Instances can be used to analyze the different indicators under different parameter conditions on the rescheduling tool.

The random generation is established as follows with different probability laws on each step. The generation is divided into two main parts:

- Generation of initial instances that will be scheduled using the scheduling genetic algorithm of the previous study with this industrial partner
- Generation of disturbances on this solution

Table 3 details the different step to generate an instance inspired from [15]. Each instance is generated with the probability law indicated in Table 3. The data generated are the number of jobs to schedule, machine eligibility for each job, processing times and setup times. This instance is scheduled with these initial data. After that, one disturbance for this instance is generated randomly among the four disturbance types. The rescheduling process used in this study is event-driven. Each time a disturbance occurs, rescheduling is triggered. Only one disturbance is generated for each instance. After disturbance, the new availability of each resource has to be calculated. In the example of Fig. 3, when disturbance occurs at time 7, M1 (resp. M3) is unavailable until time 16 (resp. time 16) to finish job 6 (resp. 1). Taking into consideration M1 and M3 availability and the unavailability of M2, the instance is rescheduled.

Table 3. Instances random generator

Step number	Detail	Probability law
Instances to scheduling operation		
1	Number of jobs to schedule	Uniform [50;350]
2	For each job, number of machines eligible (max 5 machines) and machines affectation	Discrete P(1) = 0.4; P(2) = 0.3; P(3) = 0.15; P(4) = 0.1; P(5) = 0.05
3	For each job on each machine eligible, the processing time	Uniform [450;2500]
4	For each job on each machine eligible, the setup time $a * \min(p_{mi}, p_{mj})$ where $a = U(A; B)$	Uniform [A;B] = [0.01;0.1]; [0.05;0.1]; [0.1;0.2]; [0.1;0.2] or [0.2;0.5]
Disturbed instances to rescheduling operation		
5	The kind of disturbance (4 types) 1. Arrival of new job; 2. Deletion of a job; 3. Machine breakdown; 4. Lack of human resources	Discrete P(1) = 1/4; P(2) = 1/4 P(3) = 1/4; P(4) = 1/4
6	Disturbance date	Uniform [0;c_{max}]
7	Calculation of resources availability to determine the minimum starting date on each resource	

3.2 Objective Functions

The resolution method is based on the same genetic algorithm (GA) used in [5]. The solution representation uses in this GA is given in Fig. 5. The representation chosen is an array table with two rows and as many columns as scheduled jobs. In the first row, each job is assigned once and the order will be the sequencing decoding order. The machine assigned for each job is indicated in the second row. To initialize the population of solution, as in numerous papers, a randomized initialization is used.

Job	6 4 3 5 1 2
Machine	1 2 3 2 2 3

Fig. 5. Solution chromosome example

Only the objective function is changed. Different objective functions have been studied in this paper. The first goal of planning rescheduling is to keep the efficiency of the solution, which corresponds to the optimization of the makespan value (c_{max}). This is still the main objective, as performance is more important than stability for the industrial partner. Thus, the first objective function studied is only to minimize the value of c_{max}. The other objective functions are composed of two elements to optimize: the c_{max} and a metric to guarantee as much as possible the stability between the initial planning and the rescheduling one. The indicator of stability chosen is the number of assignment job/machine differences.

A penalty cost is used for stability, to allow a tolerance threshold. A fixed number of disturbance over the total number of jobs can be tolerated. When changes occur right after the rescheduling date, it can disturb the organization already in place: setups are made, workers already have information about what they have to do next. However, when changes occur at the end of the horizon, the schedule can change. So it is not really disturbing. Finally, seven objective functions are compared, each one is linear combination:

1. Minimization of c_{max}
2. Minimization of c_{max} and assignment differences
3. Minimization of c_{max} and penalty if assignment differences are up to 10% of jobs
4. Minimization of c_{max} and penalty if assignment differences are up to 30% of jobs
5. Minimization of c_{max} and penalty if assignment differences are up to 50% of jobs
6. Minimization of c_{max} and penalty if assignment differences occur less than 24 h after rescheduling date
7. Minimization of c_{max} and penalty if assignment differences occur less than 48 h after rescheduling date

4 Results

60 instances with different disturbances have been generated. The average size of instances is 189 jobs to schedule initially. For each instance, all the objective functions have been applied and compared. After disturbance, 111 jobs in average has to be reschedule. The disturbance date is generated in average at 26% of the c_{max} value of the initial schedule optimization. Instances have been grouped, related to the type of disturbance. For each group, results are the average of the solution evaluation. Table 4 shows for each group of instances, the performance objective c_{max} reached when it is the single objective function. Stability value of the solution is the reference to evaluate the other objective functions (Sect. 3.2). This value is calculated as follow: for each job i, if the initial machine assignment is different than after rescheduling, $a_i = 1$, else $a_i = 0$. The value in Table 4 is: $\sum_{i=1}^{N} a_i$. For performance and stability, standard deviation are given. Depending on the disruption and instance data, performance and stability may be affected in different ways. Distribution of performance (c_{max}) are given by box plots (Fig. 6). The performance value distribution change from one group to another. Averages and minimum are still very closed but maximum and quartiles diverge.

In Table 5, the results with the application of the six other objective functions are given as deviation compared to the references. It can be observed that the efficiency objective is not very degraded compared to when it is the only

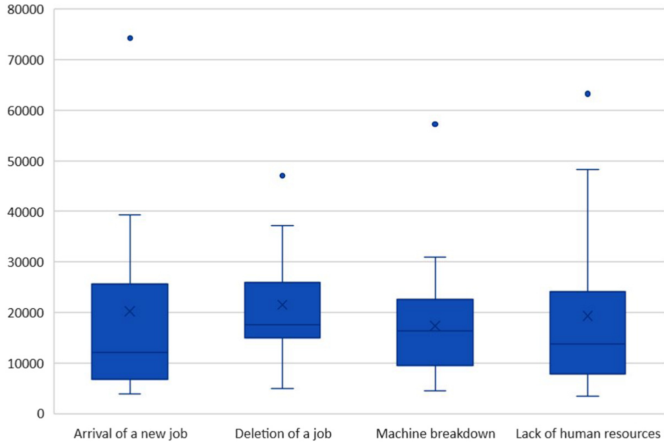

Fig. 6. Performance value distribution for each type of disturbance

one optimize. However, the stability is considerably increased by up to 97% when job/machine assignment differences are taken into consideration. As previously, for each job i, if the initial machine assignment is different than after rescheduling with the specific objective function, $b_i = 1$, else $b_i = 0$. The stability value in Table 5 is calculated by: $(\sum_{i=1}^{N} b_i - \sum_{i=1}^{N} a_i)/\sum_{i=1}^{N} a_i$. If the stability value is 0%, it means that the number of job assignment changes is the same than rescheduling optimization when stability are not considered. For instance, for group Arrival of a new job, if in Table 5, stability value reaches −100%, it means that 40 jobs has no assignment changes compared to initial scheduling planning. When the value is superior to 0%, it means that more machine assignment changes are generated than in the reference rescheduling solution. It happens when assignment changes penalized only beyond a large percentage of jobs (Table 5, Arrival of a new job, objective function 4 and 5).

Table 4. Results reference on optimization of c_{max} for the different disturbances

Objective function	Instance size	1	
Disturbance	Initial number of jobs	Performance	Stability
Arrival of a new job	183 ± 101	$23\ 129 \pm 22\ 068$	40 ± 27
Deletion of a job	204 ± 57	$21\ 492 \pm 11\ 212$	23 ± 26
Machine breakdown	181 ± 87	$21\ 607 \pm 17\ 964$	43 ± 40
Lack of human resources	188 ± 59	$18\ 966 \pm 11\ 188$	27 ± 27
Total	189 ± 72	$20\ 786 \pm 14\ 626$	32 ± 31

Table 5. Efficiency and stability results with the different objective function for the different disturbances

Objective function	2		3	
Disturbance	Performance	Stability	Performance	Stability
Arrival of a new job	$5\% \pm 9$	$-96\% \pm 6$	$8\% \pm 13$	$-67\% \pm 33$
Deletion of a job	$5\% \pm 9$	$-100\% \pm 1$	$1\% \pm 3$	$-67\% \pm 40$
Machine breakdown	$8\% \pm 14$	$-98\% \pm 4$	$0\% \pm 6$	$-59\% \pm 47$
Lack of human resources	$6\% \pm 13$	$-96\% \pm 10$	$1\% \pm 6$	$-54\% \pm 60$
Total	$6\% \pm 12$	$-97\% \pm 7$	$2\% \pm 7$	$-60\% \pm 49$
Objective function	4		5	
Disturbance	Performance	Stability	Performance	Stability
Arrival of a new job	$3\% \pm 7$	$13\% \pm 39$	$2\% \pm 5$	$13\% \pm 85$
Deletion of a job	$0\% \pm 7$	$-56\% \pm 47$	$0\% \pm 6$	$-56\% \pm 47$
Machine breakdown	$0\% \pm 6$	$-40\% \pm 49$	$0\% \pm 4$	$-56\% \pm 52$
Lack of human resources	$1\% \pm 5$	$-18\% \pm 55$	$1\% \pm 4$	$-21\% \pm 60$
Total	$1\% \pm 6$	$-27\% \pm 49$	$1\% \pm 5$	$-29\% \pm 63$
Objective function	6		7	
Disturbance	Performance	Stability	Performance	Stability
Arrival of a new job	$4\% \pm 15$	$-58\% \pm 47$	$5\% \pm 13$	$-61\% \pm 48$
Deletion of a job	$-2\% \pm 5$	$-65\% \pm 42$	$-1\% \pm 7$	$-62\% \pm 45$
Machine breakdown	$-2\% \pm 6$	$-57\% \pm 49$	$2\% \pm 6$	$-68\% \pm 40$
Lack of human resources	$1\% \pm 6$	$-70\% \pm 40$	$1\% \pm 5$	$-67\% \pm 55$
Total	$0\% \pm 8$	$-64\% \pm 43$	$1\% \pm 8$	$-65\% \pm 48$

5 Conclusion

The originality of the problem studied in this paper is the specific application to the textile industry. New evaluation methods of stability in rescheduling problem have been tested in order to offer the industrial partner an efficient solution. The continuity of this study is to allow to the company to reschedule the workshop production every time an unpredictable disruption occurs. The method has to be tested on real instances. It is the next step of this study with the industrial partner. Generated several disruptions on same instances is an other perspective. It is a very important prerequisite to have an agile and reactive production plan. It is also a first step on the road to the 4.0 factory transformation. Different evaluation functions have been tested and evaluated thanks to a random instance generator. With random generated data similar to real material, the company will be able to choose the evaluation scenario most appropriate to rescheduling, knowing the impact on both efficiency and stability. To future perspective, the Pareto front can be determine in order to let the industrial choose the solution between a set of solutions that is the most pertinent.

Acknowledgment. This research was supported by ANRT (Association National Recherche Technologie). These acknowledgements also go to our industrial partner. Their participation allows us to work on research topics applied to the current industrial context.

References

1. Arnaout, J.P.: Rescheduling of parallel machines with stochastic processing and setup times. J. Manufact. Syst. **33**(3), 376–384 (2014)
2. Azizoglu, M., Alagöz, O.: Parallel-machine rescheduling with machine disruptions. IIE Trans. **37**(12), 1113–1118 (2005)
3. Baykasoğlu, A., Karaslan, F.S.: Solving comprehensive dynamic job shop scheduling problem by using a grasp-based approach. Int. J. Prod. Res. **55**(11), 3308–3325 (2017)
4. Berthier, A., Yalaoui, A., Chehade, H., Yalaoui, F., Amodeo, L., Coquelet, G.: Machines group and load balancing: an industrial case. IFAC-PapersOnLine **52**(13), 415–420 (2019)
5. Berthier, A., Yalaoui, A., Chehade, H., Yalaoui, F., Amodeo, L., Bouillot, C.: Unrelated parallel machines scheduling with dependent setup times in textile industry. In: CIE (submitted) (2021)
6. Framinan, J.M., Ruiz, R.: Architecture of manufacturing scheduling systems: literature review and an integrated proposal. Eur. J. Oper. Res. **205**(2), 237–246 (2010)
7. Jiang, S., Ong, Y.S., Zhang, J., Feng, L.: Consistencies and contradictions of performance metrics in multiobjective optimization. IEEE Trans. Cybern. **44**(12), 2391–2404 (2014)
8. Kim, Y.-I., Kim, H.-J.: Rescheduling of unrelated parallel machines with job-dependent setup times under forecasted machine breakdown. Int. J. Prod. Res., 1–23 (2020). https://doi.org/10.1080/00207543.2020.1775910
9. Liu, Z., Lu, L., Qi, X.: Cost allocation in rescheduling with machine unavailable period. Eur. J. Oper. Res. **266**(1), 16–28 (2018)
10. Mohan, J., Lanka, K., Rao, A.N.: A review of dynamic job shop scheduling techniques. Procedia Manufact. **30**, 34–39 (2019)
11. Olteanu, A., Pop, F., Dobre, C., Cristea, V.: A dynamic rescheduling algorithm for resource management in large scale dependable distributed systems. Comput. Math. Appl. **63**(9), 1409–1423 (2012)
12. Pfeiffer, A., Kádár, B., Monostori, L.: Stability-oriented evaluation of hybrid rescheduling methods in a job-shop with machine breakdowns. In: Proceedings of the 39th CIRP ISMS, pp. 173–178 (2006)
13. Ruszczyński, A., Shapiro, A.: Stochastic programming models. Handb. Oper. Res. Manage. Sci. **10**, 1–64 (2003)
14. Sun, L., Luan, F., Pian, J.: An effective approach for the scheduling of refining process with uncertain iterations in steel-making and continuous casting process. IFAC-PapersOnLine **48**(3), 1966–1972 (2015)
15. Yalaoui, F., Chu, C.: An efficient heuristic approach for parallel machine scheduling with job splitting and sequence-dependent setup times. IIE Trans. **35**(2), 183–190 (2003)

16. Yin, J., Li, T., Chen, B., Wang, B.: Dynamic rescheduling expert system for hybrid flow shop with random disturbance. Procedia Eng. **15**, 3921–3925 (2011)
17. Zhang, S., Wong, T.N.: Flexible job-shop scheduling/rescheduling in dynamic environment: a hybrid MAS/ACO approach. Int. J. Prod. Res. **55**(11), 3173–3196 (2017)