

# Analysing the Impact of Rescheduling Time in Hybrid Manufacturing Control

Jose-Fernando Jimenez, Gabriel Zambrano-Rey, Abdelghani Bekrar, Damien Trentesaux and Paulo Leitão

**Abstract** Hybrid manufacturing control architectures merge the benefits of hierarchical and heterarchical approaches. Disturbances can be handled at upper or lower decision levels, depending on the type of disturbance, its impact and the time the control system has to react. This paper focuses particularly on a disturbance handling mechanism at upper decision levels using a rescheduling manufacturing method. Such rescheduling is more complex than the offline scheduling since the control system must take into account the current system status, obtain a satisfactory performance under the new conditions, and also come up with a new schedule in a restricted amount of time. Then, this paper proposes a simple and generic rescheduling method which, based on the satisfying principle, analyses the trade-off between the rescheduling time and the performance achieved after a perturbation. The proposed approach is validated on a simulation model of a realistic assembly cell and results demonstrate that adaptation of the rescheduling time might be beneficial in terms of overall performance and reactivity.

**Keywords** Rescheduling time · Manufacturing control · Reactivity

---

J.-F. Jimenez · A. Bekrar · D. Trentesaux  
LAMIH, UMR CNRS 8201 University of Valenciennes and Hainaut  
Cambrésis UVHC, Le Mont Houy, 59313 Famars, France  
e-mail: abdelghani.bekrar@univ-valenciennes.fr

D. Trentesaux  
e-mail: damien.trentesaux@univ-valenciennes.fr

J.-F. Jimenez (✉) · G. Zambrano-Rey  
Pontificia Universidad Javeriana, Bogotá, Colombia  
e-mail: j-jimenez@javeriana.edu.co

G. Zambrano-Rey  
e-mail: gzambrano@javeriana.edu.co

P. Leitão  
Polytechnic Institute of Bragança, Bragança, Portugal  
e-mail: pleitao@ipb.pt

P. Leitão  
LIACC—Artificial Intelligence and Computer Science Laboratory, Porto, Portugal

# 1 Introduction

Manufacturing control architectures have evolved from the traditional centralised and/or rigid hierarchical configurations to more flexible ones based on partial or full heterarchies. While the first ones are focused on optimality, the second ones are based on highly autonomous and cooperative entities, making decisions closer to the process in order to react to unexpected changes of conditions [1]. However, such reactivity, fault tolerance and dead-lock avoidance provided by heterarchical relationships require sacrificing optimality. In fact, since the 1940s, Hebert Simon [2], a pioneer in artificial intelligence, proposed a satisfying principle where models based on satisfactory or good enough decisions work much better than laboriously optimal ones. Moreover, current studies are focused on demonstrating that modelling aiming at providing satisfactory decisions may result in more sustainable, reliable and realistic models [3–5]. Particularly, hybrid control architectures (HCA), also known as semi-heterarchical architectures, provide such balance between satisfactory efficiency and reactivity, since industrials require more realistic models to reduce such a gap [6].

In HCAs, global decisional entities e.g., supervisors, schedulers, coordinators, are in charge of supervising, generating and/or coordinating pre-execution schedules with updated information, focused on one or more production objectives. Afterwards, schedules are shared with local entities which, depending on their level of autonomy, execute and keep track on these schedules [7–9]. Finding good enough schedules is highly constrained by the complexity of the scheduling problem, the required amount of information, and the limited amount of time to come up with a decision. Hence, most of HCAs advocate for artificial intelligence techniques such as metaheuristics, agent orientation, holonic or bio-inspired approaches, or a combination of these with classic mathematical formulations to ensure a certain level of reactivity [10–12].

The disruption handling in HCA's dynamics can be entirely entitled on local decision entities or shared between global and local entities. Out of these two, the latter stands out because of its benefits in terms of overall performance and responsiveness [13]. In such a case, the global entity takes care of the rescheduling of undone tasks. However, if the rescheduling time is too long and perturbations are quite frequent, this configuration may have serious consequences on the overall performance. Two insights to reduce the risk of inefficient rescheduling are: (i) to avoid fully centralized scheduling as proposed in [8], meaning partial centralized scheduling, and (ii) to analyse the impact of the rescheduling time so that this time can become a kind of governance parameter [14] for the system and be adjusted depending on the system configuration and production status; thus it evolves and adapts to new conditions.

This paper focuses on analysing the impact of the rescheduling time when rescheduling is managed jointly by global and local decisional entities. However, assuming a predictive offline and a reactive online scheduling, herein the analysis is only made on the rescheduling executed by a single global decisional entity while

the local entities just take such a schedule as an input to complete it with local decisions. In this case, the time spent for the local decision is being neglected.

This paper is organised as follows. Section 2 briefly describes the rescheduling concept. Then, Sect. 3 performs a generic method to analyse the rescheduling time impact as a rescheduling strategy. A case study based on flexible manufacturing system is presented in Sect. 4 to validate the proposed strategy. Conclusions and further research opportunities are highlighted at the end.

## 2 Rescheduling in HCA

In HCAs, the global entity starts generating optimal or satisfactory predictive schedules based on information coming from the manufacturing execution system (MES) and the current state of the shop floor. Afterwards, when the schedule execution starts, the reactive phase takes care of disturbances using two possible approaches: (i) classic scheduling repair, which may include a complete rescheduling or (ii) switching to a fully reactive configuration in which local entities are autonomous to make online schedules [7, 15] as shown in Fig. 1. The main benefit of complete/partial scheduling repair is that it re-evaluates the conditions of the system and optimises the performance given the restrictions and degradation caused by disruption. However, besides the need of defining a strategy of execution while the schedule is being recalculated, it has synchronization issues when the solution is going to be applied [16]. The main benefit of fully reactive configurations is the instantaneous reaction and thus the capability of the system to deal with various types of perturbations, dropping the rescheduling time to zero and passing to online/real-time rescheduling. Unfortunately, given the myopic behaviour of

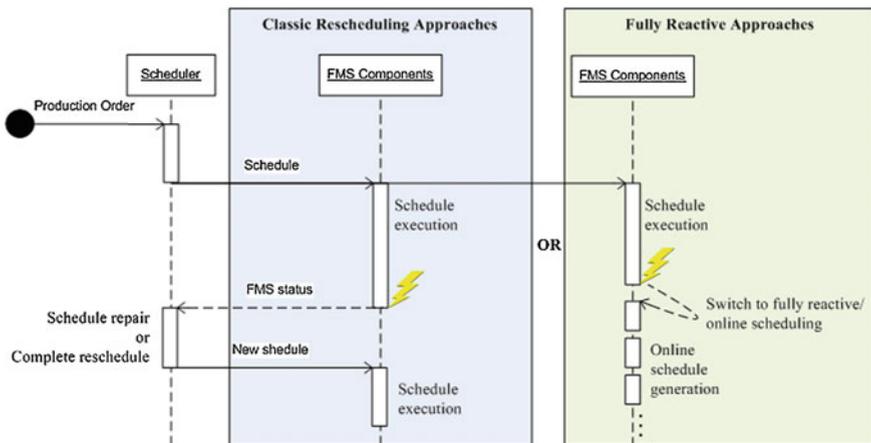


Fig. 1 Two strategies to respond to online perturbations

local entities, it becomes hard to predict the system's overall performance for the remaining production time [17].

Scheduling repair modifies only the affected portion of the original schedule. It was proposed to limit changes to the original schedule and also to reduce the rescheduling time. For instance, ElMaraghy and ElMekkawy [18] proposed a rescheduling algorithm that uses time Petri-Nets and the minimal siphons concept to deal with machine breakdowns in real-time. Similarly, Ahmadi et al. [19] used a right-shift rescheduling method, and Zakaria and Petrovic [20] employed a genetic algorithm based on a match-up approach to modify only a part of the initial schedule within a rescheduling horizon when a change occurs. When complete rescheduling is executed, it is important to ensure the applicability of the schedule by obtaining up to date information to conceive the new schedule. To do so, Novas et al. [21] proposed a collaborative framework between centralised scheduling and holonic manufacturing execution systems (HMES). The role of the HMES is to provide rightful information, in time, to centralised scheduling and use the new schedule as guidance for holon decisions. Although the authors mentioned that the updated schedule has to be as fast as possible, rescheduling time is not addressed.

The scheduling strategy has been widely addressed by researchers looking for balancing the schedule performance and stability. Event-based, periodically, or hybrid-driven strategies have been reported in literature [22]. Hybrid-driven strategies synthesize advantages of event-driven and periodic-driven because it cannot only deal with unexpected events but it also maintains a certain stability of the system. The scheduling horizon sets the rescheduling frequency and is intended to reduce complexity by dividing a wide scheduling range into small segments, which reduces the rescheduling time as reported by H. A. ElMaraghy and T.Y. ElMekkawy [18]. For instance, in this study, the author sets the rescheduling resolution to 1 s. In [23], the rescheduling time is limited as a fraction of setup and processing times and low complex heuristics are used to respect such constraint.

The rescheduling point also impacts the rescheduling time since it determines what needs to be scheduled and when. For instance, if the breakdown occurs nearly at the end of the considered scheduling horizon, low complex techniques can be used because there will be no much impact on scheduling performance. However, if the disturbance occurs right after creating the schedule, in the normal (re)scheduling point or in the middle of the scheduling horizon, it is advisable to apply a more complex technique, as proven by Pfeiffer et al. [24]. Fattahi and Fallahi [25] also studied the impact of the rescheduling point for job arrival disturbances, having the best results when efficiency and stability objectives are taken into consideration. As it has been already mentioned, an analysis on the rescheduling time has not been particularly addressed and most of the studies just mention that it must be short enough compared to processing times. In fact, in some studies, this time is ignored [26]. Since this is not necessarily true and the rescheduling times affect the quality of the schedule, in the next section a generic strategy to analyse the rescheduling time in flexible manufacturing systems (FMS) is proposed. Among the different manufacturing configurations, the FMSs were chosen because of the heterogeneity of the manufacturing resources and their high flexibility, which imply the high

complexity of such systems. This strategy is intended to work with a HCA in which a global entity (scheduler) manages rescheduling, either repair or complete rescheduling, and local entities take care of local decisions (i.e., transport between machines).

### 3 Proposal to Analyse the Rescheduling Time in FMS

From an Operations Research perspective, scheduling in FMS is a more complex version of the classical flexible job-shop scheduling problem, which is known to be NP-hard [27]. This complexity is caused by versatile manufacturing lines, redundant and reconfigurable machines, alternate routings, and flexibility in operation sequencing [28]. Basically, FMS scheduling consists of ordering products for dispatching, allocating each operation to a machine out of a set of capable machines, and sequencing the assigned operations on all machines in order to obtain a feasible schedule. These decisions are taken in the predictive phase (① in Fig. 2), being possible to estimate a certain production time ( $EPT$  in Fig. 2), and/or other production indicators if needed. Once the scheduler finishes, the order is released into the FMS to follow the schedule (② in Fig. 2). If a disruption arrives (③ in Fig. 2), the rescheduling process is triggered. At first, the scheduler needs the current FMS and product status (④ in Fig. 2) to launch the rescheduling technique (⑤ in Fig. 2). Then, a new schedule is ready for execution either until a new disruption arrives and rescheduling is again needed, or the production order is processed (⑥ in Fig. 2). As a result of disruption(s), the actual production time ( $APT$  in Fig. 2) differs from the  $EPT$  in a called lateness time  $Lt$  indicator; it is possible to determine the impact of the disruption, the capacity of the system to absorb it and the impact of the rescheduling time.

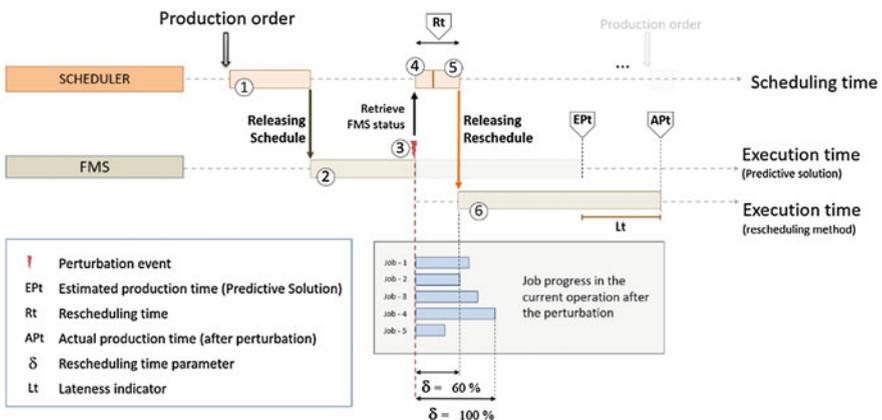
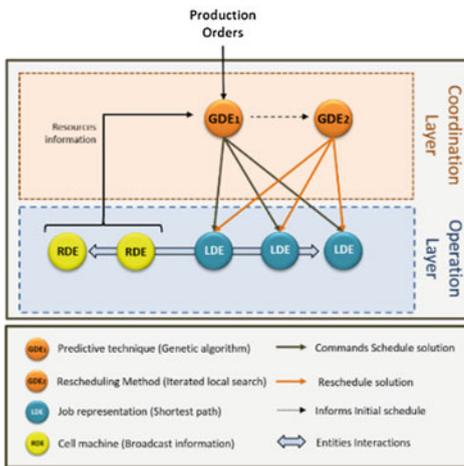


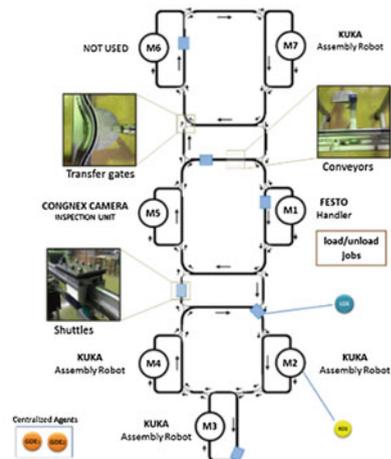
Fig. 2 Rescheduling calculation time analysis

The strategy proposed herein to analyse the rescheduling time is based on a rescheduling time parameter that can be used to limit the time the scheduler spends recalculating a new schedule or repairing it. Normally, when a disruption arrives, jobs are in the middle of a task, e.g., manufacturing, transportation, inspection, queuing, etc.; so when a job finishes the current task, it would be ideal to have the new schedule ready. Not having the schedule ready results in idleness, hence a waste of time and energy. For example, as seen in Fig. 3,  $J_5$  is the job with the shortest remaining time, so if the new schedule were ready right after  $J_5$  finishes, then there would be no idleness at all. In such case, the rescheduling time parameter, denoted by  $\delta$ , is 40% and rescheduling may run only until  $J_5$  finishes its current task. The opposite scenario happens when the scheduler takes the longest job remaining time,  $J_4$  in Fig. 2. Consequently,  $\delta$  is set at 100%, meaning that all jobs have reached idleness waiting for the new schedule. Then, the scheduler must monitor the number of jobs reaching idleness and stop rescheduling when such count equals the following expression:

(a) POLLUX Hybrid Control Architecture



(b) Manufacturing cell layout



(c) Operations for each job (Letter)

Code	Letter	Operations Sequence
901	B	07 - 01 - 01 - 01 - 02 - 02 - 05 - 03 - 06 - 08
902	E	07 - 01 - 01 - 01 - 02 - 02 - 04 - 06 - 08
903	L	07 - 01 - 01 - 01 - 05 - 05 - 03 - 03 - 06 - 08
904	T	07 - 01 - 01 - 02 - 04 - 06 - 08
905	A	07 - 01 - 01 - 01 - 02 - 04 - 05 - 03 - 06 - 08
906	I	07 - 01 - 01 - 05 - 03 - 06 - 08
907	P	07 - 01 - 01 - 02 - 04 - 06 - 08

(d) Processing times at each machine

Operation	Processing times per machine (Seconds)				
	M1	M2	M3	M4	M5
Operation 1 (O1)	20	20			
Operation 2 (O2)		20	20		
Operation 3 (O3)				20	
Operation 4 (O4)		20		20	
Operation 5 (O5)			20	20	
Operation 6 (O6)					5
Operation 7 (O7)	10				
Operation 8 (O8)	10				

Fig. 3 a Architecture of the case illustrated, b layout of the manufacturing cell, c operations sequence for each job, and d processing times of each operation per machine

$$\text{no.jobs reaching idleness} = \delta * \text{no.of unfinished jobs} \quad (1)$$

The number of unfinished jobs means the number of jobs that have not been discharged. Generally speaking, small  $\delta$  values aim at reactivity while greater  $\delta$  values aim at better global performance. In addition,  $\delta$  values can be greater than 100% if needed. Thus, the maximum rescheduling time for idleness, i.e., the longest job remaining time, is taken as reference. For instance; if the remaining time of  $J_4$  in Fig. 2 is 25 s, a  $\delta$  value of 150% would be around 37 s. After the 25th second, all jobs are idle and waiting for the new reschedule for 12 s. This particular case may arrive when the rescheduling problem is very complex and a certain level of performance must be ensured.

As presented in Fig. 2,  $\delta$  can become a parameter that needs to be setup, either statically or preferably in a dynamic way, for instance depending on the rescheduling technique, scope, strategy, rescheduling point and type of perturbation. Then, the rescheduling time parameter can be part of an adaptive FMS control. To validate the proposed strategy, the next section presents an experimental study carried out by simulating a realistic FMS.

## 4 Illustration of the Proposed Rescheduling Method

This section illustrates the inclusion of the proposed rescheduling method into a previously proposed HCA named Pollux [29]. It shows a flexible manufacturing system based on the real flexible assembly cell presented in Trentesaux's benchmark [30]. The manufacturing cell consists of four (4) partially redundant machines (M2, M3, M4, M7), one loading/unloading station (M1) and an automated inspection unit (M5) connected through a conveyor system. It can process seven different jobs ('B', 'E', 'L', 'T', 'A', 'I', 'P') composed of a unique operation sequence of a subset of 8 operations. The manufacturing problem of a given order to address is the jobs dispatching to the cell, the machine allocation for each job and the route path of each job through the conveyor system.

The hybrid control architecture model that contains the proposed rescheduling method is based on the Pollux reference control system [29] and is customised to manage the manufacturing operations within the defined FMS. The model is divided into two layers: the coordination layer, responsible for job dispatching and machine allocation, and the operation layer responsible for path routing. The coordination layer contains two global decisional entities, named  $GDE_1$  and  $GDE_2$ . While  $GDE_1$  is a decisional entity that executes the offline scheduling through a predictive decision-making approach,  $GDE_2$  is a decisional entity responsible for the rescheduling process. Located in the operation layer, the local decisional entities or LDEs are responsible for guiding the jobs in the cell and coordinating the online scheduling either when the instruction are imposed by  $GDE_1$  in the scheduling phase or by  $GDE_2$  in the rescheduling phase. For this reason, the number of LDEs

corresponds to the number of jobs in the production order. Figure 3 illustrates the architecture of the control model and the manufacturing cell layout.

**Predictive scheduling (GDE<sub>1</sub>):** Even though the studied FMS can be resolved as a Flexible Job-Shop Scheduling Problem (FJSP) in Optimization software, the reported research solves this problem through a hybrid technique using a genetic algorithm (GA) and simulation-optimization. The representation of chromosomes in the GA is a schedule that specifies the job dispatching and machine sequence to run the production execution. The GA chromosome is divided into two parts. The first part is an array where each position represents a job to process and the value in this position defines the order to be released into the cell. The second part is also an array in which each position corresponds to a job and contains a sub-array with a sequence of machines to follow. An example of a chromosome (the representation of which is also used in rescheduling) for two jobs is

$$Chromosome_i = \{[T, P][(M_1, M_2, M_2, M_3, M_4, M_5, M_1), (M_1, M_7, M_7, M_3, M_4, M_5, M_1)]\}.$$

While the selection in the GA is done by a tournament selection, the crossover and mutation are separately executed for each part of the chromosome by a one-point and an integer randomization method, respectively. The fitness function used in this instantiation is the makespan of the production order. For this, this paper presents a simulation-optimization technique because the evaluation of the fitness function is evaluated in a simulation of the manufacturing cell. A simulation model of the FMS studied is programmed in NetLogo agent-programming software [31].

Even though NetLogo is designed to simulate agents' environments, this paper uses the commands and report features of this software to run both the genetic algorithm and evaluate the fitness function by its simulation. After this process is executed, the best chromosome with minimum makespan is used for execution.

**Rescheduling method (GDE<sub>2</sub>):** the rescheduling method, which is activated when a disruption is detected (e.g., machine breakdown, urgent order arrival,...), is executed by the GDE<sub>2</sub>.

In this research, an *iterated local search* limited to fulfil a Satisfying principle was used. The iterative local search (ILS) is a hill-climbing method for discrete optimisation problems that improves searches over discrete variables. For this, starting from an initial solution it explores in a reduced space (called neighbourhood of the solution) by following a single chain of explicit set moves [32].

ILS is built as follows: when a disruption is detected, the process starts by retrieving the state of the schedule during execution and setting the initial schedule and representation as a current solution of the ILS algorithm. Then, in an iterated search, it starts improving the current solution in the two parts of the solution. In the first part, it swaps two positions of the solution changing the dispatching order. In the second part, it changes randomly the machine allocation according to the feasibility in the redundant machines.

Certainly, these moves consider the reparation of the solution to correct the consequences of the disruption and are limited to the jobs and operations not processed in both the job dispatching and the machine allocation.

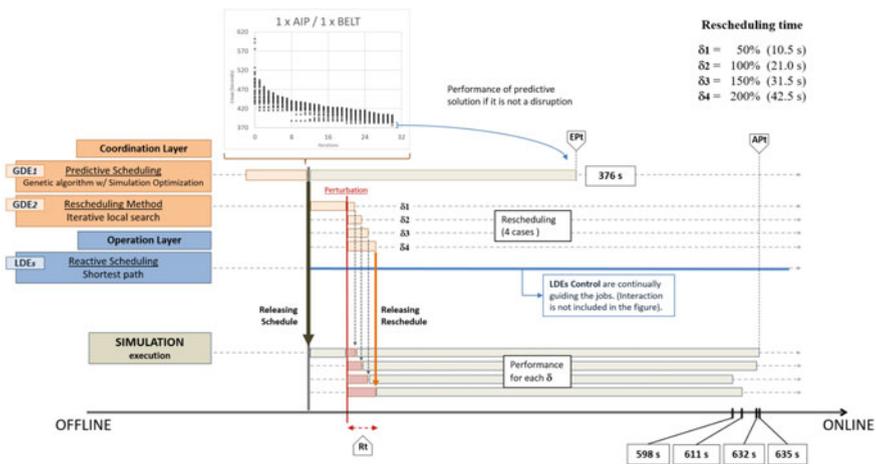
**Reactive scheduling ( $LDE_r$ ):** The reactive scheduling is executed by each LDE. The LDE has two different states according to the manufacturing environment. On one side, when the intensions are imposed by  $GDE_1$  or  $GDE_2$ , the LDE receives the instruction of the job dispatching and machine allocation and executes the processing of jobs accordingly.

In addition, in the path routing decision, the LDE uses a shortest path algorithm in order to get to the next machine. On the other hand, when a disturbance occurs, the LDE is in charge of finalising the on-going task (i.e. manufacturing, transporting, queuing, or waiting to be released) and passing to a stand-by state where the jobs loops within the cell until new instructions are received ( $GDE_2$  instructions).

For this case study, the disruption handling of the proposed HCA is analysed by comparing different values for the rescheduling time ( $\delta$ ) parameter after a fixed perturbation (Fig. 4).

For this, four scenarios were tested as defined by Trentesaux et al. [30]: B0 (2xAIP), C0 (1xAIP and 1xBELT), D0 (1xAIP and 3xBELT) and E0 (3xAIP, 2xBELT and 1xLATE). The disruption considered in the experiment is the breakdown of resource  $M_3$  and is fixed for all scenarios at 50 s after execution kick-off. The rescheduling time ( $\delta$ ) parameter is settled in percentage (%) and is tested from 50 to 400%; here, 100% represents the time elapsed from perturbation occurrence until all available jobs get to the stand-by state. The control model of the FMS was implemented in NetLogo on a PC Intel(R) Core(TM) i7-4770U CPU @ 3.40 GHz with 32.00 GB of RAM memory. Figure 5 illustrates the execution process for the C0 scenario (1 x AIP, 1xBELT).

As seen in Fig. 5, having a  $\delta$  value of 100% for scenario B0 and 150% for other scenarios helps finding a better  $C_{max}$ . In scenario B0, the perturbation arrives a



**Fig. 4** Illustration of the rescheduling method (ILS) in a hybrid control architecture (Pollux) for scenario C0

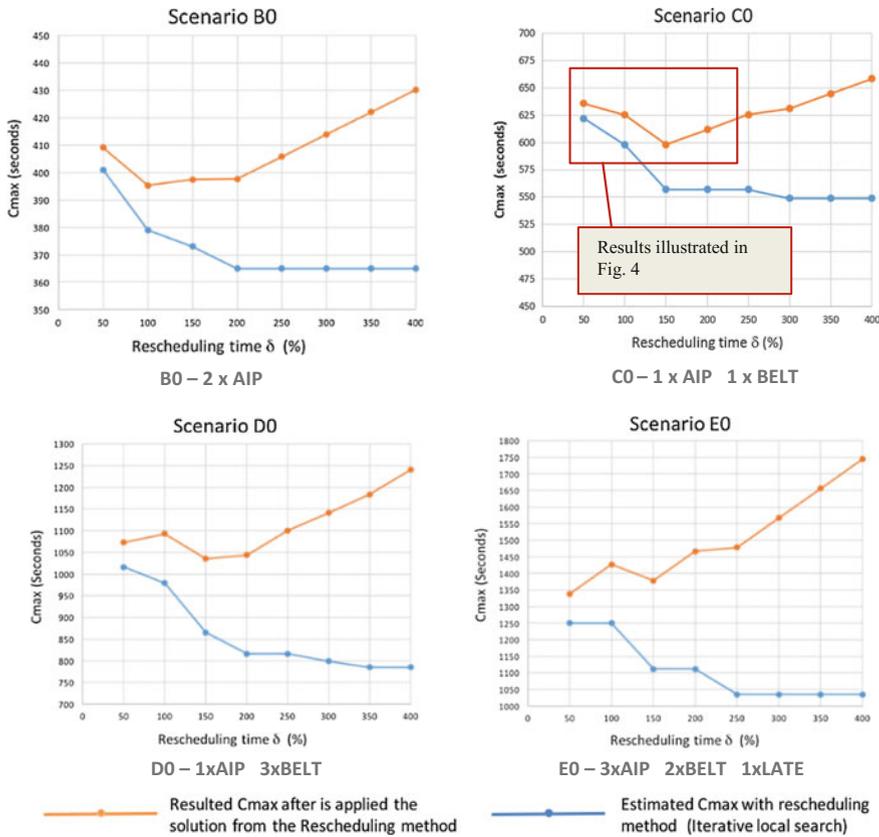


Fig. 5 Experimental results

$t = 50$  s and at  $t = 67$  s all jobs have finished their tasks and the new schedule is ready. For the other scenarios, jobs finish their current manufacturing tasks and have to wait a certain amount of time for the new improved schedule. For instance, in scenario E0, the perturbation arrives at  $t = 50$  s, all jobs finish their current tasks at  $t = 224$  but a new schedule with  $\delta = 150\%$  is ready at  $t = 316$  s.

From these preliminary results, it can be seen that actually a specific rescheduling time could be tuned that mitigates the degradation caused by certain perturbations. Even though by minimising the degradation caused by the perturbation and proving that it is worth to implement a satisfactory solution that is limited by a rescheduling time, some benefits are expected to be obtained in this research direction, it is clear that this needs to be extensively proven in different scenarios and modelling cases. For this reason, this study encourages our research to continue to explore the trade-off between the rescheduling time and the improvements achieved by the method.

## 5 Conclusions

Until now, in most works rescheduling times have been neglected and their impact on the performance indicators of manufacturing system's production has not been studied yet. Indeed, having a short rescheduling time it is needed to ensure reactivity but larger times aim at better performance. Hence, in this paper, a generic methodology was proposed to analyse the rescheduling times in terms of the current system conditions.

Using four different scenarios with increasing complexity, it was possible to show that this rescheduling time is an important parameter; moreover, it can become a dynamic parameter for the entire manufacturing control system. Therefore, the rescheduling time can be tuned up depending on the type of perturbation, current system state, maximum performance deviations, etc. Since for these particular cases an iterated local search heuristic was used to fix the schedule and find an alternative machine, future work will try to produce a thorough study using different rescheduling algorithms.

## References

1. Dilts, D.M., Boyd, N.P., Whorms, H.H.: The evolution of control architectures for automated manufacturing systems. *J. Manuf. Syst.* **10**(1), 79–93 (1991)
2. Russel, S., Norvig, P.: *Artificial Intelligence: A Modern Approach*, 3rd edn. EUA Prentice Hall (2014)
3. De Lara, M., Martinet, V., Doyen, L.: Satisficing versus optimality: criteria for sustainability. *Bull. Math. Biol.* **77**(2), 281–297 (2015)
4. Hansson, O., Mayer, A.: The optimality of satisficing solutions. In: *Proceedings of the 4th workshop on uncertainty in artificial intelligence* (1988)
5. Cardin, O., Mebarki, N., Pinot, G.: A study of the robustness of the group scheduling method using an emulation of a complex FMS. *Int. J. Prod. Econ.* **146**(1), 199–207 (2013)
6. Mourtzis, D.: Challenges and future perspectives for the life cycle of manufacturing networks in the mass customisation era. *Logistics Res.* **9**(1), 1–20 (2016)
7. Pach, C., Berger, T., Bonte, T., Trentesaux, D.: ORCA-FMS: a dynamic architecture for the optimized and reactive control of flexible manufacturing scheduling. *Comput. Ind.* **65**(4), 706–720 (2014)
8. Zambrano-Rey, G.: *Reducing myopic behavior in fms control*, PhD Dissertation. University of Valenciennes and Hainaut-Cambrésis (2014)
9. Shen, W., Maturana, F., Norrie, D.H.: MetaMorph II: an agent-based architecture for distributed intelligent design and manufacturing. *J. Intell. Manuf.* **11**(3), 237–251 (2000)
10. Barbosa, J., Leitão, P., Trentesaux, D., Adam, E.: Enhancing ADACOR with biology insights towards reconfigurable manufacturing systems. In: *IECON 2011-37th Annual Conference on IEEE Industrial Electronics Society*, pp. 2746–2751 (2011)
11. Li, X., Zhang, C., Gao, L., Li, W., Shao, X.: An agent-based approach for integrated process planning and scheduling. *Expert Syst. Appl.* **37**(2), 1256–1264 (2010)
12. Nejad, H.T.N., Sugimura, N., Iwamura, K.: Agent-based dynamic integrated process planning and scheduling in flexible manufacturing systems. *Int. J. Prod. Res.* **49**(5), 1373–1389 (2011)
13. Vieira, G.E., Herrmann, J.W., Lin, E.: Rescheduling manufacturing systems: a framework of strategies, policies, and methods. *J. Sched.* **6**(1), 39–62 (2003)

14. Jimenez, J.F., Bekrar, A., Trentesaux, D., Rey, G.Z., Leitao, P.: Governance mechanism in control architectures for flexible manufacturing systems. *IFAC-PapersOnLine* **48**(3), 1093–1098 (2015)
15. Wong, T.N., Leung, C.W., Mak, K.L., Fung, R.Y.K.: Integrated process planning and scheduling/rescheduling—an agent-based approach. *Int. J. Prod. Res.* **44**(18–19), 3627–3655 (2006)
16. Jimenez, J.F., Bekrar, A., Trentesaux, D., Leitão, P.: A switching mechanism framework for optimal coupling of predictive scheduling and reactive control in manufacturing hybrid control architectures. *Int. J. Prod. Res.* 1–16 (2016)
17. Zambrano-Rey, G.Z., Bonte, T., Prabhu, V., Trentesaux, D.: Reducing myopic behavior in FMS control: A semi-heterarchical simulation–optimization approach. *Simul. Model. Pract. Theory* **46**, 53–75 (2014)
18. ElMaraghy, H.A., ElMekkawy, T.Y.: Deadlock-free rescheduling in flexible manufacturing systems. *CIRP Ann. Manuf. Technol.* **51**(1), 371–374 (2002)
19. Ahmadi, E., Zandieh, M., Farrokh, M., Emami, S.M.: A multi objective optimization approach for flexible job shop scheduling problem under random machine breakdown by evolutionary algorithms. *Comput. Oper. Res.* **73**, 56–66 (2016)
20. Zakaria, Z., Petrovic, S.: Genetic algorithms for match-up rescheduling of the flexible manufacturing systems. *Comput. Ind. Eng.* **62**(2), 670–686 (2012)
21. Novas, J.M., Van Belle, J., Saint Germain, B., Valckenaers, P.: A collaborative framework between a scheduling system and a holonic manufacturing execution system. In: *Proceeding of SOHOMA Service Orientation in Holonic and Multi Agent Manufacturing and Robotics*. pp. 3–17. Springer (2013)
22. Yu, G.D., Yang, Y., Zhao, X., Li, G.: Multi-objective rescheduling model for product collaborative design considering disturbance. *Int. J. Simul. Model.* **13**(4), 472–484 (2014)
23. Gao, K.Z., Suganthan, P.N., Tasgetiren, M.F., Pan, Q.K., Sun, Q.Q.: Effective ensembles of heuristics for scheduling flexible job shop problem with new job insertion. *Comput. Ind. Eng.* **90**, 107–117 (2015)
24. Pfeiffer, A., Kádár, B., Monostori, L., Karnok, D.: Simulation as one of the core technologies for digital enterprises: assessment of hybrid rescheduling methods. *Int. J. Comput. Integr. Manuf.* **21**(2), 206–214 (2008)
25. Fattahi, P., Fallahi, A.: Dynamic scheduling in flexible job shop systems by considering simultaneously efficiency and stability. *CIRP J. Manufact. Sci. Technol.* **2**(2), 114–123 (2010)
26. Gao, K.Z., Suganthan, P.N., Pan, Q.K., Tasgetiren, M.F., Sadollah, A.: Artificial bee colony algorithm for scheduling and rescheduling fuzzy flexible job shop problem with new job insertion. *Knowl. Based Syst.* (2016)
27. Conway, R.W., Maxwell, L., Miller, L.W.: *Theory of scheduling*. Dover
28. Shirazi, B., Mahdavi, I., Mahdavi-Amiri, N.: iCoSim-FMS: An intelligent co-simulator for the adaptive control of complex flexible manufacturing systems. *Simul. Model. Pract. Theory* **19**(7), 1668–1688 (2011)
29. Jimenez, J.F., Bekrar, A., Zambrano-Rey, G., Trentesaux, D., Leitão, P.: Pollux: a dynamic hybrid control architecture for flexible job shop systems. *Int. J. Prod. Res.* 1–19 (2016)
30. Trentesaux, D., Pach, C., Bekrar, A., Sallez, Y., Berger, T., Bonte, T., Leitao, P., Barbosa, J.: Benchmarking flexible job-shop scheduling and control systems. *Control Eng. Pract.* **21**(9), 1204–1225 (2013)
31. Wilensky, U.: *NetLogo: Center for Connected Learning and Computer-based Modelling*, Northwest. University, Evanston (1999)
32. Glover, F., Kochenberger, G.: *Handbook of Metaheuristics 1* (2003)