SCHEDULING OF PRODUCTION AND MAINTENANCE ACTIVITIES UNDER RELIABILITY CONSTRAINT

Rachid Benmansour, Abdelhakim Artiba, David Duvivier, Eric Ramat, Daoud Ait-Kadi

9th International Conference of Modeling, Optimization and Simulation - MOSIM'12
June 06-08, 2012 – Bordeaux - France
"Performance, interoperability and safety for sustainable development"

SCHEDULING OF PRODUCTION AND MAINTENANCE ACTIVITIES UNDER RELIABILITY CONSTRAINT

Benmansour R[1,2], Artiba A[1,2]. Duvivier D[1,2]

[1]Univ Lille Nord de France,
[2]Université de Valenciennes et du Hainaut Cambrésis /
Le Mont-Houy 59313 Valenciennes cedex 9- France
rachid.benmansour@univ-valenciennes.fr

Ramat E[1,3], Aït Kadi D[4]

[3]Université du Littoral Côte d'Opale, LISIC, BP719,
62228 Calais Cedex-France
[4]Université Laval, Québec (Québec)
G1V 0A6

ABSTRACT: *This paper deals with a joint scheduling of production and preventive maintenance activities in the just-in-time context. We propose two mathematical models and a simulation model which are able to consider the maintenance and production views of a production system. The proposed models coordinate the two views so that the sum of maximum weighted earliness and tardiness cost is minimized. The mathematical models are evaluated on one machine/component subject to preventive maintenance without considering breakdowns. The simulation model is evaluated in the same context but is also able to take breakdowns into consideration. Thanks to its modular conception it is also able to easily consider several machines/components with no modification of its internal functioning. The dynamic aspects are modelled by a combination of timed petri-nets and PDEVS models and implemented in the VLE simulator.*

KEYWORDS: *Scheduling, Preventive Maintenance, Stochastic Failures, Reliability.*

1 INTRODUCTION

In today's economical context, companies are bound to exploit in an optimal way their production systems. They have to meet shipping dates that have been committed to customers and subsequently have to mediate between two conflicting objectives, namely, reducing production delays and reducing costs associated with storage. Consequently, every decision maker has to assure a maximum availability of these production tools at minimal costs (Percy and Kobbacy, 2000). To achieve this goal, we may use methods of mathematical optimization or simulation based on the assumptions considered. In this paper we propose two exact models based on a mathematical formulation of the problem. We used the Xpress optimization software to solve the resulting MIP. The obtained optimal solution is then validated through a simulation model. In order to precisely evaluate the performance criteria related to maintenance and production views, simulation is the best adapted solution. In this paper, we suggest an approach integrating optimization and simulation. This approach consists in minimizing jointly the production and the maintenance costs while keeping a reasonable level of machine reliability.

1.1 Literature review

The importance of just-in-time (JIT) scheduling has led to a wide range investigation of scheduling problems that include both earliness and tardiness penalty (Pinedo, 2008), especially for the single machine problems (such as a bottleneck machine); the single machine scheduling problem was the first to be addressed academically and its characteristics and findings have been applied to more complex problems. Most theoretical models do not take machine availability constraints into account; usually it is assumed that machines are available all the time. However, machines are not continuously available. There are many reasons why machines may not be in operation. Some of these reasons are based on a deterministic process, others on a random process. When unavailability periods are considered, there are few researchers that explicitly try to integrate preventive maintenance and scheduling decisions on a single machine. Indeed, all of them do not deal with the earliness-tardiness cost. Furthermore, in these models, the preventive maintenance cost is rarely taken into account. For instance Graves et al. (1999) consider the problem to optimize weighted completion time and they take into consideration only one preventive maintenance period. Ji et al. (2007) consider the same problem to minimize the makespan. Wang et al. (2005) consider the problem of minimizing the total weighted job completion times on a single machine with availability constraints. They show that the problem is NP-hard in the strong sense. However, they propose heuristics for the special case when the weights are proportional and when there is only a single availability constraint. Recently, Kacem et al. (2008) consider the same objective with one unavailability period. They give three exact methods for solving such a problem: a branch-and-bound method, a mixed

integer programming model, and a dynamic programming method. They carry out several computations using these approaches, and show that problems with up to 3000 jobs, can be solved within a reasonable computation time. More recently, Low et al. (2010) have addressed the same problem to minimize the makespan where the unavailability of machine results from periodic maintenance activities. Each maintenance period is scheduled after a periodic time interval and the machine should stop to be maintained after a periodic time interval or to change tools after a fixed amount of jobs processed simultaneously. They show that this problem is NP-hard in the strong sense and give some heuristic algorithms to solve it. Computational results provided by the authors show that the algorithm first fit decreasing (DFF) performs well. An excellent survey on scheduling with deterministic machine availability constraints can be found in the paper by Ma et al. (2010). In this survey, authors present recent main complexity results concerning the joint scheduling of production with unavailability periods in single machine, parallel machine, flow shop, open shop and job shop environment.

Production scheduling and preventive maintenance planning decisions are inter-dependent but most often made independently. Given that maintenance affects available production time and elapsed production time affects the probability of machine failure, this interdependency seems to be overlooked in the literature. Specifically we want to schedule a set of *n* jobs on a single machine to minimize simultaneously:

a) The sum of maximum weighted earliness and tardiness cost,
b) The wasted production,
c) The maintenance costs.

The first objective aims to reduce production delays and the costs associated with storage. The second one penalizes the units of time related of unachieved jobs due to maintenance events (non-resumable job).
The last objective describes the incurred costs corresponding to preventive maintenance.

The remainder of this paper is organized as follows. The second section gives the problem formulations; the third section depicts the simulation paradigms, formalisms and tools that constitute the bases of our simulation model; the fourth section describes an application of our optimization-simulation hybrid model. Finally several conclusions and perspectives are given.

2 PROBLEM FORMULATIONS

Suppose we have a set N of *n* available jobs, each job i requires a given positive processing time p_i. Completion time of job i is presented by C_i. Earliness (E_i) and tardiness (T_i) of job i, maximum earliness (E_{max}), maximum tardiness (T_{max}), and the sum of maximum earliness and tardiness (ET_{max}) in each sequence are evaluated against the common due date D as follows:

$$E_i = \max(0, D - C_i)$$
$$T_i = \max(0, C_i - D)$$
$$T_{\max} = \max_{i \in N}(T_i)$$
$$E_{\max} = \max_{i \in N}(E_i)$$
$$ET_{\max} = E_{\max} + T_{\max}$$

The objective is to jointly minimize the sum of maximum weighted earliness and tardiness penalties while guaranteeing that the reliability of the machine is above a certain level R^*. This goal is often encountered in food industries when companies must deliver their customers in time with fresh food. Storage and late delivery of foods are highly undesirable in case of perishable products.

The problem is solved thanks to two mixed-integer linear programs. The difference between the two models described hereafter is located in the strategy that is used to schedule preventive maintenance actions.

In the first model, the machine must undergo a preventive maintenance before it reaches the age t^*. A preventive maintenance will never interrupt a job (non preemption) and a job is not started if the duration of the sequence of contiguous jobs starting from the latest preventive maintenance (i.e. a batch) is greater than t^*. This means that in this model a maintenance might be done anticipatively (i.e. with a period less than t^*).

In the second model preventive maintenance occurs at a predetermined fixed time $k.t^*$ (k is a natural number). If a job is running at time $k.t^*$ this job is stopped. It is considered as wasted production and will be restarted at the end of preventive maintenance.

2.1 First mathematical model

We propose the following mixed-integer linear program to solve the studied problem to optimality. In this case, the machine must undergo a preventive maintenance before it reaches the age t^* to ensure that the reliability of the machine is above a predetermined level R*.

Min

$$f(S) = c_a E_{\max} + c_d T_{\max} + c_m d_m \left(\sum_{j=1}^{n} y_j - 1 \right) \quad (1)$$

Where *S* denotes a feasible schedule of the jobs, c_a is the per-unit earliness cost, c_d is the per-unit tardiness cost and c_m is the cost of a single preventive maintenance. d_m is the duration of the preventive maintenance and $\sum_{j=1}^{n} y_j - 1$ is the number of preventive maintenances.

Subject to the constraints:

$$\forall\, j \in N \quad \sum_{i=1}^{n} p_i x_{ij} \le t^* y_j \tag{2}$$

$$E_{\max} \ge D - \sum_{i=1}^{n} p_i - d_m \left(\sum_{j=1}^{n} y_j - 1 \right). \tag{3}$$

$$T_{\max} \ge \sum_{i=1}^{n} p_i + d_m \left(\sum_{j=1}^{n} y_j - 1 \right) - D. \tag{4}$$

$$\forall i \in N \quad \sum_{j=1}^{n} x_{ij} = 1. \tag{5}$$

$$\forall i, j \in N^2 \quad x_{ij} \in \{0,1\}$$

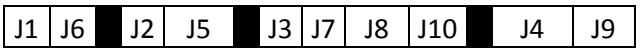
$$\forall\, j \in N \quad y_j \in \{0,1\}$$

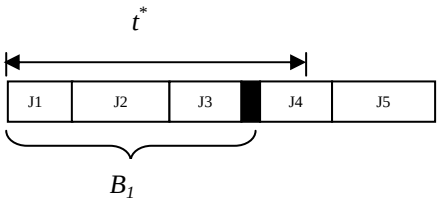
$$E_{\max}, T_{\max} \ge 0$$

$$x_{ij} = \begin{cases} 1 & if \quad Job\, i \in B_j \\ 0 & else \end{cases}$$

$$y_j = \begin{cases} 1 & if\ B_j \quad is\ used \\ 0 & else \end{cases}$$

Where B_j is the j^{th} batch in the sequence. It is the time window of length less or equal than t^* between two consecutive preventive maintenances.





represent the preventive maintenance.

Equation (2) requires that each Batch B_j, if used, may contain jobs whose total duration is less that t^*. The values for maximum earliness and maximum tardiness are calculated by restrictions (3) and (4). Equation (5) assures that each job i is assigned to a specific batch B_j. The next section describes the second model.

2.2 Second mathematical model

In order to take wasted production into account, we propose the following mixed-integer linear program to solve the studied problem to optimality. In this model the preventive maintenance occurs at a predetermined fixed time *k*.t* (*k* is a natural number).

Min

$$f(S) = c_a E_{\max} + c_d T_{\max} + c_w \sum_{j=1}^{n} e_j + c_m d_m \left(\sum_{j=1}^{n} y_j - 1 \right) \tag{6}$$

Where *S* denotes a feasible schedule of the jobs and c_w is the per-unit waste cost.
Subject to the constraints:

$$\forall\, j \in N \quad \sum_{i=1}^{n} p_i x_{ij} + e_j = t^* y_j. \tag{7}$$

$$E_{\max} \ge D - \sum_{i=1}^{n} p_i - d_m \left(\sum_{j=1}^{n} y_j - 1 \right) - \sum_{j=1}^{n} e_j \tag{8}$$

$$T_{\max} \ge \sum_{i=1}^{n} p_i + d_m \left(\sum_{j=1}^{n} y_j - 1 \right) - D + \sum_{j=1}^{n} e_j \tag{9}$$

$$\forall i \in N \quad \sum_{j=1}^{n} x_{ij} = 1 \tag{10}$$

$$\forall\, 1 \le i \le n-1 \quad y_i \ge y_{i+1} \tag{11}$$

$$\forall i \in N \quad e_i \ge 0 \tag{12}$$

$$\forall i, j \in N^2 \quad x_{ij} \in \{0,1\}$$

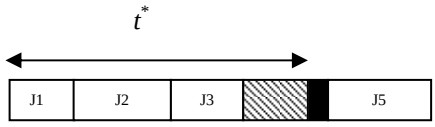
$$\forall\, j \in N \quad y_j \in \{0,1\}$$

$$E_{\max}, T_{\max} \ge 0$$

$$x_{ij} = \begin{cases} 1 & if \quad Job\, i \in B_j \\ 0 & else \end{cases}$$

$$y_j = \begin{cases} 1 & if\ B_j \quad is\ used \\ 0 & else \end{cases}$$

The variables e_j represent the duration of the wasted production in batch B_j (12). Equation (7) requires that the machine undergo a preventive maintenance after exactly an operating time equal to t^*. The values for maximum earliness and maximum tardiness are calculated by restrictions (8) and (9). Equation (10) is equivalent to equation (5) in the first model and equation (11) requires that the first batches are used first.



Represents the wasted production.

The next section briefly introduces our simulation model.

3 SIMULATION MODEL

This section briefly presents the simulation model which relies on the VLE software (Virtual Laboratory Environment; http://www.vle-project.org). This simulator relies on strong concepts and intrinsically provides multimodeling capabilities.
VLE (Quesnel et al., 2009, 2007) is a software and an API (Application Programming Interface) which supports multimodeling and simulation by implementing the DEVS abstract simulator. VLE is oriented toward the integration of heterogeneous formalisms. Furthermore, VLE is able to integrate specific models developed in most popular programming languages into one single multimodel. VLE implements the dynamic structure discrete event (DSDE) formalism (Barros, 1997) which provides the abstract simulators for parallel DEVS (PDEVS) (Zeigler et al., 2000) for the parallelization of atomic models and dynamic structure DEVS (DSDEVS) (Barros, 1996) for the M&S of systems where drastic changes of structures and behaviours can occur over time. DSDE abstract simulators gives to VLE the ability to simulate distributed models and to load and/or delete atomic and coupled models at runtime. VLE proposes several simulators for particular formalisms; for instance, cellular automata, ordinary differential equations (ODE), spatialized ODE, difference equations, various finite state automata (Moore, Mealy, UML statecharts, Petri-nets, etc.) and decision (scheduler with precedence and temporal constraints and predicates for activity activation). VLE can be used to model, simulate, analyze with R software (http://www.r-project.org) and visualize dynamics of complex systems. His main features are: multimodeling abilities (coupling heterogeneous models), a general formal basis for modeling dynamic systems and an associated operational semantic, a modular and hierarchical representation of the structure of coupled models with associated coupling and coordination algorithms, coupling of pre-existing models, distributed simulations, a component based development for the acceptance of new visualization tools, storage formats and experimental frame design tools, and free and open source software.
In this paper VLE is used to implement a simulation model in order to verify and to extend the results of the mathematical models. These models are used to compute the optimal sequence of jobs which is provided to the simulation model. It is noteworthy that this simulation model can handle both breakdowns and preventive maintenance. It comprises a "Breakdowns generator" which can be disabled to precisely compare the results of the simulation model with those of the mathematical models. It is also possible to integrate the randomness of processing times. Thanks to its modular conception it is also able to easily consider several machines/components with no modification of its internal functioning. The dynamic aspects are modelled by a combination of timed petri-nets and PDEVS models and implemented in the VLE simulator. Due to lack of space, this model is not described in this paper. Interested reader can find a complete description of a similar simulation model in (Roux et al., 2010). However, the objective function of the later model is exclusively based on the availability of the machine and a continuous sequence of randomly generated jobs.

The modular conception of the simulation model relies on the "CMSP component" (see Figure 1). Each CMSP is a coupled model (using DEVS terminology) which is composed of several interchangeable and parametrized models. These models are briefly presented in the following paragraphs.

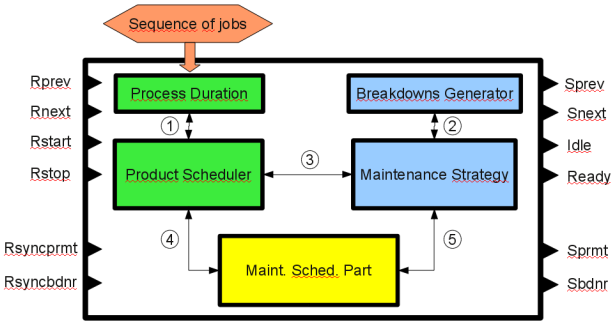


Figure 1: Coupled Maintenance-Scheduling Production building bloc (CMSP)

The "Maint Sched. Part" model (MSP for short) is a Petri-Net model. It simulates the functioning of a machine subject to production, maintenance and breakdowns. It is a "passive model" which needs to be fed by the "Product Scheduler" to work properly.

The "Product Scheduler" model (MSC for short) is also a Petri-Net model in this paper. In the presented study, its aim is to maximize the load of the production process through the MSP model. This allows us to concentrate on maintenance aspects when considering heavily loaded periods. This basic scheduler acts as an infinite loop that sends sequences of events to the MSP model to process as many as possible jobs while taking into account the events provided by the "maintenance strategy". It might be replaced by a more sophisticated model if needed.

The MSC uses the "Process Duration" model to generate process durations. In the presented result, the (optimal) sequence of jobs is provided by one of the mathematical models described in previous sections.

The "Maintenance Strategy" models (MS<Strategy>) are based on various maintenance strategies. Several MS<Strategy> can be used. For instance, the well-known "Bloc Replacement Policy" strategy (Barlow and Proshan, 1976) is available through the MSBloc model. It relies on a "Breakdowns generator" model to "compute" breakdown occurrences. In the presented example,

the MSBloc model is directly implemented in C++. However, it might also be implemented via a Petri-net, a finite state automaton or a more sophisticated coupled-model.

Inside a CMSP the models work together thanks to five links:

(1) The Process Duration model provides MSC with a sequence of jobs to produce;
(2) The Breakdowns Generator provides MS<Strategy> with the next amount of time before breakdown;
(3) MSC and MS<Strategy> are synchronized through this link by a communication of their current states (idle, producing, begin/end of maintenance, breakdown...);
(4) MSC sends orders to MSP and retrieve its responses (start production, stop production...);
(5) MS<Strategy> sends preventive maintenances and breakdowns events to MSP and retrieve its responses (maintenance in progress, ready, processing a job...).

In order to link several CMSP to build a multi-component model, each CMSP has a set of input-ports and a set of output-ports so as to be synchronized with other CMSP. These ports are not detailed in this paper since we are focusing on one CMSP.
The next section presents numerical results.

4 NUMERICAL RESULTS

We illustrate our model by a numerical example with 10 jobs as given in table 1. The value of each parameter is given in table 2. In order to ensure a reliability level above 90.3%, we set t^* to 70 in all presented results. This value is deduced from the Weibull distribution with a shape parameter β=3 and a scale parameter θ=150. In the mathematical models we do not consider the breakdowns. This is justified because the reliability of the machine is never below 90.3%.

Jobs	**p_i**
Job1	26
Job2	17
Job3	8
Job4	20
Job5	34
Job6	25
Job7	15
Job8	18
Job9	40
Job10	26
Sum of processing times	**229**

Table 1: Processing time of jobs

In increasing order, the considered costs c_a, c_w, c_d, c_m have been set to these values to be as close as possible of actual problems encountered in various projects. These costs are also defined on the basis of the same scale depending on the considered "time-unit". For instance, if the time-unit is set to one hour, c_a=100 means that the earliness cost is 100$ per hour. The small value of d_m means that a short amount of time is needed for preventive maintenance. However, the preventive maintenance also corresponds to the highest cost amongst the considered costs. The value of the due date D is not given in Table 2 since the objective of the models is to compute the optimal cost while D varies. The sequence of jobs is also optimized by the models to obtain the best schedule at the lowest cost for each considered value of D. This means that the decision maker can choose the best value of D depending on the list of orders.

Description	**Value**
Duration of preventive maintenance	d_m=4
Optimal age	t^*=70
Per time-unit earliness cost	c_a=100
Per time-unit tardiness cost	c_d=400
Per time-unit waste cost	c_w=300
Per time-unit maintenance cost	c_m=600

Table 2: Parameters

In table 3, D varies from 200H to 280H with a step of 10H. Numerical results show that there is a high cost (23600$) when D=200. This is explained by the fact that it is not possible to produce "in-time" with this value (please remember that the sum of processing times is equal to 229H). Consequently the delays are penalized through the cost c_d which induces such a high cost. As D is increased to a value of 241, the cost decreases to a minimum/optimum value of 7200$. Then the cost increases again as D is greater than 241. This is explained by the fact that we are producing more and more in-advance and are penalized by the cost c_a. In figure 2, the differences in the slope where D is less than 241 in comparison with the slope where D is greater than 241 is explained by the differences between c_d and c_a. The optimal value for D corresponds to the sum of processing times (i.e. 229) increased by the total duration of preventive maintenances (i.e. 3*4).

D	200	210	220	230	240	250	260	270	280
f(S)	23600	19600	15600	11600	7600	8100	9100	10100	11100

Table 3: Results of model 1

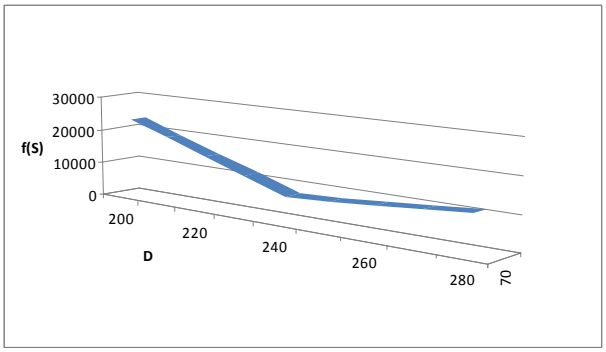


Figure 2: Processing time of jobs (model 1)

Since the sequence of jobs is also adjusted by the model to obtain the best schedule at the lowest cost for each considered value of D, it might be interesting to give this sequence and to compare the sequences obtained by the various models. This sequence is given in Figure 3.

J4	J7	J8	J1	J3	J6	J9	J10	J2	J5

Figure 3. Optimal sequence of model 1

In table 4, numerical results show that there is a high cost (43300$) when D=240. These results are similar to those of the previous model without wasted production. However the optimal value of the due date D is higher than the previous model as shown in figure 4. The optimal value is obtained for a due date equal to 293. This is explained by the fact that there is an additional cost due to the waste of production.

D	240	250	260	270	280	290	300	310	320
f(S)	43300	39300	35300	31300	27300	23300	23300	24300	25300

Table 4: Results of model 2

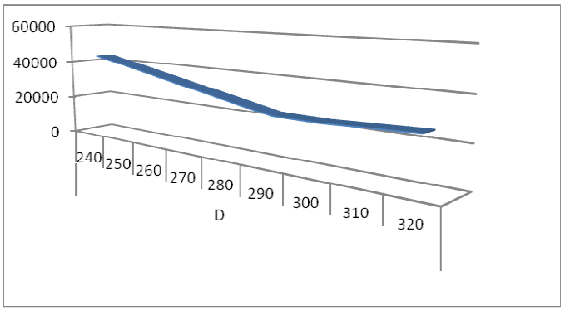


Figure 4: Processing time of jobs (model 2)

The optimal sequence generated by model 2 is given in Figure 5.

J4	J9	J6	J1	J10	J7	J3	J5	J8	J2

Figure 5. Optimal sequence of model 2

Similar results have been obtained thanks to the simulation model. It might also be interesting to compare the results obtained on the basis of the same optimal sequence when considering the classical "Block Replacement Policy" (BRP) for the maintenance. This means that the preventive maintenance occurs at the end of a job if there is at least t^* unit-times since the last preventive maintenance (i.e. no anticipation of the maintenance and no wasted production). In figure 6, the optimal sequence given by model 2 is used by the simulation model and a "Block Replacement Policy" is applied, where t^* is set to 2*70 and the duration of each breakdown is set to d_b=8H. A preventive maintenance event is shown as a "P" in a green block whereas a breakdown is shown as a "B" in a red block. Since the value of t^* is very large, two breakdowns are visible in this Gantt-chart.

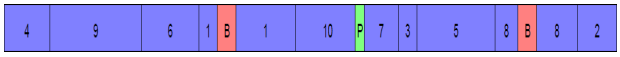


Figure 6: One example of a Gantt-Chart

In this example, the results of the simulation model are as follows: end of last job at t=267H, 16 hours of wasted production, one (1) maintenance event and two (2) breakdowns. In order to evaluate this schedule, we introduce c_b=600 as the cost of a breakdown per time-unit. Considering that D=267H is given by the end of last job, the cost of this schedule is $c_w*16+c_m*d_m*1+c_b*d_b*2$, that is 16800$.

	Model 1	**Model 2**	**Simulation**
Maintenance strategy	anticipative	preemptive	BRP
Cost ($)	7200	22600	≥7200
Reliability (%)	≥90.3	=90.3	≤90.3
t^*	≤70	=70	≥70
D (H)	241	293	≥241

Table 5: Comparisons of the models

Table 5 compares the results of the two mathematical models with those of the simulation model with t^*=70. The results of the simulation model are the same than those of the mathematical models when using the same maintenance strategy. They are also close to those of the mathematical models when using BRP strategy but with a lower reliability due to the fact that maintenance events cannot be anticipated nor preemptive. There is also a small probability that at least one breakdown occurs, leading to increase D and the cost f(S).

CONCLUSIONS AND PERSPECTIVES

In the context of just-in-time production, we have presented two mathematical models and a simulation model applied to one machine subject to preventive maintenance and under a reliability constraint. Several results have been given and we have also shown how to extend the results of the mathematical models through a simulation model. Indeed, the hybrid model composed of one of the mathematical models and the simulation model is able to experimentally demonstrate the efficiency of the presented approach based on various maintenance strategies. Also the simulation model can handle breakdowns.
In future research, we intend to take into account the stochastic aspects of the presented problem in two ways; either we consider the processing times deterministic but the completion time of each job is stochastic due to failures or we consider durations of jobs are initially stochastic. Another immediate extension of this study can be done by considering several machines in the simulation models.

REFERENCES

Barlow, R.E., Proshan, F., 1976. *Theory of Modeling and Simulation*. Wiley Interscience.

Barros, F.J., 1996. *Dynamic structure discrete event system specification: formalism, abstract simulators and applications*. Winter Simulation 13 (1), 3546.

Barros, F.J., 1997. *Modeling formalisms for dynamic structure systems*. ACM Transactions on Modeling and Computer Simulation (TOMACS) 7, 501515.

Graves, G.H. and C.Y. Lee, 1999. *Scheduling maintenance and semiresumable jobs on a single machine*. Naval Research Logistics, 46(7):845-863.

Ji, M., Y. He, and TCE Cheng, 2007. *Single-machine scheduling with periodic maintenance to minimize makespan*. Computers & operations research, 34(6):1764-1770.

Kacem, I., C. Chu, and A. Souissi, 2008. *Single-machine scheduling with an availability constraint to minimize the weighted sum of the completion times*. Computers & operations research, 35(3):827-844.

Low, C., M. Ji, C.J. Hsu, and C.T. Su, 2010. *Minimizing the makespan in a single machine scheduling problems with fexible and periodic maintenance*. Applied Mathematical Modelling, 34(2):334-342.

Ma Y., C. Chu, and C. Zuo. 2010. *A survey of scheduling with deterministic machine availability constraints*. Computers & Industrial Engineering, 58(2):199-211.

Pinedo, M., 2008. *Scheduling: Theory, Algorithms, and Systems*, 3rd edition, Springer.

Quesnel, G., Duboz, R., Ramat, E., Traore, M., 2007. *VLE - a multi-modeling and simulation environment. In: Moving Towards the Unified Simulation Approach*, Proceedings of the Summer Simulation 2007 Conference, SCS, ACM, San Diego, USA, pp. 367-374.

Quesnel, G., Duboz, R., Ramat, E., 2009. *The virtual laboratory environment-- a operational framework for multi-modelling, simulation and analysis of complex dynamical systems*. Simulation Modelling Practice and Theory 17, 641-653.

Roux O., D. Duvivier, E. Ramat, G. Quesnel, 2010. *Optimization of preventive maintenance through a combined maintenance-production simulation model*, International Journal of Production Economics (IJPE), available on line.

Wang G., H. Sun, and C. Chu, 2005. *Preemptive scheduling with availability constraints to minimize total weighted completion times*. Annals of Operations Research, 133(1):183-192.

Zeigler, B.P., Kim, D., Praehofer, H., 2000. *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*. Academic Press.