Modeling flexible workshops scheduling problems: evaluating a timed automata based approach vs MILP

Sara Himmiche, Alexis Aubry, Pascale Marangé, Jean-Francois Pétin

Modeling Flexible Workshops Scheduling problems: evaluating a Timed Automata based approach vs MILP

S. Himmiche [*] A. Aubry [**,***] P. Marangé [**,***] J-F Pétin [**,***]

[*] *Master Ingénierie de Systèmes Complexes, Université de Lorraine (e-mail: sara.himmiche2@etu.univ-lorraine.fr)*
[**] *Université de Lorraine, CRAN, UMR 7039, Campus Sciences, BP 70239, 54506 Vandœuvre-lès-Nancy, France (e-mail: firstname.name@univ-lorraine.fr)*
[***] *CNRS, CRAN, UMR 7039, France*

Abstract: This paper investigates the relevance of modeling workshop scheduling problems using a Discrete Event System (DES) approach based on timed automata (TA). To realize this study, the DES approach is compared with a classical approach based on Mixed Integer Linear Programming (MILP). In order to compare these two modeling approaches, an evaluation system is proposed based on: (i) a problem instances generation system using the classical three-fields notation of Graham ($\alpha|\beta|\gamma$), (ii) a measurement system based on three criteria and associated metrics: complexity, genericity and scalability. This system covers the most common problems when dealing with flexible workshop scheduling. The results obtained by the application of the method is finally discussed.

Keywords: Scheduling Problems Modeling, Discrete event systems in manufacturing, Flexible and reconfigurable manufacturing systems, Timed Automata.

1. INTRODUCTION

Production Scheduling consists in: (i) defining the allocation of operations to machines and (ii) defining the starting and completion dates of a given sequence of operations, performed on dedicated machines to produce a certain quantity of product (usually called job). This definition opens a large extent of problems. The characteristics of scheduling problems are linked to the type of the production system (Flow-Shop, Open Shop, Job-Shop, flexible Shop ...). In this paper, two types of approaches are considered. The first one (the classical approach) is based mainly on Operational Research tools. The second approach is based on Discrete Event Systems (DES) modeling tools and appears in the last few years (Behrmann et al., 2005).

The existence of this second type of approaches led us to a challenging question: which method is the most suitable for modeling and solving scheduling problems? (Marangé et al., 2016)) investigated the efficiency of using DES tools for solving the scheduling problem (i.e. the ability of DES-based approaches for obtaining an optimal schedule) but does not give any idea about the modeling perspective of the question. This is why, in this paper, we are going to investigate the following question: which type of approaches is the more efficient for modeling a workshop scheduling problem?

For answering to this question, the following of the paper is structured as follows. The section 2 defines the basic flexible jobshop scheduling problem and the two modeling approaches. In the section 3, an evaluation method is presented. The section 4 addresses the experimental phase of the evaluation method and gives the associated results. The section 5 concludes this paper and provides some perspectives for a further work.

2. BACKGROUND

2.1 Problem Formulation

This problem is a Flexible Job Shop Scheduling Problem (FJSSP). The choice of this problem was not randomly done. The FJSSP is in fact very often discussed in the literature and finds numerous application in industrial environments.

According to (Rajkumar et al., 2010), the FJSSP can be defined as follows.

J jobs are to be scheduled on M machines and each job j needs a number of O^J_j ordered operations. $\mathcal{J}$ is the set of jobs j and $J = |\mathcal{J}|$ is the number of jobs. $\mathcal{M}$ is the set of machines m and $M = |\mathcal{M}|$ is the number of machines. $\mathcal{O}$ is the set of all the operations that can be realized by the workshop. $\mathcal{O}^J_j$ is the set of the operations o_{jk} in the route of the job j where $k \in \{1..O^J_j\}$ and $O^J_j = |\mathcal{O}^J_j|$ is the number of operations defining the route of the job j. The execution of the k^{th} operation of the job j (denoted as o_{jk}) requires one machine selected from a set of qualified machines called $\mathcal{M}_{jk}$ and will occupy the machine m during d_{jkm} time units until the operation is completed (time is assumed to be an integer clock).

$\mathcal{M}_{jk} = \{m/Q(o_{jk}, m) = 1\}$ is the set of the qualified machines for the operation o_{jk} where Q is an application that is equal to 1 when a given machine m is qualified for a given operation o_{jk} or is equal to 0 otherwise.

A schedule (solution) can be modeled by three decision variables. The allocation decision can be modeled by a binary variable x_{jkm} that decides if the operation o_{jk} is allocated to the machine m. The sequencing decision of the operations can be modeled by a variable y_{jk} (resp. c_{jk}) that fixes the starting (resp. completion) date of the operation o_{jk}.

Moreover, a schedule has to satisfy 4 constraints for being qualified as feasible:

- (C1) **Machine capacity**: each machine can process only one operation at a given time.
- (C2) **Operation route**: for each job j there are some precedence constraints that must be satisfied (job-shop). For each couple $(k_1, k_2) \in \{1..O_j^J\}^2 | k_1 < k_2$, o_{jk_1} must be completed before starting the operation o_{jk_2}.
- (C3) **Machine qualification**: an operation cannot be allocated to a machine m that is not qualified for this operation.
- (C4) **Non-preemption**: an operation cannot be preempted. When started, the operation must be executed until its completion.
- (C5) **Non-splitting allocation**: a given operation cannot be split and executed on two different machines.

2.2 Modeling approaches

The FJSSP is proved to be NP-Hard (Garey et al., 1976) . It is usually discussed by constraint solving methods using scheduling and operational research theories (the classical approach). In the literature, various operational research tools are used to tackle scheduling problems. In this paper, we focus on MILP-based methods. MILP can be defined as a 4-tuple $M = (C, X, P, F)$ where the feature C is a finite set of constraints, the feature X is a finite set of decision variables, the feature P is a finite set of parameters and the feature F is the objective function of the linear program. Using the MILP formulation, the FJSSP can be modeled by the mathematical program given in the figure 1.
The equations (1a) and (1b) model the constraint (C1), ensuring that the two qualified operations $o_{j_1k_1}$ and$o_{j_2k_2}$are not processed at the same time on the machine m. The variable $\sigma_{j_1k_1j_2k_2m}$ is a binary variable that allows to choose which operation is first executed on a machine m. If $\sigma_{j_1k_1j_2k_2m} = 1$ (resp. $\sigma_{j_1k_1j_2k_2m} = 0$), then the equation (1a) (resp. equation (1b)) must be satisfied and the operation $o_{j_1k_1}$ (resp. $o_{j_2k_2}$) is executed before $o_{j_2k_2}$ (resp. $o_{j_1k_1}$). Moreover, the variable H is a real big number that ensures that, in this case, the constraint (1b) (resp. constraint (1a)) is always satisfied. The equation (2) ensures the satisfaction of constraint (C2). The equation (3) models the constraint (C3). $Q(o_{jk}, m)$ is a boolean function that defines if the machine m is qualified to process the operation o_{jk} and the equation (4) models the constraint (C4) and (C5). The equation (5) links the completion date of an operation to its starting date and the equation (6) models the property that the Makespan is always bigger or equal to any completion date. The set of equations (7,8,9,10) defines the respective domain of variation of the variables.

DES-based methods emerge increasingly to model and solve scheduling problems (Kobetski and Fabian, 2009). The core idea is, (i) to model the scheduling problem as a state-transition model that represents operation sequencing, (ii) to use the reachability analysis (Clarke et al., 2000) for finding a possible path to reach a defined state. The DES approach opens a large window on several types of automata. Timed automata are part of communicating automata. They are defined by (Alur and Dill, 1994) as a 8-tuples $A = (S, V, X, L, T, S_m, S_0, v_0)$. The feature S is a finite set of locations, the feature V is a finite set of integer variables, the feature X is a finite set of clocks, the feature L is a set of synchronizing labels, the feature T is the set of transitions, S_m is the set of marked locations, S_0 is the initial location and v_0 is the initial valuation of variables.

To represent the scheduling problem, we define two models: a job model and a machine model. The job model, denoted as α^j , is composed of generic operation patterns (Figure 3), each one models a given operation o_{jk} of the job j and is composed of four locations (Figure 2).

In the first location, the operation o_{jk} is waiting to be executed. The output transition of this location corresponds to the emission of a request asking a machine to execute the operation o_{jk}. Once this request has been emitted, the pattern will wait in the second location for an answer from a machine model which can be:

- *Rejected* if the asked machine cannot execute the operation for unqualification or unavailability reasons; the pattern returns to the first location in which a new request may be done,
- *Accepted* if the machine m is able to perform the operation; the allocation application (x_{jkm}) is then updated and the used machine m is recorded; the automaton then reaches the third location.

In the third location, the pattern is waiting for a completion of operation acknowledgment from the machine m model.

According to the logical route of the job j, this pattern can be applied as many times as there is one operation to be realized in job j, i.e. for all the operations $o_{jk} \in \mathcal{O}_j^J$, as shown in the figure 3. The last location of the pattern instance for the operation o_{jk} (*End of operation* o_{jk}) overlaps with the first location of the pattern instance for the next operation $o_{j(k+1)}$ (*Waiting for operation* $o_{j(k+1)}$ *to be performed*). The complete model of the job ends with a marked location *Finished job*, which indicates that all operations have been executed for this job.

The machine m model β^m interacts with the job j model α^j through the reception of messages (*Requested*) and the emission of messages (*Accepted* or *Rejected*). The machine model is composed of 3 locations (see Figure 4). In the initial location, the machine is idle. Once a request is received from an operation o_{jk} (i.e., from a job model α^j), the machine model evolves to *Computing answer* location and checks if the machine is qualified for the operation (by interrogating the function F). From the location *Computing answer*, if one of the previous

$$
\begin{cases}
\min(C_{max}) & \\
c_{j_1k_1} - y_{j_2k_2} \leq ((1-\sigma_{j_1k_1j_2k_2m}) + (1-x_{j_1k_1m}) + (1-x_{j_2k_2m})) \times H & \forall (m,j_1,j_2,k_1,k_2) \in \mathcal{M} \times \mathcal{J}^2 \times \{1 \dots O^J_{j_1}\} \times \{1 \dots O^J_{j_2}\}, j_2 > j_1 \quad (1a) \\
c_{j_2k_2} - y_{j_1k_1} \leq (\sigma_{j_1k_1j_2k_2m} + (1-x_{j_1k_1m}) + (1-x_{j_2k_2m})) \times H & \forall (m,j_1,j_2,k_1,k_2) \in \mathcal{M} \times \mathcal{J}^2 \times \{1 \dots O^J_{j_1}\} \times \{1 \dots O^J_{j_2}\}, j_2 > j_1 \quad (1b) \\
c_{jk} \leq y_{j(k+1)} & \forall (j,k) \in \mathcal{J} \times \{1 \dots O^J_j - 1\} \quad (2) \\
x_{jkm} \leq Q(o_{jk}, m) & \forall (j,k,m) \in \mathcal{J} \times \{1 \dots O^J_j\} \times \mathcal{M} \quad (3) \\
\sum_{m \in \mathcal{M}_{jk}} x_{jkm} = 1 & \forall (j,k) \in \mathcal{J} \times \{1 \dots O^J_j\} \quad (4) \\
c_{jk} = y_{jk} + \sum_{m \in \mathcal{M}_{jk}} x_{jkm} \times d_{jkm} & \forall (j,k) \in \mathcal{J} \times \{1 \dots O^J_j\} \quad (5) \\
c_{jk} \leq C_{max} & \forall (j,k) \in \mathcal{J} \times \{1 \dots O^J_j\} \quad (6) \\
x_{jkm} \in \{0,1\} & \forall (j,k,m) \in \times \mathcal{J} \times \{1 \dots O^J_j\} \times \mathcal{M} \quad (7) \\
y_{jk} \geq 0 & \forall (j,k) \in \times \mathcal{J} \times \{1 \dots O^J_j\} \quad (8) \\
c_{jk} \geq 0 & \forall (j,k) \in \times \mathcal{J} \times \{1 \dots O^J_j\} \quad (9) \\
\sigma_{j_1k_1j_2k_2m} \in \{0,1\} & \forall (m,j_1,j_2,k_1,k_2) \in \mathcal{M} \times \mathcal{J}^2 \times \{1 \dots O^J_{j_1}\} \times \{1 \dots O^J_{j_2}\} \quad (10)
\end{cases}
$$

Fig. 1. MILP of the Flexible Job Shop Scheduling Problem

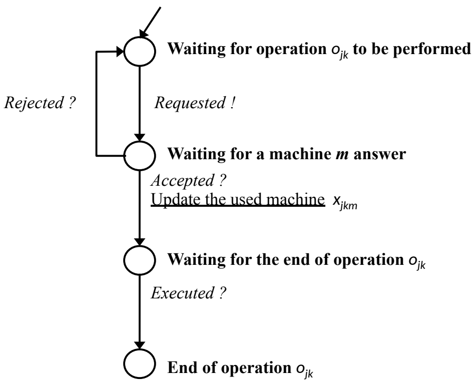

Fig. 2. Pattern for the operation o_{jk}

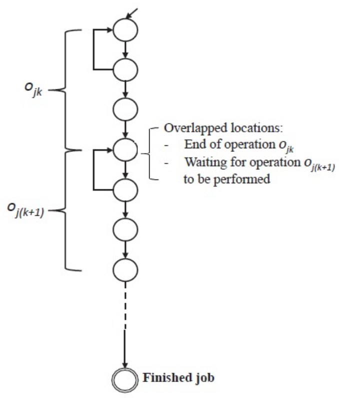

Fig. 3. Job model α^j

conditions is satisfied, an acceptance message is sent to α^j, and the machine model β^m reaches the *Operation* o_{jk} *execution* location; otherwise, a rejection message is sent to α^j and the machine model β^m returns back to the initial location. In the *Operation* o_{jk} *execution* location, the counter is used to assess the evolution of the time. As soon as the spent time since the beginning of the operation reaches the nominal value of the operation duration d_{jkm}, the machine model goes back to the initial location and emits the message *Executed*.

The two models take into account the problem constraints as follows:

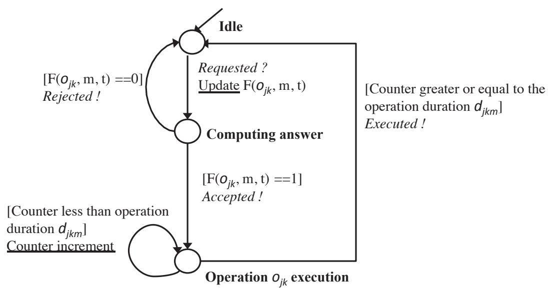

Fig. 4. Machine model β^m

- (C1): this constraint is structurally satisfied because the machine model must be in the initial location *idle* to be able to receive a new request.
- (C2): this constraint is structurally satisfied by the job model: it must receive the message that the operation o_{jk} is completed before sending the request to the operation $o_{j(k+1)}$.
- (C3): this constraint is satisfied by the condition on F
- (C4) and (C5): these constraints are structurally satisfied because a machine will stay in the location *Operation* o_{jk} *execution* until the counter isn't equal at the operation duration.

3. EVALUATION METHOD

The two modeling approaches presented before and their associated solving methods are able to find a solution with a comparable efficiency (Rasmussen et al., 2004). The question here is: How can these two approaches be distinguished regarding their power of modeling?

In this paper, we propose an evaluation based on two systems: a problem instances generation system and a measurement system.

3.1 Problem instances generation system

The goal of this system is to define a set of problems for experimenting each modeling approach. We propose to build this set according to the three-fields notation $(\alpha|\beta|\gamma)$ (Graham et al., 1979). The field α represents the structure of the workshop and is decomposed into two values $\alpha = \alpha_1\alpha_2$. The value α_1 refers to the type of the workshop (type of machines, multipurpose machines or not...). In this paper, only the following values will be considered:

- JMPM for flexible job shop.
- FMPM for flexible flow shop.
- OMPM for flexible open shop.
- XMPM for flexible mixed shop (Hybrid)

The value α_2 refers to the number of machines in the workshop. If this number is not fixed in the problem, like in our case, then $\alpha_2 = \emptyset$.

The field β describes the set of constraints associated with the workshop scheduling problem. The constraints studied in this paper are:

- $s_j, \tilde{d_j}$, there is an imposed earliest starting time and a deadline for each job.
- R_{sd}, S_{sd}, there is a disassembly time of a machine depending on the sequence of operations realized on that machine.
- $unavail_j$, there are some periods of unavailability for the machines.
- $pmtn$, operations can be preempted.

The field γ fixes the optimisation criterion linked to the scheduling problem. This field will be considered as static in the experiment and fixed to $\gamma = C_{max}$, which represents the Makespan or maximum completion time on the machines. As an example, the basic problem defined in the section 2.1 is denoted as $JMPM||C_{max}$. The idea of this system is to make the fields evolve for defining a set of particular workshop scheduling problems.

3.2 Measurement System

For measuring the performance of each modeling approach on the different problems that can be generated using the Problem instances generation system, we propose to define three criteria : Complexity, Genericity and Scalability. Complexity is defined by (Edmonds, 1999) as the property of a model which makes it difficult to formulate its overall behaviour in a given language. In other words, the complexity of a model defines the difficulty to model a problem and the required effort that should be deployed to succeed. The second criterion is Genericity. Genericity is defined by (Gaffuri, 2009) as the capacity of a model to be applicable to a large set of problem instances. The last criterion is Scalability. It has been demonstrated in the literature that scalability has no generally accepted definition (Hill, 1990). The scalability is here defined as the capacity of a model to progress despite the changes of situation in the production workshop. The model should maintain its functioning and performances in terms of finding the solution of scheduling problems.

After defining these criteria, it is necessary to quantify them. The quantification is the answer to the question: How can complexity, genericity and scalability be measured? For each criterion, the idea is to start with the most basic measure when considering a given model which is its size and then to build some situations so that the criteria can be evaluated. The size measure is based on the features of each model. For a TA model, the size can be evaluated through the number of patterns, the number of locations per pattern, the number of transitions per pattern, the number of guards per pattern, the number of labels per pattern, the number of variables per pattern and the number of global variables. For a MILP model, the size can be evaluated through the number of constraints, the number of decision variables and the number of parameters. Moreover we define two indexes that enable to compare two situations regarding a given model.

The first index is called $Rl(f)$. This index enables to compare the size of the modified model regarding a given feature f with the size of the initial model.

$$Rl(f) = \frac{AS(f)}{IS(f)} \tag{1}$$

where $AS(f)$ is the actual size of the model regarding a given feature and $IS(f)$ is the initial size. Three conclusions can be given over the value of $Rl(f)$:

- if $0 \leq Rl(f) < 1$, then the size of the model decreased.
- if $Rl(f) = 1$, then the size of the model is identical.
- if $Rl(f) > 1$, then the size of the model increased.

The second index is called $Rm(f)$. It enables to evaluate the effort provided, regarding a given feature f, to modify a given model for taking into account new considerations/constraints.

$$Rm(f) = \frac{NbA(f) + NbS(f) + NbM(f)}{IS(f)} \tag{2}$$

where $NbA(f)$ is the number of added elements regarding the feature f, NbS is the number of suppressed elements regarding the feature f and $NbM(f)$ is the number of modified elements regarding the feature f. Two conclusions can be given following the value of $Rm(f)$.

- If $Rm(f) = 0$, then there is no modification and there is no effort provided for the modification of the model.
- If $Rm(f) > 0$, then the model has been modified with some required efforts. The higher Rm is, the more the effort of modeling is important.

In the following, it will be defined how the metric and the two indexes $Rl(f)$ and $Rm(f)$ can be used to evaluate the complexity, the genericity and the scalability.

Complexity measure For the complexity, the idea should be to measure the time and intellectual efforts when modeling a given flexible workshop scheduling problem according to the two modeling approaches. This experience has not yet been lead but should be lead for going further and having complete results. However, a first qualitative analysis can be made. In fact, there is an important background in the literature involving the MILP models for modeling and solving scheduling problems. On the contrary, the SED based approach is not frequently used to model this kind of problems. Consequently, the time spent for modeling with the MILP approach is certainly less important than the time spent for modeling with TA if the literature is available to the modeler.

Genericity measure In terms of genericity, we have to measure (i) how a given model is capable to model different production workshops and (ii) how much it costs to have this general model in comparison with a dedicated model. The measure of genericity is binary, a model is general if it is able to model $\alpha_1||C_{max}$ with $\alpha_1 \in \{JMPM, FMPM, OMPM, XMPM\}$, and it is not general otherwise.

For measuring the cost of this genericity, the idea is to evaluate the oversize of the general model in comparison with model dedicated to a particular workshop ($JMPM$,$FMPM$,$OMPM$,$XMPM$). For instance, we take the initial MILP model for $JMPM||C_{max}$ presented in the figure 1. This model has been modified to be general so that other types of workshop can be modeled. In this context, for a given feature f of the model, $Rl(f)$ evaluates the cost of the general model in comparison with the Jobshop model in term of size and $Rm(f)$ evaluates the modeling efforts to be made for going from the jobshop model to the general model.

Scalability measure In term of scalability, we have to measure (i) how a given model is able to catch new constraints/rules/particular cases and (ii) how much it costs to obtain the new model that catches the particularities. A model is scalable if it is able to model $JMPM|\beta|C_{max}$ with $\beta \in \{(s_j, \tilde{d}_j), (R_{sd}, S_{sd}), unavail_j, (unavail_j, pmtn)\}$ and it is not otherwise. To measure the cost of scalability, we use once again the indexes $Rl(f)$ and $Rm(f)$. For example,we have the initial MILP model for $JMPM||C_{max}$ presented in the figure 1. This model has been modified to model $JMPM|s_j, \tilde{d}_j|C_{max}$. In this context, for a given feature f of the model, $Rl(f)$ evaluates the cost of the modified model in comparison with the initial model in term of size and $Rm(f)$ evaluates the modeling efforts that was made to take into account this new constraint.

4. APPLICATION AND DISCUSSIONS

4.1 Genericity evaluation

The first step to evaluate the genericity is to find a general TA model and MILP model that is able to represent not just the basic problem (flexible jobshop scheduling problem) but also the flexible flow shop, the flexible open shop and the flexible hybrid shop. Even though, only the flexible job shop TA model and MILP model were presented in the section 2, the other types of workshops were modeled with the two modeling approaches but are not presented here due to space restriction.

The goal here is to explain how the initial models (for the FJSSP) have been modified for obtaining a general model able to represent also the other types of workshops and then to evaluate the genericity of the two approaches (TA and MILP).

First we can remark that a flowshop is a particular case of a jobshop where all the jobs share the same steps in the workshop. That means that the initial model for the FJSSP is able to represent the flexible flowshop problem ; only the input parameters are different.

Considering the openshop, that means that the jobs are defined by routes where the order of the operations is not predefined. That means that the constraint (C2) is no more

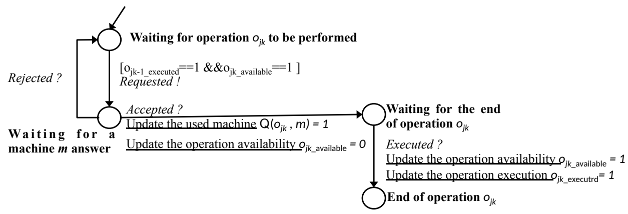

Fig. 5. General operation model

true. And the flexible hybrid shop is a mixed situation: some operations are constrained, some not.

Thus, for generalizing the MILP model, it is necessary to:

(1) Add a parameter $\beta_{jk_1k_2}$ that defines if there exists a precedence constraint between two given operations in a route: $\beta_{jk_1k_2}$ is equal to 1 if the operation o_{jk_1} must be executed before o_{jk_2} (case of the jobshop) and is equal to 0 if such a constraint does not exist.
(2) Add a variable $\alpha_{jk_1k_2}$ that will be constrained by the value of the parameter $\beta_{jk_1k_2}$: $\alpha_{jk_1k_2}$ is equal to 1 if it is decided to execute the operation o_{jk_1} before the operation o_{jk_2} and is equal to 0 if the opposite is decided.
(3) Modify the constraint (2) of the model given in the figure 1
(4) Add two constraints (2'a) and (2'b) and also define the domain of $\alpha_{jk_1k_2}$ in (11)

Finally, the constraint (2) of the figure 1 can be replaced by:

$$\begin{cases} \alpha_{jk_1k_2} \geq \beta_{jk_1k_2} & \forall(j,k_1,k_2) \in J \times \{1 \ldots O_j^J\}^2 \quad (2) \\ c_{jk_1} - y_{jk_2} \leq (1-\alpha_{jk_1k_2}) \times H & \forall(j,k_1,k_2) \in J \times \{1 \ldots O_j^J\}^2 \quad (2'a) \\ c_{jk_1} - y_{jk_2} \leq \alpha_{jk_1k_2} \times H & \forall(j,k_1,k_2) \in J \times \{1 \ldots O_j^J\}^2 \quad (2'b) \\ \alpha_{jk_1k_2} \in \{0,1\} & \forall(j,k_1,k_2) \in J \times \{1 \ldots O_j^J\}^2 \quad (11) \end{cases}$$

We can note that when $\beta_{jkk+1} = 1$ (jobshop case), then α_{jkk+1} must be equal to 1 according to equation (2) and thus only constraints (2'a) is useful and becomes $c_{jkk+1} \leq y_{jkk+1}, \forall(j,k) \in J \times \{1 \ldots O_j^J - 1\}$ which is exactly the constraint (2) of the figure 1.

For generalizing the TA models, the machine model is not modified. For the job model, we do not model any more the logical route but only the operation with its precedence rules. Thus, on the operation model (figure 2), a guard is added between the location *Waiting for operation o_{jk} to be performed* and the location *Waiting for machine m answer* to check that, if exists, the precedent operation for the job is executed and that no other operation of the considered job is being executed. This new model is presented in the figure 5.

To measure the cost of genericity, the size of each workshop specific models and the size of the general models are first computed for each feature f of each modeling approach. This enables to compute the two criteria Rl and Rm. We compute the average value of Rl and Rm for each approach (TA and MILP) over the types of workshop and over the set of features. These values are given in the figure 6. It can be concluded that the size of the MILP model has increased (Rl) for being general while the size of the TA model has decreased. Regarding the efforts for

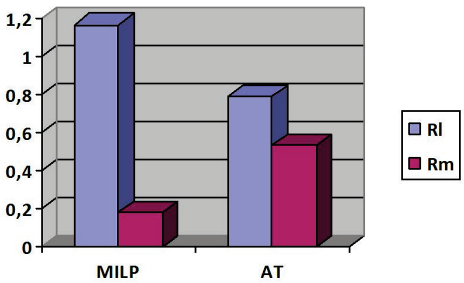

Fig. 6. Global values of *Rl* and *Rm* regarding the generecity evaluation

transforming the particular models to a general one, it can be concluded that this effort is less important for the MILP than for the TA ($Rm^{TA} > Rm^{MILP}$).The changes that allow to find the general model are made only once in the study. Even if the cost of genericity is high for the TA model, the real interest remain in modeling different constraints with this general model (evaluated through the scalability criterion).

4.2 Scalability evaluation

For evaluating the scalability of each modeling approach, We instantiate the general models to have a flexible job-shop problem and then, to add some constraints: i.e. modifying the field β. Each result (model) of this instantiation cannot be detailed here. Results are presented in table 2: the size of each modified model and then the indexes *Rl* and *Rm* for each model have been computed.

Table 1. Values of *Rm* and *Rl* regarding the scalability evaluation

Problem	*Rl*		*Rm*	
	TA	MILP	TA	MILP
$JMPM\|s_i, \tilde{d}_i\|C_{max}$	1.019	1.24	0.049	0.24
$JMPM\|unavail_j\|C_{max}$	1.046	1.30	0.085	0.30
$JMPM\|unavail_j, pmtn\|C_{max}$	1.19	1.77	0.36	0.80

The objective of these measures is to know if the general model is able to evolve and take to into account new constraints that can be found in real industrial production systems. To discuss the results of the three problems presented in table 2, it can be noticed that, for the three problems, the values of *Rl* and *Rm* of the TA models are lower compared to those of the MILP models. In other words, the size of TA models does not increase as much as the MILP models. Moreover, both models require some modeling efforts ($Rm > 0$), but we can remark that $Rm^{TA} < Rm^{MILP}$. That means that the TA modeling approach needs less efforts.

5. CONCLUSION AND PERSPECTIVES

This paper proposed a study on two different approaches that are used to model and solve scheduling problems. It proposed a complete system for evaluating and then for comparing the modeling power of each approach. This evaluation system is first defined by a problem instances generation system based on the three-fields notation of Graham. Secondly, it is completed by a measurement system based on three criteria and associated metrics. Moreover, this evaluation approach has been applied and the results show that Discrete-Events-Systems-based modeling approach is interesting for modeling Flexible Workshop Scheduling problems in comparison with a classical approach like Mixed Integer Linear Programming (MILP). The main drawback of MILP modeling is its high risk of over-sizing and its need of big modeling efforts for taking into account new constraints.

For a further work, it will be interesting to make a sociological study in order to evaluate the complexity of such approach: i.e. evaluate the time that can take a group of people that are in the same level of knowledge to model a problem with the two approaches using TA and MILP tools. It is also important to enrich the presented study by taking into account other constraints (make varying the field β). The development of new measures can also help to enrich the evaluation and the comparison of the two approaches.

REFERENCES

Alur, R. and Dill, D. (1994). A theory of timed automata. *Theoretical Computer Science*, 126(2), 183–235.

Behrmann, G., Brinksma, E., M, M.H., and Mader, A. (2005). Production scheduling by reachability analysis - a case study. In *International Parallel and Distributed Processing Symposium*, 3, 140–147.

Clarke, E., Grumberg, O., and Peled, D. (2000). *Model Checking*. MIT Press.

Edmonds, B. (1999). *Syntactic Measures of Complexity*. Ph.d. thesis, University of Manchester, UK.

Gaffuri, J. (2009). Trois exemples d'adaptation d'un modèle générique de déformation en généralisation cartographique. Publications du Comité Français de Cartographie.

Garey, M., Johmson, D., and Sethi, R. (1976). The complexity of flow shop and job shop scheduling. *Mathematics of Operations Research*, 1(2), 117–129.

Graham, R., Lawler, E., Lenstra, J., and Kan, A.R. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. In *Proceedings of the Advanced Research Institute on Discrete Optimization and Systems Applications of the Systems Science Panel of NATO and of the Discrete Optimization Symposium*, number 5 in 1979, 287–326.

Hill, M.D. (1990). What is scalability. *ACM SIGARCH Computer Architecture News*, 18, 18–21.

Kobetski, A. and Fabian, M. (2009). Time-optimal coordination of flexible manufacturing systems using deterministic finite automata and mixed integer linear programming. *Journal of Discrete-Event Dynamic Systems: Theory and Applications*, 19(3), 287–315.

Marangé, P., Aubry, A., and Pétin, J.F. (2016). Ordonnancement d'ateliers à partir de patrons de modlisation bass sur les automates communicants. In *11th International Conference on Modeling, Optimization and SIMulation, MOSIM'2016*.

Rajkumar, M., Asokan, P., and Vamsikrishna, V. (2010). A grasp algorithm for flexible job-shop scheduling with maintenance constraints. *International Journal of Production Research*, 48(22), 6821–6836.

Rasmussen, J.I., Larsen, K.G., and Subramani, K. (2004). Resource-optimal scheduling using priced timed automata. In *Tools and Algorithms for the Construction and Analysis of Systems*, 220–235. Springer.