



**HAL**  
open science

## An approach for joint scheduling of production and predictive maintenance activities

Ghita Bencheikh, Agnès Letouzey, Xavier Desforges

► **To cite this version:**

Ghita Bencheikh, Agnès Letouzey, Xavier Desforges. An approach for joint scheduling of production and predictive maintenance activities. *Journal of Manufacturing Systems*, 2022, 64, pp.546-560. 10.1016/j.jmsy.2022.08.005 . hal-03847010

**HAL Id: hal-03847010**

**<https://hal.science/hal-03847010>**

Submitted on 10 Nov 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# An approach for joint scheduling of production and predictive maintenance activities

Ghita BENCHEIKH, Agnès LETOUZEY, Xavier DESFORGES  
{agnes.letouzey@enit.fr}

Laboratoire Génie de Production, Université de Toulouse, INP-ENIT, 65016 Tarbes, France

## Abstract

The Industry 4.0 paradigm, thanks to the deployment of cutting-edge technologies enabling the deployment of new services, contributes to improve the agility of productive organizations. Among these services, the Prognostic and Health Management (PHM) contributes to the health assessment of the manufacturing resources and to prognose their future conditions by providing decision supports for production and predictive maintenance management. However, the future conditions of technical production resources depend on the productive tasks they will have to carry out. If their future conditions will not satisfy production criteria, maintenance tasks will have to be planned and productive tasks will be delayed or assigned to other resources for which their future conditions considering these new tasks must be assessed. In this context, a multi-agent system SCEMP (Supervisor, Customers, Environment, Maintainers and Producers) is here proposed in which production scheduling and predictive maintenance planning collaborate and exploit decision supports provided by PHM modules. The proposed multi-agent system provides a framework in which production and the predictive maintenance activities can be scheduled simultaneously by compromising on their objectives. During the scheduling process, SCEMP enables to identify the needed predictive maintenance from the assignments of production tasks to machines, the machine component prognoses and machine models. It schedules production tasks and predictive maintenance activities according to the number, competencies and availabilities of production and maintenance resources. The SCEMP framework is described and presented in the tough job shop context. For this context, case studies have been generated and scheduled within acceptable computation times. To illustrate the SCEMP functioning, some case studies are detailed with the obtained performances. It is flexible and can be adapted to various manufacturing situations. It can also be used to assess the interest of implementing prognostic functions for machine components.

## Keywords

Scheduling problem, Predictive Maintenance, Prognostics and Health Management, Multi-agent systems.

## 1 Introduction

The Industry 4.0 paradigm takes advantage of the deployment of technologies that are mainly Information and Communication Technologies (ICTs) developed in the context of Cyber-Physical Systems (CPSs) leading to Cyber-Physical Production Systems (CPPSs) [1]. The CPPSs can be seen as ecosystems made of interconnected entities such as machine-tools, robots, employees, numerical services [2]. Based on the digitalization, this paradigm aims at providing to manufacturing industries solutions to face the new challenging economic, social and environmental issues by gaining, among other features, agility with keeping productivity high [3]. Aiming at those objectives, waiting for technical production resources failures to occur before undertaking their maintenance is no longer a suitable policy for companies because of unpredicted downtimes disturbing production scheduling and because of maintenance operations often led in emergency-lasting longer than the ones for which logistics were anticipated thanks to plans. Therefore, in the domain of Condition Based Maintenance (CBM), predictive maintenance policies have gained a growing interest by enabling to plan ahead maintenance or replacements of machine components according to their health states and expected lifetimes [4]–[6]. In this context, the Prognostics and Health Management (PHM) systems process data provided by sensors to provide decision support for optimal uses of maintenance resources. The implementation of PHM systems can be done thanks to technologies that already support Industry 4.0 such as the Internet of Things for sensors, the cloud computing or the Internet of Services for data processes [1], [7]. In addition to diagnostics, they must also predict (prognose) the components failures that consists in assessing as accurately as possible their Remaining Useful Lifetimes (RULs) and the RULs of the system functionalities they implement [8]. The assessments of component RULs and consequently of technical production resources times to failure enhance maintenance performances by reducing the number of maintenance operations [9]. This reduction is essential to decrease the total maintenance costs, to cut the number of production interruptions and, also, to reduce the unproductive times. Nevertheless, a joint consideration of both predictive maintenance and production operations is important and guarantees punctuality and quality of production to increase the benefits of the companies.

The management of the production and maintenance is a very challenging problem since these two kinds of activities act on the same resources and are strongly linked to each other [10]. A review of literature dealing with the joint consideration of production and maintenance activities to highlight the importance of such joint considerations is

given in [11]. Several studies have been devoted to optimize the scheduling of production tasks and maintenance activities. A major part of early studies adopts the approach called machines scheduling with availability constraints [12]–[15]. The drawback of these methods is that maintenance activities are planned in a first stage and are considered as constraints in production scheduling in a second stage. This approach does not really satisfy the objectives of both production and maintenance. That is why the scheduling of both production and maintenance activities should be made at the same level in the planning, as proposed by several studies like [16], [17], [18], [19].

For this purpose, Fitouri et al. have proposed a decision making heuristic based on rules [20]. In their study the job shop scheduling problem is considered where the health state of machines is operation-dependent and each machine has a minimum and maximum degradation threshold. Each operation has a specific RUL for each machine thus the cumulative machine degradation imposed by an operation is calculated according to its RUL. Zhu have also considered the joint scheduling problem of production and predictive maintenance (PdM) operations in the article [21]. He proposed an integrated optimization model of production scheduling and maintenance planning, where the degradation of machines is based on Weibull distribution. A Machine learning model with algorithms for scheduling is proposed in [22]. In this work, the authors have considered a PdM model for optimizing production schedule using deep neural networks with database simulations. Another method for joint scheduling of production and PdM operations is proposed in [23]. The proposed method is based on a two stages genetic algorithm integrating flexible job shop scheduling problem using non-linear health predictions.

With the predictive maintenance policy, the time to failures of machines depends on how long these machines have been used for production and, sometimes, on how rough the production tasks were. This requires the determinations of the expected usages of components during the future operations to assess the components RULs [24], leading to the notion of the severity of operations, or the more integrated approach of operation specific health prognostics [25].

Only few studies have integrated the results of PHM systems for production/operation and predictive maintenance planning; many of them are classified in a review of post-prognostics decision making [26]. However, these works do not consider the eligibility constraint (tasks can be manufactured only by some machine). The fact that the duration of the production tasks may depend on the resources on which they are planned is also ignored. Indeed, due to the development of production systems, various kinds of machines have been designed to perform several operations (e.g. turning, milling, drilling, etc.) [9], [11], [27], [28], [29]. For metal removal machining operations, cutting and feed speeds can also vary according to the desired mission [30]. The different forces and motions induce different levels of degradation and thus, the aging of the machine depends on the materials and the types of operations being performed. In this context generic methods based on Bayesian networks were proposed [31], [32]. These methods are designed for complex systems, providing the ability of machine functions to carry out planned production tasks. Local prognoses are combined to provide decision supports for maintenance management (by indicating the components deserving maintenance) and for production planning (by indicating the probability or chance the production tasks are achieved in satisfying conditions). An approach for joint production and predictive maintenance planning has also been developed for flow-shop problem with multifunctional machines [33], [34].

Most of the studies deal with particular cases of scheduling problems considering specific assumptions in a given production system, reducing the portability of the contributions made in the study, thus restricting the domains for which they can be applied. Also, any change in the environment can lead to a general redefinition of the method. To overcome this problem, it is necessary to implement a flexible generic method that can be easily adapted to any changes in the environment. Therefore, a method that has a modeling tool to take into account the adaptation and evolution of a production system is proposed in this paper. The proposed planning tool allows to consider each entity of the problem separately, while keeping the links between them. This simplifies its adaptation to changes that may occur in the environment because each entity of the problem may have its own characteristics, constraints and objectives to consider, which is generally the case in production systems.

In this paper, the joint job shop production and predictive maintenance scheduling is addressed. Most of production scheduling problems are NP-hard [35], which means that the time needed to find a solution grows exponentially with the number of instances (jobs, resources...). Indeed, it have been proved that the scheduling problem on one machine with availability constraints is an NP-hard problem even when they are previously known [14]. The problem of minimizing the makespan with availability constraints on parallel machines is also an NP-hard problem [36]. The problem becomes even more complex if various uncertainties related to the random demands and failures are introduced [37]. As mentioned previously, the duration of operations can also be a variable that depends on the resource features. In addition, and especially when considering networked companies, customers, producers, maintainers have their own objectives, constraints and structures that should be taken in account.

Considering the complexity of the problem, the use of exact methods to solve it within a reasonable time is almost impossible. Therefore, rather than solving the whole problem at once, it could be more convenient to divide it into several subproblems. Each element or stakeholder of the problem (customer, producer, maintainer...) can be considered as a specialized entity. As these entities are more or less autonomous, a process of coordination between

them is necessary. These considerations lead to the notion of agent to implement these entities and to Multi-Agent Systems (MASs) to make these entities interact and interacting entities is part of the functioning of CPPSs [2]. MASs offer a convenient framework to implement the different services and resources of manufacturing systems where each agent can have its own structure and its own reasoning process. MASs enable to solve complicated problems by cooperation between the agents and to solve conflicts using negotiation techniques. They have shown their effectiveness regarding scheduling problems. Therefore, they have been widely accepted in applications for production and/or maintenance scheduling [29], [38]–[41], [42], [29] as the autonomous, distributed and dynamic nature of the agents fits the requirements to design complex, flexible, robust and dynamic manufacturing scheduling methods [43]. An agent is a software entity that has a knowledge base, a reasoning process, and an explicit model of the problem to solve. The agents are able to operate without any human intervention and have the control over their actions and their own state (autonomy). Among their different features [44], the agents interact with each other, and possibly with humans, in order to complete their own problem solving and/or to help others (social ability), this requires means by which communication can be held [45]. The design of communication and negotiation protocols between agents is therefore essential for the proper functioning of decentralized control. These protocols must be simple, while being able to perform all the operations assigned to the agents [46]. The Contract-Net Protocol (CNP) is one of the most recognized negotiation protocols. Introduced by Smith in 1980 [47], it was the first work that used a negotiation process involving a mutual selection by the agents, using a coordination environment that provides a shared memory space for communicating data in an asynchronous way. CNP is based on the contracting mechanism used by businesses to govern the exchange of goods and services [48]. The manager agent announces the tasks to be performed by the other agents and requests bids for the execution of the tasks. These bids are evaluated according to specified criteria (price, quantity, due date, distribution of tasks), and the winning bidder gets the contract. However, there was no generic protocol for task announcements, auctions and validation decisions [49]. CNP has demonstrated its ability to test various scheduling strategies [50] and to schedule decentralized and integrated manufacturing systems [38]. Among the MASs based on CNP, SCEP (Supervisor, Customers, Environment and Producers) framework allows to deal with flexible manufacturing routings and different machine structures and it can associate different objectives and constraints to the agents [50]. SCEP has shown its effectiveness for different kinds of scheduling problems [51]–[54]. However, it does not consider the joint scheduling of production and predictive maintenance operations.

In this work, we propose a new MAS framework called SCEMP (Supervisor, Customers, Environment, Maintainers and Producers), inspired by SCEP. Like in [38], SCEMP considers decentralized manufacturing systems that can be made of networked companies but, in SCEMP, new maintainer agents are added and the predictive maintenance strategy is implemented. The producers, considered as production resources (e.g. machines), can be complex systems that can perform different functionalities with different aptitudes and each functionality uses a set of the machine devices or components. The devices of technical production resources are continuously observed by PHM modules in order to provide information about their health states (diagnoses) and their future conditions (prognoses) [1]. The processing of these information (diagnoses and prognoses) by methods as the ones proposed in [31], [32] enables the assessments of the machines abilities to carry out production tasks as well as the identification of the devices to maintain. This enables the producer agents to schedule the production operations according to their machines current and future health states and to call the maintenance agents for maintenance activities when the scheduled tasks on their machines cannot be carried out in acceptable condition. Therefore, production and maintenance tasks are planned simultaneously aiming at better compromising between production and maintenance objectives. SCEMP allows to implement, for the different producer and maintainer agents, different scheduling methods based on different criteria as it can be the case for small networked companies. The aim is here to present the design and principles of SCEMP and to show how it works in the tough case of joint job shop and predictive maintenance scheduling that, to our knowledge, has seldomly been addressed. Indeed, the only work we found in literature is the one described in [55]. In this work, the developed method is able to define the needed maintenance but it does not consider the availability of maintenance resources.

The description of the considered problem is formulated in section 2 whose modeling is then presented in section 3. The section 4 is devoted to the presentation of the MAS framework SCEMP proposed to solve the problem. First, we introduce the different agents and, then, the communication protocol between these agents is described. In section 5, a first illustrative example is given to evaluate the behavior of SCEMP, involving few machines, having different functionalities whose devices are not all monitored by a PHM system, and few maintenance agents in a job shop context. Another example based on a single machine, is used to compare performances between a systematic preventive maintenance policy and policy based on PHM thanks to the proposed SCEMP framework. Finally, a conclusion of this work is given in which some directions for further research are listed.

## 2 Problem statement

This study addresses the job shop scheduling problem of manufacturing workshops under availability constraints. A workshop consists of  $m \geq 1$  machines able to perform one or several kinds of tasks and where the machines are subject to predictive maintenance. A manufacturing workshop scheduling belongs to a class of scheduling problems in which the jobs are composed of several tasks [56]. Sets of production jobs, machines and maintenance resources are given. A job requires one or more tasks that must be carried out on specific machines capable to perform them within a given time frame. Workshop scheduling problems are then classified according to the sequences in which the jobs tasks are carried out but also according to the types of machines in the workshop. Three types of workshops are defined and described in [57]: flow shops, job shops and open shops. In this paper job shops are considered where the tasks of each job must be executed in a given order specific to each job. The objective of these workshops is to schedule jobs as a set of Manufacturing Orders (MO) requested by a set of customers. Each MO (see Figure 1) has a routing (composition a) which is a linear sequence of production tasks, denoted by TF (composition b). The treatment to be performed for each MO depends on its routing. Each TF of the routing requires at least one manufacturing operation called here “activity” (composition c) and it can be executed by a machine  $m$  if it has all the corresponding functionalities. In the workshop, there can be several machines that have the needed functionalities to carry out a TF. However, capabilities of machines to perform a given activity are not necessarily the same. Therefore, the duration of a TF depends on the capability of the machine to perform the activity the TF requires. A production competency of a machine is the association between a specific functionality and the capability of the machine to perform the activity (association d and g). The duration of execution  $p_{i,j,k}$  of a  $TF_{i,j}$  of  $MO_i$  on a machine  $m_k$  is calculated by the equation (1), where  $cap_{k,F}$  is the capability of machine  $m_k$  to perform the functionality  $F$  and  $\Gamma_{i,j}$  is the set of functionalities required by the operation  $TF_{i,j}$ .

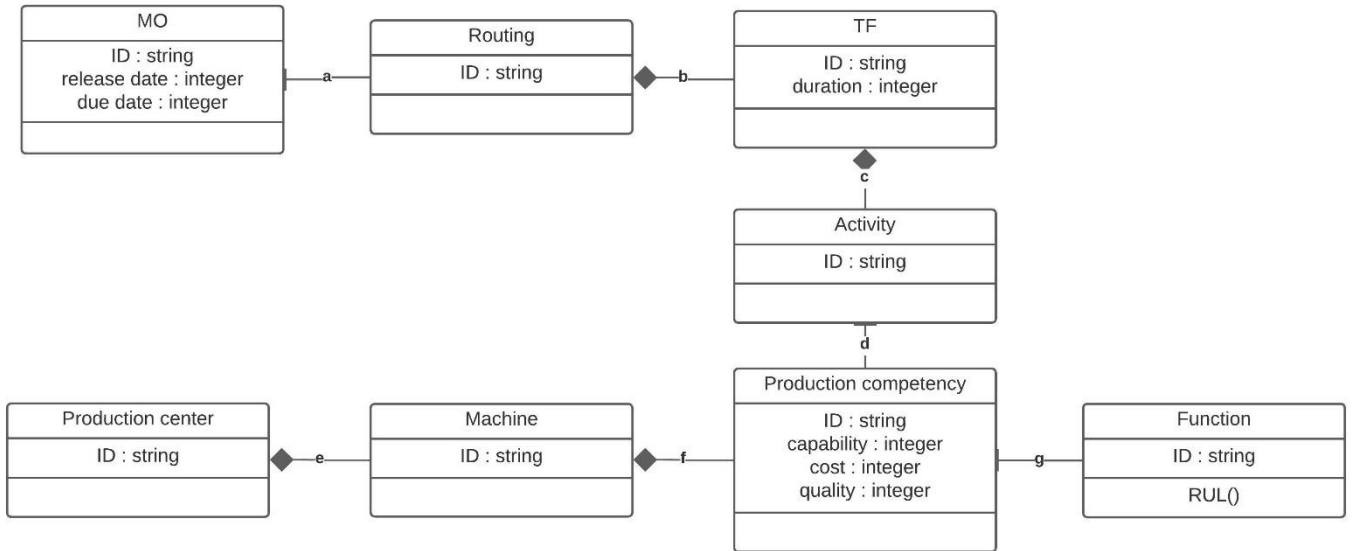


Figure 1: Relation between MO and machines

$$p_{i,j,k} = \text{duration}_{i,j} \times \max(\text{cap}_{k,F} | F \in \Gamma_{i,j}) \quad (1)$$

The workshop is here represented as sets of machines considered as complex systems able to perform different functionalities but can only execute one at a time. To carry out a TF, the machine uses specific groups of devices (or components) that implement the functionalities required by the TF.

In order to increase machines availabilities and to avoid downtimes or to reduce their durations, PHM functionalities must be implemented according to the seven-layer frame proposed by the OSA-CBM (Open System Architecture for Condition Based Monitoring) project [58] leading to standard ISO 13374-1. Distributed implementations of this frame based on holonic architecture have shown their effectiveness in the domain of health assessment (diagnosis) of complex systems [59] but also in making decision to maintain devices from a fleet monitoring [60]. Knowing the importance of predicting and preventing failures, the machines are supposed to be continuously monitored by PHM modules whose implementation can be compliant with architecture proposed in [1]. These modules provide the current health states of the machine devices, the predictions of their degradations and therefore the assessments of their RULs. From the information about the devices provided by the PHM modules, methods like the ones proposed in [31], [32], assess the ability of sub-functions, functionalities, sub-systems and, so, of the machine to perform the TF to plan. Thus, production management has decision supports for assigning a TF to a machine or not. These methods also make it possible to define the components or devices that need to be maintained to improve these abilities. Thus, maintenance management has the information it needs to define the necessary resources and the durations of the

maintenance tasks to plan them. Because this scheduling problem is NP-hard, we do not propose an optimal method but a collaboration framework between stakeholders (customers associated to MO, producers associated to machines and maintainers). In this framework, each stakeholder can implement its own scheduling method aiming at satisfying its own objectives. The studied problem is modeled from the following assumptions:

- at any time, each machine can handle at most one activity (production or maintenance),
- each production task TF can be processed by at least one machine in satisfying operating conditions,
- the processing time of a production task TF depends on the machine capabilities to perform its needed functionalities,
- machines devices or components are monitored by PHM modules that provide their RULs at each time t.

### 3 UML Generic model

For modeling the problem, the object-oriented Unified Modeling Language (UML) have been selected [61]. Indeed, the object-oriented approach encompasses concepts like inheritance, polymorphism, abstraction, encapsulation, communication, etc. [62]. These concepts allow to consider a large variety of resources (manufacturing, maintenance...) and the definition of interfaces and services those entities share even if their implementations are different. This offers a generalization ability of the models and an adaptability to many cases.

The problem is presented by the UML class diagram shown in Figure 2, where the three main kinds of agents (customers, producers, and maintainers) and the association between them are shown. A producer agent is made of a machine (association e) that is used to perform production activities (association d). A machine gathers a set of entities and a set of competencies (association f' and f). The different devices bring into operation the functionalities of the machines which are associated to production competencies. In fact, a machine is made of a set of interconnected elements called "entities". An entity can be either a device (a component) or a function (generalization m and m'). A function is composed of a set of entities (composition h), and thus, according to the definition of an entity, each function can be made of a group of devices, a group of functions or both. In other words, devices bring into operation a whole function and those functions can bring into operation other functions of higher levels. The highest-level functions are called functionalities and they are associated to production competencies (association g) which are characterized by their capabilities to execute the corresponding functionalities. Two types of functions are distinguished:

- Simple functions are characterized by the fact that if one involved entity fails, the function becomes unavailable and cannot be executed.
- Functions called redundancy functions are conceived with several entities used for the same purpose (generalization n and n'). Thus, if one or several entities fail, depending on the order of the redundancy and of the number of redundant entities, the function can still be available as long as there are enough entities able to carry the service out.

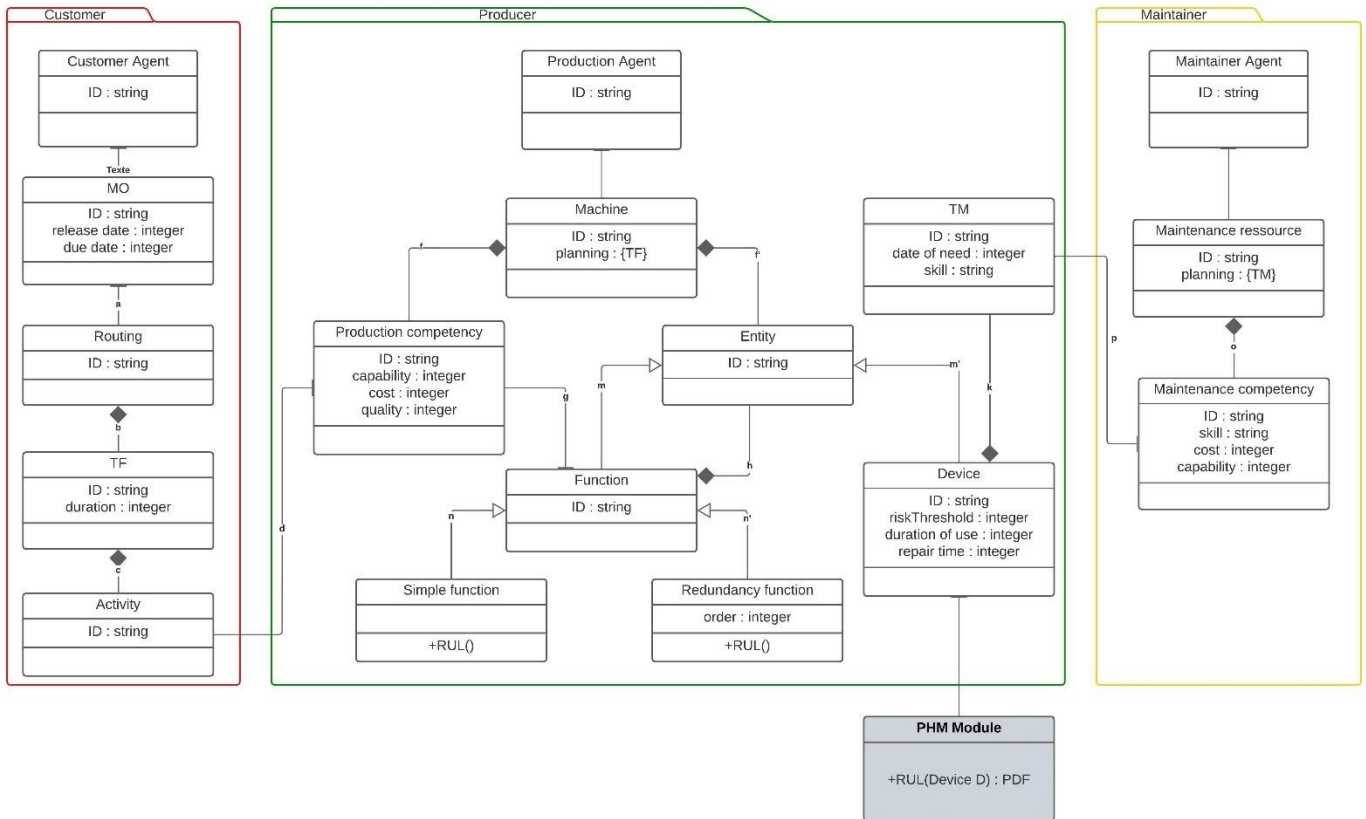


Figure 2: UML class diagram modeling

Each device degrades when used and has a degradation threshold beyond which the failure risk becomes too high for a proper functioning. Reaching this threshold is considered as the device failure. To prevent this failure, a device must undergo a maintenance task (TM), if it leads to the machine inability to carry out a planned sequence of TF within acceptable conditions that can be characterized by a risk threshold. Each device is supposed to be monitored by a PHM module (association l) that returns the Probability Density Function (PDF) of its RUL at each instant  $t$  from which the time remaining before its failure and, so, the time to reach its degradation threshold, can be predicted statistically (see Figure 3). The provided PDF may depend on the severity of the TF. However, implementing prognostic systems dealing with degradation observations and their extrapolations for all the machine devices would be too costly. Thus, PHM modules can use data, possibly provided by the device suppliers, such as the ones leading to the definition of Mean Time To Failure (MTTF) or Mean Time Between Failures (MTBF) and associated to the features of their imprecisions in the absence of prognostic system for some devices [31]. Therefore, such devices are considered to be subject to systematic preventive maintenance. RULs of the functionalities are calculated from the RULs of the devices bringing them into operation. A function can be seen as a tree structure of entities. So, it is obvious that the RUL of a function is based on the RULs of the devices belonging to its structure, the levels of the devices in its structure and the combination of sub-functions (redundancy and/or simple functions).

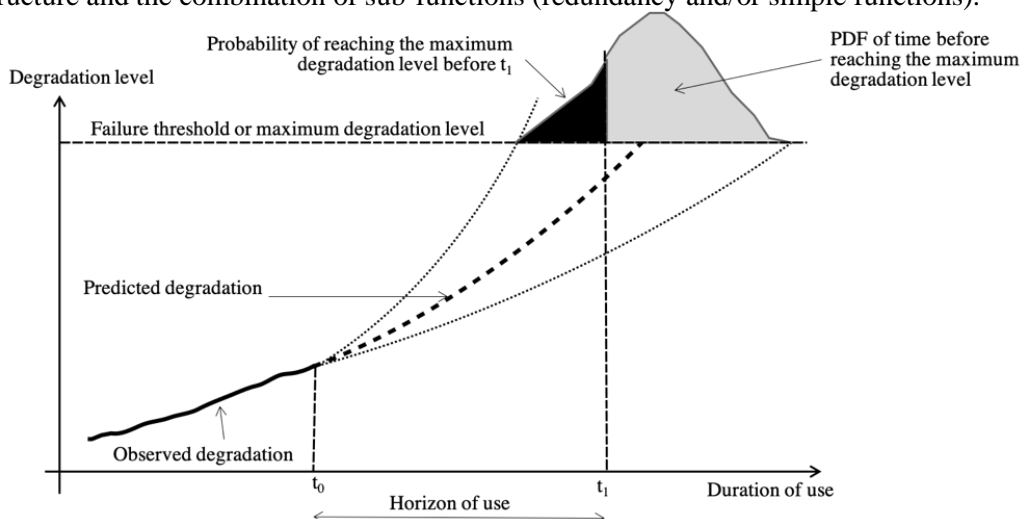


Figure 3: PDF of the prediction of RUL

Knowing the probabilities of failures enables prior scheduling of production and maintenance tasks since it informs about the future availabilities of the machine functionalities and the future needs of maintenance operations. The need of a maintenance task (TM) involving a specific device is declared at a time  $t$  when its failure probability makes the required machine functionalities unable to fully achieve within acceptable conditions the planned sequence of TFs under a given probability limit ( $p_s$ ). A method to identify the devices that should undergo maintenance to better improve the probability that the functionality achieves in good condition a production operation is described in [31], [32]. A device can be repaired by a maintainer if this one has the corresponding competencies to carry out the needed TM (association  $p$ ). Moreover, one maintainer can only repair one machine at a time, but many maintainers can work on a machine at the same time if this one has several devices that should undergo maintenance.

## 4 SCEMP framework

The multi-agent system framework, Supervisor Customers Environment Maintainers and Producers (SCEMP), proposed in this paper is made of several agents. Each agent has its own objectives to satisfy and its own constraints to respect. SCEMP is a framework in which the agents cooperate indirectly to carry out the scheduling process (Figure 4). This process consists of:

- a set of jobs (MO), each managed by a customer agent,
- a set of machines, each managed by a producer agent,
- a set of maintenance resources, each managed by a maintainer agent.

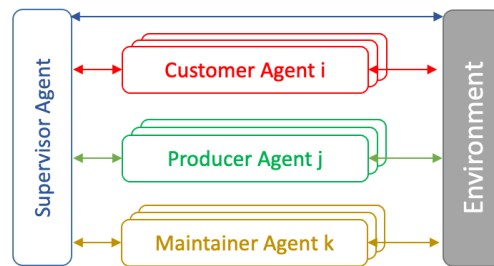


Figure 4: SCEMP framework

SCEMP is inspired by the Contract-Net protocol with the notions of call for tenders and auctions, but its communication is done through a blackboard, called environment, on which the agents write their demands and proposals. The access to this blackboard is controlled by the supervisor agent whose main role is to protect the information of the blackboard, such as there is no interference or loss of information during the scheduling process. When a call for orders is launched in the environment, the concerned agents propose bids for the tasks in question and write their proposals in the environment. The bidding agent compares the different proposals and chooses the one that best suits its objectives. When a proposal is validated, a contract between the bidding agent and the winning agent is established and the Final Position (FP) for the item is thus set. If the FP is empty at the end of the process, this means that the object is still in the “free” state, i.e. it is not yet assigned to any resource. Otherwise, it goes to the “validated” state and cannot be modified later.

### 4.1 Agents description

The **Environment agent** is a blackboard in which the different agents write and retrieve the information they need to solve their sub-problems. The environment is therefore a critical resource in which information must be ordered and written in a specific way that can be understood by the other agents. It is made up of a set of objects all having the same structure. Each object corresponds to a TF or a TM and is formed by a 3-tuple (WP, FP, PL), where WP is the Wished Position defined by the agent who made the offer (customer agent in the case of a TF and the producer agent in the case of a TM), FP is the Final Position chosen for the task in question and PL is a list of proposals given by the agents concerned (see equation(2)) where  $Pr_{i,j,k}$  is the proposal given by the agent  $k$ , and  $\mu_{i,j}$  is the set of resources (producers for a TF, or maintainers for a TM) having the competencies required by the task. For each task  $TF_{i,j}$  or  $TM_{i,j}$ , the auction proposed by an agent  $k$  consists of an Effective Position ( $EP_{i,j,k}$ ) and a Potential Position ( $PP_{i,j,k}$ ).

$$PL_{i,j} = \{ Pr_{i,j,k} = (EP_{i,j,k}, PP_{i,j,k}) / \forall m_k \in \mu_{i,j} \} \quad (2)$$

- The EP is determined by the producer or maintainer considering all other tasks that can be scheduled on its resource. These are the tasks that are in the “validated” state as well as the tasks the agent’s resource can perform (Figure 5 (a)).



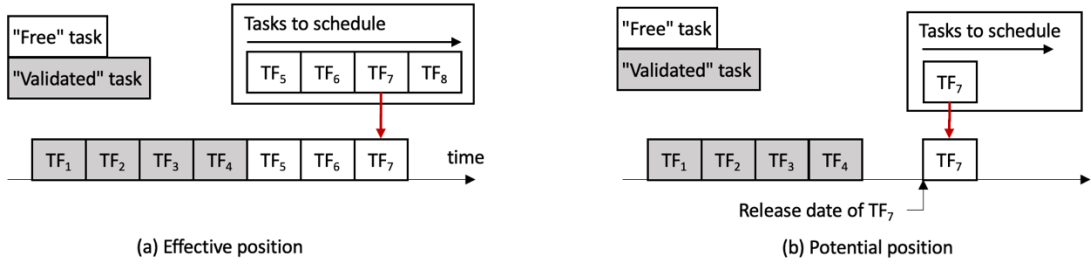


Figure 5: Example of an effective position (a) and a potential position (b)

- The PP is determined by considering only the “validated” tasks and by assuming that the other tasks on the list will be performed by other resources (Figure 5 (b)).

The **Supervisor agent** controls the access of all agents to the environment in a predefined sequence.

A **Customer agent**  $C_i$  (see Figure 6) corresponds to one manufacturing order  $MO_i$ . It is characterized by a release date  $R_i$ , a due date  $D_i$  and a sequence of manufacturing tasks to perform  $S_i = \{TF_{i,1}, TF_{i,2}, \dots, TF_{i,n}\}$ . Each production operation  $TF$  requires one machine functionality at least. All the  $TF$ s can be carried out by one machine, at least, when it is in good operating conditions but that will degrade while being used. The customer agent has requirements for the execution of its sequence of tasks to achieve the objectives assigned by the sales department. The customer agent verifies the order of the tasks so that no task starts before the previous one is completed.

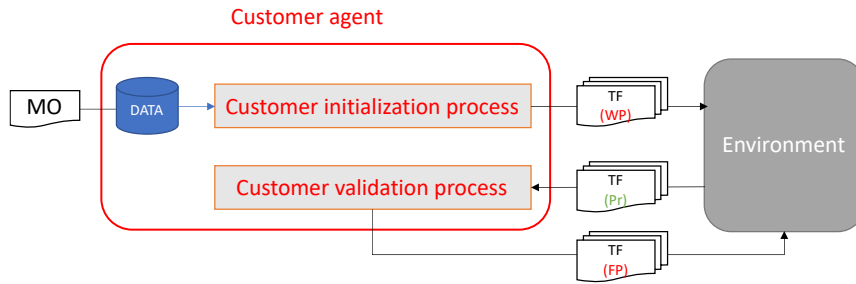


Figure 6: Customer agent description

A **Producer agent**  $P_j$  (see Figure 7) manages one machine  $m_j$  and has knowledge of all its devices (or components) and its attributes. This knowledge encompasses the machine competencies, its capabilities to perform the corresponding functionalities, its devices RULs and also its scheduling method. Each producer agent implements its scheduling method depending on the production system features and its optimization criteria. The scheduling method can be based, among many possible methods, on genetic algorithm [63], constraint satisfaction problems [64] or priority rules [65]. The producer agent ensures the feasibility of the planning. It verifies that no overlap of operations can appear. The producer agent also monitors its machine functionalities health states by calculating their probabilities of failure according to the results given by the PHM modules of the devices contributing to the functionalities required to perform the  $TF$ s to schedule. When the probability that at least one needed functionality fails while accomplishing the planned sequence of  $TF$ s is over the threshold  $p_s$ , the producer calls for maintenance tasks (TMs) for the devices that are identified by, e.g. the method proposed in [31]. Each producer agent can define its own value for  $p_s$ . Each TM is device-specific. Regarding the health status of the machine, the producer not only proposes a suitable position for each  $TF$  but also make the calls for maintenance activities aiming at its objectives.

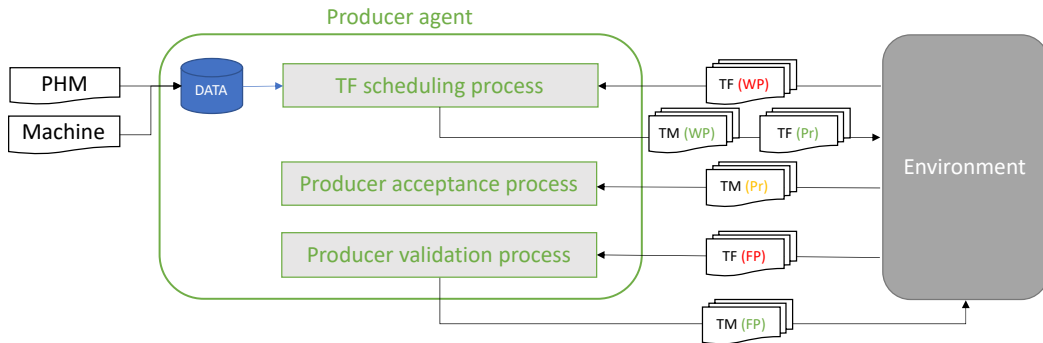


Figure 7: Producer agent description

A **Maintainer agent**  $M_k$  (see Figure 8) manages one maintenance resource  $m_k$ . A maintenance resource represents, for example, the maintenance team and/or special equipment that must be mobilized to carry out the maintenance task (TM). Each maintenance resource is modeled by a maintainer agent. The notion of competency of the maintainer agents is similar to the one of producer agents without being related to the notion of functionality but to the notion of skill. Each maintainer agent has a list of competencies which enable it to perform some of maintenance tasks needed

by producers with a specific speed (capability). Each maintainer agent has its own method for scheduling the maintenance tasks and its optimization criteria.

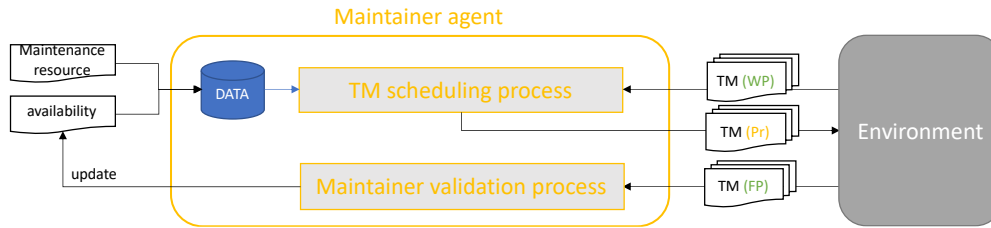


Figure 8: Maintainer agent description

## 4.2 Scheduling process

As previously stated, this framework needs the cooperation of the different agents. Because producers and maintainers operate on the same resources (the machines), conflicts often appear during the scheduling process. However, these conflicts can be managed by a negotiation process. The proposed negotiation process is based on the Contract-Net protocol using the concepts of tendering and auction [66]. The communication between the different agents is here a sequence of messages whose meanings are in relation with the involved agents. This sequence of messages must be structured according to a protocol. The UML sequence diagram of the Figure 9 illustrates the cooperation protocol between the different agents that the supervisor agent controls.

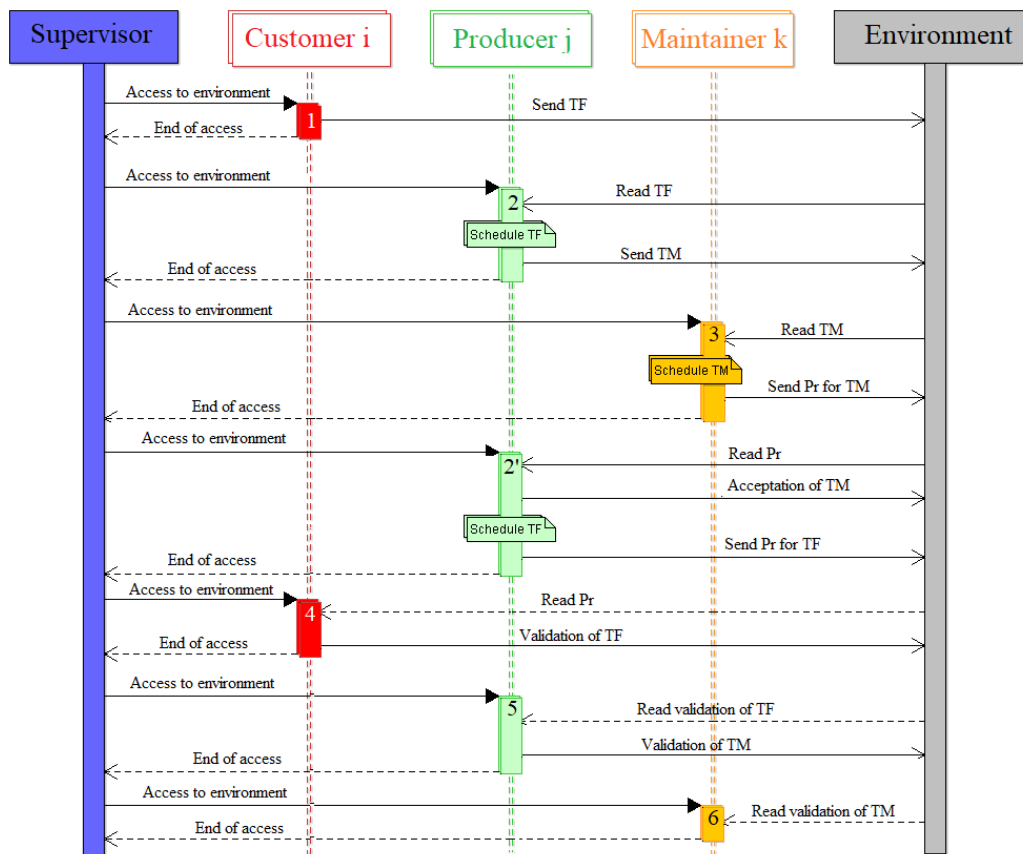


Figure 9: Sequence diagram of SCEMP framework

- Action 1 (Customer-red): at the beginning of the process, the supervisor agent instantiates and initializes the population of agents around the environment (blackboard). First, the supervisor gives the customers the access to the environment. Each customer plans the different operations of its sequence. Since the customer agents do not have the knowledge of the different machines capabilities to perform the TFs and the TFs already planned on them, only the precedence constraint is considered by the customers. This initialization process is described in section 4.2.1. Then each customer  $C_{i,v}$  associated to a  $MO_{i,v}$ , writes on the environment its WPs for each  $TF_{i,v}$  of its sequence ( $v$  is the position of the task in the sequence). The WP for each  $TF_{i,v}$  is given by a temporal interval  $WP_{i,v} = [r_{i,v}, r_{i,v} + p_{i,v}]$ , where  $r_{i,v}$  and  $p_{i,v}$  represent respectively the release date and the duration of the task  $TF_{i,v}$  that could be determined from the machine that has the best capability for  $TF_{i,v}$ .
- Action 2 (Producer-green): when all the customers have written their wishes, the supervisor agent dismisses them and gives access to the environment to the producers. Each producer agent reads the information about the unplanned TFs published on the blackboard and selects the operations that are compatible with its

functionalities and adjusts their durations according to their capabilities related to their competencies. The producer agent then starts the process of TFs scheduling described in section 4.2.2. This scheduling leads to the determination of their starting and ending dates and updates its machine health status. When the health status of one needed functionality of the machine leads to the inability to perform entirely the tasks it has planned under the probability of failure  $p_s$ , the producer agent sends a call for proposals to maintainers for each device needing maintenance it identifies (critical device). Therefore, new objects  $TM_{j,d}$  are added to the blackboard environment that corresponds to the resource (machine)  $m_j$  critical device  $d$ . At this point, the producer's scheduling process is suspended until it obtains proposals from maintainers.

- Action 3 (Maintainer-yellow): when the supervisor gets the signal that there is at least one unplanned TM in the environment, it activates the maintainer agents in order to get proposals for this unplanned task (i.e. action 3 begins). The initialization of maintainers is similar to the one of the producers, each maintainer agent selects from the blackboard the TMs that its maintenance resource is able to perform. It adjusts the durations of the TM according to its capabilities to perform them. It then inserts them into its list of tasks to schedule. After that, the maintainer agent runs its own scheduling method and assigns to each TM a position into its planning. By the end of this action, the maintainer writes on the blackboard its proposals for the TMs of its list. Once the maintainers have written their proposals on the blackboard, they no longer have access to it.
- Action 2' (Producer-green): when all maintainers have written their proposals, each unplanned maintenance operation  $TM_{j,d}$  on the blackboard gets a list of proposals  $Pr_{j,d,k}$  (proposal made by maintainer  $M_k$  to maintain the device  $d$  of the machine  $m_j$ ). The supervisor then activates the producers again. They so can accept one of the proposals. If there is no proposal corresponding to its needs, a producer may not accept any of them. In this case, it awaits the proposals that will be made at the end of the next iteration hoping to get better proposals the maintainers will define during the next sub-cycle (see section 4.2.4). Action 3 will be repeated if at least one TM is not accepted. The exchange of messages between the producers and maintainers ends when a proposal is accepted for each TM called during the current cycle. Once a producer has accepted a proposal for each TM it needs, it ends the scheduling of the remaining TFs. Each producer agent then writes its proposals for each  $TF_{i,v}$  of its list on the blackboard.
- Action 4 (Customer-red): The supervisor reactivates the customer agents to make them validate proposals for their unplanned TFs of their sequences. For each unplanned operation  $TF_{i,v}$  the corresponding customer agent  $C_i$  receives a list of proposals  $PL_{i,v}$ . Each proposal  $Pr_{i,v,j}$  is given by one producer agent  $P_j$ . Then the customer validation process begins. The customers compare the different proposals to their objectives, if there is one proposal that suits them, then the proposed position is validated for this TF. Otherwise, the TF remains invalidated while the customer agent is hoping for better proposals at the next cycle. Section 4.2.3 describes in detail the validation process. All customers then update their wishes for their remaining TFs.
- Action 5 (Producer-green): at this stage, the supervisor activates the producers to let them read the decisions made by the customers. If the proposal  $Pr_{i,v,j}$  is validated then the producer  $P_j$  will add the task  $TF_{i,v}$  into its planning. Naturally, if all the proposals given by a producer are not selected, then there will be no need to perform the TM launched by this producer during this cycle. Therefore, in this action, the producers execute the validation process (discussed in section 4.2.5) to decide whether an accepted TM should be held or dropped.
- Action 6 (Maintainer-yellow): at this stage, the maintainers check if their proposals have been validated by the producers in order to add them into their planning.

#### 4.2.1 Customers initialization process

Once a customer agent  $C_i$  is created, an initialization of all the operations of its  $MO_i$  routing is made. A job  $MO_i$  cannot start before a date  $R_i$  (the release date) and must ideally end at a date  $D_i$  (the due date). The agent calculates the release  $r_{i,v}$  and due date  $d_{i,v}$  for each operation (Figure 10), meaning that for each  $TF_{i,v}$  of the sequence an interval of the feasible time window  $[r_{i,v}, d_{i,v}]$  is calculated, noting that:  $r_{i,v} = R_i + \sum_{k=1}^{v-1} p_{i,k}$  and  $d_{i,v} = D_i - \sum_{k=v+1}^n p_{i,k}$ .

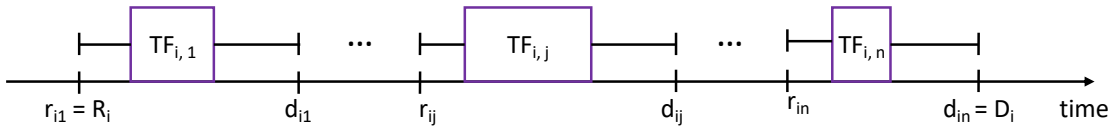


Figure 10: Customers initialization

#### 4.2.2 TF scheduling process

In this section we address the problem of scheduling several independent jobs on a single machine subject to predictive maintenance based on PHM. The use of PHM allows the estimation of machine functionality failure time while processing a production operation and enables prior planning for maintenance of the devices before they fail as mentioned earlier in section 4. Thus, maintenance orders and production activities can be scheduled simultaneously using this information about future availabilities of the machine functionalities. In this section, each producer agent

solves the sub-problem of joint scheduling on a single machine with availability constraint. We consider the following assumptions:

- Each TF has a release date before which it cannot be executed.
- Each TF has a set of required functionalities, and its process duration will depend on the competencies of the machine on which it will be executed.
- The machines are not continuously available due to predictive maintenance activities leading to TMs.
- Only one TF can be assigned to a machine at a time.
- We assume that the probability of failure of each device is known thanks to the features of its PDF provided by its prognostic system.

Whatever the scheduling method implemented in the producer agent is, it will not valid its sequence of planned tasks if its health status does not allow to achieve them within acceptable conditions with a probability higher than  $1 - p_s$ .

The algorithm carrying out the schedule of a producer  $P_j$  is shown in Figure 11.

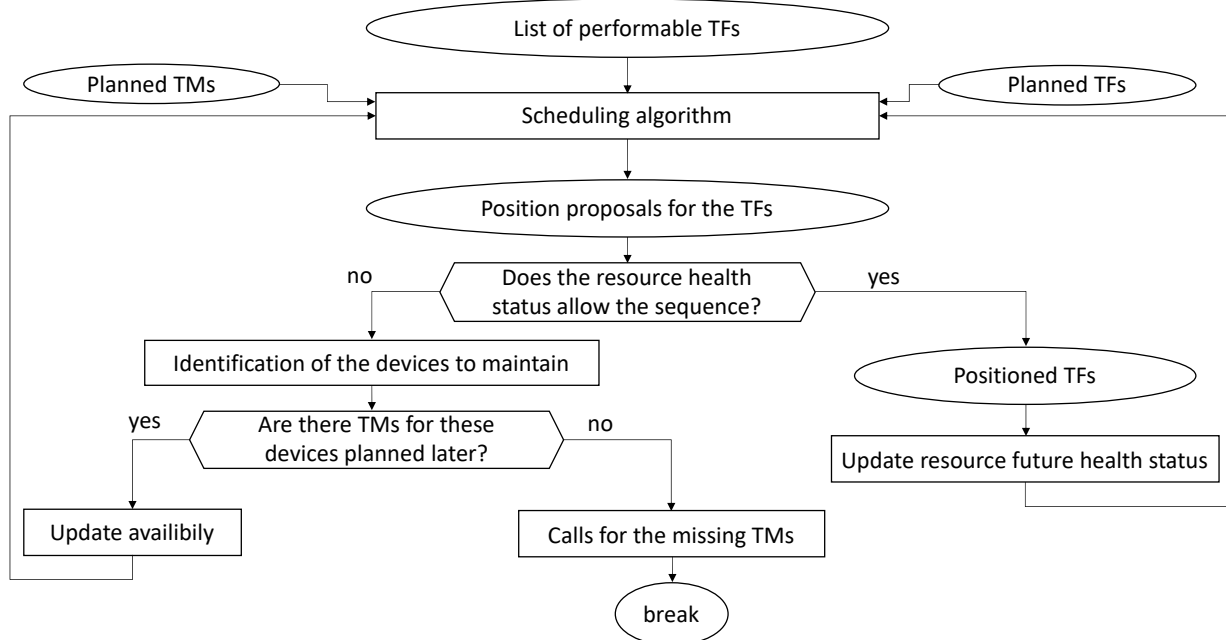


Figure 11: Producers scheduling process

From the list of TFs performable by the resource  $m_j$ , the producer agent  $P_j$  runs its scheduling method. Each time a producer schedules a sequence of TFs with their proposals (containing the effective and potential positions) while executing its planning process, it checks whether the resource  $m_j$  future health status will allow to achieve the sequence or not. Two cases can occur:

- The health status of the machine will allow to achieve the sequence. That means that the probability of failure of the required functionalities will not exceed  $p_s$  while carrying out the TFs. In this case, the effective and potential positions determined are assigned to the TFs.
- The health status of the machine will not allow to perform entirely the sequence of TFs under the risk threshold  $p_s$ . The devices whose maintenance will best improve the ability of the functionalities to carry the planned sequence of TFs out are then identified. If there are TMs already planned for such devices, then the producer tries to find other positions for the TFs that cannot be achieved, after the planned TMs and updates the availability of  $m_j$ . Otherwise, it makes calls for TMs for the devices requiring maintenance. In this case, the scheduling process stops till those TMs are planned.

#### 4.2.3 Customer validation process

Following the order of tasks in the sequence, the customer agent selects consecutively a final position FP for each of its unplanned TFs that is called the validation process. For each TF, the customer agent compares the different EPs to its WP:

- It validates the first proposal for which  $EP=WP$  ensuring the precedence constraint.
- If all the proposed EPs and PPs are after its WP and if the PPs are not better than the EPs in the proposals, the customer validates the proposal closest to its WP.
- If all the proposed EPs and PPs are after its WP and if there are PPs are better than the EPs in the proposals, the customer does not validate anything and waits for the next cycle hoping for better proposals.

The validation process ends when all the customer agents have studied their proposals. Let us note that one task, at least, has to be validated for each cycle to prevent dead end in the process. Forced validations may thus occur [52].

#### 4.2.4 Producer acceptance process

The acceptance process is similar to the customer validation process. For each TM, the producer agent compares the different EPs with each other. If all the EPs do not satisfy the objective and if there are better PPs, the agent does not accept any of the proposals given by the maintainer agents.

#### 4.2.5 Producer validation process

Once the customer agents have validated some TFs, the producer agents have to decide if they validate their accepted TMs or not. The producer agents validate their accepted TMs if the preceding TFs are validated. Otherwise, they cancel the TMs.

## 5 Computational experiments and results

Computational experiments have been carried out to test the behavior of the proposed joint scheduling approach. To our knowledge, there is not any industrial implementation of joint scheduling of production and predictive maintenance problem for a job shop context. That is why we have generated data sets for the tests. SCEMP has been implemented using the object-oriented C++ programming language and all tests have been performed on a computer with Core i5 @ 2.40 GHz and 8.00 GB RAM. The purpose is not to evaluate the configurability of SCEMP (e.g. different scheduling methods with criteria and constraints the producer and maintainer agents can implement, other contexts like flow shop, *etc.*) but to assess its ability to jointly plan production tasks in job shop context and the needed predictive maintenance activities to carry them out. The presented tests have been led with the following assumptions:

- the machines cannot be moved,
- the tasks TFs are exclusive and only one task can be performed at a time by a producer or a maintainer.
- several maintainers can work on a machine at the same time (an electrical maintainer can work in the machine's control cabinet while a mechanical maintainer replaces a bearing),
- the transport times of the parts between the machines are neglected (considering small and medium-sized workshops, travel times between machines are often very short compared to the time taken to complete production tasks),
- the travel times of the maintainers are neglected.

We have purposely used a general machine structure to keep the tests general and not domain specific. Two kinds of tests are detailed in this paper to show how SCEMP behaves and prove that it can be used in different environments. Without loss of generality, these tests were performed from simple case studies:

- a job shop scheduling problem of 10 to 100 MOs on 3 machines, to show SCEMP behavior considering different performance indicators.
- scheduling of 500 MOs with one TF on one machine having two functionalities, monitored with PHM modules or not, to compare predictive and systematic preventive maintenance strategies.

### 5.1 Evaluation of SCEMP behavior

The first illustrative example has been led to check the behavior of SCEMP in a job shop context with different machines that can perform different functionalities with different capabilities.

For that, three machines  $m_1$ ,  $m_2$  and  $m_3$  (Figure 12) have been instantiated. The machine  $m_1$  can perform two functionalities  $F_1$  and  $F_2$  corresponding respectively to competencies  $Comp_1$  and  $Comp_2$  with respective capabilities of 1.5 and 1. The functionality  $F_1$  is a simple function that requires only one simple function (a rhombus located under the function marked with an "and" indicates a simple function)  $f_1$  from the machine and it is composed of three devices  $Dv_1$ ,  $Dv_2$  and  $Dv_3$ , while  $F_2$  is a redundancy function (a rhombus under the function marked with an "or" indicates a first order redundancy) composed of two functions  $f_1$  and  $f_2$ . In  $m_1$  the odd numbered devices are subject to predictive maintenance (denoted PM) tasks and even numbered devices are subject to systematic preventive maintenance (denoted SM) tasks. The machine  $m_2$  can perform only one functionality corresponding to the competency  $Comp_3$  within a capability of 1.5. Its functionality is a simple function composed of two functions, a simple function  $f_1$  and a redundancy function  $f_2$ . The whole machine gathers four devices where the even numbered ones are subject to PM tasks and the odd numbered devices are subject to SM tasks. The machine  $m_3$  can perform two functionalities corresponding to competencies  $Comp_1$  and  $Comp_3$  with a same capability of 1. Both the functionalities are simple functions made of only one function. The function  $f_1$  is a simple function made of two devices  $Dv_1$  and  $Dv_2$  and the function  $f_2$  is a redundancy function that can use the device  $Dv_2$  (shared with  $f_1$ ) and device  $Dv_3$ . Only the device  $Dv_2$  is subject to PM tasks, while the other two devices are subject to SM tasks.

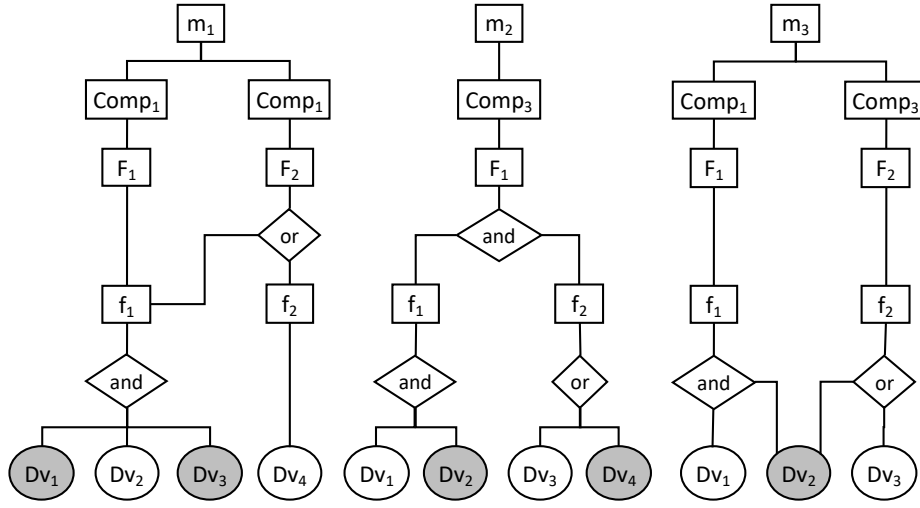


Figure 12: Machines structures

In the maintenance center, we consider two maintainers having only one competency each allowing to repair only one type of devices. The maintainer  $M_1$  can repair all the even numbered devices and maintainer  $M_2$  can repair all odd numbered devices.

In this example, 10 instances of several MOs data have been generated according to a uniform random law. The number  $n$  of MOs varies from 10 to 100, with a step of 10 MOs between each instance. The routing size  $rs$  of each MO is ranging from 5 to 10 TFs ( $rs \in [5,10]$ );  $rs$  is determined from a uniform random law. Each TF is generated randomly:

- Its duration can vary uniformly from 1 to 10 TU (Time Unit).
- It can require  $Comp_1$ ,  $Comp_2$ ,  $Comp_3$ ,  $Comp_1$  and  $Comp_2$ , or,  $Comp_1$  and  $Comp_3$ .

Two sets of data are generated to observe the behavior of SCEMP considering the complexity of the problem. The release dates  $R$  are generated randomly such as  $R \in [0, 10 \times k_i]$  where  $k_i$  is an integer determined from a uniform random law such as  $k_1 \in [0, 100]$  and  $k_2 \in [0, 50]$ . The due dates  $D$  are determined from  $D = R + 10 \times rs$  since the maximum duration of a TF is 10 TU. Examples of random generations of instances are shown in Figure 13 where only the release dates are displayed.

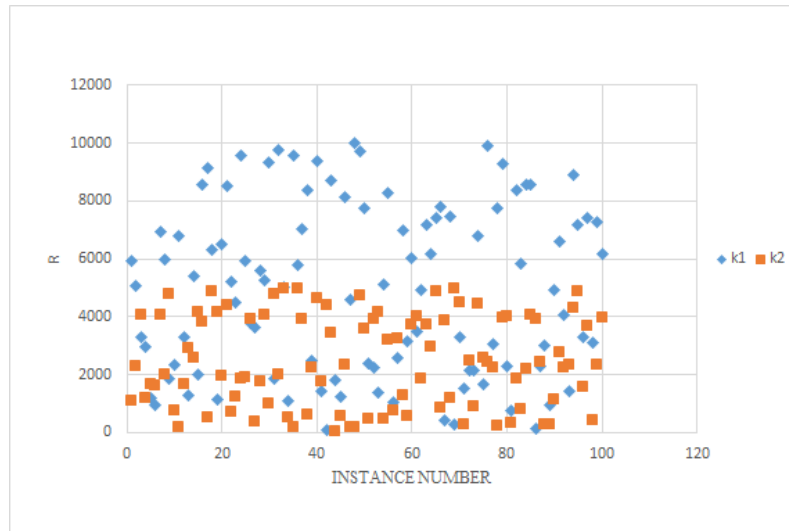


Figure 13 : Release dates of MOs of each set of data

In this case study, several assumptions are done:

- At the beginning of the process, the agents (producers and maintainers) do not have any TF or TM assigned (or planned) and the durations of use of all the devices are equal to 0.
- All the agents (producers and maintainers) schedule their tasks from the FIFO (First In First Out) priority rule.
- The durations of all the TMs are equal to 5 TU.
- All devices subject to SM tasks have a MTTF of 100 TU. Their mean failure rates are therefore  $\lambda_m = 100$ . These devices follow a normal probability law  $N_s(\lambda_s = \lambda_m, \sigma_s = \lambda_m/10)$  where  $\sigma_s$  is the standard deviation. The probability of failure of each of these devices  $P_f(t)$  is computed from the equation (3) where  $t_s$  is the

duration of use of the device since its last maintenance. Severities of TFs are not considered in the examples presented in this paper.

$$P_{fs}(t) = \frac{1}{\sigma_s \sqrt{2\pi}} \cdot \int_0^{t_s} \exp \left[ \frac{-1}{2\sigma_s^2} \cdot (u - \lambda_s)^2 \right] \cdot du \quad (3)$$

- Each device subject to PM is monitored by a PHM module providing the probability distribution function of its failure according to its duration of use since its last maintenance. This probability distribution function is also a normal probability law  $N_p(\lambda_p, \sigma_p)$ . At the initialization stage and after each maintenance of those devices, their values  $\lambda_p$  are set randomly from the distribution  $N_s$  and  $\sigma_p$  are set to a constant value  $\sigma_p = \sigma_s/5$ . The probabilities of failure of each of these devices  $P_{fs}(t)$  are computed from the equation (4). Thus, the distribution  $N_p$  of such a device changes after each maintenance.

$$P_{fp}(t) = \frac{1}{\sigma_p \sqrt{2\pi}} \cdot \int_0^{t_p} \exp \left[ \frac{-1}{2\sigma_p^2} \cdot (u - \lambda_p)^2 \right] \cdot du \quad (4)$$

- For all the machines, the failure thresholds  $p_s$  of their functionalities are fixed at 1%; i.e. the TFs cannot be achieved when the probability of failure of their functionalities exceed  $p_s$ . The producer agent will make calls for maintenance for the devices identified thanks to the method proposed in [31].
- At the initialization stage and after maintenance actions the devices are considered as good as new, i.e. their durations of use after their last maintenance is zero.

At the end of the SCEMP program execution, a Gantt chart is obtained in which the production and maintenance tasks are planned without overlap, as it is shown in Figure 14. The production tasks are represented by lines and the maintenance tasks are represented by rectangles containing the ID of the TM.

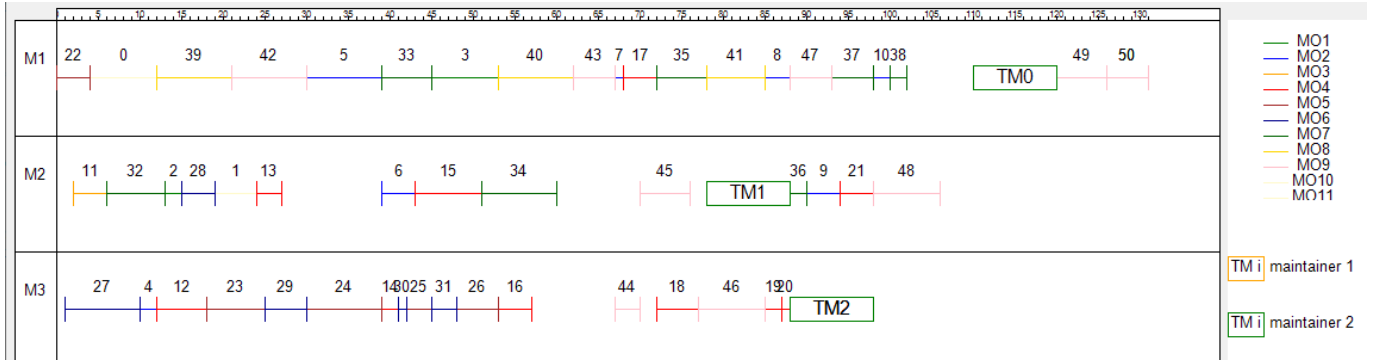


Figure 14 : Gantt Chart

Different performance indicators have been chosen to evaluate SCEMP:

- the execution time  $E$  in seconds,
- the number of cycles necessary to reach a solution  $nC$ ,
- the total number of maintenance tasks  $nTM$ ,
- the number of late jobs  $nR$ ,
- the total job tardiness  $R$  in TU,
- the load of each machine  $L$  in TU.

Each instance (from 10 to 100 MOs) has been generated randomly 10 times. The average results are shown in Table 1. The number of cycles needed to plan all the MOs and the total job tardiness increases considerably when the number of MOs becomes high but remains acceptable comparing to the complexity of the problem and can be considered as relatively good for industrial managers. Indeed, according to the results presented in Table 1 the highest computation time is about 5 minutes for a case that can correspond to the scheduling of a mechanical manufacturing cell of three machines on a one- or two-day horizon. All the tests have been solved and we have not been confronted to the case where SCEMP does not converge. We can see that SCEMP guarantees a good load balance of the machines, although some machines have better capabilities allowing them to perform the operations quicker than the others. The machine  $m_2$  is the less solicited one because it can perform only one functionality. This is because its list of performable TFs contains fewer tasks than the ones of the two other machines having two functionalities. However, we notice from Table 1 that both the number of cycles ( $nC$ ) and the execution time ( $E$ ) increase considerably when the problem becomes more complex. This means that SCEMP is not polynomial and its execution time highly depends on the jobs of the considered instance. The execution time depends strongly on the number of cycles needed to validate all the production tasks, whereas since each producer schedules all the production tasks listed in its list of performable TFs, most of the tasks will not be validated and therefore they will be rescheduled during the next cycle. A planning method that limits the number of tasks in each cycle is under investigation to avoid these unnecessary reschedules.



This will considerably reduce the computation time of each producer and consequently the total execution time of the program.

Table 1 – Results of joint scheduling on 3 machines using SCEMP

$nMO$	$E(s)$		$nC$		$nTM$		$nR$		$R(TU)$		$L_1(TU)$	$L_2(TU)$	$L_3(TU)$
	$k_1$	$k_2$	$k_1$	$k_2$	$k_1$	$k_2$	$k_1$	$k_2$	$k_1$	$k_2$			
10	0.52	1.02	16.6	32	7.2	6.7	2.2	4	13.35	62,6	161.68	40.13	183.30
20	3.25	6.08	21.5	48,3	19.6	15	8	8,6	81.65	213,7	330.70	122.55	382.00
30	7.68	19.33	24.9	70,4	37.8	21	12.2	17,2	168.1	672,5	498.95	230.55	490.50
40	14.63	24.43	30.3	49,7	40.8	43.5	11.7	13,8	123.05	318,8	653.70	209.40	630.10
50	20.95	78.32	28.4	60,7	50.9	60.7	16.7	27,3	189.35	526,4	923.98	255.98	799.70
60	38.25	82.80	34.3	45,4	64.3	77.3	13.5	26,6	124.1	361,85	854.63	375.90	1042.65
70	53.63	163.95	34	64,2	72.6	82	20	28,1	272.6	511,85	1133.53	415.43	1119.70
80	92.80	217.49	39.2	61,5	91.6	111.6	26.3	42	352.2	616,2	1340.98	514.95	1259.05
90	106.26	268.47	33.7	71,1	106.2	115.8	25.6	38,4	366.5	819,4	1369.63	508.05	1522.55
100	128.56	282.29	38.4	59	112.5	126.6	15.7	45,1	145.05	917,45	1592.35	557.93	1587.65

## 5.2 Systematic preventive maintenance vs Predictive Maintenance

This section presents a use case to show how SCEMP can be exploited to compare the implementations of the predictive maintenance policy, denoted PM, and of the systematic preventive maintenance policy, denoted SM, and to evaluate the interests of using the PHM module results provided from degradation observations and extrapolation techniques. For that, we have considered two situations with an identical machine for which the TFs are the same. In the first situation, the machine is subject to SM tasks and in the second one it is subject to PM tasks. The machine structure is shown in Figure 15, it can perform two types of functionalities  $F_1$  and  $F_2$  respectively corresponding to two competencies  $Comp_1$  and  $Comp_2$ . Both functionalities require one function of the machine, each function is a simple function and requires two devices having one common device  $Dv_2$ . For both situations, the producer agent has a scheduling method based on a FIFO priority rule, no TF are pre-planned and the devices are assumed to be as good as new, with a duration of use of zero. For both situations, two maintainer agents are considered, with a scheduling method based on FIFO priority rule. The first maintainer agent has the skill, so the competency, to maintain devices with an odd number ( $Dv_1$  and  $Dv_3$ ) whereas the second maintainer agent has the skill, so the competency, to maintain devices with an even number ( $Dv_2$ ). The devices have a mean time to failure of 100 TU (Time Unit), the mean failure rate is therefore  $\lambda_m = 100$ . 100 replicates of this instance have been created, each being scheduled 4 times, with the following maintenance strategies.

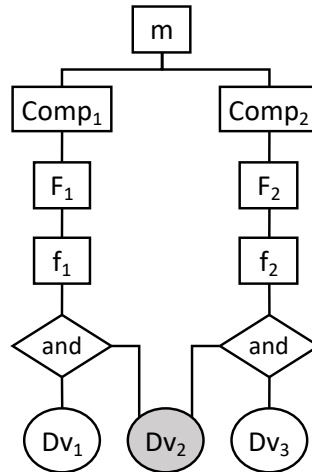


Figure 15: Machine structure

In the first situation, the devices of the machine, subject to SM tasks, have all a failure rate that follows a normal probability law  $N_s(\lambda_s = \lambda_m, \sigma_s = \lambda_m/10)$ .

As Vogl et al. wrote in [6]: “Perhaps the main challenge for prognostics is that there will always be a limit to the accuracy and precision of condition-based estimation of RUL due to the inherent uncertainty of predicting the future”. Thus, it is interesting to assess the gain that can be obtained thanks to the accuracy improvement of prognostic modules compared to statistical data that device manufacturers can give and also dealing with their MTTF or MTBF definitions. Works on device prognoses also aim to quantify the uncertainty of their forecasts [67][68]. That is why, in the second situation, the devices of the machine, subject to PM tasks, are supposed to be monitored by a PHM module providing the probability distribution function corresponding to the new state of the device which is considered in this



test as a normal probability law too  $N_p(\lambda_p, \sigma_p)$ , where  $\sigma_p \leq \sigma_s$ . To show how SCEMP can be used to assess how improving the accuracy of component prognoses can improve production and maintenance performance, we conducted three sets of tests with different values of  $\sigma_p$ :

- $\sigma_p = \sigma_s$ , in this case the information provided from PHM is at most as precise as the information used in SM,
- $\sigma_p = \sigma_s/2$ , in this case the information provided from PHM module is two times more precise than the information used in SM,
- $\sigma_p = \sigma_s/5$ , in this case the information provided from PHM module is five times more precise than the information used in SM.

For each device, the value of  $\lambda_p$  is set randomly from the distribution  $N_s$  at its initialization and after each of its maintenances. For each device, its probability of failure  $P_{ip}(t)$  at a time  $t$  is computed from equation (4) where  $t_p$  is the duration the device is used since its last maintenance:

At the machine level, the failure threshold of the functionalities is fixed at a probability of 1% and if the production cannot be achieved under a probability of 0.01, the producer agent will make calls for maintenances of devices it will identify thanks to the method proposed in [31].

The problem is to schedule a set of TFs. Each TF requires one of the two functionalities of the machine that are randomly defined equiprobably. Several data sets of 500 production tasks were generated randomly. The duration of the TFs of those sets were also set randomly among the equiprobable discrete values 1, 2, ..., 10 TU. The maintenance tasks TMs have a duration of 10 TU.

To compare the maintenance strategies, the criteria of the total number  $n_{TM}$  of maintenance operations planned for each machine and mean duration  $T$  of use of the three devices of each machine have been chosen to evaluate the quality of the planning of the two machines. The results of the simulations are given in Table 2 and Table 3.

Table 2 – Comparison of use of devices

	<i>SM</i>	<i>PHM(<math>\sigma_p = \sigma_s</math>)</i>	<i>PHM(<math>\sigma_p = \sigma/2</math>)</i>	<i>PHM(<math>\sigma_p = \sigma/5</math>)</i>
<i>Mean time (in TU)</i>	74.06	73.72	85.00	92.20
<i>Standard deviation</i>	0.30	2.70	2.74	3.59
<i>Median time (in TU)</i>	74.09	73.49	84.99	91.74

Table 3 – Comparison of the numbers of TMs

	<i>SM</i>	<i>PHM(<math>\sigma_p = \sigma_s</math>)</i>	<i>PHM(<math>\sigma_p = \sigma/2</math>)</i>	<i>PHM(<math>\sigma_p = \sigma/5</math>)</i>
<i>Mean number of TMs</i>	72.63	73.23	63,15	58.19
<i>Standard deviation</i>	1.80	3.08	2.61	2.72
<i>Median number of TMs</i>	72.50	73.00	63.00	58.00

We can notice on Table 3 that the total number of planned maintenance tasks are nearly identical in the case of systematic preventive maintenance and the first case of predictive maintenance ( $\sigma_p = \sigma_s$ ). This number is remarkably smaller in the second ( $\sigma_p = \sigma_s/2$ ) and third ( $\sigma_p = \sigma_s/5$ ) cases of predictive maintenance.

As we can see from Table 2, the mean time of use of the devices of the machines subject to predictive maintenance are higher than the mean time of use of the device of the machine subject to systematic maintenance except for the case where  $\sigma_p = \sigma_s$ . In addition to that, the total number of maintenance tasks planned on the machines and the mean duration of use of the devices decrease while the precision of the information provided by PHM modules increases. These results tend to show that maintenance performances are improved with the implementation of effective PHM modules. In this case study, the implementation of PHM modules is useless, if they do not lead to reduce the uncertainty about the RULs. Nevertheless, it also shows the relevance of implementing accurate PHM modules, since they allow to reduce the number of planned maintenance tasks while insuring at the same time the availability of the machine. This allows us to state that the use of PHM modules results can be a key stone for a better joint scheduling of production and maintenance operations.

SCEMP was tested in a job shop context under the same conditions as the one described in section 5.1 with more machines and so more producer agents (till 100 by replicating the three machines presented in Figure 12), with more MOs (made of 5 to 10 TFs) and so more customer agents (till 10000) and only two maintainer agents. In this situation, scheduling of TFs and TMs is obtained in about 4 hours. This computation time can be considered acceptable relatively to the scheduling of a mechanical manufacturing workshop for a horizon of one or two weeks and for which maintenance resources are certainly undersized (only two maintainers for one hundred machines). Other tests have been led by implementing different scheduling methods in producer agents (priority rules and the genetic algorithm proposed in [63]). During all these tests, SCEMP always converged to a scheduling of TFs and TMs. These tests lead

us to state that SCEMP seems to be able to schedule complex problems (more MOs, more machines, different scheduling methods for each producer/maintainer agent, complex functionalities requiring numerous devices).

## 6 Conclusion

In this paper we have proposed SCEMP, a multi-agent system framework, to solve the joint scheduling of production and maintenance operations on machines subject to predictive maintenance and/or to preventive maintenance. The machines are complex systems gathering several devices which are monitored by PHM modules providing PDFs of their RULs. The framework SCEMP is made of several customer agents, producer agents and maintainer agents which collaborate with each other through a blackboard environment onto which all agents write their demands and proposals. Collaboration between agents involves broadcasting calls for proposals for production and maintenance tasks, then comparing the bids received to award a contract to the agent who offers the best bid. Customer agents write their manufacturing orders on the blackboard then the producer agents schedule the production tasks according to their machine health states and call for the maintenance tasks when their probabilities of failure become critical. The maintenance operations are performed by maintainer agents who schedule the maintenance tasks according to their availabilities. SCEMP shows its ability to schedule job shops with machines subject to predictive maintenance which does not seem to have been addressed before. As shown in section 5.2, SCEMP can also be used to test the opportunity to implement PHM modules for devices according to their expected accuracies, to production resources, to production tasks and to maintenance resources and tasks. The execution time of SCEMP is relatively reasonable for managers. It can be adapted to different situations, moreover, as mentioned in section 5.1 a new planning method limiting the number of tasks of each producer is under investigation which will allow to reduce considerably this computation time. SCEMP is flexible and offers to the different agents the possibility to have their own structure, constraints, objectives, scheduling method and decision making ability. Despite the advantages of SCEMP, we have noticed that for machines with many components, many TMs are launched and the gap between them is not very large, which means that machines are often stopped for maintenance. To overcome this issue, strategies for grouping maintenance tasks are under investigations.

Other investigations will extend the SCEMP system to consider other types of production systems for which maintenance tasks are done by distant maintenance centers for which transport time of maintenance resources must be considered. In such context, the maintainer agents would certainly aim at grouping maintenance tasks to do in a same production center. In another context, like transportation systems, the productive resources (trucks, trains, aircrafts...) must go to maintenance centers. In such context, the producer agent would certainly aim to group maintenance of the system devices so that its health state enables it to function properly in the long term (e.g. several weeks or months). Eventually, SCEMP should also be compared to methods in the literature that also allow joint planning of production and predictive maintenance operations. However, those we have identified are more context specific. Their specificity may make them more effective in their context than SCEMP, which aims at being more generic. Those methods may also be less effective than SCEMP for other contexts.

## 7 References

- [1] J. Lee, B. Bagheri, and H.-A. Kao, 'A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems', *Manuf. Lett.*, vol. 3, pp. 18–23, Jan. 2015, doi: 10.1016/j.mfglet.2014.12.001.
- [2] D. Mourtzis and E. Vlachou, 'A cloud-based cyber-physical system for adaptive shop-floor scheduling and condition-based maintenance', *J. Manuf. Syst.*, vol. 47, pp. 179–198, Apr. 2018, doi: 10.1016/j.jmsy.2018.05.008.
- [3] O. Cardin, F. Ounnar, A. Thomas, and D. Trentesaux, 'Future Industrial Systems: Best Practices of the Intelligent Manufacturing and Services Systems (IMS2) French Research Group', *IEEE Trans. Ind. Inform.*, vol. 13, no. 2, pp. 704–713, Apr. 2017, doi: 10.1109/TII.2016.2605624.
- [4] T. P. Carvalho, F. A. A. M. N. Soares, R. Vita, R. da P. Francisco, J. P. Basto, and S. G. S. Alcalá, 'A systematic literature review of machine learning methods applied to predictive maintenance', *Comput. Ind. Eng.*, vol. 137, p. 106024, 2019.
- [5] B. Nikolic, J. Ignjatic, N. Suzic, B. Stevanov, and A. Rikalovic, 'Predictive Manufacturing Systems in Industry 4.0: Trends, Benefits and Challenges', in *DAAAM Proceedings*, 1st ed., vol. 1, B. Katalinic, Ed. DAAAM International Vienna, 2017, pp. 0796–0802. doi: 10.2507/28th.daaam.proceedings.112.
- [6] G. W. Vogl, B. A. Weiss, and M. Helu, 'A review of diagnostic and prognostic capabilities and best practices for manufacturing', *J. Intell. Manuf.*, vol. 30, no. 1, pp. 79–95, Jan. 2019, doi: 10.1007/s10845-016-1228-8.
- [7] P. Zheng *et al.*, 'Smart manufacturing systems for Industry 4.0: Conceptual framework, scenarios, and future perspectives', *Front. Mech. Eng.*, vol. 13, no. 2, pp. 137–150, 2018.
- [8] G. Vachtsevanos, F. Lewis, M. Roemer, A. Hess, and B. Wu, *Intelligent fault diagnosis and prognosis for engineering systems*, Wiley. Hoboken, New Jersey: John Wiley & Sons, Inc., 2006.
- [9] A. Ladj, F. B.-S. Tayeb, and C. Varnier, 'An integrated prognostic based hybrid genetic-immune algorithm for scheduling jobs and predictive maintenance', in *Evolutionary Computation (CEC), 2016 IEEE Congress on*,

- 2016, pp. 2083–2089.
- [10] J. P. Kenné, A. Gharbi, and M. Beit, ‘Age-dependent production planning and maintenance strategies in unreliable manufacturing systems with lost sale’, *Eur. J. Oper. Res.*, vol. 178, no. 2, pp. 408–420, 2007.
- [11] D. Pandey, M. S. Kulkarni, and P. Vrat, ‘Joint consideration of production scheduling, maintenance and quality policies: a review and conceptual framework’, *Int. J. Adv. Oper. Manag.*, vol. 2, no. 1–2, pp. 1–24, 2010.
- [12] J. Kaabi, C. Varnier, and N. Zerhoni, ‘Heuristics for scheduling maintenance and production on a single machine’, in *Systems, Man and Cybernetics, 2002 IEEE International Conference on*, 2002, vol. 5.
- [13] W. Pujadas and F. F. Chen, ‘A reliability centered maintenance strategy for a discrete part manufacturing facility’, *Comput. Ind. Eng.*, vol. 31, no. 1–2, pp. 241–244, 1996.
- [14] X. Qi, T. Chen, and F. Tu, ‘Scheduling the maintenance on a single machine’, *J. Oper. Res. Soc.*, vol. 50, no. 10, pp. 1071–1078, 1999.
- [15] N. Fnaiech, C. Fitouri, C. Varnier, F. Fnaiech, and N. Zerhouni, ‘A New Heuristic Method for Solving Joint Job Shop Scheduling of Production and Maintenance’, *IFAC-Pap.*, vol. 48, no. 3, pp. 1802–1808, Jan. 2015, doi: 10.1016/j.ifacol.2015.06.348.
- [16] K. Kang and V. Subramaniam, ‘Integrated control policy of production and preventive maintenance for a deteriorating manufacturing system’, *Comput. Ind. Eng.*, vol. 118, pp. 266–277, 2018.
- [17] S. Zhao, L. Wang, and Y. Zheng, ‘Integrating production planning and maintenance: an iterative method’, *Ind. Manag. Data Syst.*, vol. 114, no. 2, pp. 162–182, 2014.
- [18] S. Wang and M. Liu, ‘Multi-objective optimization of parallel machine scheduling integrated with multi-resources preventive maintenance planning’, *J. Manuf. Syst.*, vol. 37, pp. 182–192, Oct. 2015, doi: 10.1016/j.jmsy.2015.07.002.
- [19] Z. Zhang, Q. Tang, and M. Chica, ‘Maintenance costs and makespan minimization for assembly permutation flow shop scheduling by considering preventive and corrective maintenance’, *J. Manuf. Syst.*, vol. 59, pp. 549–564, Apr. 2021, doi: 10.1016/j.jmsy.2021.03.020.
- [20] C. Fitouri, N. Fnaiech, C. Varnier, F. Fnaiech, and N. Zerhouni, ‘A decision-making approach for job shop scheduling with job depending degradation and predictive maintenance’, *IFAC-Pap.*, vol. 49, no. 12, pp. 1490–1495, 2016.
- [21] K. Zhu, ‘A Joint Optimization Model of Production Scheduling and Maintenance Based on Data Driven for a Parallel-Series Production Line’, *J. Math.*, vol. 2021, p. e7588559, Sep. 2021, doi: 10.1155/2021/7588559.
- [22] T. Zonta, C. A. da Costa, F. A. Zeiser, G. de Oliveira Ramos, R. Kunst, and R. da Rosa Righi, ‘A predictive maintenance model for optimizing production schedule using deep neural networks’, *J. Manuf. Syst.*, vol. 62, pp. 450–462, Jan. 2022, doi: 10.1016/j.jmsy.2021.12.013.
- [23] S. Zhai, M. G. Kandemir, and G. Reinhart, ‘Predictive maintenance integrated production scheduling by applying deep generative prognostics models: approach, formulation and solution’, *Prod. Eng.*, vol. 16, no. 1, pp. 65–88, Feb. 2022, doi: 10.1007/s11740-021-01064-0.
- [24] S. Ferreira, A. Arnaiz, B. Sierra, and I. Irigoien, ‘Application of Bayesian networks in prognostics for a new Integrated Vehicle Health Management concept’, *Expert Syst. Appl.*, vol. 39, no. 7, pp. 6402–6418, Jun. 2012, doi: 10.1016/j.eswa.2011.12.027.
- [25] S. Zhai, B. Gehring, and G. Reinhart, ‘Enabling predictive maintenance integrated production scheduling by operation-specific health prognostics with generative deep learning’, *J. Manuf. Syst.*, Mar. 2021, doi: 10.1016/j.jmsy.2021.02.006.
- [26] O. Bougacha, C. Varnier, and N. Zerhouni, ‘A Review of Post-Prognostics Decision-Making in Prognostics and Health Management’, *Int. J. Progn. Health Manag.*, vol. 11, no. 15, p. 31, 2020.
- [27] M. Ghaleb, S. Taghipour, M. Sharifi, and H. Zolfagharinia, ‘Integrated production and maintenance scheduling for a single degrading machine with deterioration-based failures’, *Comput. Ind. Eng.*, vol. 143, p. 106432, 2020.
- [28] Q. Liu, M. Dong, F. F. Chen, W. Lv, and C. Ye, ‘Single-machine-based joint optimization of predictive maintenance planning and production scheduling’, *Robot. Comput.-Integr. Manuf.*, vol. 55, pp. 173–182, 2019.
- [29] P. Rokhforoz and O. Fink, ‘Distributed joint dynamic maintenance and production scheduling in manufacturing systems: Framework based on model predictive control and Benders decomposition’, *J. Manuf. Syst.*, vol. 59, pp. 596–606, Apr. 2021, doi: 10.1016/j.jmsy.2021.04.010.
- [30] A. Ladj, C. Varnier, F. B. S. Tayeb, and N. Zerhouni, ‘Exact and heuristic algorithms for post prognostic decision in a single multifunctional machine’, *Int. J. Progn. Health Manag.*, vol. 8, no. 2, pp. 1–15, 2017.
- [31] X. Desforges, M. Diévar, and B. Archimède, ‘A prognostic function for complex systems to support production and maintenance co-operative planning based on an extension of object oriented Bayesian networks’, *Comput. Ind.*, vol. 86, pp. 34–51, Apr. 2017, doi: 10.1016/j.compind.2017.01.002.
- [32] E. L. M. González, X. Desforges, and B. Archimède, ‘Assessment method of the multicomponent systems future ability to achieve productive tasks from local prognoses’, *Reliab. Eng. Syst. Saf.*, vol. 180, pp. 403–415, 2018.
- [33] A. Ladj, F. B.-S. Tayeb, and C. Varnier, ‘Tailored Genetic Algorithm for Scheduling Jobs and Predictive

- Maintenance in a Permutation Flowshop’, in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2018, vol. 1, pp. 524–531.
- [34] A. Ladj, F. B.-S. Tayeb, C. Varnier, A. A. Dridi, and N. Selmane, ‘A Hybrid of Variable Neighbor Search and Fuzzy Logic for the permutation flowshop scheduling problem with predictive maintenance’, *Procedia Comput. Sci.*, vol. 112, pp. 663–672, 2017.
- [35] A. Berrichi, L. Amodeo, F. Yalaoui, E. Châtelet, and M. Mezghiche, ‘Bi-objective optimization algorithms for joint production and maintenance scheduling: application to the parallel machine problem’, *J. Intell. Manuf.*, vol. 20, no. 4, p. 389, 2009.
- [36] L. Shen, D. Wang, and X.-Y. Wang, ‘Parallel-machine scheduling with non-simultaneous machine available time’, *Appl. Math. Model.*, vol. 37, no. 7, pp. 5227–5232, 2013.
- [37] M. A. Abdel-Aal and S. Z. Selim, ‘Robust optimization for selective newsvendor problem with uncertain demand’, *Comput. Ind. Eng.*, vol. 135, pp. 838–854, 2019.
- [38] F. Gehlhoff and A. Fay, ‘On agent-based decentralized and integrated scheduling for small-scale manufacturing’, - *Autom.*, vol. 68, no. 1, pp. 15–31, Jan. 2020, doi: 10.1515/auto-2019-0105.
- [39] M. E. Leusin, M. Kück, E. M. Frazzon, M. U. Maldonado, and M. Freitag, ‘Potential of a multi-agent system approach for production control in smart factories’, *IFAC-Pap.*, vol. 51, no. 11, pp. 1459–1464, 2018.
- [40] P. Rokhforoz, B. Gjorgiev, G. Sansavini, and O. Fink, ‘Multi-agent maintenance scheduling based on the coordination between central operator and decentralized producers in an electricity market’, *Reliab. Eng. Syst. Saf.*, vol. 210, p. 107495, 2021.
- [41] L. Shi, G. Guo, and X. Song, ‘Multi-agent based dynamic scheduling optimisation of the sustainable hybrid flow shop in a ubiquitous environment’, *Int. J. Prod. Res.*, vol. 59, no. 2, pp. 576–597, 2021.
- [42] L. Wang, H.-Y. Feng, and N. Cai, ‘Architecture design for distributed process planning’, *J. Manuf. Syst.*, vol. 22, no. 2, pp. 99–115, Jan. 2003, doi: 10.1016/S0278-6125(03)90008-2.
- [43] W. Xiang and H. P. Lee, ‘Ant colony intelligence in multi-agent dynamic manufacturing scheduling’, *Eng. Appl. Artif. Intell.*, vol. 21, no. 1, pp. 73–85, 2008.
- [44] M. Wooldridge and N. R. Jennings, ‘Intelligent agents: Theory and practice’, *Knowl. Eng. Rev.*, vol. 10, no. 2, pp. 115–152, 1995.
- [45] N. R. Jennings and M. Wooldridge, ‘Applying agent technology’, *Appl. Artif. Intell. Int. J.*, vol. 9, no. 4, pp. 357–369, 1995.
- [46] N. K. C. Krothapalli and A. V. Deshmukh, ‘Design of negotiation protocols for multi-agent manufacturing systems’, *Int. J. Prod. Res.*, vol. 37, no. 7, pp. 1601–1624, 1999.
- [47] R. G. Smith, ‘The contract net protocol: High-level communication and control in a distributed problem solver’, *IEEE Trans. Comput.*, no. 12, pp. 1104–1113, 1980.
- [48] L. Monostori, J. Váncza, and S. R. Kumara, ‘Agent-based systems for manufacturing’, *CIRP Ann.*, vol. 55, no. 2, pp. 697–720, 2006.
- [49] T. Sandholm, ‘An implementation of the contract net protocol based on marginal cost calculations’, in *AAAI*, 1993, vol. 93, pp. 256–262.
- [50] B. Archimède and T. Coudert, ‘A multi-agent scheduling approach for the flexible manufacturing production systems’, *IFAC Proc. Vol.*, vol. 31, no. 32, pp. 143–148, 1998.
- [51] B. Archimède, A. Letouzey, M. A. Memon, and J. Xu, ‘Towards a distributed multi-agent framework for shared resources scheduling’, *J. Intell. Manuf.*, vol. 25, no. 5, pp. 1077–1087, Oct. 2014, doi: 10.1007/s10845-013-0748-8.
- [52] B. Archimède and T. Coudert, ‘Reactive scheduling using a multi-agent model: the SCEP framework’, *Eng. Appl. Artif. Intell.*, vol. 14, no. 5, pp. 667–683, Oct. 2001, doi: 10.1016/S0952-1976(01)00025-2.
- [53] K. Ishak, B. Archimède, and P. Charbonnaud, ‘SCEP-SOA: An applicative architecture to enhance interoperability in multi-site planning’, in *OTM Confederated International Conferences "On the Move to Meaningful Internet Systems"*, 2008, pp. 313–322.
- [54] M. A. Memon and B. Archimède, ‘Towards a distributed framework for transportation planning: A food supply chain case study’, in *2013 10th IEEE International Conference on Network, Sensing and Control (ICNSC)*, 2013, pp. 603–608.
- [55] C. Fitouri, N. Fnaiech, C. Varnier, F. Fnaiech, and N. Zerhoune, ‘A Decision-Making Approach for Job Shop Scheduling with Job Depending Degradation and Predictive Maintenance’, *IFAC-Pap.*, vol. 49, no. 12, pp. 1490–1495, Jan. 2016, doi: 10.1016/j.ifacol.2016.07.782.
- [56] D. P. Williamson *et al.*, ‘Short shop schedules’, *Oper. Res.*, vol. 45, no. 2, pp. 288–294, 1997.
- [57] T. Gonzalez and S. Sahni, ‘Flowshop and jobshop schedules: complexity and approximation’, *Oper. Res.*, vol. 26, no. 1, pp. 36–52, 1978.
- [58] G. Provan, ‘An open systems architecture for prognostic inference during condition-based monitoring’, in *2003 IEEE Aerospace Conference Proceedings (Cat. No. 03TH8652)*, 2003, vol. 7, pp. 3157–3164.
- [59] A. Le Mortellec, J. Clarhaut, Y. Sallez, T. Berger, and D. Trentesaux, ‘Embedded holonic fault diagnosis of

- complex transportation systems', *Eng. Appl. Artif. Intell.*, vol. 26, no. 1, pp. 227–240, 2013.
- [60] A. Fadil, D. Trentesaux, and G. Branger, 'Event management architecture for the monitoring and diagnosis of a fleet of trains: a case study', *J. Mod. Transp.*, pp. 1–19, 2019.
- [61] R. Miles and K. Hamilton, *Learning UML 2.0*. O'Reilly Media, Inc., 2006.
- [62] J. J. Odell, *Advanced object-oriented analysis and design using UML*, vol. 12. Cambridge University Press, 1998.
- [63] N. Sortrakul, H. L. Nachtmann, and C. R. Cassady, 'Genetic algorithms for integrated preventive maintenance planning and production scheduling for a single machine', *Comput. Ind.*, vol. 56, no. 2, pp. 161–168, 2005.
- [64] C. Le Pape, 'Implementation of resource constraints in ILOG SCHEDULE: a library for the development of constraint-based scheduling systems', *Intell. Syst. Eng.*, vol. 3, no. 2, pp. 55–66, Summer 1994.
- [65] S. S. Panwalkar and W. Iskander, 'A survey of scheduling rules', *Oper. Res.*, vol. 25, no. 1, pp. 45–61, 1977.
- [66] T. Coudert, B. Grabot, and B. Archimède, 'Production/maintenance cooperative scheduling using multi-agents and fuzzy logic', *Int. J. Prod. Res.*, vol. 40, no. 18, pp. 4611–4632, Jan. 2002, doi: 10.1080/00207540210159545.
- [67] F. Zhao, Z. Tian, and Y. Zeng, 'Uncertainty Quantification in Gear Remaining Useful Life Prediction Through an Integrated Prognostics Method', *IEEE Trans. Reliab.*, vol. 62, no. 1, pp. 146–159, Mar. 2013, doi: 10.1109/TR.2013.2241216.
- [68] M. A. Zaidan, R. F. Harrison, A. R. Mills, and P. J. Fleming, 'Bayesian Hierarchical Models for aerospace gas turbine engine prognostics', *Expert Syst. Appl.*, vol. 42, no. 1, pp. 539–553, Jan. 2015, doi: 10.1016/j.eswa.2014.08.007.