Prognostic Decision Making to Extend a Platform Useful Life under Service Constraint

Nathalie Herr, Jean-Marc Nicod and Christophe Varnier*†‡
FEMTO-ST Institute UFC/CNRS/ENSMM/UTBM,
24 rue Savary, 25000 Besancon, France.
[nathalie.herr/jean-marc.nicod/christophe.varnier]@femto-st.fr

March 21, 2014

Abstract

This paper adresses the problem of optimizing the useful life of a heterogeneous distributed platform which has to produce a given production service. The purpose is to provide a production scheduling that maximizes the production horizon. The use of Prognostics and Health Management (PHM) results in the form of Remaining Useful Life (RUL) allows to adapt the schedule to the wear and tear of equipment. This work comes within the scope of Prognostics Decision Making (DM). Each considered machine is supposed to be able to provide several throughputs corresponding to different operating conditions. The key point is to select the appropriate profile for each machine during the whole useful life of the platform. Many heuristics are proposed to cope with this decision problem and are compared through simulation results. Simulations assess the efficiency of these heuristics. Distance to the theoretical maximal value comes close to 10% for the most efficient ones. A repair module performing a revision of the schedules provided by the heuristics is moreover proposed to enhance the results. First results are promising.

1 Introduction

The problem tackled in this paper concerns the scheduling of a platform composed of many heterogeneous machines. These are supposed to be of similar type and to run in parallel. At each time the global throughput provided by the platform is determined by the sum of each machine throughput that is currently running. The purpose is to manage the platform and implement a production schedule which allows to provide at least a given service level requested by consumers. In literature on scheduling theory, machines are commonly assumed to be continuously available [15]. This assumption may not be valid in a real production situation due to wear and tear on machines. In the considered problem,

*This work has been supported by the Labex ACTION project (contract "ANR-11-LABX-01-01").
†Computations have been performed on the supercomputer facilities of the *Mésocentre de calcul de Franche-Comté*.
‡Authors in alphabetic order

as each machine is assumed to be independent, the breakdown of one of them does not necessarily entail a shutdown of the whole platform. Maintenance is nevertheless required in the long term and generates significant costs. These can be minimized by optimizing maintenance strategies [4]. In some cases, one solution is to group maintenance operations in order to minimize the costs due to the use of material and human resources or production shutdown periods. To manage such a grouping, some maintenance actions might need to be postponed. Dietl et al. [10] proposed for instance to match the time to failure of different tools used in each station of a transfer line by derating them in such a way that a maximum of tools can be maintained at the same time. Grouping maintenance actions can also be necessary because maintenance is challenging. For instance, many works have been carried out to optimize the maintenance of wind farms [13, 18]. For the maintenance of such systems, especially offshore ones, complex aspects like weather conditions, requirement of non-traditional resources, skilled technicians, expensive hired services or spare parts have indeed to be considered. Kovacs et al. [18] proposed a mixed-integer programming formulation for the problem of optimizing the scheduling of maintenance actions for wind farms. Haddad et al. [13] provided an optimization model based on real options and stochastic dynamic programming to optimize the maintenance of offshore wind turbines. Minimization of maintenance costs has been studied by Besnard et al. [4] who proposed an opportunistic maintenance optimization model for offshore wind power systems. Vieira et al. [27] proposed a new variable health threshold that helps to re-schedule and to optimize the maintenance plan of the assets at a wind farm. The objective was to maximize the wind turbine component life-cycle. They concluded that optimizing maintenance of wind turbine components can help achieving a better use of this wind turbine.

The study presented in this paper is close to the one developed in [27]. However our approach goes further in that not only one machine is considered, but a set of machines The objective is then to maximize the production horizon of a whole platform. We assume that the platform can be totally shutdowned for maintenance and that the needed service is provided by an other platform during the maintenance period. All the machines can then be maintained in the same time. The key point is to be able to take the wear and tear of machines into consideration and to know the time left before occurrence of a failure. Prognostics and Health Management (PHM) can comply with these needs. Prognostics phase is indeed dedicated to estimate the Remaining Useful Life (RUL) of machines in service [19, 23]. In a PHM context, the production horizon can be seen as the RUL of the whole platform, which depends on both the RUL of each machine and the schedule. The use of PHM results is furthermore consistent with our objectives in that PHM aims at maintaining equipment operational performance over time, improving their usage while minimizing their maintenance costs [5].

The organisation of the paper is as follows: Section 2 discusses related work. The tackled problem is detailed in Section 3 and is illustrated through a motivating example in Section 4. Then heuristics are proposed to solve the problem in Section 5 and are compared through simulation results (Section 6). A way to enhance the obtained results is described in Section 7. Finally, we conclude in Section 8.

2 Related work

As pointed out by Haddad et al. [13], PHM has been shown to provide many benefits for the health management of systems such as avoiding failures, increasing availability, minimizing loss of remaining life, optimizing resource usage or reducting no-fault-founds. These benefits are strongly tied to the decision part of PHM process whose main purpose is to determine appropriate maintenance actions in response to prognostics predictions [1, 14]. The post-prognostics decision process concentrates appropriate decisions onto one equipment whereas Prognostic Decision Making (PDM) extends decisions to a whole system. Prognostic Decision Making aims also at choosing an appropriate system configuration [2]. Our work falls within this latter case. Temporal segmentation for decision framework has been introduced by Bonissone et al. [5]. They identified three types of decisions in the segment dealing with multiple and repeated decisions: tactical, operational and strategic. According to the frequency on which the decisions have to be taken, diagnostics and prognostics fit with tactical level (seconds, minutes, hours). Decisions for process control in such timeframes suits with on-line scheduling and rescheduling. The part that concerns frequency from nanoseconds to seconds describes configurations that are encountered in electronic, electro-mechanical and control domains. Operational level is adapted to lower frequency decision process as production or maintenance planning and off-line scheduling. Our work falls within these two short-term and mid-term levels of the decision making process, in which many applications are studied. We can cite the aerospace domain [2, 7] and applications on wind turbines [13], electronic systems [23] or cutting tools [6].

Most of the studies proposed in the literature focus on maintenance planification. PHM enables indeed maintenance to be planned on the basis of actual component or system health state [7]. Many contributions are proposed in the form of maintenance policies that minimize life cycle costs. Sandborn et al. [23] endeavor to determine when scheduled maintenance makes sense for electronic systems. Haddad et al. [13] proposed an optimization consisting in finding an optimum subset of offshore turbines to be maintained, given information on their degradation, availability requirement and costs constraints. Balaban et al. [2] developed a prototype algorithm that uses probabilistic methods and prognostics information in generation of action policies for aerospace applications. In the same area, a PHM and Maintenance data integration tool that enables various available diagnostic and prognostics methods to be used in a real environment has been proposed by Camci et al. [7] for fighter aircrafts. Asmai et al. [1] used the data-driven approach to implement an intelligent maintenance prognosis tool. Incorporated into the maintenance decision process, this tool can be used to recommend better maintenance planning. In the same paper, it is pointed out that acknowledging the *RUL* information can also be very useful for production scheduling. Indeed, this quantity gives information about the status of equipment before proceeding with new production jobs. This can help avoiding material waste and production loss due to equipment breakdown in the middle of an operation. Decisions could therefore take several forms: immediate machine shutdown in order to avoid further damage, machine operation modification in such a way as to reduce the load, continuation of normal operation [13], preventive intervention, production rescheduling, etc. The use of prognostics results in the form of *RUL* can then be extended to modify operational conditions or mission profiles in order to accomplish the main objectives of the mission [16, 19]. Balaban et al. [3] proposed such an application on a hardware testbed based on a planetary rover platform and considering many fault modes such as mechanical deterioration, electronic faults or low remaining battery charge. The objective is not only to determine the RUL of a component, but also to suggest actions that can optimize vehicle maintenance, ensure mission safety, or extend mission duration. The idea that is conveyed is the following: if decisions are made with respect to the system health evolution over time, the mission effectiveness can be maximized before energy and health budgets are exceeded. In case of a fault occurrence, a new mission plan may have to be defined. Reconfiguration of the vehicle can also be considered in order to extend the RUL of the affected component as long as needed to ensure achievement of the mission objectives.

Such kind of reconfiguration that affects the system production rate can be found in scheduling literature. Variable-speed scheduling is for instance a generalization of standard multiple machine scheduling because not only the assignment of jobs to machines has to be managed, but also the time used by jobs on machines [20, 25, 26]. Tooling machines are for example variable-speed machines, insofar as they can be run at different speeds [10]. The notion of reconfiguration can also be found in the field of scientific computing and more precisely in scheduling of multiprocessor tasks. Processors capable of global Dynamic Voltage and Frequency Scaling (DVFS) have been developed and allow the manipulation of the voltage and frequency when the computational load is not perfectly balanced [17, 24]. Many applications have been proposed in the literature [8, 22, 28]. Three main objectives can be highlighted among these papers using reconfiguration. The first one is the makespan, i.e., total length of the schedule, minimization [20, 22, 26]. The second one is the minimization of energy consumption [8, 17, 24, 28]. This objective is linked with the third one that consists in minimizing production costs [24, 26].

The objective set in this paper can not be classified in these three categories. The point is to configure a set of machines so as to maximize the production horizon. A second objective is to use all the considered machines to their full potential in order to minimize maintenance costs by grouping maintenance operations. Scheduling which is taken into consideration differs furthermore from the general definition. We seek indeed to schedule considering prognostics information. We consider prognostics-based scheduling, which can be defined as a scheduling that takes the wear and tear of equipments into account and that adapts to remaining useful life (RUL). Scheduling appears then to be part of the PHM Decision Process, as far as prognostics results are used to determine the length of time intervals between two maintenance operations. Prognostics-based scheduling complies with main goal of scheduling that is achieving an optimal usage of resources. Such a kind of scheduling could furthermore be adaptive, as it may respond to disruptions or to knowledge of new informations dynamically over time [9].

3 Problem statement

As developed in previous section, the performance of a machine may vary during its use and this variation can be controlled, for instance through voltage, power or speed scaling. We propose to exploit this characteristic to optimize the use of the considered platform. Each machine is supposed to be able to provide several throughputs. In a PHM context, each throughput corresponds to a certain operating condition. We assume moreover that each machine is monitored and associated with a prognostics module that gives a RUL value depending on both its past and its future usage. As highlighted by Elghazel and al. [11], the way to consider operating conditions, especially future ones in RUL estimation, still needs deep studying. So we assume in this paper that each needed RUL value is known and that RUL evolution depends on the operating conditions, that is on the running profile in which the machine is used.

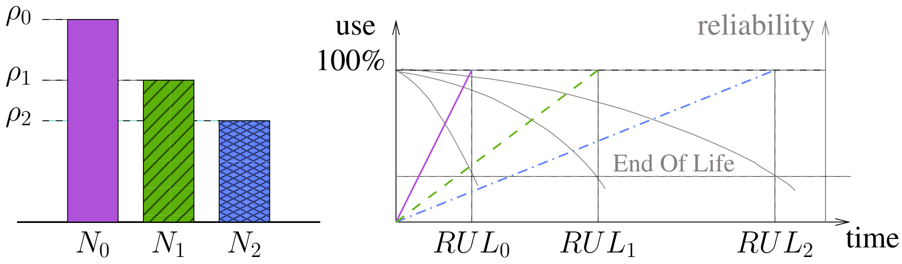
3.1 Controlled running profiles in a PHM context

We define a running profile as a controlled machine profile involving a certain throughput and associated with a certain RUL: $N_i = (\rho_i, RUL_i)$, with $i \in I = \{0, \ldots, n-1\}$. Let N_0 be the nominal running profile, where immediate throughput is the most significant. This nominal running profile has the minimum RUL. By comparison a sub-nominal profile provides a lowest throughput, but its associated RUL is longer (see Figure 1(a)). Each running profile corresponds to an operating condition and impacts differently the wear and tear of the machine and therefore its operational time. Taking several running profiles into consideration seems to be interesting in that the combination of two or more profiles allows to reach an operational time that is greater than the one obtained with its nominal running profile. As example, the Figure 1(b) shows that the use of three different running profiles allows to run the same machine for longer than the RUL of nominal profile. Of course, in counter part, the amount of work done with this machine is lower than it would be with the nominal profile.

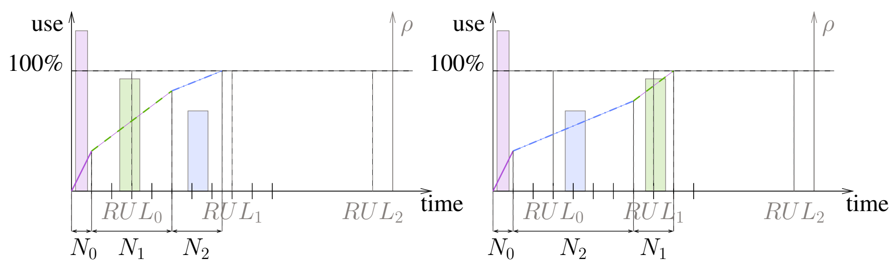
We assume that the order in which the running profiles are selected among the machine lifespan does not impact the RUL evolution. The second scenario proposed in Figure 1(b) shows that the machine lifespan can not only exceed the RUL of the nominal profile, but also the one associated to the second defined running profile. As showed in the motivating example in Section 4, one can take advantage of the use of many running profiles to optimize the scheduling of a set of machines.

3.2 Application framework

The application that is addressed in this paper is a platform composed of m independent machines M_j, with $j \in J = \{1, \ldots, m\}$, performing independent and identical tasks. All the machines can be used in parallel as a global system. Machines are supposed to be always supplied with power or raw material required for the production. The provided result is a given service level that is measured as a throughput, i.e., number of pieces performed or amount of matter (a) treated per unit of time (ut). At each time the global throughput ρ_{tot} provided by the platform is determined by the sum of each throughput ρ_j of machines M_j that are currently running. Note that the platform has to deliver a given global throughput $\sigma = \sigma(t)$. This latter one is based on a customer demand, which can be variable and defined as a function of time. The platform can be seen as a distributed environment where machines that are currently running fulfill a shared global task such that $\rho_{tot} \geq \sigma$.



(a) Characteristics of running profiles



(b) Use of several running profiles

Figure 1: Running profiles

All the machines are not supposed to be in use at any time because of their RUL or because the target throughput σ can be achieved by using only a subset of the available machines within the platform. RUL is assumed to be constant in time when the machine is not used. Moreover, we assume that overproduction should be avoided as far as possible. Overproduction leads indeed either to costly stocks or to losses if the production can not be stored. Allowing overproduction can however allow to extend a platform useful life (see scenario 2 in Section 4). The key point of the considered problem is then to be able to find the appropriate profile for each machine at each period of time, as described in the next section.

3.3 Decision problem

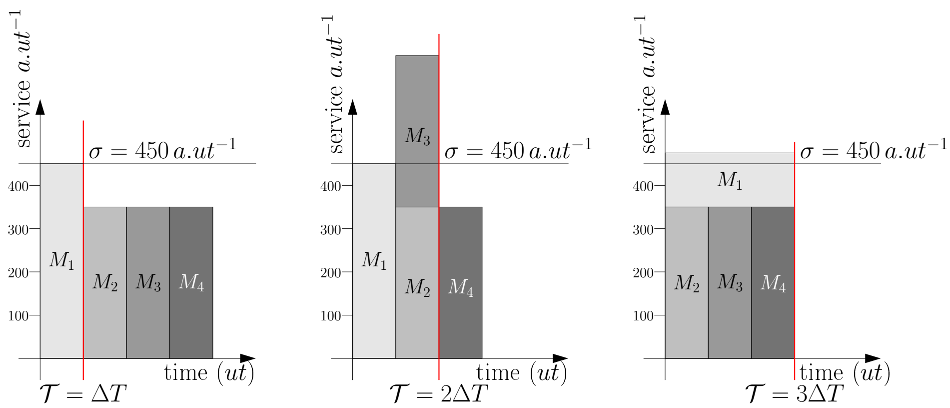
The problem tackled in this paper is the optimization of the useful life of a platform such as defined in the application framework (see Section 3.2). The objective is to provide a prognostics-based schedule as defined in Section 2 by configuring the platform so as to reach the demand as long as possible. One way to tackle the problem consists in discretizing the time into periods. This approach is not so far from realistic constraints, since one can imagine that one period could be one day or one week in a real case. The production horizon $\mathcal{T}$ can then be expressed as follows: $\mathcal{T} = K \times \Delta T$, with ΔT the length of one time period and K the number of periods for which the demand level σ is reached. We assume that, if the demand is a variable function of time $\sigma(t)$, then $\sigma(t) = \sigma_k$ is a constant value within the period k for all t and all k such that $(k-1)\Delta T < t \leq k\Delta T$ and $k \leq K$.

Considering discretized time, the problem consists in choosing, for each period of time k, a subset of machines to be used and an associated running profile for each of them. Using the notations defined in this section, the problem that is tackled here can be described by a notational form: $\textsc{max}\mathrm{K}(\sigma_k \mid \rho_{i,j} \mid RUL_{i,j})$. This general notation stands for the problem of finding an optimal schedule, that is maximizing the production horizon $K\Delta T$, considering a variable needed service level σ_k in terms of throughput for each time period k $(1 \leq k \leq K\})$ and several machines M_j $(1 \leq j \leq m)$ with many running profiles N_i $(0 \leq i < n)$, different throughputs $\rho_{i,j}$ and different states of health $RUL_{i,j}$ at time 0 (time 0 refers to the schedule at starting time). Many alternatives can be used to represent sub-problems by adapting the subscripts of the different parameters.

In the following, for the readability of the resolution methods developed in Section 5 and without loss of generality, the demand $\sigma(t) = \sigma_k$ is assumed to be constant in time and equal to σ. The problem is then the following: $\textsc{max}\mathrm{K}(\sigma \mid \rho_{i,j} \mid RUL_{i,j})$. The proposed methods could nevertheless easily be adapted for a variable demand, provided that this demand is known in advance for the whole production horizon.

4 Motivating example

To illustrate our purpose let us describe a small example. Consider four machines, each of them being able to provide a given throughput during only one period (ΔT). Let's assume that a given global throughput σ has to be reached. A first scenario, where overproduction is not allowed, consists in using each machine one after another. One can see in Figure 2(a) that in this case the platform runs for four periods but the targeted throughput is achieved only for one period. Considering the objective, the scheduling horizon $\mathcal{T}$ is then one period ($\mathcal{T} = \Delta T$).



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3

Figure 2: Motivating example

When overproduction is allowed, two machines can be used in parallel and the scheduling horizon is increased to two periods (see Figure 2(b) with $\mathcal{T} = 2\Delta T$). If the production is stopped after two periods, some potential still remains. The machine M_4 has indeed never been used and does not need maintenance yet. The schedule proposed in Figure 2(b) is optimal under the previous assumptions.

Machines should then be used in another profile to extend the production horizon. One can see in Figure 2(c) that using the machine M_1 with a lower throughput allows to reach the targeted throughput for three periods. This example shows that using machines with different running profiles allows to extend the useful life of the set of machines while respecting a given targeted global throughput.

5 Resolution methods

An optimal approach based on an exact resolution method has been proposed to cope with the problem defined in Section 3.3 by Nicod et al. in [21]. The decision problem has been described as follows: does a schedule exist to fulfill a given constant demand σ during a given number of periods K, considering the machines health states? An Integer Linear Program (ILP) using binary variables has been proposed. A binary solver [12] has been used in parallel with a dichotomic search to find the maximal value of K. As solving such a Binary Integer Linear Program (BILP) is NP-complete, solutions can be found in limited time only for small size instances of the problem.

In order to deal with large scale problems, we propose here five polynomial time heuristics that allocate for each period of time enough machines to reach the targeted throughput as long as possible. Each heuristics follows its own strategy to select the machines and a running profile for each of them so as to define its contribution to the global production within the current period. A first strategy consists in defining the schedule period by period. A new selection of machines is performed for each period and is applied only for one period. An other strategy consists in applying the same selection on many periods. The number of periods on which a solution can be applied is limited by the selected machine having the smallest RUL. One can see that the strategy working by group of periods cannot be used as it is when a variable demand σ_k is considered. It could nevertheless easily be adapted by applying each selection on the minimum of the two values: smallest RUL of the selected machines and length of the time interval in which the targetted demand remains constant. Regardless of the two strategies, three different types of heuristics can be distinguished. First heuristics provides a random schedule. Second type of heuristics are greedy ones and last one uses a dynamic programming based algorithm. An illustration based on the initial set of machines described in Figure 3 is given for each heuristics in the following sections.

5.1 H–RAND: Random assignment heuristics

The first heuristics works period by period. The following process is iterated as long as a solution is found:

1. Choose randomly a machine and an associated running profile among these that are available

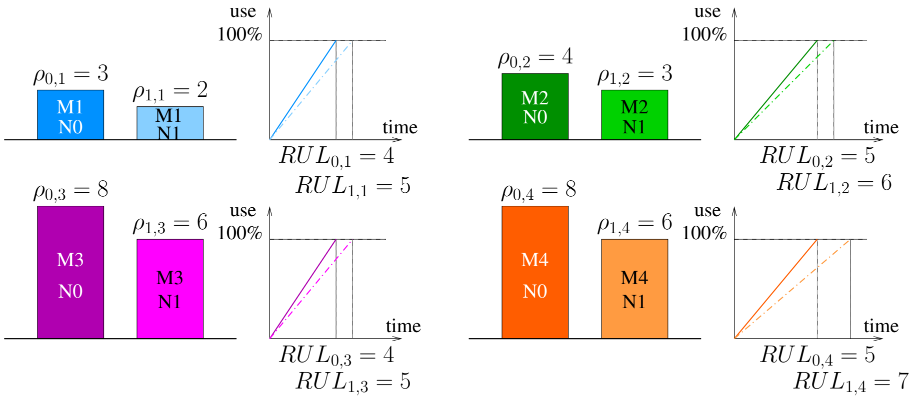


Figure 3: Initial set of machines

2. Go to the first step as long as the global throughput does not reach the demand σ
3. Use the selected machines for one period of time
4. Update the RUL of the selected machines to take their usage into account

The process is stopped as soon as a period can not be completed. This can happen even if there is enough potential left. In each period, the remaining machines depends indeed on the first choices made by the heuristics.

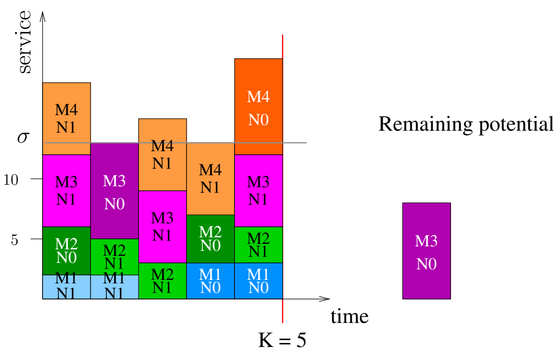


Figure 4: Schedule obtained with H–RAND

5.2 H–LRF: Largest RUL First heuristics

This second heuristics works by group of periods and aims at considering each machine M_j using its profile associated to the largest RUL, that is the profile providing the lowest throughput: $p_{n-1,j} = (\rho_{min,j}, RUL_{max,j})$. The following process (see illustration in Figure 5) is iterated as long as there is enough potential left to reach the demand for minimum one time period:

1. Select all available machines in their running profile providing the lowest throughput

2. While the global throughput ρ_{tot} is less than the demand σ, increase the contribution of the machine having the maximal RUL by modifying its running profile from the chosen one N_i to the previous one providing a higher throughput N_{i-1}

3. While the global throughput ρ_{tot} exceeds the demand σ, erase the machine providing the maximal throughput ρ_{max} from the solution such that ρ_{max} is lower or equal to the overproduction (i.e., $\rho_{max} < \rho_{tot} - \sigma$)

4. Use the selected machines for a number of periods equal to the smallest RUL of the solution

5. Update the RUL of the selected machines to take their usage into account

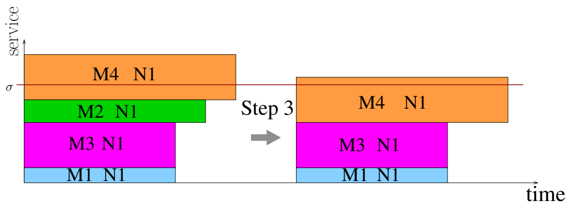


Figure 5: H–LRF operating principle

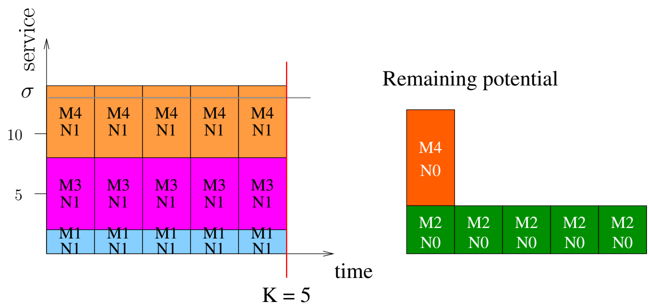


Figure 6: Schedule obtained with H–LRF

5.3 H–HOF: Highest Output First heuristics

The heuristics H–HOF is based on the same principle as H–LRF but each machine M_j is configured with its most efficient profile $p_{0,j} = (\rho_{max,j}, RUL_{min,j})$. Two options can be considered. First one, H–HOFlt (Highest Output First, lowest throughput first), selects the machines having the lowest throughput first and second one, H–HOFht (Highest Output First, highest throughput first), these having the highest throughput. The following process (see illustration in Figure7) is iterated as long as there is enough potential left to reach the demand for a minimum of one period:

1. Select the smallest subset of machines providing the smallest (resp. the highest) throughputs in their most efficient profile such that ρ_{tot} reaches at least σ for the first option H–HOFlt (resp. second option H–HOFht)

2. While the global throughput ρ_{tot} exceeds the demand σ, decrease the contribution of the machine having the minimal RUL by modifying its running profile from the chosen one N_i to the following one providing a lowest throughput N_{i+1}, only if ρ_{tot} remains greater or equal to σ

3. Use the selected machines as long as possible (corresponds to the smallest RUL of the solution)

4. Update the RUL of the selected machines to take their usage into account

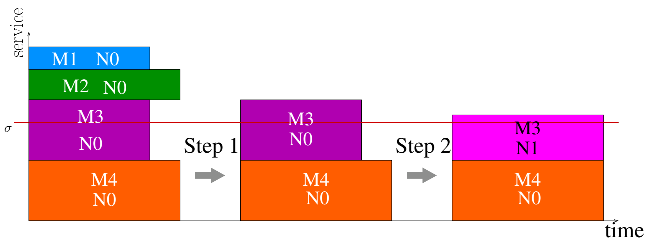


Figure 7: H–HOFht operating principle

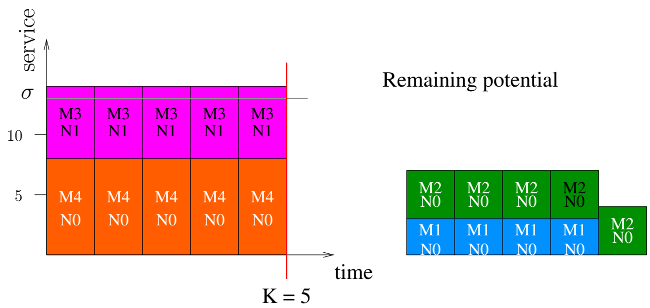


Figure 8: Schedule obtained with H–HOFht

5.4 H–DP: Dynamic Programming based heuristics

H–DP is a more sophisticated heuristics. It aims at minimizing the production loss. If one period is considered, the problem is to find a subset of couples machine/running profile that is able to reach at least the production demand with the smallest overproduction. A Knapsack-like algorithm is proposed so as to make the choice between all the available couples within the current period. The differences with the classical Knapsack problem is first that the sum of the value ($\rho_{i,j}$) of the selected objects (subset of couples machine/running profile) should be greater or equal to the knapsack weight (σ). Secondly, each object (M_j) can have several values ($\rho_{i,j}$, $0 \leq i \leq n-1$) and at most one could be selected. The objective of this Knapsack-like problem is to minimize the sum of the machine values in the case where this sum exceeds the knapsack weight σ.

The algorithm developed to implement H–DP is the classical dynamic programming based approach. Each available machine is successively considered, following an ascending order of their throughput. This sorting allows to minimize the number of recorded solutions and therefore minimizes both the memory needed and the processing time. Machines with the same throughput are also sorted in descending order of their RUL. Each running profile of each machine is successively considered, from the last one providing the minimal throughput to the most nominal one. Performing this sorting before each search for solution allows to wear out the set of machines homogeneously. Due to this turnover in the use of machines, a maximum of different machines are kept available for the last periods and the production horizon is extended.

For each machine M_j, the targeted throughput σ' is iterated from 1 to σ. For each value of σ', each available profile $p_{i,j} = (\rho_{i,j}, RUL_{i,j})$ $(0 \le i \le n-1)$ of M_j is considered to select or not the current machine with its better configuration regarding the objective. To define the objective value let's introduce some notations: let $ov_i(\sigma', j)$ be the overall throughput obtained by the j first machines using both the j^{th} machine with its i^{th} profile and the optimal configuration considering the $j-1$ first machines obtained for a target throughput of $\sigma' - \rho_{i,j}$; let $OV_i(\sigma, j)$ be a valide overall throughput and $+\infty$ otherwise; finally let $OV(\sigma', j)$ be the optimal (minimal) throughput that exceeds the target demand σ' using a subset of the j first machines. The expression of the optimal value is the following:

$$ov_i(\sigma', j) = OV(\sigma' - \rho_{i,j}, j-1) + \rho_{i,j} \text{ with } 1 \le i \le n$$

$$OV_i(\sigma', j) = \begin{cases} ov_i(\sigma', j) & \text{if } ov_i(\sigma', j) \ge \sigma' \\ +\infty & \text{otherwise} \end{cases}$$

$$OV(\sigma', j) = \min\left(OV(\sigma', j-1), \min_{1 \le i \le n} OV_i(\sigma', j)\right)$$

The minimal throughput for the current period is given at the position $OV(\sigma, m)$ of the 2D matrix OV used by the algorithm. Thanks to the storage of each choice that is made for every couple (σ', j) when the algorithm is running, the algorithm is able to reconstruct the way to obtain the optimal schedule. Should two or more equivalent schedules be found, the algorithm chooses the solution with fewer machines.

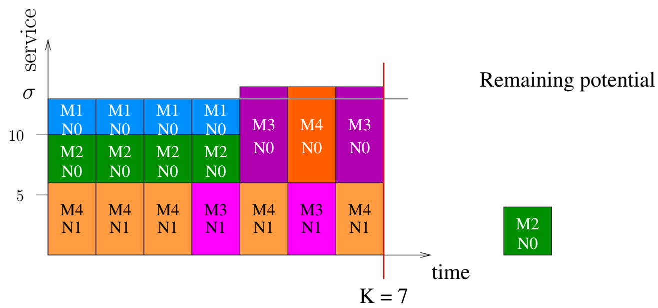


Figure 9: Schedule obtained with H–DP

As illustrated in Figure 9, H–DP minimizes the overproduction as long as possible. While the schedule found for each time period is optimal, the global schedule is not necessary optimal. This can be seen in Section 6, in which all the proposed heuristics are compared to upper bounds and among themselves through their reached production horizon.

6 Simulation results

6.1 Benchmark generation

Both proposed approaches previously described (optimal and heuristics ones) have been validated on random problem instances. These have been generated using a simulator and configured with many parameters. First one is the number of machines constituting the platform: $M \in \{10, 25, 50\}$. Second one sets the number of running profiles with which each machine can be used: $N \in \{1, 2, 5, 10\}$. As pointed out in Section 3.3, the demand σ is considered to be constant during the whole scheduling process. Only one demand value is then associated to each problem instance. Many demand values corresponding to different problem instances have however been tested. These values have been defined as follows: $\sigma = \alpha \times \rho_{tot,max}$, with $\rho_{tot,max} = \sum_{1 \leq j \leq m} \rho_{max,j}$ the maximal total throughput available with the considered set of machines and α a load varying between 30% and 90%.

The protocol used to run the experiments consists in generating 20 problem instances with the same parameters values. Each instance corresponds to a different platform.

In the next sections, we present the results obtained with this benchmark generation. On the basis on many tests on different sets of machines, it appears that one version of H–HOF is as efficient as the other one. For the rest of the study, only the version selecting the machines having the highest throughputs first, H–HOFht, will then be considered.

6.2 Comparison to the optimal

As developed in Section 5 and in [21], results obtained with the heuristics can be compared to optimal ones only for small size instances of the problem. Tests have been performed for cases with $N \leq 2$, $M \leq 5$ and $K \leq 20$. Solutions obtained with H–DP are on average at 5% from the optimal one.

6.3 Comparison of heuristics

In the following figures, the production horizon K is represented as a function of the load $\alpha = \sigma / \rho_{tot,max}$ varying between 30% and 90%.

It appears that the random assignment heuristics provides results that are not so bad when only one running profile is considered. One can see in Figure 10 that H–RAND provides the greater K value for loads between 40% to 80%. This can be explained by the fact that this heuristics does not select useless machine. The maximal overproduction is then equal to the maximal available throughput ρ_{max} minus 1. H–RAND is however not reliable for high loads ($\alpha \geq 50\%$) when the number of running profiles is increased. The number of possibilities for the choice of couples machine/profile increases with the number of profiles. If many machines are selected in profiles providing to low throughputs, the remaining

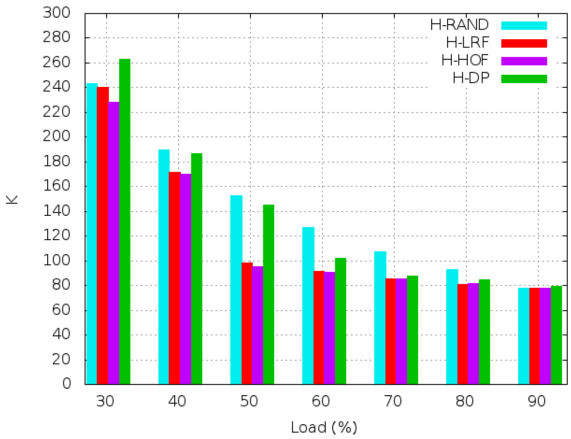


Figure 10: Average number of periods completed (K) depending on the load - N = 1 running profile, M = 10 machines

machines may not be sufficient to reach the demand, even in their nominal profile. Results already decrease when taking into account two running profiles (see Figure 11). The heuristics H–RAND will then not be considered in next simulation results.

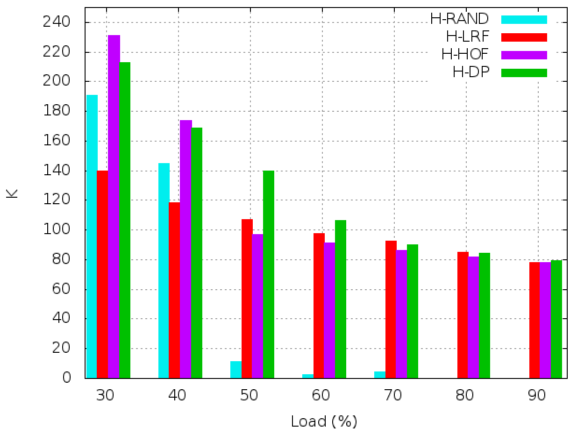


Figure 11: Average number of periods completed (K) depending on the load - N = 2 runnnig profiles, M = 10 machines

Considering many running profiles can however be interesting. One can indeed see in Figure 12 that the production horizon K increases with N when H–LRF is used.

Variation of the number of running profiles and of the number of machines seems to have no significant effect on the results provided by H–HOF (see Figures 12, 13 and14). This heuristics favours indeed the nominal running profiles. Considering the same machine, the nominal running profile provides the same throughput and is associated with the same RUL whatever the number of running profiles considered.

H–DP appears to give the best results for low loads α varying between 30% and 50% (see Figures 13 and 14). For high loads ($\alpha > 50\%$), the highest production horizons are obtained with H–LRF (see Figures 13 and 14).

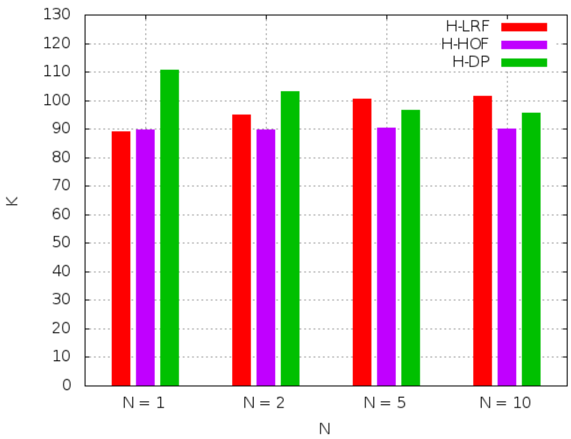


Figure 12: Average number of periods completed (K) depending on the number of running profiles (N) - M = 10 machines, load = 60%

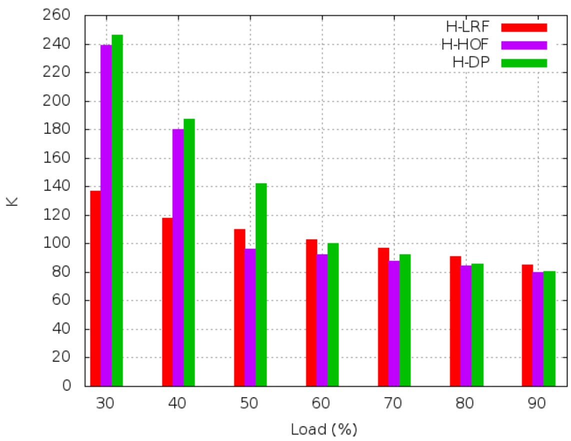


Figure 13: Average number of periods completed (K) depending on the load - N = 5 running profiles, M = 10 machines

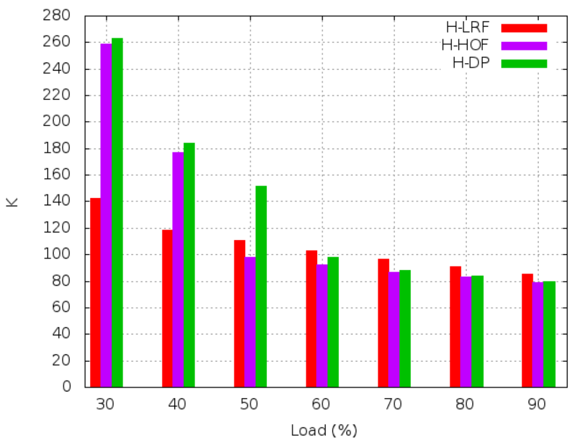


Figure 14: Average number of periods completed (K) depending on the load - N = 5 running profiles, M = 25 machines

6.4 Comparison to an upper bound

An upper bound KMAX can be provided as defined in Equation 1. If all the machines are used with their running profile that provides the best output ($Q = \rho \times RUL$) and if the total required throughput σ is constant over time, then KMAX is the theoretical longest duration for which the demand σ can be reached (see Equation (1)). This upper bound is only reachable under a very restrictive condition, if no overproduction is performed during the whole scheduling horizon.

$$\text{KMAX} = \left\lfloor \frac{\sum_{1 \le j \le m} \max_{0 \le i < n} (\rho_{i,j} \times RUL_{i,j})}{\sigma} \right\rfloor \tag{1}$$

In the following figures, distance of the production horizon K to the theoretical maximal horizon KMAX is represented as a function of the load $\alpha = \sigma/\rho_{tot,max}$ varying between 30% and 90%.

One can see in Figures 15, 16 and 17 that all the heuristics excepting H–RAND are at least at 50% from KMAX, 30% for H–DP. This is promising since the upper bound KMAX is reasonably not reachable. In the best cases, H–DP is at 10% from the maximal value. With high loads and a great number of running profiles, H–LRF gets also close to 10%.

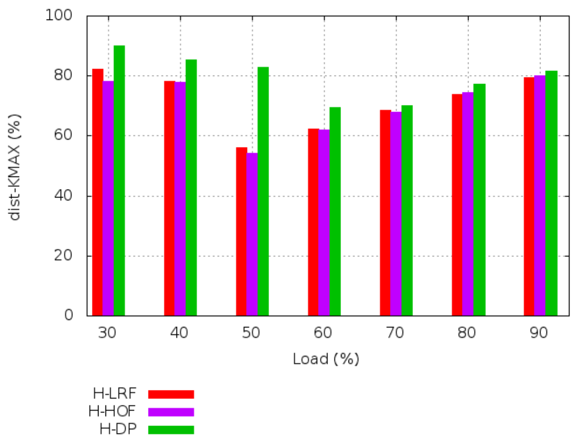


Figure 15: Distance to the theoretical maximal value (KMAX) depending on the load - N = 1 running profile, M = 10 machines

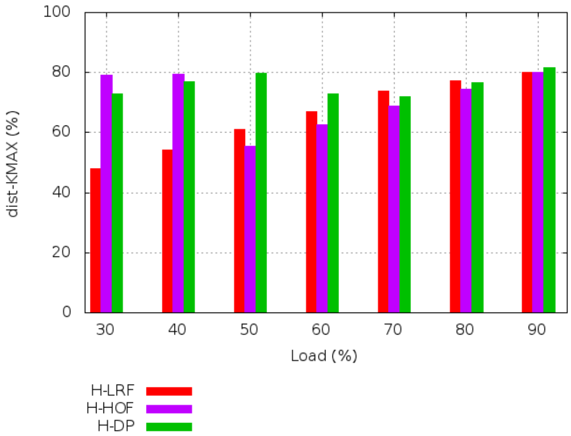


Figure 16: Distance to the theoretical maximal value (KMAX) depending on the load - N = 2 running profiles, M = 10 machines

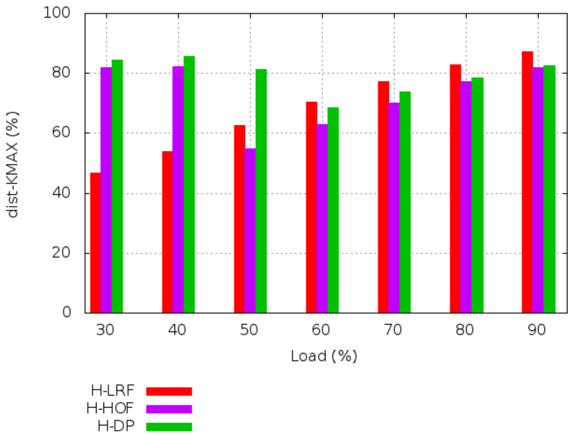


Figure 17: Distance to the theoretical maximal value (KMAX) depending on the load - N = 5 running profiles, M = 10 machines

7 Enhancement: Repair module

According to the results presented in Section 6, optimal solutions can not be found using the sub-optimal approaches proposed in Section 5. The distance to the optimum is indeed always positive, which means that there is a scope for improvement. There is also usually remaining potential, i.e., many machines can still be used as their RUL is greater than 0 at the end of most of the schedules obtained with sub-optimal approaches. The corresponding production horizons could then be extended by using this remaining potential.

7.1 Strategy

We propose to enhance the results obtained with the previous heuristics by performing a revision of the schedules. This can be done because the schedules are built up offline. Repair will be performed on the results of the H–DP as it gives globally the best results in terms of production horizon. There is furthermore less remaining potential at the end of H–DP schedules, so less to repair. We saw that the random strategy achieved good results even if it allows overproduction. The idea here is then to relax the first criterion of the dynamic programming by allowing overproduction. In concrete terms, some remaining machines are exchanged for other machines used in the initial schedule. The recovery of these machines allows to increase the number of machines that can be used in parallel and allows potentially to reach the demand for one or more additional period(s).

The repair process can be seen on the following very simple example. Let's consider three machines with one running profile and the caracteristics showed in Figure 18. The schedule obtained with H–DP can be seen in Figure 19(a). One can see that the machine M_3 is never used. There is a remaining potential, but no additional period can be completed because the remaining machine is not powerful enough to reach the demand alone. The machine M_3 is not used in the first period of the schedule, so it can be exchanged with machine M_2 for one period. There is now an overproduction in the first period of the schedule, but also two different machines available. The demand can then be reached for one more period by using machines M_3 et M_2 in parallel (see Figure 19(b)). The same exchange can be done in the second scheduled period. This allows to get the machine M_2 back for one period and to increase anew the production horizon K by one (see Figure 19(c)). On this example, applying the repair on the H–DP schedule allows to use all the machines entirely and to extend the horizon production from 4 to 6 time periods.

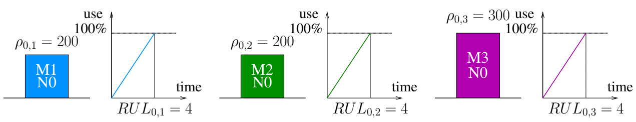
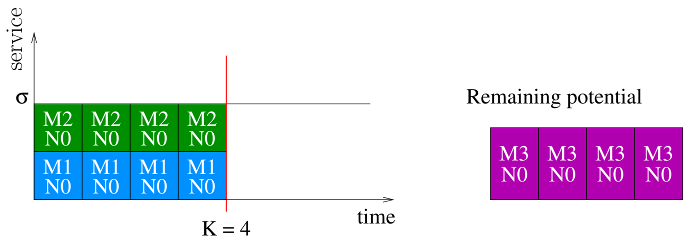


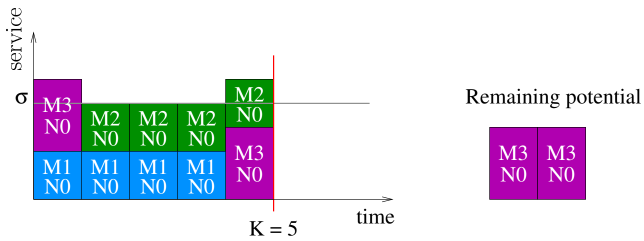
Figure 18: Set of machines for repair illustration

7.2 Results

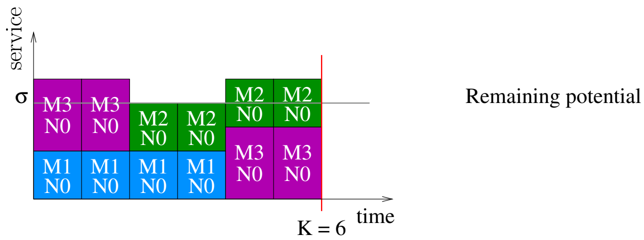
The repair efficiency has been studied on small problem instances, with 5 machines and 1 running profile. Figure 20 compares the repair results to the initial



(a) H–DP schedule



(b) H–DPR - Step 1



(c) H–DPR - Step 2

Figure 19: Repair strategy

ones provided by H–DP. One can see that the repair module allows to increase the number of completed periods K. The repair is more efficient for low loads. In case of high loads, more machines have to be used in parallel to reach the demand. Even if the remaining potential is high, only few remaining machines can then be exchanged.

The first results showed in Figure 21 are promising. Results obtained by the dynamic programming based heuristics are indeed improved and brought closer to the theoretical maximal value. The new results are between 5% and 25% from this maximal bound. We recall that the optimal solution KOPT is less than KMAX. Repair results are then actually better than showed in Figure 21.

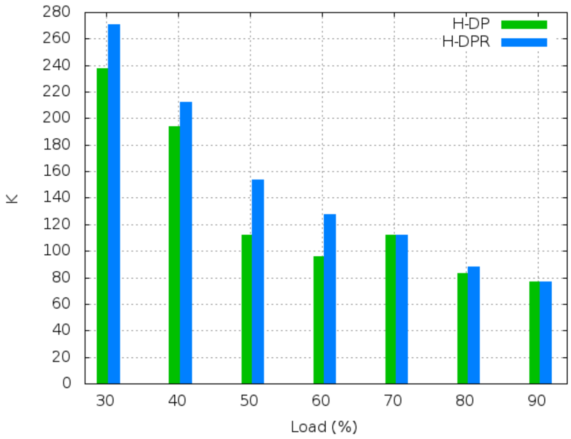


Figure 20: Average number of periods completed (K) depending on the load - N = 1 running profile, M = 5 machines

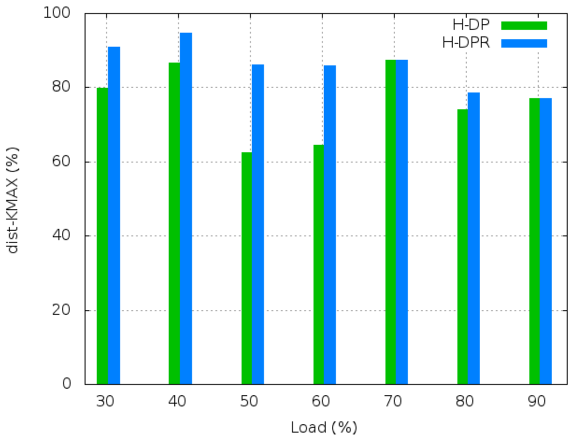


Figure 21: Distance to the theoretical maximal value (KMAX) depending on the load - N = 1 running profile, M = 5 machines

8 Conclusion and future work

A new approach of scheduling using prognostics results has been investigated in this paper. We have proposed scheduling algorithms using several operating conditions for each machine of a heterogeneous platform so as to extend the global operational time. We have shown that we are able to prolong as long as possible the production horizon by managing the usage of the resource thanks to the knowledge of each machine remaining useful life.

Prognostics-based scheduling has been proposed to configure sets of machines in compliance with the objective. This particular scheduling makes use of prognostics results in the form of RUL to adapt the provided schedule to the real state of the machines. It is part of the last step of the PHM process, i.e., decision making. Since the optimal solution can only be reached by running a time consuming Binary Integer Linear Program, several sub-optimal heuristics have been presented to solve the considered decision problem in polynomial time. Efficiency of these heuristics has been assessed by numerous exhaustive simulations.

As future work, we plan to explore continuous use of machines. None of the proposed solutions guarantees that a machine will be used during its whole operational time without a planned shutdown. Taking this constraint into account is challenging in some production context. When some machines are running, as fuel cells, shutting down their production for a short period incurs extra costs.

Taking maintenance tasks into account within prognostics-based schedules is also a very interesting issue. In the best case scenario, optimization of the maintenance policy could allow to provide a steady-state scheduling.

References

[1] Siti Azirah Asmai, Burairah Hussin, and Mokhtar Mohd. Yusof. A framework of an intelligent maintenance prognosis tool. In *2nd International Conference on Computer Research and Development, ICCRD 2010*, number art. no. 5489525, pages 241 – 245, 2010.

[2] Edward Balaban and Juan J. Alonso. An approach to prognostic decision making in the aerospace domain. In *Annual Conference of the Prognostics and Health Management Society*, 2012.

[3] Edward Balaban, Sriram Narasimhan, Matthew Daigle, José Celaya, Indranil Roychoudhury, Bhaskar Saha, Sankalita Saha, and Kai Goebel. A mobile robot testbed for prognostic-enabled autonomous decision making. In *Annual Conference of the Prognostics and Health Management Society*, 2011.

[4] François Besnard, Michael Patriksson, Ann-Brith Strömberg, Adam Wojciechowski, and Lina Bertling. An optimization framework for opportunistic maintenance of offshore wind power system. In *IEEE Powertech*, 2009.

[5] Piero P. Bonissone and Naresh Iyer. Soft computing applications to prognostics and health management (phm): Leveraging field data and domain knowledge. In *IWANN*, pages 928–939, 2007.

[6] Fatih Camci and Ratna B. Chinnam. Health-state estimation and prognostics in machining processes. *IEEE Transactions on Automation Science and Engineering*, 7 (3):581 – 597, July 2010.

[7] Fatih Camci, G. Scott Valentine, and Kelly Navarra. Methodologies for integration of phm systems with maintenance data. In *IEEE Aerospace Conference*, volume 1 - 9 of *IEEE AEROSPACE CONFERENCE PROCEEDINGS*, pages 4110 – 4118, 2007.

[8] G. Chen, K. Malkowski, M. Kandemir, and P. Raghavan. Reducing power with performance constraints for parallel sparse applications. In *Proceedings of 19th IEEE International Parallel and Distributed Processing Symposium, Boston, USA*, 2005.

[9] Jie Chen and F. Frank Chen. Adaptive scheduling in random flexible manufacturing systems subject to machine breakdowns. *International Journal of Production Research*, 41, No. 9:1927–1951, 2003.

[10] Clemens Dietl and Uwe K. Rakowsky. An operating strategy for high-availability multi-station transfer lines. *International Journal of Automation and Computing*, 2:125 – 130, 2006.

[11] Wiem Elghazel, Noureddine Zerhouni, Jacques Bahi, Kamal Medjaher, M. Hakem, and C. Guyeux. Dependability of sensor networks for prognostics and health management. Technical report, FEMTO-ST, dpt AS2M, 25000 Besançon, FRANCE, 2013.

[12] Inc. Gurobi Optimization. Gurobi optimizer reference manual. `http://www.gurobi.com`, 2014.

[13] Gilbert Haddad, Peter Sandborn, and Michael Pecht. A real options optimization model to meet availability requirements for offshore wind turbines. In *MFPT: The Applied Systems Health Management Conference, Virginia Beach, Virginia*, 2011.

[14] Naresh Iyer, Kai Goebel, and Piero Bonissone. Framework for post-prognostic decision support. In *Proceedings of 2005 IEEE Aerospace Conference*, 2005.

[15] Min Ji, Yong He, and T.C.E. Cheng. Single-machine scheduling with periodic maintenance to minimize makespan. *Computers & Operations Research*, 34:1764–1770, 2007.

[16] Mohamed-Hedi Karray, Brigitte Chebel-Morello, and Noureddine Zerhouni. A trace based system for decision activities in cbm process. In *Proceedings of IEEE International Conference on Prognostics and Health Management, Gaithersburg, Maryland*, 2013.

[17] Hideaki Kimura, Mitsuhisa Sato, Yoshihiko Hotta, Taisuke Boku, and Daisuke Takahashi. Empirical study on reducing energy of parallel programs using slack reclamation by dvfs in a power-scalable high performance cluster. In *Proceedings of IEEE International Conference on Cluster Computing, Barcelona, Spain*, 2006.

[18] Andras Kovacs, Gabor Erdos, Laszlo Monostori, and Zsolt Janos Viharos. Scheduling the maintenance of wind farms for minimizing production loss. In *Proceedings of the 18th IFAC World Congress, Milano (Italy)*, 2011.

[19] M. Lebold and M. Thurston. Open standards for condition-based maintenance and prognostic systems. In *5th Annual Maintenance and Reliability Conferance (MARCON), Gatlinburg, USA*, 2001.

[20] C.-Y. Lee and V.J. Leon. Machine scheduling with a rate-modifying activity. *European Journal of Operational Research*, 128:119–128, 2001.

[21] Jean-Marc Nicod, Léo Obrecht, and Christophe Varnier. Task scheduling to extend platform useful life using prognostics. In *Proceedings of IFAC Conference on Manufacturing Modelling, Management, and Control, Saint Petersburg, Russia, June 19-21, 2013*, 2013.

[22] Thomas Rauber and Gudula Rünger. Compiler support for task scheduling in hierarchical execution models. *Journal of Systems Architecture*, 45:483–503, 1998.

[23] Peter Sandborn. A decision support model for determining the applicability of prognostic health management (phm) approaches to electronic systems. In *Proc. Reliability and Maintainability Symposium (RAMS), Arlington, VA*, 2005.

[24] Greg Semeraro, Grigorios Magklis, Rajeev Balasubramonian, David H. Albonesi, Sandhya Dwarkadas, and Michael L. Scott. Energy-efficient processor design using multiple clock domains with dynamic voltage and frequency scaling. In *8th International Symposium on High-Performance Computer Architecture (HPCA), Cambridge, MA*, February 2002.

[25] Jiri Sgall. *Online algorithms*, volume 1442 of *Lecture notes in computer science*, chapter On-line Scheduling, pages 196–231. Springer-Verlag, Berlin, Germany, 1998.

[26] Michael A. Trick. Scheduling multiple variable-speed machines. *Operations Research*, 42:234–248, 1994.

[27] Rodrigo J.A. Vieira, Matteo Lietti, and Miguel A. Sanz-Bobi. New variable health threshold based on the life observed for improving the scheduled maintenance of a wind turbine. In *2nd IFAC Workshop on Advanced Maintenance Engineering*, 2012.

[28] Lizhe Wang, Gregor von Laszewski, Jai Dayal, and Fugang Wang. Towards energy aware scheduling for precedence constrained parallel tasks in a cluster with dvfs. In *Proceedings of 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid), Melbourne, Australia*, 2010.