Use of (max, +) algebra for scheduling and optimization of HVLV systems subject to preventive maintenance

Imed NASRI\textsuperscript{a}, Georges HABCHI\textsuperscript{a}, Reda BOUKEZZOULA\textsuperscript{b}

\textsuperscript{a}Univ. Savoie, SYMME, F-74000 Annecy
\textsuperscript{b}Univ. Savoie, LISTIC, F-74000 Annecy

Abstract

High-Variety, Low-Volume (HVLV) manufacturing systems are built to produce parts of several types in small quantities and under multiple production objectives. They relate to job-shop systems well known by researchers. One of the most studied assumptions of HVLV systems scheduling is considering that machines may be periodically unavailable during the production scheduling. This article deals with an analytical integrating method using \((\text{max}, +)\) algebra to model HVLV scheduling problems subject to preventive maintenance (PM) while considering machines availability constraints. Each machine is subject to PM while maintaining flexibility for the start time of the maintenance activities during the planning period. The proposed model controls the placement of maintenance activities along the production operations. Indeed, the sequencing of maintenance activities on the machines depends on the criteria to minimize and may be different for each criteria value. For preventive maintenance, the proposed model aims to generate the best sequencing between activities while respecting the planning program that satisfy the optimal criteria values. In order to illustrate the performance of the proposed methodology, a simulation example is given.

Keywords: HVLV manufacturing systems, \(\text{max}+\) scheduling model, decision variables, non-linear optimization, simulation, preventive maintenance, makespan

1. Introduction

It is well recognized that High-Variety, Low-Volume (HVLV) manufacturing systems are a class of dynamic systems where their behavior can be assimilated to Discrete Event Dynamic Systems (DEDS). HVLV systems are characterized by a wide variety of products using shared machines, a weak and personalized demand, relatively long processing times and frequent change over and set-up times. Consequently, a continuous approximation of the production flow by continuous flow systems \cite{1} is not appropriate for HVLV systems. In this framework, it seems very interesting to handle the kind of system such as Job-Shop \cite{2} systems due to the wide variety of processed products and routes.

For DEDS modeling, two major categories are distinguished \cite{3} \cite{4}, \ie untimed and timed models (Petri nets models) where two different kinds of algebraic objects have been introduced (untimed and timed behaviors). If the untimed behavior can be generally represented by languages, the timed one is represented by dater functions which provide the occurrence time of all the possible events in the system. As discussed in \cite{5}, timed Petri Nets models cannot represent scheduling problems in decision-free manufacturing systems. It is well known that the
subclass of timed event graphs can be modeled as a finite dimensional recurrent linear systems over the (max, +) algebra. Thus, where a state representation is obtained the (max, +) algebra model can be used to describe and analyze the behavior of the graphical model. Moreover, using this analytical representation enables us to control the system and to evaluate its performances. Indeed, as demonstrated in [5] [6], using (max, +) algebra a decision-free manufacturing system is similar to the one employed in Continuous Variable Dynamic Systems (CVDS). The obtained models are linear in the sense of (max, +) algebra and allow the formulation and the application of control concepts such as tracking problems, predictive control [7] [8] [9] [10] [11], stability, controllability, and observability [12] in a similar manner to that of CVDS. For more details on (max, +) algebra in DEDS modeling and control, we refer the reader to [5] [13] [14].

During the last few decades, several researchers have tried to represent and solve scheduling problems in dynamic systems with non-free decision using (max, +) algebra. In [15], an optimal scheduler for a printer has been presented. The scheduling is based on the max-plus modeling framework. It allows modeling the schedule of multiple sheets as discrete events in a system described by max-plus linear state-space equations. The optimal scheduler uses the feeding and handling times of each sheet as the design variables. It is shown that the proposed method successfully finds the overall optimal schedule for different types of the sheets. Also, max-plus algebra has been used for the rescheduling of trains on a large-scale railway network in the case of perturbations [16]. The authors studied the structure of the matrix system and then described how this structure can be manipulated by the control variables. In addition, it has been shown that this leads to an affine matrix system depending on the control variables and the temporal parameters. The application of (max, +) algebra has been extended to flow-shop manufacturing systems scheduling. Yardakul and Odrey proposed a dioid algebraic model for flow-shop scheduling [17]. A decision making capability has been made to the developed dioid algebraic model. With this new capability, sequencing decisions at each machine could be represented in the model. A lower and upper bounds approach for the $F_2/maxdelay/C_{max}$ problem has been introduced in [18]. These bounds allow the building of a branch-and-bound procedure for the two-machine flow-shop scheduling. The same authors proposed in [19] a polynomial time optimization algorithm for $2 \times 2$ (max, +) matrices products. It has been shown that the problem under consideration generalizes numerous scheduling problems, like single machine problems or two-machine flow-shop problems. Also, the authors showed that for $3 \times 3$ matrices, the problem is NP-hard and a branch-and-bound algorithm has been proposed to solve it. In [20], a mathematical formulation for cyclic permutation flow-shop problem using max-plus algebra has been proposed. The authors showed that this formulation makes it easier to compute the period of a cyclic system and can be used to evaluate the solutions in a cyclic flow-shop scheduling problem. Another subclass of manufacturing systems is widely studied in the literature, it is known as Job-Shop systems. The Job-Shop scheduling problems are proven to be NP-hard. Max-plus algebra has been used to solve cyclic job-shop problem in [21]. Cury et al. proposed a (max, +) model dealing with job-shop problems with time lags [22]. As far as we know, there is no research dealing with HLVV scheduling systems subject to maintenance using (max, +) algebra until now.

The objective of this paper is to propose an analytical formulation for scheduling of Job-Shop HLVV systems with maintenance tasks using the (max, +) algebra In order to deal with sequencing decisions, control variables have been introduced in the model according to an optimization procedure where performance criteria are minimized. These control variables are used to solve the conflicts between concurrent operations processed on the same machine. Moreover, Preventive Maintenance (PM) is integrated into the proposed model. Compared to the conventional
Due to the introduction of decision control variables, the proposed scheduling model is non-linear in the sense of (max, +) algebra. This non-linearity is caused by the multiplication between the control vector $V$ and the state vector $X$. So we cannot deal with this model as a classical linear (max, +) model proposed in the literature. We cannot use residuation theory to directly calculate the control vector $V$ in dioid algebra. It is important to note here that, in the (max, +) literature, no generic solution techniques are available for non-linear systems [5]. So, the scheduling (max, +) optimization problem is transformed as a conventional non-linear mathematical optimization problem with constraints.

Generally, production scheduling is done to allocate a limited set of resources to a limited number of jobs thus optimizing the system’s performances according to one or more criteria where various constraints are taken into consideration. In order to deal with sequencing decisions, control variables must be introduced in the model [25]. Moreover, most research dealing with job-shop problems assume that machines are always available. However, in most real life industrial settings a machine can be unavailable for many reasons, such as unforeseen breakdowns (stochastic unavailability) or due to a scheduled Preventive Maintenance (PM) where the period of unavailability is known in advance (deterministic unavailability). The problem of integrating production and preventive maintenance has been generally approached in the literature in two different ways. Some authors approached this problem by determining the optimal preventive maintenance schedule in the production system and others by taking maintenance as a constraint to the production system [26] [27] [28].

In this paper, a (max, +) algebraic model is developed to generate all feasible schedules by choosing different values for decision variables and according to the optimization procedure where performance criteria are minimized. Moreover, Preventive Maintenance (PM) is integrated into the proposed model [29] [30]. Compared to the methodology given in [31] and [32], where the authors consider equal durations of maintenance tasks in the case of single machine scheduling, in this article all the considered machines are subject to preventive maintenance and the allocated times for maintenance operations can be different on each machine. Moreover, the sequencing of maintenance tasks to their corresponding machines is not known in advance. Indeed, the (max,+) model gives an analytic representation where only sequencing decisions type are needed to solve the conflicts between concurrent operations. In this framework, the sequencing of operations and maintenance activities are determined by adjusting the decision control variables according to the minimization of specified performance criteria (makespan).

When considering a HVLV manufacturing system where its dynamic is represented by a (max, +) algebra model, the originality of this work with regard to other existing methods lies in the answers given to the following questions:

- Given a desired performance criterion, is it possible to synthesize a (max, +) algebra analytical scheduling methodology able to guarantee good performance (a minimal value of the makespan) according to the specified criterion optimization? Therefore, our intent in this article is to find a schedule subjected to preventive maintenance that minimizes the makespan. The proposed scheduling methodology is based on an algebraic (analytical) technique using (max,+) algebra.

- Is it possible to solve the scheduling problem of HVLV systems as a mathematical programming problem according to the choice of underlying decision variables? In other words, how the decision variables are handled in the optimization procedure according to the constraints imposed by the context?
• How can a preventive maintenance policy be incorporated in the scheduling methodology?
• What is the influence of this preventive maintenance on the scheduling strategy performance?

This article is proposed with two main objectives. The first and the most important is to give the answers to the previous questions where the advantages of using (max, +) algebra in scheduling of HVLV manufacturing systems are illustrated. The second is clearly to exhibit the theoretical assumptions that are required to have a well-defined scheduling problem when (max, +) algebra is used to represent HVLV dynamic systems.

The remainder of this article is organized as follows: after introducing some definitions and notations in section 2, we focus, in section 3, on the problem statement. In section 4, the scheduling methodology is proposed. Next, examples are presented to show the effectiveness and the feasibility of the proposed method. Concluding remarks and future research directions are presented in section 6.

2. Concepts and notations

For the sake of rigor and clarity, let us define the basic notions and notations used in the model.

2.1. (Max, +) algebra

The purpose of this section is not to review the properties of (max, +) algebra, but merely to introduce the necessary notations and discuss the aspects which will be used in modelling. The reader may refer to [4] and other references therein for details of (max, +) algebra properties.

A dioid $D$ is a set provided with two laws of intern composition $\oplus$ (addition) and $\otimes$ (multiplication) with the following equivalences for the traditional algebra (table 1), where $\epsilon$ and $e$ are respectively the neutral elements of $\oplus$ and $\otimes$.

<table>
<thead>
<tr>
<th>Classical algebra</th>
<th>Max</th>
<th>+</th>
<th>$-\infty$</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Max,+) algebra</td>
<td>$\oplus$</td>
<td>$\otimes$</td>
<td>$\epsilon$</td>
<td>$e$</td>
</tr>
</tbody>
</table>

2.1.1. Matrix operations

Max-plus algebra can be used for matrix operands $A, B$ likewise with the same conditions used in classical algebra. To perform the $A \oplus B$ operation, the elements of the resulting matrix at (row $i$, column $j$) have to be set up by the maximum operation of both corresponding elements of the matrices $A$ and $B$:

$[A \oplus B]_{ij} = [A]_{ij} \oplus [B]_{ij} = \max([A]_{ij}, [B]_{ij})$

The $\otimes$ operation is similar to the algorithm of matrix multiplication in classical algebra, however, every “+” calculation has to be substituted by an $\oplus$ operation and every “.” calculation by a $\otimes$ operation. More precisely, to perform the $A \otimes B$ operation, where $A$ is a $m \times p$ matrix and $B$ is a $p \times n$ matrix, the elements of the resulting matrix at (row $i$, column $j$) are determined by matrices $A$ (row $i$) and $B$ (column $j$):

$[A \otimes B]_{ij} = \bigoplus_{k=1}^{p} [A]_{ik} \otimes [B]_{kj} = \max([A]_{i1} + [B]_{1j}, \ldots, [A]_{ip} + [B]_{pj})$
2.1.2. (Max, +) Algebra properties

- **Associativity:**
  \[(a \oplus b) \oplus c = a \oplus (b \oplus c)\]
  \[(a \otimes b) \otimes c = a \otimes (b \otimes c)\]

- **Commutativity:**
  \[a \oplus b = b \oplus a\]
  \[a \otimes b = b \otimes a\]

- **Distributivity:**
  \[(a \oplus b) \otimes c = a \otimes c \oplus b \otimes c\]

2.1.3. Notations

Notations used in the model:

- **T**: current period.
- **T − 1**: last period.
- **m**: number of machines.
- **l**: number of jobs.
- **r**: total number of operations in the system.
- **q**: number of maintenance tasks on machine \(k\).
- **s**: total number of maintenance tasks in the system.
- **k**: machine index.
- **i**: job index.
- **j**: operation index.
- **M_k**: machine \(k\).
- **J_i**: job \(i\).
- **O_{ijk}**: operation \(j\) for job \(i\) on machine \(k\).
- **u_i**: availability time of raw materials used for the production of job \(i\).
- **x_{ijk}**: start time of \(O_{ijk}\).
- **y_{ijk}**: completion time of \(O_{ijk}\).
- **p_{ijk}**: processing time of \(O_{ijk}\).
- **PM_{hk}**: maintenance task \(h\) on machine \(k\).
- **x_{hk}**: start time of \(PM_{hk}\).
- **y_{hk}**: completion time of \(PM_{hk}\).
• \( t_{hk} \): duration of \( PM_{hk} \).
• \( \Delta_k \): time interval (period) between two consecutive repetitive periodic maintenance tasks.
• \( \Delta_{hk,zk} \): time interval (period) between two consecutive flexible periodic maintenance tasks \( PM_{hk} \) and \( PM_{zk} \) \((h \neq z)\).

Decision variables:
• \( v_{ijk';j''k} \): decision variable between \( O_{ijk} \) and \( O_{ij'k''} \) \((i' \neq i \text{ et } j'' \neq j)\).
• \( v_{ijk;hk} \): decision variable between \( O_{ijk} \) and \( PM_{hk} \).
• \( v_{hk;zk} \): decision variable between \( PM_{hk} \) and \( PM_{zk} \) \((h \neq z)\).

2.2. HLV system description

A HLV system is usually viewed as a network of machines and buffers having job-shop characteristics \([2]\). Let us consider a HLV system composed of a set of \( m \) machines \( \{M_k\} \), \((k = 1, \ldots, m)\) and a set of \( l \) jobs \( \{J_i\} \), \((i = 1, \ldots, l)\). Each job is characterized by a route composed of \( r \) operations. In this class of systems, each job has its own routing. According to \([33]\), HLV systems employ functional layout configurations whereby equipment carrying out the same type of processing is grouped together and positioned in distinct areas of the shop-floor. The machines \( M_k, k \in \{1, \ldots, m\} \) form the load centers for identical functions and the stocks represent outstanding locations. In this kind of system, the two prevalent production control modes are push and pull with their names pointing to the way the system responds to actual customer demand.

For the sake of simplicity and without loss of generality, in this article a square HLV Job-Shop system operating in push production mode is assumed. Moreover, the proposed methodology can still be used for non square HLV Systems. In this situation, the number of products, the number of machines and the number of operations are equal \((r = m = l)\). Each job \( \{J_i\} \) consists of a sequence of \( m \) operations on \( m \) different machines (routes). When considering \( x_{ijk}(T) \) to be the starting time of the operation \( O_{ijk} \) in period \( T \), then the completion time \( y_{ijk}(T) \) of the operation \( O_{ijk} \) in period \( T \) is equal to its processing time \( p_{ijk} \) plus \( x_{ijk}(T) \), i.e., \( y_{ijk}(T) = x_{ijk}(T) + p_{ijk} \). Let \( u_i(T) \) be the date at which the raw material of job \( J_i \) is fed into the system for the \( T^{th} \) time.

The HLV system considered for modelling takes into account the following properties and constraints:

• The transfer times of products between the different machines are equal to zero.
• The set-up times are not considered. Indeed, this hypothesis is inconsistent with the characteristics of HLV systems that require frequent change-over and set-up times. However, the aforementioned assumption was considered in order to simplify the model equations.
• Unique routes for jobs are given.
• The case of breakdowns and subsequent repairs of machines is not considered.
• A machine has a single capacity, thus it processes one task at a time.
• The durations allocated to the maintenance activities are given and not necessarily equal.
• The sequencing order of maintenance tasks on machines is not known in advance.
3. (Max, +) algebra for HVLV systems modelling

In order to solve the scheduling problem of a square HVLV system, we focus on the design of a (max, +) algebraic model. The overall model is composed of two interactive sub-models. The first one represents the dynamic behavior of the HVLV system without maintenance and the second one aims at describing the dynamic behavior of the maintenance operations.

For the sake of clarity, the sub-model without maintenance is presented first. Then, the sub-model of maintenance behavior is detailed. Finally, the unification of the two sub-models and their interactions are proposed.

3.1. (Max, +) sub-model without maintenance

For a given period \( T \), the model is presented in the form of dynamic equations describing the times at which an operation starts and ends. Where the scheduling problem has to be solved, the result concerning period \((T - 1)\) is already known, thus in order to have a simplified model, the vector \( X(T - 1) \) is denoted by \( X(0) \), and the period \((T)\) is omitted from all the vectors that are dependent \((X, Y, U, V)\).

Then if the following three conditions are satisfied, the operation \( O_{ijk} \) can start on machine \( M_k \):

1. All the preceding operations of \( O_{ijk} \) are processed: the precedence constraints of operations for the same job are respected and all the operations \( O_{ij'k} (i' \neq i \text{ and } j' \neq j) \) requiring a processing on the same machine \( M_k \) are carried out. The sequence between \( O_{ijk} \) and \( O_{ij'k} \) is determined by the decision variable \( v_{ijk,ij'k} \).

2. The machine concerned is available to execute \( O_{ijk} \) (the machine must be released of all products carried out).

3. The dates \( u_i \) at which the raw materials are fed to the system are respected.

According to the precedence constraint of operation \( O_{ijk} \) two situations can be distinguished:

- **Case 1:**
  - If operation \( O_{ijk} \) is the first operation of the job \( J_i \), then its processing start time \( x_{ijk} \) is determined by the maximum of either \( x_{ij'k}(0) + p_{ij'k}, x_{ij'k} + p_{ij'k} + v_{ij'k,ij''k} \) or \( u_i \). \( x_{ij'k}(0) \) is the starting time of all operations \( O_{ij'k} \) executed on machine \( M_k \). \( x_{ij'k} \) is the starting time of operation \( O_{ij'k} \). An expression for this situation can be formalized mathematically as follows:
    \[
    x_{ijk} = \max(x_{ij'k}(0) + p_{ij'k}, x_{ij'k} + p_{ij'k} + v_{ij'k,ij''k}, u_i) \tag{1}
    \]
  - Using the \((\max, +)\) notations, the equation \((1)\) becomes:
    \[
    x_{ijk} = x_{ij'k}(0) \oplus p_{ij'k} \otimes x_{ij'k} \oplus p_{ij'k} \otimes v_{ij'k,ij''k} \oplus u_i \tag{2}
    \]

- **Case 2:**
  - If operation \( O_{ijk} \) is not the starting operation on the job \( J_i \), then the expression of \( x_{ijk} \) is determined by the maximum of either \( x_{ij'k}(0) + p_{ij'k}, x_{ij'k} + p_{ij'k} + v_{ij'k,ij''k}, x_{ij''k} + p_{ij''k} + v_{ij''k,ij''k} \) or \( u_i \).
  - \( x_{ij''k} + p_{ij''k} + v_{ij''k,ij''k} \) is the starting time of the direct preceding operation \( O_{ij''k} \) of \( O_{ijk} \). \( O_{ij''k} \) is the \((j - 1)th\) operation carried out on machine \( t \) and \( O_{ijk} \) is the \(pth\) operation carried out on the same job \( i \) executed on machine \( k (k \neq t) \). It means that: \( x_{ijk} \geq x_{ij''k} + p_{ij''k} + v_{ij''k,ij''k} \).
  - An expression for this situation can be formalized mathematically as follows:
    \[
    x_{ijk} = \max(x_{ij'k}(0) + p_{ij'k}, x_{ij'k} + p_{ij'k} + v_{ij'k,ij''k}, x_{ij''k} + p_{ij''k} + v_{ij''k,ij''k}) \tag{3}
    \]
The equation (3) can be rewritten using (max, +) algebra as follows:

\[ x_{ijk} = x'_{ijk}(0) \otimes p'_{ijk} \oplus x'_{ijk-1} \otimes p'_{ijk-1} \oplus x_{ijk} \otimes p_{ijk} \otimes v_{ijk} \]  \hspace{1cm} (4)

In both cases the completion time of \( O_{ijk} \) is determined by the following equation:

\[ y_{ijk} = x_{ijk} + p_{ijk} \] \hspace{1cm} (5)

Using (max,+ ) notations, the equation (5) becomes:

\[ y_{ijk} = x_{ijk} \otimes p_{ijk} \] \hspace{1cm} (6)

In equations 1–4 and the rest of the paper, the value of \( x_{ijk} \) is the maximization over all \( i, j, k \) such that \( i \neq i' \) and \( j \neq j' \). This means that the start date of operation \( j \), for job \( i \), on machine \( k \) depends on three quantities:

1. At the beginning of a new schedule, machine \( k \) must be idle \((x'_{ijk}(0) + p'_{ijk})\).
2. The preceding operation (if there is one) on the same job \( i \) is completed \((x_{ijk-1} + p_{ijk-1})\).
3. All the other operations, for all the other jobs, that need to be performed on the same machine \( k \) \((x''_{ijk} + p''_{ijk} + v_{ijk})\).

Equations (2) and (4) show a multiplication in the terms of (max+, +) algebra between the decision variables \( v_{ijk} \) and the state variables \( x_{ijk} \). This multiplication causes a non-linearity in the model from a (max,+ ) algebra point of view. The absence of techniques for solving non-linear models in (max,+ ) algebra makes direct problem resolution, as is done in classical algebra, impossible. Indeed, the resolution techniques available in (max,+ ) can only resolve linear systems, the “Residuation” is one of these techniques.

3.2. (Max, +) sub-model of maintenance

In this section, the HVLV system is subject to a maintenance policy. Let us assume \( PM_{hk} \) be the preventive maintenance \( h \) on machine \( k \) and \( t_{hk} \) is its duration. In this case, its starting time and its completion time are respectively denoted \( x_{hk} \) and \( y_{hk} \).

By adopting the same principle detailed in section 3.1 two decision control variables denoted \( v_{ijk,hk} \) and \( v_{hk,zk} \) are used. \( v_{ijk,hk} \) is used to solve and control the conflicts between operations \( O_{ijk} \) and maintenance tasks \( PM_{hk} \) processed on the same machine \( M_k \). \( v_{hk,zk} \) is used to solve and control the conflicts between maintenance activities \( PM_{hk} \) and \( PM_{zk} \).

We consider two kinds of preventive maintenance, i.e., repetitive and flexible periodic maintenance. In order to specify the nature of the maintenance, the parameter \( \Delta_{ijk} \) is introduced. Indeed:

- If \( \Delta_{ijk} = \Delta_k, k = 1, \ldots, q \), then the maintenance is repetitive, i.e. the time intervals between two consecutive maintenance tasks are equal (Figure 1).
- If \( \Delta_{ijk} = \Delta_{ijk,k}, i, j = 1, \ldots, q, i \neq j \), then the maintenance is flexible, i.e. the time intervals between two consecutive maintenance tasks may not be equal (Figure 2).
In order to describe the dynamic of a maintenance task $PM_{hk}$ on a machine $M_k$, the following two conditions must be satisfied. The \textit{maximum} of these conditions provides the starting time of the maintenance activity $PM_{hk}$:

1. All the preceding operations and maintenance tasks of $PM_{hk}$ ($O_{ijk}$ and $PM_{h'} (h' \neq h)$) are carried out in period 0 on machine $M_k$.
2. The machine concerned is available to carry out $PM_{hk}$ (the machine must be released of all products carried out).

Consequently, the dynamics of maintenance operations can be described by the following equation:

$$
\begin{align*}
    x_{hk} &= \max(x_{ijk}(0) + p_{ijk}; x_{zjk}(0) + t_{zk}; x_{ijk} + p_{ijk} + v_{hk;jjk}; x_{zjk} + t_{zk} + \Delta_{i} + v_{hk;zk}) \\
    y_{hk} &= x_{hk} + t_{hk}
\end{align*}
$$

\(7\)

Using (max,+) notations, the equation (7) can be written as follows:

$$
\begin{align*}
    x_{hk} &= x_{ijk}(0) \odot p_{ijk} \odot x_{zjk}(0) \odot t_{zk} \odot x_{ijk} \odot p_{ijk} \odot v_{hk;jjk} \odot x_{zjk} \odot t_{zk} \odot \Delta_{i} \odot v_{hk;zk} \\
    y_{hk} &= x_{hk} \odot t_{hk}
\end{align*}
$$

\(8\)

3.3. (Max, +) overall model

In order to build the (max,+) overall scheduling model subject to preventive maintenance, the two sub-models proposed in sections 3.1 and 3.2 are unified.
In this model, the dynamics of $O_{ijk}$ described by its starting time $x_{ijk}$ is obtained by the augmentation of the sub-model without maintenance.

In order to consider the maintenance in the model, the expression of $x_{ijk}$ in equations (1) and (3) is augmented by the term “$x_{hk} + t_{hk} + V_{ijk, hk}$”. This term represents the sequence between $O_{ijk}$ and $PM_{hk}$. This relation is determined by the decision variable $v_{ijk, hk}$.

We find the following two situations:

**Case 1:** if $O_{ijk}$ is the first operation on the job $J_i$, then its processing start time $x_{ijk}$ is determined by the following expression:

$$x_{ijk} = \max(x_{ijk}(0) + p_{ijk}; x_{hk}(0) + t_{hk}; x_{r,j-k} + p_{r,j-k} + V_{ijk, r,j-k}; u_{ij}; x_{hk} + I_{hk} + v_{ijk, hk}) \quad (9)$$

Using the (max, +) notations, the equation (9) becomes:

$$x_{ijk} = x_{ijk}(0) \otimes p_{ijk} \otimes x_{hk}(0) \otimes t_{hk} \otimes x_{r,j-k} \otimes p_{r,j-k} \otimes v_{ijk, r,j-k} \otimes u_{ij} \otimes x_{hk} \otimes I_{hk} \otimes v_{ijk, hk} \quad (10)$$

**Case 2:** If operation $O_{ijk}$ is not the starting operation on the job $J_i$, then the expression of $x_{ijk}$ is determined by the following expression:

$$x_{ijk} = \max(x_{ijk}(0) + p_{ijk}; x_{hk}(0) + t_{hk}; x_{i(j-1)r} + p_{i(j-1)r}; x_{r,j-k} + p_{r,j-k} + V_{ijk, r,j-k}; u_{ij}; x_{hk} + I_{hk} + v_{ijk, hk}) \quad (11)$$

The equation (11) can be written in (max, +) algebra as follows:

$$x_{ijk} = x_{ijk}(0) \otimes p_{ijk} \otimes x_{hk}(0) \otimes t_{hk} \otimes x_{i(j-1)r} \otimes p_{i(j-1)r} \otimes x_{r,j-k} \otimes p_{r,j-k} \otimes v_{ijk, r,j-k} \otimes x_{hk} \otimes I_{hk} \otimes v_{ijk, hk} \quad (12)$$

In both cases the completion time of $O_{ijk}$ is determined by the following equation:

$$y_{ijk} = x_{ijk} + p_{ijk} \quad (13)$$

Using (max, +) notations, the equation (13) becomes:

$$y_{ijk} = x_{ijk} \otimes p_{ijk} \quad (14)$$

The model equations (10) and (12) can be summarized in the following compact form (Figure 3):

$$\begin{align*}
    X &= A \otimes X(0) \otimes B \otimes U \otimes V \otimes X \\
    Y &= C \otimes X
\end{align*} \quad (15)$$

where:

- $X$ is the state vector of the system. Its dimension is $(N \times 1)$. It collects the starting times of operations or jobs $J_i$ and maintenance tasks $PM_{hk}$ ($N = s + r$ is the total number of operations for the jobs and maintenance activities).
- $X(0) = \epsilon$. 

10
- $U$ is the input vector. Its dimension is $(l \times 1)$. It collects the dates at which the raw materials are fed into the system ($l$ is the total number of jobs).
- $Y$ is the output vector. Its dimension is $(N \times 1)$. It is composed of the completion times of operations.
- $A \in \mathbb{R}^{N \times N}$ is an appropriate square matrix whose elements are in $(\max, +)$ algebra.
- $B \in \mathbb{R}^{N \times l}$ is an appropriate $(\max, +)$ matrix.
- $C \in \mathbb{R}^{N \times N}$ is a square matrix composed of the processing times of operations for jobs and maintenance tasks and $\epsilon$.
- $V$ is a $(\max, +)$ control matrix. Its dimension is $(N \times N)$. $V$ is composed of all the decision variables, the processing times and $\epsilon$.

Equation (15) calculates the starting and ending times for all jobs and all maintenance activities in period $T$. Once the obtained production planning is performed in the workshop for period $T$, a new calculation is started using the appropriate model and data for a new production program that can be completely different from those of the previous periods. Then, operations may be scheduled differently in the new period to minimize the makespan.

![Figure 3: Schematic representation of HVLV model with maintenance](image)

**Example:** The construction of the model represented by equation (15) is illustrated by a simple example. This example is a non-free decision HVLV system with two machines ($M_1$ and $M_2$) and two kinds of products ($P_1$ and $P_2$) (Figure 4).

![Figure 4: HVLV system with two kinds of products](image)

In this example, we assume that the system is empty and has no jobs at the beginning and that machine $M_2$ is subject to two repetitive periodic maintenance tasks $PM_{12}$ and $PM_{22}$. Let:
- $\Delta_2$ be the time interval between maintenance activities,
- $x_{12}$ be the starting time of $PM_{12}$ and $x_{22}$ the starting time of $PM_{22}$ and
The above equations can be written in (max, +) algebra as follows:

\[ x_{11} = \max(x_{111}(0) + p_{111}; u_1) \]

\[ x_{22} = \max(x_{122}(0) + p_{122}; x_{212}(0) + p_{212}; x_{12}(0) + t_{12}; x_2(0) + t_{22}; x_{111} + p_{111}; x_{212} + p_{212} + v_{122,12}; x_{12} + t_{12} + v_{122,12}; x_{22} + t_{22} + v_{122,22}) \]

\[ x_{212} = \max(x_{122}(0) + p_{122}; x_{212}(0) + p_{212}; x_{12}(0) + t_{12}; x_2(0) + t_{22}; x_{122} + p_{122} + v_{122,12}; x_{12} + t_{12} + v_{122,12}; x_{22} + t_{22} + v_{122,22}) \]

\[ x_{12} = \max(x_{122}(0) + p_{122}; x_{212}(0) + p_{212}; x_{12}(0) + t_{12}; x_2(0) + t_{22}; x_{122} + p_{122} + v_{122,12}; x_{12} + p_{212} + v_{22,12}; x_{12} + t_{12} + \Delta_2 + v_{22,12} \]

\[ y_{11} = x_{111} + p_{111} \]

\[ y_{12} = x_{122} + p_{111} \]

\[ y_{212} = x_{212} + p_{212} \]

\[ y_{12} = x_{12} + t_{12} \]

\[ y_{22} = x_{22} + t_{22} \]

The above equations can be written in (max, +) algebra as follows:

\[ x_{111} = x_{111}(0) \otimes p_{111} \otimes u_1 \]

\[ x_{212} = x_{212}(0) \otimes p_{122} \otimes x_{12}(0) \otimes p_{212} \otimes x_{12}(0) \otimes t_{12} \otimes x_{22}(0) \otimes t_{22} \otimes x_{111} \otimes p_{111} \]

\[ x_{212} = x_{212}(0) \otimes p_{122} \otimes x_{12}(0) \otimes p_{212} \otimes x_{12}(0) \otimes t_{12} \otimes x_{22}(0) \otimes t_{22} \otimes x_{122} \otimes p_{122} \]

\[ x_{12} = x_{122}(0) \otimes p_{122} \otimes x_{12}(0) \otimes p_{212} \otimes x_{12}(0) \otimes t_{12} \otimes x_{22}(0) \otimes t_{22} \otimes x_{122} \otimes p_{122} \]

\[ x_{22} = x_{222}(0) \otimes p_{122} \otimes x_{12}(0) \otimes p_{212} \otimes x_{12}(0) \otimes t_{12} \otimes x_{22}(0) \otimes t_{22} \otimes x_{122} \otimes p_{122} \]
\[ \begin{align*}
  y_{111} &= x_{111} \otimes p_{111} \\
  y_{122} &= x_{122} \otimes p_{122} \\
  y_{212} &= x_{212} \otimes p_{212} \\
  y_{12} &= x_{12} \otimes t_{12} \\
  y_{22} &= x_{22} \otimes t_{22} 
\end{align*} \]

\( x_{111} \) is replaced by its value in the expression of \( x_{122} \). Using the distributivity of \( \otimes \) over \( \oplus \), the following representation is obtained:

\[
X = \begin{bmatrix}
  x_{111} \\
  x_{122} \\
  x_{212} \\
  x_{12} \\
  x_{22}
\end{bmatrix}
\]

\[
U = \begin{bmatrix}
  u_1 \\
  u_2
\end{bmatrix}
\]

\[
Y = \begin{bmatrix}
  y_{111} \\
  y_{122} \\
  y_{212} \\
  y_{12} \\
  y_{22}
\end{bmatrix}
\]

\[
A = \begin{bmatrix}
  p_{111} & \epsilon & \epsilon & \epsilon & \epsilon \\
  p_{111} \otimes p_{111} & p_{122} & p_{212} & t_{12} & t_{22} \\
  \epsilon & p_{122} & p_{212} & t_{12} & t_{22} \\
  \epsilon & p_{122} & p_{212} & t_{12} & t_{22} \\
  \epsilon & p_{122} & p_{212} & t_{12} & t_{22}
\end{bmatrix}
\]

\[
B = \begin{bmatrix}
  \epsilon & \epsilon \\
  \epsilon \\
  \epsilon & \epsilon \\
  \epsilon & \epsilon \\
  \epsilon & \epsilon \\
  \epsilon & \epsilon & \epsilon & t_{12} & \epsilon \\
  \epsilon & \epsilon & \epsilon & \epsilon & t_{22}
\end{bmatrix}
\]

The control matrix \( V \) is as follows:

\[
\begin{bmatrix}
  \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\
  \epsilon & \epsilon & p_{212} \otimes v_{122;212} & t_{12} \otimes v_{122;12} & t_{23} \otimes v_{122;22} \\
  \epsilon & p_{122} \otimes v_{122;12} & \epsilon & t_{23} \otimes v_{122;22} \\
  \epsilon & p_{122} \otimes v_{122;12} & p_{212} \otimes v_{122;212} & \epsilon & t_{23} \otimes v_{12;22} \\
  \epsilon & p_{122} \otimes v_{122;12} & p_{212} \otimes v_{122;212} & t_{12} \otimes \Delta_2 \otimes v_{22;12} & \epsilon
\end{bmatrix}
\]
4. Proposed scheduling methodology

As shown in section 3, the developed (max, +) algebraic model (equation (15)) is used in formulating the HVLV scheduling problem. A state-space scheduling control model is investigated via (max, +) algebra. In the case of makespan minimization, the vector $U$ gives information about the date of the availability of raw materials used to manufacture the jobs. Therefore, it determines the start date of production on the machines concerned by the first operations of routes. While, the variable $V$ is used to determine the sequence of operations on machines. In this section, the scheduling control model is used to solve a non-linear optimization problem subject to constraints in order to achieve the following two objectives:

1. Generation of a feasible schedule.
2. Performance of production measures (the makespan is minimised).

4.1. Generation of a feasible schedule

In the proposed (max, +) model (equation (15)), the decision variables $v_{ijk}$, $v_{hk}$, and $v_{hkz}$ allow the optimizer to allocate time in a production planning horizon to process the jobs $J_i$ and maintenance tasks $PM_j$ on their corresponding machines. They are incorporated into the model as conflict resolvers. Their values determine the starting times $x_{ijk}$ and $x_{hk}$. On the one hand, if $v_{ijk}$ is equal to $e$, then $O_{ijk}$ is processed before $O_{ij'}k_j$ on $M_k$. On the other hand, if $v_{ijjk}$ is equal to $e$, then $O_{jk}$ is processed after $O_{ij'}k_j$. The same approach is considered for the decision variables $v_{hk}$, $v_{hkz}$, and $v_{hkz}$ to solve the conflicts between $O_{ijk}$ and $PM_j$ and between $PM_j$ and $PM_{jk}$ respectively. To get feasible schedules between job operations and maintenance activities competing on the same machine $M_k$, the different control variables must be bounded and must satisfy the following (max, +) relations:

$$\begin{align*}
\left\{ \begin{array}{l}
v_{ijk} + v_{ij'}k_j = -\infty \\
v_{hk} + v_{hk} = -\infty \\
v_{hkz} + v_{hkz} = -\infty
\end{array} \right. & \text{ and } \left\{ \begin{array}{l}
\max(v_{ijk}, v_{ij'}k_j) = 0 \\
\max(v_{hk}, v_{hk}) = 0 \\
\max(v_{hkz}, v_{hkz}) = 0
\end{array} \right.
\end{align*}$$

(16)

where $(i \neq i', j \neq j'$ and $z \neq h$).

Using (max, +) notation, the above equations become:

$$\begin{align*}
\left\{ \begin{array}{l}
v_{ijk} + v_{ij'}k_j = e \\
v_{hk} + v_{hk} = e \\
v_{hkz} + v_{hkz} = e
\end{array} \right. & \text{ and } \left\{ \begin{array}{l}
\max(v_{ijk}, v_{ij'}k_j) = e \\
\max(v_{hk}, v_{hk}) = e \\
\max(v_{hkz}, v_{hkz}) = e
\end{array} \right.
\end{align*}$$

(17)

The Constraints given in equations (17) apply to all $i'j'k$ not equal to $ijk$ $(i \neq i'$ and $j \neq j')$, i.e. the values of decision variables $v_{ijk}$, $v_{ij'}k_j$ and $v_{ijjk}$ generate the schedule between two concurrent operations $O_{ijk}$ and $O_{ij'}k_j$ to be processed on the same machine $k$.

4.2. Performance of production measures: non-linear optimisation for makespan minimisation

The model represented by equation (15) is non-linear in the terms of (max,+) algebra. This non-linearity is due to the multiplication between the state vector $X$ and the control matrix $V$. As far as we know, there are no techniques that can resolve this model in dioid algebra. All techniques proposed like residuation are used to solve linear (max,+) model for free-decision systems. For this reason, the control problem of HVLV systems is transformed in this section to a non-linear optimization problem subject to constraints.
The makespan criterion denoted $C_{max}$ is used in order to allow the finishing of the given set of jobs $J_i$ in a short period of time and to start the processing of new jobs associated with the following planning horizon. As a result, the highest utilization of the resources of the system is provided. The decision variables are $v_{ijk}$, $r_{ik}$, $w_{ijk}$, and $v_{ik}$. We suppose that $A \otimes X(0) = \epsilon$ and raw materials are available at the beginning of the planning period. That means: $u_i = 0 \forall i = 1 \ldots l$ (Figure 5), where $1 \leq i \leq l$, $1 \leq k \leq m$, $v_{(a)}$ represents the decision variables of the control matrix $V$ and $y_{ink} = x_{ink} + p_{ink}$ is the completion time of the job $J_i$.

As mentioned in section 3.2, two policies of maintenance integration are considered: Repetitive Periodic Maintenance (RPM) and Flexible Periodic Maintenance (FPM). Two cases are distinguished:

- **Case 1**: maintenance tasks are generated periodically with equal periods $\Delta_k$ on the corresponding machine $M_k$. In this case, we need to incorporate the period $\Delta_k$ in the scheduling model, such that the duration between two consecutive periodic maintenance activities is equal to $\Delta_k$. In addition, the first maintenance activity on machine $M_k$ must start at date $\Delta_k$. Moreover, the last RPM activity finishes at $\sum_{h=1}^{q} t_{hk} + \Delta_k q$. An expression for this situation can be formalized mathematically as follows:

$$\Delta_k \leq x_{hk} \leq \sum_{h=1,h \neq z}^{q} t_{zk} + \Delta_k q$$  \hspace{1cm} (18)

- **Case 2**: the time intervals between two consecutive flexible periodic maintenance activities $PM_{h,k}$ and $PM_{l,k}$, denoted $\Delta_{h,k,z}$ ($h,z = 1 \ldots q, h \neq z$) are not necessarily equal, but they are given. The starting date $\Delta_{0k}$ of the first flexible maintenance is known. Moreover, the duration between two consecutive FPM tasks must be respected. An expression for this situation can be formalized mathematically as follows:

$$\Delta_{0k} \leq x_{hk} \leq \Delta_{0k} + \sum_{h=1,h \neq z}^{q} t_{zk} + \max(\sum_{h=1,h \neq z}^{q} \Delta_{h,k,z})$$  \hspace{1cm} (19)

The term $\sum_{h=1,h \neq z}^{q} \Delta_{h,k,z}$ in the inequality (19) is an addition of $(q - 1)$ by $(q - 1)$ of periods $\Delta_{h,k,z}$.

If $\Delta_{0k} = \Delta_{h,k,z} = \Delta_k$ (the time intervals between the maintenance tasks are equal), then the inequality (19) is reduced to the inequality (18).

In this case, the optimisation problem given in Figure 5 can be formalized as follows:

$$J^* = \min_{y_{ink}} \{ C_{max} = \max_{y_{ink}} \}$$

subject to Equations (6), (11), (7), (16), (18) and (19).

The proposed optimization methodology is applied to find $V$ that minimizes $J$ (solving the traditional makespan problem). Without maintenance, the optimization problem is reduced and can be formalized as follows:

$$J^* = \min_{y_{ink}} \{ C_{max} = \max_{y_{ink}} \}$$

subject to Equations (1), (3) and the following constraints :
\begin{equation}
\begin{aligned}
V_{ijk}r'p'k \otimes V_{r'p'k,ijk} &= \epsilon \\
V_{ijk}r'p'k \otimes V_{r'p'k,ijk} &= \epsilon
\end{aligned}
\end{equation}

\forall i \neq i' \text{ and } j \neq j''.

Figure 5: Full diagram of makespan optimisation subject to maintenance

5. Examples of application

The \((\max,+)\) model has been tested and validated on several examples of production systems. In this section, the aim is first to apply the approach used to automatically generate the model equations for an example discussed in the literature and then to present the effectiveness of the model by comparing our simulation results with those obtained in literature [34]. For this purpose and, in order to be able to compare the obtained results, the proposed method is implemented using a software of non-linear optimization (LINGO). Indeed, the obtained performance for makespan optimisation is compared to the one produced by the method detailed in [34].

For the sake of simplicity and without loss of generality, an optimization problem in one production period is solved in this section. \(U\) is assumed to be known (it is equal to zero). In this example, the proposed non-linear programming approach calculates \(V\) for one period and then finds the corresponding feasible schedule that minimizes the makespan.

If the size of the system increases, then the number of equations in the proposed model explodes in a non-linear manner. Consequently, the automatic generation of the model is crucial. As mentioned in section 5.2, only square job-shop systems are considered in this article \((m = l = r)\). Consequently, the dimension of the state vector \(X\) is equal to \(m^2\). Moreover, the total number \(Q\) of decision variables \(V_{ijk}r'p'k\) in the control matrix \(V\) is equal to \((r - 1) \times r \times l\). As \(m = l = r\), then \(Q = (m - 1) \times m^2\). Consequently, the number of equations describing the dynamic of the system explodes which make their implementation very hard. For example, for a \(6 \times 6\) job-shop the number of constraints is equal to 607 and for a \(10 \times 10\) job-shop the problem is composed of 2891 constraints. For these reasons, the automatic generation of the model equations is necessary. An algorithm for equations generation is implemented using MATLAB.

The example considered in this section deals with a square job-shop scheduling problem [34]. This example is a very complicated problem. It is an industrial (10x10) HLV system with job-shop characteristics. There is a wide variety of products (10 kinds of products). Each job has a specific route on the machines. There are ten operations per each type of job. Consequently, the size of the state vector \(X\) in equation (15) is \(100 \times 1\). Table 2 shows the different routes of jobs \(J_i\) and the different processing times of operations \(O_{ijk}\).

Two kinds of PM are considered (RPM and FPM). The affectation of maintenance to each machine is as follows:
Table 2: Production data of considered system

<table>
<thead>
<tr>
<th>Operation's index  \ Job</th>
<th>machine's index k/processing time p&lt;sub&gt;ijk&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>J&lt;sub&gt;1&lt;/sub&gt;</td>
<td>1/6</td>
</tr>
<tr>
<td>J&lt;sub&gt;2&lt;/sub&gt;</td>
<td>3/1</td>
</tr>
<tr>
<td>J&lt;sub&gt;3&lt;/sub&gt;</td>
<td>2/11</td>
</tr>
<tr>
<td>J&lt;sub&gt;4&lt;/sub&gt;</td>
<td>4/3</td>
</tr>
<tr>
<td>J&lt;sub&gt;5&lt;/sub&gt;</td>
<td>7/2</td>
</tr>
<tr>
<td>J&lt;sub&gt;6&lt;/sub&gt;</td>
<td>3/3</td>
</tr>
<tr>
<td>J&lt;sub&gt;7&lt;/sub&gt;</td>
<td>3/3</td>
</tr>
<tr>
<td>J&lt;sub&gt;8&lt;/sub&gt;</td>
<td>5/3</td>
</tr>
<tr>
<td>J&lt;sub&gt;9&lt;/sub&gt;</td>
<td>10/8</td>
</tr>
<tr>
<td>J&lt;sub&gt;10&lt;/sub&gt;</td>
<td>3/6</td>
</tr>
</tbody>
</table>

1. RPM:
   - on M<sub>1</sub>: PM<sub>11</sub> and PM<sub>21</sub>.
   - on M<sub>2</sub>: PM<sub>12</sub>, PM<sub>22</sub> and PM<sub>32</sub>.
   - on M<sub>3</sub>: PM<sub>15</sub>, PM<sub>25</sub> and PM<sub>35</sub>.
   - on M<sub>4</sub>: PM<sub>16</sub>, PM<sub>26</sub> and PM<sub>36</sub>.
   - on M<sub>5</sub>: PM<sub>17</sub>, PM<sub>27</sub>, PM<sub>37</sub> and PM<sub>47</sub>.
   - on M<sub>6</sub>: PM<sub>18</sub> and PM<sub>28</sub>.
   - on M<sub>7</sub>: PM<sub>19</sub>, PM<sub>29</sub> and PM<sub>39</sub>.
   - on M<sub>8</sub>: PM<sub>110</sub>, PM<sub>210</sub> and PM<sub>310</sub>.

2. FPM:
   - on M<sub>3</sub>: PM<sub>13</sub>, PM<sub>23</sub> and PM<sub>33</sub>.
   - on M<sub>4</sub>: PM<sub>14</sub>, PM<sub>24</sub> and PM<sub>34</sub>.

The durations t<sub>tkd</sub> allocated to the different maintenance tasks are shown in Table 3.

Table 3: Durations of maintenance activities

<table>
<thead>
<tr>
<th>Duration</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>t&lt;sub&gt;11&lt;/sub&gt;, t&lt;sub&gt;21&lt;/sub&gt;, t&lt;sub&gt;33&lt;/sub&gt;, t&lt;sub&gt;36&lt;/sub&gt;, t&lt;sub&gt;47&lt;/sub&gt;, t&lt;sub&gt;29&lt;/sub&gt;, t&lt;sub&gt;10&lt;/sub&gt;</td>
<td>4</td>
</tr>
<tr>
<td>t&lt;sub&gt;21&lt;/sub&gt;, t&lt;sub&gt;14&lt;/sub&gt;, t&lt;sub&gt;34&lt;/sub&gt;, t&lt;sub&gt;16&lt;/sub&gt;, t&lt;sub&gt;26&lt;/sub&gt;, t&lt;sub&gt;27&lt;/sub&gt;, t&lt;sub&gt;28&lt;/sub&gt;</td>
<td>3</td>
</tr>
<tr>
<td>t&lt;sub&gt;12&lt;/sub&gt;, t&lt;sub&gt;15&lt;/sub&gt;, t&lt;sub&gt;25&lt;/sub&gt;, t&lt;sub&gt;17&lt;/sub&gt;, t&lt;sub&gt;19&lt;/sub&gt;, t&lt;sub&gt;11&lt;/sub&gt;</td>
<td>2</td>
</tr>
<tr>
<td>t&lt;sub&gt;23&lt;/sub&gt;, t&lt;sub&gt;18&lt;/sub&gt;</td>
<td>5</td>
</tr>
<tr>
<td>t&lt;sub&gt;32&lt;/sub&gt;</td>
<td>7</td>
</tr>
<tr>
<td>t&lt;sub&gt;13&lt;/sub&gt;, t&lt;sub&gt;37&lt;/sub&gt;</td>
<td>1</td>
</tr>
<tr>
<td>t&lt;sub&gt;24&lt;/sub&gt;, t&lt;sub&gt;35&lt;/sub&gt;, t&lt;sub&gt;210&lt;/sub&gt;</td>
<td>6</td>
</tr>
<tr>
<td>t&lt;sub&gt;59&lt;/sub&gt;</td>
<td>7</td>
</tr>
</tbody>
</table>
Table 4 shows the different values of periods $\Delta_k$ for RPM and Table 5 shows the values of periods $\Delta_{0k}$ and $\Delta_{hk}$ for FPM.

### Table 4: Periods $\Delta_k$

<table>
<thead>
<tr>
<th>Period</th>
<th>$\Delta_1$</th>
<th>$\Delta_2$</th>
<th>$\Delta_5$</th>
<th>$\Delta_6$</th>
<th>$\Delta_7$</th>
<th>$\Delta_8$</th>
<th>$\Delta_9$</th>
<th>$\Delta_{10}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>30</td>
<td>25</td>
<td>17</td>
<td>20</td>
<td>15</td>
<td>10</td>
<td>19</td>
<td>40</td>
</tr>
</tbody>
</table>

### Table 5: Periods $\Delta_{0k}$ and $\Delta_{hk}$

<table>
<thead>
<tr>
<th>Period</th>
<th>$\Delta_{03}$</th>
<th>$\Delta_{04}$</th>
<th>$\Delta_{13,23}$</th>
<th>$\Delta_{13,33}$</th>
<th>$\Delta_{23,33}$</th>
<th>$\Delta_{14,24}$</th>
<th>$\Delta_{14,34}$</th>
<th>$\Delta_{24,34}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>15</td>
<td>20</td>
<td>7</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>9</td>
<td>15</td>
</tr>
</tbody>
</table>

When applying the non-linear optimization problem with maintenance for $A \otimes X(0) = \epsilon$ and $u_i = 0 \forall 1 \leq i \leq 10$, the optimal value of makespan is $C_{\text{max}}^* = 129$ time units. The Gantt chart of operations and maintenance tasks on the different machines is shown in Figure 6. The completion dates $C_i$ of jobs $J_i (i = 1 \ldots 10)$ are summarized in Table 6.

![Gantt chart](image)

**Figure 6:** Scheduling of operations and maintenance tasks

### Table 6: Completion dates of jobs

<table>
<thead>
<tr>
<th>Jobs</th>
<th>$J_1$</th>
<th>$J_2$</th>
<th>$J_3$</th>
<th>$J_4$</th>
<th>$J_5$</th>
<th>$J_6$</th>
<th>$J_7$</th>
<th>$J_8$</th>
<th>$J_9$</th>
<th>$J_{10}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_i$</td>
<td>100</td>
<td>129</td>
<td>127</td>
<td>125</td>
<td>122</td>
<td>116</td>
<td>129</td>
<td>129</td>
<td>115</td>
<td>115</td>
</tr>
</tbody>
</table>

Figure 6 shows that the results obtained by the proposed $(\text{max,} +)$ model are quite convincing.
which prove the validity of our model in the case of HVLV systems subject to maintenance. Indeed, the maintenance tasks constraints described in equations (18) and (19) are respected. The operations of jobs and the maintenance activities are scheduled simultaneously in order to make the makespan minimal. For example, machine $M_7$ is subject to 4 RPM maintenance tasks. First, we observe on the Gantt chart that the intervals between two consecutive RPM are equal and the value of the period is $\Delta_7 = 15$ time units. Secondly, as machine $M_4$ is subject to FPM, the periods between maintenance tasks are not equal which is coherent with the obtained simulation results. Indeed, the maintenance task $PM_{34}$ begins at $t = 20$. This value is equal to $\Delta_{04}$. Moreover, $PM_{34}$ ends at $t = 23$. The starting time $\chi_24$ of $PM_{24}$ is equal to 38. Consequently, the time interval between $PM_{34}$ and $PM_{24}$ is equal to 15. This value is equal to $\Delta_{14,24}$ proposed in Table 5. Moreover, the results obtained show that the completion date $\gamma_{24}$ of $PM_{24}$ is equal to 44 and the starting time $\chi_{14}$ of $PM_{14}$ is equal to 52. So, the period between $PM_{24}$ and $PM_{14}$ is equal to $\Delta_{14,24} = 8$. Consequently, the results are respected and show that the incorporation of maintenance into the phase of scheduling design does not affect the quality of the results.

When dealing with maintenance free scheduling by application of equation (21), the obtained optimal value of the makespan is $C_{max} = 85$ time units (Figure 7). The proposed optimization methodology was run on a DELL OPTIPLEX 380 (processor: 2.93 Ghz, RAM: 2048 Mo). The computational time to find the optimal solution is 10 hours. The Gantt chart shown in Figure 7 describes the completion times $C_i$ of the different jobs $J_i$ presented in Table 7. The optimised value of makespan (85 time units) is very interesting since it is much lower than the optimised values described in the literature. Indeed, the optimal values obtained using the Traditional and the Adaptive Modified Genetic Algorithms (AGA) are respectively 114 and 94 time units [35]. Furthermore, the optimal value obtained using the method of Improved Adaptive Genetic Algorithm (IAGA) [34] is equal to 92 time units. Hence, these results prove that the proposed (max,+)$^*$ approach is a more robust and stable solution.

<table>
<thead>
<tr>
<th>Jobs</th>
<th>$J_1$</th>
<th>$J_2$</th>
<th>$J_3$</th>
<th>$J_4$</th>
<th>$J_5$</th>
<th>$J_6$</th>
<th>$J_7$</th>
<th>$J_8$</th>
<th>$J_9$</th>
<th>$J_{10}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_i$</td>
<td>76</td>
<td>85</td>
<td>85</td>
<td>69</td>
<td>73</td>
<td>85</td>
<td>85</td>
<td>85</td>
<td>85</td>
<td>85</td>
</tr>
</tbody>
</table>

Table 7: Completion dates of products
6. Conclusion and feature research

The objective of this work is to build a \((\text{max}, +)\) algebraic model for the scheduling, optimization, and control of HVLV systems while periodic preventive maintenance is considered. Two situations concerning maintenance are investigated simultaneously in this article. In the first one, maintenance tasks are fixed periodically: maintenance is required after a periodic time interval (all periods are equal on each machine). In the second one, time intervals between two consecutive maintenance activities are not equal (flexible periodic maintenance). The jobs and both situations of maintenance operations are scheduled simultaneously. Moreover, the maintenance tasks are scheduled between them, so that a regular criterion is optimized.

A non-linear optimization problem with constraints is then solved using \((\text{max}, +)\) algebra to minimize the makespan. The simulation results show that the proposed model can be a good tool for the control and optimization of HVLV systems subject to maintenance.

In real-world applications for HVLV systems, various uncertainty aspects of the system may perturb its behavior (processing times, set-up times, breakdowns, etc). In this context, further research work will be done to improve the proposed model to make it more robust when confronted with unplanned events, in such a way it can deal with changeover in HVLV systems. Also, analytical techniques for differentiation and optimization in \((\text{max}, +)\) algebra should be developed to use dioid algebraic models in solving scheduling problems effectively.

References

Internationale de Modélisation, Optimisation et SIMulation (MOSIM 2012), Bordeaux, France.


