Scheduling of energy flows for parallel batch processes using max-plus systems

Mark Mutsaers, Leyla Özkan, Ton Backx *

* Control Systems group, Department of Electrical Engineering, Eindhoven University of Technology, P.O. Box 513, 5600MB, Eindhoven, The Netherlands
E-mail: {m.e.c.mutsaers, l.ozkan, a.c.p.m.backx}@tue.nl

Abstract: A common approach to increase reliability and efficiency of manufacturing systems is optimal scheduling. In this paper, we consider scheduling problems for a specific type of systems where batch processes are running in parallel. The goal is to schedule the interaction between the parallel processes, in the form of energy flows, to minimize a desired objective. We make use of a max-plus approach in the formulation of the scheduling problem and use this formulation in the optimization subsequently. This framework has been used to formulate and optimize a case study for the production of calcium silicate stones.

Keywords: scheduling problems, parallel batch processes, energy exchange, max-plus algebra, optimization

1. INTRODUCTION

Manufacturing systems are becoming complex since they consist of a large number of complex processes. In order to deal with this complexity, advanced control and scheduling techniques are used to ensure that the complete production process runs in a more efficient and reliable way than controlling and making decisions manually. The questions of reliability and efficiency in manufacturing processes is becoming more important considering the growing demands of economic and environmental constraints. Therefore, in this article, we consider an energy efficiency problem for a manufacturing system consisting of several batch processes, running in parallel.

These batch processes can interact with each other, and therefore influence the production of different batches in these parallel processes. A possible interaction between the processes can be the exchange of energy, which then will be used as the decision variable in the scheduling problem. This kind of problems can be represented using complex discrete event systems, which in general lead to a nonlinear description when using the conventional algebra. Therefore, we propose to use the max-plus approach which results in systems that are “linear” in the max-plus algebra.

Discrete event systems are already being represented using this approach by a number of researchers as e.g. Giffler [1960] or Cuninghame-Green [1979]. An overview of obtained results can also be found in Baccelli [1992]. Due to the linear property of these systems, we can use more efficient analytic methods that are available for simulation, optimization and analyzing scheduling problems. We will show results for a case study, where the production of calcium silicate stones is done in parallel batches, and where the steam flow between the processes will be the scheduling variable.

The outline of the paper is as follows. First, the scheduling problem is explained and formulated in Section 2. Section 3 presents the background information on max-plus algebra and systems, which is applied to the discussed problem, namely scheduling of parallel batch processes, in Section 4. In Section 5, optimization results for this problem are given. We conclude the paper in Section 6, where also some recommendations for future research will be given.

2. PROBLEM FORMULATION

Scheduling of processes in large manufacturing systems is an important aspect for keeping plants running in a reliable way. With scheduling tools, one is able to increase production, reduce bottlenecks in the plant, and to keep inventories between sub-processes at a minimal level as possible. Since these phenomena influence the reliability of a factory (see e.g. Goldratt [2004]), it is therefore important that the problem of scheduling processes in large factories is completely understood. In this paper, we focus on a specific type of scheduling problems.

In order to make the problem of scheduling more constructive, a manufacturing system consisting of three processes is illustrated in Fig. 1. The processes 1 and 2 are producing raw material 1 and raw material 2, respectively. Process 1 produces an intermediate product 1, which is then used in Process 3 to produce product 2. The plant output is then the final product 2.

Fig. 1. Illustrative manufacturing system for scheduling.

* This work was supported by AgentschapNL, which falls under the Dutch Ministry of Economic Affairs, Agriculture and Innovation, and is done in corporation with Xella, The Netherlands.
intermediate products from raw materials that are later fed in process 3, which converts them to the final product. With scheduling techniques, one tries to optimize the time instances when processes 1 and 2 are starting with a batch of raw material in such a way that the inventory in front of process 3 is not becoming too large, or, even worse, too small such that the process 3 becomes idle and the throughput of the factory becomes less.

In this paper, we consider a slightly different scheduling problem than the one illustrated above. Instead of having multiple different processes that produce different intermediate products, we consider the production of equivalent products in parallel using batch processes, which have similar behavior. This kind of scheduling problems should be easier to be setup and to solve, since no processes have to wait for intermediate products. This is only true when there is no interaction between the in parallel running batch processes. Here, we focus on manufacturing plants where batch processes run in parallel and where energy can be exchanged in, and between, each batch of products that is produced. This is illustrated in Fig. 2, where three similar batch processes are running in parallel, however energy exchange with the other processes is possible.

An example of an industry that has this kind of scheduling problem is the manufacturing of calcium silicate stones, as produced by the company Xella. A key step of producing calcium silicate stones is the hardening phase, where the product needs to stay in an autoclave, which is filled with high pressure steam for a certain period. This production requires large amounts of energy that will be consumed. Since multiple autoclaves run in parallel, it is possible to re-use steam that becomes available when one autoclave is ready, for starting up another autoclave. This will reduce the total energy consumption of the factory as a whole, and is, therefore, a representative case study for scheduling parallel batch processes.

Different methods that could be used to define and solve scheduling problems are using e.g. Petri nets (see e.g. Petri [1966] or Petri [1996]) or using Max-Plus systems (see e.g. Heidegger [2006] or de Schutter [2001]). We have chosen to use the latter strategy, since it has the desired linear property that can efficiently be exploited in modeling and optimization and it has been successful in solving complex scheduling problems as e.g. the Dutch railway network (see van den Boom [2004]). This methodology, which will be explained in detail in Section 3, is used to formulate and optimize scheduling problems as depicted in Fig. 2. More explicitly, we are going to solve the following problem in this paper:

**Problem formulation:** Use the max-plus strategy for the representation and optimization of scheduling problems where batch processes are running in parallel, with inter-batch interaction as e.g. the exchange of energy. Especially, we formulate the problem of the production process of calcium silicate stones, where multiple autoclaves are running in parallel. The inter-batch interaction will be the flow of energy, in the form of high pressure steam.

3. MAX-PLUS ALGEBRA AND SYSTEMS

In order to understand the behavior of a complete factory, or a smaller subsystem of such a process, one requires a mathematical model that represents its behavior. This model can then be used to perform simulations or could be used for designing optimal operations that makes the system behave in a more favorable manner. A model has adjustable parameters, which are denoted as inputs, and has signals that can be measured, defined as outputs, denoted by $u$ and $y$ as depicted in Fig. 3, respectively.

$$u[k] = \sum \alpha + y[k]$$

Fig. 3. Block scheme of a dynamical system.

In “classical” system and control theory, this kind of models has inputs and outputs which are (multi-dimensional) signals over time. In discrete time, one uses the variable $k$ to denote the sample number, which is part of the general mathematical representation of a system as depicted in Fig. 3, namely:

$$x[k+1] = Ax[k] + Bu[k],$$

$$y[k] = Cx[k],$$

(1)

where $x$ denotes the state (vector) of the system and the relations between the inputs and outputs are embedded in the matrices $A$, $B$ and $C$.

We have introduced the concept of “classical” discrete time dynamical systems in this paper to denote that the modeling method, using max-plus systems, is different from the traditional, well known modeling approaches. Scheduling problems are, generally, discrete-event systems, which can not be modeled easily using the classical modeling framework since it will result in complex, highly non-linear models. It turns out that the concept of inputs and outputs in discrete-event systems will be viewed as time instances when an event happens, instead of being time trajectories. This concept will become more clear in the next subsections, where the max-plus algebra and max-plus systems are introduced. Modeling using max-plus systems results in an elegant way of representing systems in scheduling problems in an equivalent form as in (1).

To illustrate modeling discrete-event systems, we use the example introduced in Section 2, which is depicted in more detail in Fig. 4. Denote that the “outputs” illustrated in this figure denote the finishing times of the processes. Process $P_3$ produces the final product at time $y$ from the two intermediate products, which are ready at $u_1$ and $u_2$ and are produced by the processes $P_1$ and $P_2$ from
the raw materials, that are available at times $r_1$ and $r_2$. The variables $d_1$ and $d_2$ denote the time it takes before the products are ready, and so they influence $u_1$ and $u_2$. When both products are available, it takes $d_3$ time before the output product is ready. We assumed that there is no inventory before process $P_3$. The total time to produce a product $y$ is thus the sum of the production time $d_1$ and the time that it takes before both $u_1$ and $u_2$ are available, which is given by the mathematical expression:

$$y[k] = d_3 + \max(r_1[k] + d_1, r_2[k] + d_2, y[k-1]).$$  

(2)

Remark: An important difference between this description and (1) is that $k$ does not denote the sample number, but is used to indicate the number of a batch.

One can also observe that the used operations in (2) are not similar to the ones in linear dynamical systems, as in (1). By introducing a different kind of algebra than the one we are used to, we can overcome this difference and end up with max-plus systems that will have the same structure as (1).

### 3.1 Max-plus algebra

In (2), the basic operations are taking the maximum value of two elements, or adding them together. Therefore, we would like to have an algebra which has these two operations as basics. In a max-plus algebra (see e.g. Heidergott [2006]), these operations are represented using symbols $\oplus$ and $\otimes$, respectively. More specifically,

$$x \oplus y = \max(x, y) \quad \text{and} \quad x \otimes y = x + y,$$

(3)

for all $x, y \in \mathbb{R} \cup \{-\infty\} := \mathbb{R}_\epsilon$, where $\epsilon := -\infty$. The structure $(\mathbb{R}_\epsilon, \oplus, \otimes)$ is then called a max-plus algebra. We have included $\epsilon = -\infty$ as it is the neutral element for the operator $\oplus$ (e.g. $x \oplus \epsilon = \max(x, -\infty) = x$ for all $x \in \mathbb{R}$). For the operator $\otimes$, the neutral element is zero, and $\epsilon$ is the absorbing element.

We can then use this algebra to represent the simple example in (2) as:

$$y[k] = (r_1[k] \otimes d_1 \oplus r_2[k] \otimes d_2) \otimes d_3,$$

which can be seen as a simple dynamical system that is linear in its input.

In order to extend this to the multi-variable case, we need to define the maximum of a finite sequence of numbers $a_1, a_2, \ldots, a_N \in \mathbb{R}_\epsilon$ as:

$$\bigoplus_{k=1}^{N} a_k = a_1 \oplus a_2 \oplus \cdots \oplus a_N = \max_{k=1,\ldots,N} a_k.$$  

(4)

Extension to the multi-variable case then implies for $A, B \in \mathbb{R}_{\epsilon}^{m \times n}$ and $C \in \mathbb{R}_{\epsilon}^{n \times p}$ that

$$(A \oplus B)_{ij} = a_{ij} \oplus b_{ij} = \max(a_{ij}, b_{ij}),$$

$$(A \otimes C)_{ij} = \bigoplus_{k=1}^{n} a_{ik} \otimes c_{kj} = \max_{k} (a_{ik} + c_{kj}),$$

for all $i, j$.

The notation used might look a bit complicated, but in the next subsection it will be shown to be useful for representing (batch) processes in scheduling problems. The reason for including these details here is to clarify the relation between max-plus linear systems and classical systems that are used in general modeling.

### 3.2 Max-plus linear systems

We can now extend the use of max-plus algebra to systems with more equations and restrictions. In the simple example, we could introduce auxiliary variables, since not all processes are directly controlled by inputs, or measured as outputs. These variables are similar to the state vector $x$ in classical dynamical systems. As an example in Fig. 4, when two processes $P_1$ and $P_2$ are connected in series, we can define their times of finishing a batch as $x_1[k]$ and $x_3[k]$. The production of a batch in $P_3$ depends on the previously produced batch of $P_1$, which mathematically can be formulated as $x_3[k] = f(x_1[k-1])$, or equivalently, $x_3[k+1] = f(x_1[k])$. In a more general setting, we define this as the following set of equations:

$$x[k+1] = A \otimes x[k] \oplus B \otimes u[k],$$

$$y[k] = C \otimes x[k],$$

(5)

where $A, B$ and $C$ are matrices.

This representation looks similar to the one used for linear dynamical systems in (1). Note that the $\oplus$ and $\otimes$ operators, which are used in (5), make this system linear in the max-plus algebra. Therefore, this system is called a max-plus linear system.

### 3.3 Example of a MISO max-plus system

In order to illustrate the use of max-plus linear systems, the example in Fig. 4, taken from de Schutter [2001], will be discussed but now with the assumption that there are enough buffers between all processes. This means that a process can already start working on a new batch directly when the previous one is finished.

Due to the inventories between the processes, we need to introduce "state" variables. These are the time instances when the intermediate products leave the three processes, denoted by $x_1, x_2$ and $x_3$. Starting with process $P_1$, it starts working on a batch of raw material as soon as this material is available and when the previous batch has been processed. This can be expressed as:

$$x_1[k+1] = \max(x_1[k], r_1[k]) + d_1,$$

or using the max-plus algebra in the more compact notation:

$$x_1[k+1] = d_1 \otimes x_1[k] \oplus r_1[k].$$

Similarly, we can also describe the two other outputs of processes $P_1$, $P_2$ and $P_3$:

$$x_2[k+1] = d_2 \otimes x_2[k] \oplus r_2[k],$$

$$x_3[k+1] = d_3 \otimes \{ d_1 \otimes (x_1[k] \oplus r_1[k]) \oplus d_2 \otimes (x_2[k] \oplus r_2[k]) + x_3[k] \}.$$  

(6)
The set of equations can be reformulated in the matrix form, as given in (5), resulting in the system matrices:
\[ A = \begin{bmatrix} d_1 & \epsilon & \epsilon \\ \epsilon & d_2 & \epsilon \\ d_1 + d_3 & d_2 + d_3 & d_3 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} d_1 & \epsilon & \epsilon \\ \epsilon & d_2 & \epsilon \\ d_1 + d_3 & d_2 + d_3 & d_3 \end{bmatrix}, \]
where \( u[k] = [r_1[k] \ r_2[k]]^\top \). The output of the system is produced by process \( P_3 \). Therefore, \( y[k] = x_3[k] \), hence the matrix \( C = [\epsilon \ \epsilon \ 0] \).

4. MAX-PHASES FOR PARALLEL BATCH PROCESSES

As mentioned in the introduction and problem formulation of this paper, we are going to apply the max-phase method to manufacturing systems where parallel batch processes are interacting with each other, and between produced batches. This task requires the following three steps to be performed:

1. **Modeling of the batch process**: Each batch process that is running in parallel needs to be modeled. This implies that the different stages that need to be completed for each batch of product are represented using the max-plus algebra. Also the interaction moments with other parallel running processes need to be included in this model. The result of this first step will be a max-plus system describing each batch process.

2. **Defining interaction constraints**: It is not possible to define interactions between arbitrary processes at any time or batch. This can, for example, be due to physical constraints in the factory. Therefore, we need to define these constraints, possibly also using the max-plus algebra, so that we can take them into account when solving the scheduling problem.

3. **Defining the objective**: Now the model and the constraints are available, an objective function needs to be specified that needs to be minimized during optimization.

In the next subsections, these three steps are applied to define the scheduling problem for the production of calcium silicate stones, as done in the factories of Xella. In step (1), the batch process of hardening the stones in an autoclave will be modeled, which is the process that is carried out in parallel. Step (2) will result in the interaction constraints, which are the flows of energy between the batch processes, in the form of high pressure steam. Finally, step (3) will define the objective that needs to be minimized, which is the production time of an average batch of stones.

4.1 **Modeling of the batch process**

The process of hardening the stones using an autoclave can be split up into six different phases. This has been illustrated in Fig. 5 for an autoclave, which has a number \( i \). This process corresponds to one of the parallel running batch processes in Fig. 2.

The input of this batch process is the supply of to-be-hardened stones, which we can assume to be unlimited. This means that process \( i \), going to produce batch \( k \) and denoted using \( x_i[k] \), can start immediately when the previous batch \( k-1 \) was produced. The six different phases, which are in the rectangular boxes in Fig. 5, need some processing time however, and even can depend on interactions from other processes (in the form of energy exchange). For modeling the finishing times of each phase in a batch, we use the subscripts \( x_{ij} \) where \( i \) is the autoclave number and \( j \) is the phase number. We are intending to produce batch \( k+1 \), hence we use the notation \( x_{ij}[k+1] \) to describe time instances for the production of the intermediate products. We neglect the batch number in Fig. 5 to keep the figure simple. The interaction between the different batch processes is denoted using \( w_{ij}[k+1] \) and \( w_{ji}[k+1] \), which will be explained later in this subsection.

The instances when process \( i \) is ready with the intermediate phases for producing batch \( k+1 \) are defined as follows:

- **Loading an autoclave**, \( x_{i1}[k+1] \): Before an autoclave \( i \) is filled with stones it takes some time, namely \( d_{i1} \). Therefore, the finishing time of this phase is the starting time of the autoclave \( u_i[k] \) plus this delay \( d_{i1} \). Mathematically, this reads as:
  \[ x_{i1}[k+1] = x_{i1}[k] + d_{i1}. \]

- **Heating up an autoclave**, \( x_{i3}[k+1] \): After loading the autoclave, steam is required to start the hardening process. This steam comes partly from a source, however steam from other batches that run in parallel will be re-used. The connection with other autoclaves will be discussed in the next subsection, together with its the constraints.

For now, we model the time when steam becomes available for the autoclave as \( w_{i1}[k+1] \). The time of starting the heating is then the maximum value of \( w_{i1}[k+1] \) and the time when the autoclave is filled with stones, \( x_{i1}[k+1] \), which is obviously necessary. It takes \( d_{i2} \) time before the steam is exchanged, and additional energy from a source is added, so the finishing time is the maximum of \( x_{i1}[k+1] \) and \( w_{i1}[k+1] \) added to \( d_{i2} \). This can be formulated as:
  \[ x_{i3}[k+1] = \max(x_{i1}[k+1], w_{i1}[k+1]) + d_{i2}. \]

The artificial signal \( \tilde{x}_{i1}[k+1] \) is the time instance when process \( i \) starts received steam from another autoclave, which will be used in the next subsection.

- **Waiting**, \( x_{i5}[k+1] \): The stones need to stay in the autoclave for \( d_{i3} \) time, hence, \( x_{i5}[k+1] \) is the sum of \( x_{i2}[k+1] \) and \( d_{i3} \), written as:
  \[ x_{i5}[k+1] = x_{i2}[k+1] + d_{i3}. \]

- **Batch of product ready**, \( x_{i4}[k+1] \): After waiting a certain time, namely \( x_{i5}[k+1] \), the batch of stones is done with the hardening process. Hence, \( x_{i4}[k+1] = x_{i3}[k+1] \).

Since the autoclave still has high-pressure steam in it, which needs to be exchanged with another autoclave, we introduce the artificial signal \( \tilde{w}_{i2}[k+1] \).

![Flow chart of the steps performed in each autoclave batch process with number i.](image-url)
Cooling down an autoclave, \( x_{i5}[k + 1] \): When another autoclave is ready to receive steam, an exchange of energy will take place. Using the interaction variables, this exchange starts at \( w_{i2}[k + 1] \), which obviously should be later in time than when the previous phase is finished. It still takes \( d_{i5} \) time to exchange the steam and to get rid of the remaining steam, hence the batch \( k + 1 \) is processed after \( w_{i2}[k + 1] \), which should be larger than \( x_{i4}[k + 1] \), added with \( d_{i5} \), or in short, \( x_{i5}[k + 1] = \max(x_{i4}[k + 1], w_{i2}[k + 1]) + d_{i5} \).

Unloading an autoclave, \( x_{i6}[k + 1] \): When the autoclave \( i \) does not contain steam anymore, the stones can be unloaded from it, which takes \( d_{i6} \) time units. Therefore, \( x_{i6}[k + 1] \) is the sum of \( x_{i5}[k + 1] \) and \( d_{i6} \), and is in fact equal to the finishing time of the complete batch, formulated as

\[
x_{i}[k + 1] = x_{i6}[k + 1] = x_{i5}[k + 1] + d_{i6}.
\]

The six phases can be modeled using the max-plus algebra, which results in:

\[
x_{i1}[k + 1] = d_{i1} \otimes x_{i}[k],
\]

\[
x_{i2}[k + 1] = d_{i2} \otimes (x_{i1}[k + 1] \oplus w_{i1}[k + 1]),
\]

\[
x_{i3}[k + 1] = d_{i3} \otimes x_{i2}[k + 1],
\]

\[
x_{i4}[k + 1] = x_{i3}[k + 1],
\]

\[
x_{i5}[k + 1] = d_{i5} \otimes (x_{i4}[k + 1] \oplus w_{i2}[k + 1]),
\]

\[
x_{i6}[k + 1] = d_{i6} \otimes x_{i5}[k + 1] = x_{i}[k + 1].
\]

This batch process can be represented as a linear max-plus system:

\[
x_{i}[k + 1] = (d_{i1} \otimes d_{i2} \otimes d_{i3} \otimes d_{i5} \otimes d_{i6} x_{i}[k] \\
\quad \otimes [d_{i2} \otimes d_{i3} \otimes d_{i5} \otimes d_{i6} \quad d_{i5} \otimes d_{i6}]) u_{i}[k],
\]

where \( u_{i}[k] := [w_{i1}[k + 1] \quad w_{i2}[k + 1]]. \)

4.2 Defining interaction constraints

Each process that is producing batches in parallel is modeled using the max-plus linear system in (6), we focus on how they are interacting with each other. For each autoclave that is producing a batch, we can distinguish two possible interactions, namely:

1. requesting for energy from others,
2. supplying energy to another autoclave.

We explain how these energies are exchanged between autoclaves using the illustration in Fig. 6, where two autoclaves are interacting and exchanging steam with each other. At a certain time, \( \tilde{w}_{i2} \), the left autoclave with number \( i \) is done with the hardening, and it checks which autoclave wants to use its steam. Here, autoclave \( j \) will be ready to receive it at time \( x_{j1} \), which is later than \( \tilde{w}_{i2} \), so autoclave \( i \) has to wait. To “reserve” the steam to be used for autoclave \( j \), we set \( w_{i1} \) equal to \( \tilde{w}_{i2} \). When autoclave \( j \) is then ready for receiving the steam, autoclave \( i \) starts with supplying steam (depicted in gray) at time instance \( w_{i2} \). Afterwards autoclave \( i \) gets rid of the remaining steam that is left over, and autoclave \( j \) gets additional steam from a source, such that the two batch processes reach \( x_{i5} \) and \( x_{j2} \), respectively (Note that if \( \tilde{w}_{i2} \geq x_{j1} \), the transfer of steam starts directly).

Summarizing, \( w_{i1} \) depends on a time instance \( \tilde{w}_{i2} \) from another autoclave, and \( w_{i2} \) depends on \( x_{j1} \).

Fig. 6. Interaction (lines) and steam exchange (grayed) between autoclaves \( i \) and \( j \).

Till now we did not mention the batch number in which the autoclaves \( i \) or \( j \) are processing when transferring energy. It could be possible that autoclave \( i \) is producing batch \( k \) and is providing steam to an autoclave \( j \) producing batch \( k' \), with \( k' \) being smaller, equal or larger than the batch number \( k \). This can easily occur when the batch processes that run in parallel have different processing times.

The decision on making the choice of which autoclaves are being connected should be made during optimization, such that the objective is minimal. To do this, we introduce the binary decision variables \( \alpha_{ij}^k \). This variable equals 1 when an autoclave \( i \), that is producing batch \( k \), is delivering steam to autoclave \( j \) which is going to heat up batch \( l \). This immediately results in the first constraint, since it is not possible to provide or get steam from yourself in the same, the previous, and the next batch. This means that:

\[
\alpha_{ii}^{k-1}[k] = 0, \quad \alpha_{ii}^{k}[k] = 0 \quad \text{and} \quad \alpha_{ii}^{k+1}[k] = 0,
\]

for all parallel batch processes \( i \) and batches \( k \).

For the definition of the other constraints, we define the number of parallel batch processes as \( N_p \) and the number of batches that needs to be scheduled to be \( N \). Using these horizons and the decision variable \( \alpha_{ij}^k \), we express the time instances \( w_{i1}[k] \) and \( w_{i2}[k] \) as:

\[
w_{i1}[k] = \sum_{l=1}^{N_p} \sum_{j=1}^{N} \alpha_{ij}^l[k] \tilde{w}_{ij2}[l],
\]

\[
w_{i2}[k] = \sum_{l=1}^{N_p} \sum_{j=1}^{N} \alpha_{ij}^l[k] \tilde{w}_{ij1}[l].
\]

To ensure that an autoclave can maximally make one connection for exchanging energy in a batch, we obtain the constraints:

\[
\sum_{l=1}^{N_p} \sum_{j=1}^{N} \alpha_{ij}^l[k] = 1 \quad \text{and} \quad \sum_{l=1}^{N_p} \sum_{j=1}^{N} \alpha_{ij}^l[k] = 1,
\]

for all \( i = 1, \ldots, N_p \) and \( k < N \). For the last batch in the prediction horizon, the equality conditions become inequality conditions, since it can be possible that an autoclave can not deliver steam to another autoclave anymore, since there are no requesting batch processes available.

The expressions above are not in the max-plus framework, since the multiplication with the binary decision variable can not be converted to the max-plus algebra. The optimization tool used to solve the scheduling problem can, however, deal with this kind of binary decision variables, therefore, this it will not be a problem.
$d_{ij} \ i = 1 \ i = 2 \ i = 3
j = 1 \ 1 \ 2 \ 1
j = 2 \ 1 \ 1 \ 1
j = 3 \ 5 \ 8 \ 6
j = 5 \ 1 \ 1 \ 1
j = 6 \ 1 \ 3 \ 2$

Table 1. Parameters used during optimization.

<table>
<thead>
<tr>
<th>Batch</th>
<th>Decisions exchanging steam</th>
<th>$x_1[k]$</th>
<th>$x_2[k]$</th>
<th>$x_3[k]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1 full</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1 → 3, 3 → 2, 2 → 1</td>
<td>10</td>
<td>28</td>
<td>18</td>
</tr>
<tr>
<td>2</td>
<td>1 → 3, 3 → 2, 2 → 1</td>
<td>32</td>
<td>50</td>
<td>40</td>
</tr>
<tr>
<td>3</td>
<td>1 → 3, 3 → 2, 2 → 3</td>
<td>54</td>
<td>72</td>
<td>62</td>
</tr>
<tr>
<td>4</td>
<td>3 → 1, 1 → 2, 2 → 1</td>
<td>83</td>
<td>94</td>
<td>78</td>
</tr>
<tr>
<td>5</td>
<td>1 → 2, 2 → 3, 3 → 1</td>
<td>98</td>
<td>109</td>
<td>115</td>
</tr>
<tr>
<td>6</td>
<td>1 → 2, 2 → 3, 3 full</td>
<td>120</td>
<td>131</td>
<td>$\infty$</td>
</tr>
</tbody>
</table>

Table 2. Results of the scheduling problem.

4.3 Defining the objective

The goal is to minimize the average processing time for a batch. In that case, the parallel running autoclaves do not have to wait long before they can make use of the steam, and we achieve the most optimal throughput of the complete factory. When adding all intervals between the batches, we get:

$$\sum_{k=1}^{N} x[k] - x[k-1] = \frac{1}{N} (x[N] - x[0]),$$

where $x[0]$ is a (fixed) initial condition. Hence, we are going to optimize the time of the final batch. Since it might be possible that steam of one or multiple processes can be utilized, these processes never finish with their last batch, and therefore the minimal value of the objective function will be infinity. Therefore, neglecting the final batches results in the complete problem:

Scheduling problem:

$$\min_{\alpha^i_{n,k}} x[N-1]$$

subject to:

- dynamics of the parallel batch processes in (6);
- constraints on $\alpha^i_k$ in (7);
- expressions for $w_{11}$ and $w_{12}$ as in (8);
- additional constraints on $\alpha^i_{n,k}$ in (9);

for $i = 1, \ldots, N_p$.

5. OPTIMIZATION AND RESULTS

In this section, the above defined scheduling problem needs to be solved for a case study. To solve this kind of problems, the YALMIP toolbox in MATLAB is used, which is able to deal with max-plus problems as well as problems with binary decision variables (see e.g. Lofberg [2004]). We are going to optimize and solve the scheduling problem using the parameters in Table 1 with $N = 6$ and $N_p = 3$. An assumption we make is that an autoclave can only interact (so can give an get steam) from parallel processes in the same, the previous or the next batch. This reduces the number of decision variables.

The initial conditions considered are that autoclave 1 gets fresh steam from the source initially, and that the others start at different time instances. More specifically, $x[0] = [0 \ 4 \ 7]^T$. The results of the optimization can be found in Table 2, where the value of the objective function became 322 time units. Unfortunately, the optimization problem that needs to be solved is not convex, so we can not be sure whether this is the global optimal value.

6. CONCLUSIONS AND RECOMMENDATIONS

In this paper, we have shown scheduling problems having parallel batch processes that are interacting with each other. We have used the max-plus approach to model such a process, together with the interconnection constraints resulted in a problem that could be used for optimization.

This approach has been applied to a case study of production of calcium silicate stones, which are produced in parallel batch processes. This process involves large amounts of high-pressure steam, which is re-used between the parallel processes, and therefore needs to be scheduled as interaction variable.

The developed model is a simplification of the complete process, and can be extended to consider a more general situation. For future research, the use of more processes that run in parallel, or multiple instances of interaction (as e.g. exchanging energy) between them, could be done which will make optimization more complex.

ACKNOWLEDGEMENTS

We would like to thank dr.ir. Ton J.J. van den Boom from Delft University of Technology, The Netherlands, for the discussions on max-plus algebras and systems.

REFERENCES