General Lower Bounds for the Total Completion Time in a Flowshop Scheduling Problem
MaxPlus Approach

Nhat-Vinh Vo¹, Pauline Fouillet² and Christophe Lenté¹

¹Université François-Rabelais de Tours, CNRS, LI EA 6300, OC ERL CNRS 6305, Tours, France
²Université François-Rabelais de Tours, Polytech, Tours, France

Abstract: The flowshop scheduling problem has been largely studied for 60 years. As a criterion, the total completion time also receives a great amount of attention. Many studies have been carried out in the past but they are limited in the number of machines or constraints. MaxPlus algebra is also applied to the scheduling theory but the literature focuses on some concrete constraints. Therefore, this study presents a new method to tackle a general permutation flowshop problem, with additional constraints, to elaborate on lower bounds for the total completion time. These lower bounds can take into account several constraints, like delays, blocking or setup times, but they imply to solve a Traveling Salesman Problem. The theory is developed, based on a MaxPlus modeling of flowshop problems and experimental results are presented.

1 INTRODUCTION

The m-machine flowshop scheduling problem has been largely studied for 60 years. The makespan is the most studied criterion, especially for permutation flowshop problems. However, the total completion time criterion also receives a great amount of attention. It reflects “the total manufacturing waiting time experienced by all customers” (Emmons and Vairaktaras, 2013). Even with only two machines, problem $F_2||\sum C_i$ is $NP$ − hard in the strong sense and so are problems with more machines. Therefore, results that help to solve these problems are interesting.

Total completion time criterion has been studied largely. A branch-and-bound algorithm, incorporating a lower bound, dominance relation and an upper bound is presented by Allahverdi and Al-Anzi in (Allahverdi and Al-Anzi, 2006). That study solves total completion time minimization problem $F_2||\sum C_i$ where separate setup times are taken into account. The number of visited nodes and the percentage between this number and that of possible nodes are considered. This percentage shows us that their lower bound is effective as the number of visited nodes is quite small. Separate setup times are also investigated by Su and Lee (Su and Lee, 2008) in a two-machine flowshop no-wait scheduling problem with a single server in order to minimize total completion time. In another research, eleven heuristics based on the Shortest Processing Time (SPT) rule are implemented by Aydilek and Allahverdi (Aydilek and Allahverdi, 2010). Their study is to minimize total completion time of a two-machine flowshop scheduling problem, in which processing times are bounded. A lower bound based on the first machine of problem $F_2||\sum C_i$ is presented as the sum of a previously existing lower bound and the optimum of an asymmetric traveling salesman problem (ATSP) (Della Croce et al., 1996). These aforementioned studies only deal with limited number of machines and few constraints. It is not easy to generalize them to any number of machines or any constraint.

In this study, the proposed approach is based on MaxPlus algebra (see 2.1). It has been widely used in control systems, especially in relation with Petri Nets but rarely in the scheduling theory. Nevertheless, some articles can be cited on project scheduling problems (Giffler, 1963), on cyclic parallel machine problems (Hanen and Munier, 1995), on cyclic flowshop scheduling problems (Cohen et al., 1985; Gaubert, 1992) and on cyclic jobshop scheduling problems (Gaubert and Maisresse, 1999). The MaxPlus algebra is applied to modeling of flowshop problems and experimental results are presented.
maximal delays for two-machines (Bouquard and Lenté, 2006) or for any number of machines (Augusto et al., 2006). In these studies, jobs are associated to MaxPlus square matrices and lower bounds, upper bounds and/or dominance conditions are derived by applying transformations to those matrices. Moreover, this approach is used effectively to model flowshop problems with minimal-maximal delays, setup and removal times and to highlight a central problem (Vo and Lenté, 2013).

The objective of this study is to address a general permutation flowshop problem in terms of constraints taken into account. We elaborate on lower bounds for the total completion time that are based on the resolution of two sub-problems: one problem similar to the one machine total completion time minimization problem and the other similar to a travelling salesman problem. These lower bounds are incorporated in a branch-and-bound procedure to be tested.

The next section presents the background of the study: MaxPlus algebra and flowshop scheduling problem. We recall in section 3 how MaxPlus algebra can be used to model a general flowshop problem. The lower bound construction is then explained in section 4. Finally, a branch-and-bound algorithm is explained and some tests concerning problem $F3|\text{perm}, \text{Snnd}| \sum C_i$ and problem $Fm|\text{perm}| \sum C_i$ are presented as experimental results.

2 CONTEXT AND DEFINITIONS

2.1 MaxPlus Algebra

MaxPlus algebra is briefly described as follows; a more detailed presentation can be found in (Gunawardena, 1998).

In MaxPlus algebra, the maximum is denoted by $\oplus$ and the addition by $\otimes$. The former, $\oplus$, is idempotent, commutative, associative and has a neutral element (−∞) denoted by $\Theta$. The latter, $\otimes$, is associative, distributive on $\oplus$ and has a neutral element (0) denoted by $\Theta$. The null element, $\Theta$, is an absorbing element for $\otimes$. These properties can be summarized by stating that $\mathbb{R}_{\text{max}} = (\mathbb{R} \cup \{-\infty\}, \oplus, \otimes)$ is a dioid. It is important to note that in MaxPlus algebra in particular, and in dioids in general, the first operator $\oplus$ cannot be simplified, that is $a \oplus b = a \oplus c \neq b = c$. Furthermore, in $\mathbb{R}_{\text{max}}$, the second operator $\otimes$ is commutative, and except $\Theta$, every element is invertible: the inverse of $x$ is denoted by $x^{-1}$ or $\Theta/x$. For simplicity, we denote the ordinary subtractions by $x/y$ instead of $x \otimes y^{-1}$ and by $xy$ the product $x \otimes y$.

It is possible to extend these two operators to $m \times m$ matrices of elements of $\mathbb{R}_{\text{max}}$. Let $A$ and $B$ be two matrices of size $m \times m$, operators $\oplus$ and $\otimes$ are defined by

$$\forall (i,j) \in \{1, \ldots, m\}^2, [A \oplus B]_{ij} = [A]_{ij} \oplus [B]_{ij}$$

$$\forall (i,j) \in \{1, \ldots, m\}^2, [A \otimes B]_{ij} = \bigoplus_{k=1}^{m} [A]_{ik} \otimes [B]_{kj}$$

where $[.]_{ij}$ is the element at the $i^{th}$ row and $j^{th}$ column of the corresponding matrix. It is not difficult to show that the set of $m \times m$ matrices in $\mathbb{R}_{\text{max}}$ is a dioid. However, $\otimes$ is not commutative and not every matrix is invertible.

The two following lemmas can be derived from the previous definitions. They will be useful for the development of the lower bound.

Lemma 1. $\forall j \in \{1, \ldots, m\}$ :

$$[A \otimes B]_{1j} = \bigoplus_{k=1}^{m} [A]_{1k} \otimes [B]_{kj}$$

Lemma 2. $\forall k, j \in \{1, \ldots, m\}$ :

$$[A \otimes B]_{kj} \geq [A]_{i\ell} \otimes [B]_{\ell j}$$

$$[A \otimes B]_{kj} \geq [A]_{i\ell} \otimes [B]_{\ell j}$$

2.2 Flowshop Scheduling Problem

Since the paper of Johnson (Johnson, 1954), flowshop problems have been studied largely (Emmons and Vairaktarakis, 2013). Basically, a flowshop scheduling problem consists of a set of $n$-jobs $J = \{J_1, \ldots, J_n\}$ and another set of $m$-machines $\{M_1, \ldots, M_m\}$. Each job must go through all machines in the same predefined order, let us say from $M_1$ to $M_m$ and each machine can load only one job at a time (Brucker, 2006). If all jobs must be executed in the same order over all machines, the problem is called a permutation flowshop problem. In this case, there exists an ordered list of jobs (or a sequence) $\sigma$ that is identically scheduled on all machines. We limit our current study to permutation flowshop problems. Each job $J_i$ is composed of $m$ operations $O_k$ ($1 \leq k \leq m$): one per machine. An operation is at least described by a processing time $p_k$: the processing time of job $J_i$ on machine $M_k$ (or equivalently, the processing time of the $k^{th}$ operation of job $J_i$). The completion time of job $J_i$ on machine $M_k$ ($C_{ik}$) and the completion time of job $J_i$ ($C_i$) are related by $C_i = C_{im}$.

Over the years, several additional constraints have been taken into consideration (Emmons and Vairaktarakis, 2013). Some of them can be modeled using MaxPlus algebra (Vo and Lenté, 2013). One of
the most common constraints is the permutation constraint (perm) which has just been mentioned above. A constraint of no – wait appears in problems where there is no delay allowed between two successive operations of a job. On the contrary, constraints of min – delay, max – delay, min – max delay indicate a flowshop problem with delays between two successive operations of a job. Depending on the case, these delays may have to meet a lower bound, an upper bound or both. It may also exist separate non- sequence dependent setup times ($S_{nd}$) and/or removal times ($R_{nd}$) before and after each operation. Finally, some authors have considered blocking constraints, due to the non-existence of intermediate storage between consecutive machines or to specific interactions between machines. These constraints are referred to as $R_{sb}$, $R_{cb}$ and $R_{cb}^b$ in (Trabelsi et al., 2012).

The most studied criterion is the makespan, or the maximal completion time ($C_{max}$). It is defined by the completion time of the last operation scheduled on the last machine ($M_m$). In this article we focus on the total completion time ($\sum C_i$) which is the sum of the completion times of the different jobs in a given schedule.

3 MaxPlus MODELING OF FLOWSHOP SCHEDULING PROBLEMS

Our problem can be noted $F_m|perm|\beta|\gamma$ using notations proposed by Graham et al. (Graham et al., 1979). It is a $m$ machine permutation flowshop problem with a set of constraints $\beta$ that is a subset of \{min – max delay, no – wait, $S_{nd}$, $R_{nd}$, $R_{sb}$, $R_{cb}$, $R_{cb}^b$\}. Criterion $\gamma$ can be whatever we desire since it does not interfere in the modeling process. The total completion time criterion is investigated in the following of this article.

Basically, the modeling process follows this scheme:

- Given the $k$th operation $O_{ik}$ of a job $J_i$, four dates are considered: date $\delta_k$ of availability of machine $M_k$ (before execution of operation $O_{ik}$), starting time $S_{ik}$ of operation $O_{ik}$, its completion time $C_{ik}$ and date of liberation $D_{ik}$ of machine $M_k$ (after execution of operation $O_{ik}$), that is the date when job $J_i$ leaves machine $M_k$ to be placed in a stock or on the following machine. In most flowshop problems, dates $C_{ik}$ and $D_{ik}$ are equal; however, they can be different in case of blocking constraints or removal times. Date of liberation of the last machine ($D_{im}$) is equal to the completion time $C_i$ of the job, except if there exist removal times. In this case $D_{im}$ is equal to $C_i$ plus the removal time of operation $O_{im}$.

4 Figure 1 shows an example of flowshop problem $F_m|perm, min – max delay, S_{nd}, R_{nd}|\sum C_i$ where triangles illustrate setup and removal times and rectangles illustrate processing times.

- Formulate the system $(S)$ of inequalities that link these different variables.
- Calculate the smallest $(D_k)$ $(1 \leq k \leq m), (1 \leq i \leq n)$ solutions of the system $(S)$.

![Figure 1: Example of flowshop problem: $F_m|perm, min – max delay, S_{nd}, R_{nd}|\sum C_i$](image)

Whatever the set of constraints $\beta$ is, these calculations lead to a MaxPlus linear relation between dates of liberation $D_{ik}$ and dates of availability $\delta_k$ (Lenté, 2011), (Vo and Lenté, 2013). More precisely, we can state the following proposition:

**Proposition 1** (Matrix associated to a job). Let $\vec{\delta}$ (resp. $\vec{D}$) be the line vector of the $m$ dates $\delta_k$ (resp. $D_k$): it exists a $m \times m$ MaxPlus matrix $T_i$ computed from data of job $J_i$ such that

$$\vec{D}_i = \vec{\delta} \otimes T_i$$

Matrix $T_i$ is called the associated matrix of job $J_i$: it entirely characterizes job $J_i$.

Various elements of matrix $T_i$ will be denoted $t_{ij}$, in other words, $t_{ij} = [T_i]_{ij}$. This matrix sums up the job data (processing times, setup times, delays and so on) and the flowshop constraints.

$$T_i = \begin{pmatrix} t_{i1} & t_{i2} & \cdots & t_{im} \\ t_{i2} & t_{i2} & \cdots & t_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ t_{im} & t_{im2} & \cdots & t_{mm} \end{pmatrix}$$

These results can be generalized to a sequence of jobs (Lenté, 2011), (Bouquard et al., 2006).
Definition 1 (Matrix associated to a sequence). Let \( \sigma \) be a sequence of \( v \) jobs: its associated matrix is matrix \( T_\sigma \) defined by
\[
T_\sigma = \bigotimes_{i=1}^{v} T_{\sigma(i)} \quad (6)
\]

Proposition 2. If \( \bar{\delta} \) is the vector of dates of availability of machines and \( \bar{D}_\sigma \) the vector of dates of liberation of machines, after the execution of sequence \( \sigma \), we have the relation
\[
\bar{D}_\sigma = \bar{\delta} \otimes T_\sigma \quad (7)
\]

4 PROPOSED LOWER BOUNDS

This section presents lower bounds for problem \( Fm|\text{perm.}, \beta \sum C_i, \alpha \in \{\text{min, max delay, no wait}\}, S_{\text{head}}, R_{\text{load}}, R_{\text{RB}}, R_{\text{CB}} \} \). To develop the calculations, we assume that \( C_i = D_{im} \) (1 \( \leq i \leq n \)). It is true unless there exist removal times: this particular case will be discussed at the end of this section.

4.1 The first lower bound

We first present a lower bound of the completion time of the \( k \)th job in a sequence before elaborating on a lower bound for the total completion time.

4.1.1 Lower bound of completion time of a job

Proposition 3. Let \( \sigma \) a sequence of jobs and \( \bar{\delta} \) the line vector of dates of availability of the machines (\( \bar{\delta} = (\delta_1, \delta_2, \ldots, \delta_n) \)). The completion time of the job in \( k \)th position in the sequence verifies relation:
\[
\begin{align*}
&\text{if } k = 1:\quad C_{\sigma(1)} \geq \delta_1 \left[T_{\sigma(1)}\right]_{1m} \\
&\text{if } k = 2:\quad C_{\sigma(2)} \geq \delta_1 \left[T_{\sigma(1)}T_{\sigma(2)}\right]_{1m} \\
&\text{if } k > 2:\quad C_{\sigma(k)} \geq \delta_1 \bigotimes_{j=1}^{k-2} \left[T_{\sigma(j)}T_{\sigma(k)}\right]_{1m} 
\end{align*}
\]

Proof. Let \( \tau \) be the sub-sequence composed of the first \( k \) jobs of sequence \( \sigma \). Proposition 2 and definition 1 result in:
\[
\bar{D}_\tau = \bar{\delta} \otimes T_\tau = \bar{\delta} \otimes \bigotimes_{i=1}^{k} T_{\tau(i)} = \bar{\delta} \otimes \bigotimes_{i=1}^{k} T_{\sigma(i)} \quad (8)
\]
Moreover \( \bar{D}_{\sigma(k)} = \bar{D}_\tau \) and by assumption \( C_{\sigma(k)} = D_{\sigma(k)\text{m}} \), which is the last element of vector \( \bar{D}_{\sigma(k)} \), we have,
\[
C_{\sigma(k)} = \left[D_{\sigma(k)\text{m}}\right]_{1m} = \left[\bar{\delta} \otimes \bigotimes_{i=1}^{k} T_{\sigma(i)}\right]_{1m} \quad (9)
\]
For \( k = 1 \), the application of lemma 1 results in:
\[
C_{\sigma(1)} \geq \left[\delta_1 \left[T_{\sigma(1)}\right]_{1m}\right]_{1m} \quad (10)
\]
For \( k \geq 2 \), by iteratively applying this lemma into equation (9), we obtain:
\[
C_{\sigma(k)} \geq \delta_1 \left[T_{\sigma(1)\text{m}}\right]_{1m} \left[T_{\sigma(2)\text{m}}\right]_{1m} \left[T_{\sigma(3)\text{m}}\right]_{1m} \cdots \left[T_{\sigma(k)\text{m}}\right]_{1m} \quad (11)
\]
Inequality (11) can be rewritten as
\[
C_{\sigma(k)} \geq \delta_1 \bigotimes_{j=1}^{k-2} \left[T_{\sigma(j)\text{m}}\right]_{1m} \left[T_{\sigma(k)\text{m}}\right]_{1m} \quad (12)
\]

4.1.2 Lower bound of the Total Completion Time

Definition 2 (Lower Bound \( LB_{1,FL} \)).

Given a sequence \( \sigma \) of \( n \) jobs, we define:
\[
A_1(\sigma) = \bigotimes_{j=1}^{n-1} t_{11}^{\sigma(j)} n^{-j} \\
B_1(\sigma) = \left[T_{\sigma(1)\text{m}}\right]_{1m} \left[\bigotimes_{j=2}^{n} \frac{\left[T_{\sigma(j-1)\text{m}}\right]_{1m} \quad (13)}{t_{11}^{\sigma(j-1)}}\right]
\]

Proposition 4.
\[
\forall \sigma \text{ sequence : } \bigotimes_{i=1}^{n} C_{\sigma(i)} \geq \delta_1 \otimes A_1(\sigma) \otimes B_1(\sigma) \quad (14)
\]

Proof. Considering proposition 3, we have:
\[
\bigotimes_{i=1}^{n} C_{\sigma(i)} \geq \delta_1 \left[T_{\sigma(i)\text{m}}\right]_{1m} \bigotimes_{i=2}^{n} \delta_1 \left[t_{11}^{\sigma(i)}\right]_{1m} \left[T_{\sigma(i)\text{m}}\right]_{1m} \quad (13)
\]
Rearranging the factors on the right side of inequality (13):
\[
\bigotimes_{i=1}^{n} C_{\sigma(i)} \geq \left[\delta_1 t_{11}^{\sigma(i)}\right]_{1m} \bigotimes_{i=2}^{n} \left[T_{\sigma(i-1)\text{m}}\right]_{1m} \left[T_{\sigma(i)\text{m}}\right]_{1m} \quad (14)
\]
and then multiplying the inequality (14) by $\prod_{i=1}^{n-1} t_{i1}^{-1}$ we complete the proof.

At this point, we can obtain a lower bound of the Total Completion Time by computing the optimal values of factors $A_1$ and $B_1$. The two following propositions explain how to do it.

**Proposition 5 (Minimisation of $A_1$).**
Let $\sigma_{SPT}^1$ the sequence obtained by sorting jobs in non-decreasing order of the coefficient $t_{11}$. This sequence minimizes criterion $A_1$.

**Proof.**

$$A_1(\sigma) = \prod_{j=1}^{n-1} t_{11}^{(j)} n-j = \prod_{j=1}^{n} t_{11}^{(j)} n-j+1$$

(15)

The second factor is a constant, so we have to minimize $\prod_{j=1}^{n-1} t_{11}^{(j)} n-j+1$. It is similar to the total completion time criterion in a one-machine problem (1) where processing times are $t_{11}s$. This criterion is minimized by using Smith's rule (Smith, 1956).

**Proposition 6 (Minimisation of $B_1$).**
Let us consider an Asymmetric Traveling Salesman Problem (ATSP) defined by the following distances between $n + 1$ towns, numbered from 0 to $n$:

$$\forall i \in \{1, \ldots, n\} : d(0,i) = |T_i|_1$$

$$\forall i \in \{1, \ldots, n\} : d(i,0) = |T_i|_1$$

<table>
<thead>
<tr>
<th>$\forall (i,j) \in {1, \ldots, n}^2$</th>
<th>$d(i,j)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d(i-j)$</td>
<td>$\prod_{j=1}^{n} t_{11}^{(j)} n-j+1$</td>
</tr>
</tbody>
</table>

(16)

Let sequence $\sigma_{ATSP}^1$, be an optimal cycle of this ATSP: $B_1(\sigma_{ATSP}^1)$ is the optimal value of criterion $B_1$

**Proof.**

With these notations, $B_1(\sigma)$ can be rewritten as the length of a cycle:

$$B_1(\sigma) = d(0,\pi(1)) \prod_{i=1}^{n-1} d(\sigma(i),\sigma(i+1)) d(\sigma(n),0)$$

(17)

All these results lead to the next proposition.

**Proposition 7 (Lower Bound $LB_{VFL}^1$).**
Let $LB_{VFL}^1 = (\delta_1 n \otimes A_1(\sigma_{SPT}^1) \otimes B_1(\sigma_{ATSP}^1)$: $LB_{VFL}^1$ is a lower bound of the Total Completion Time.

In usual notations, this lower bound is defined by:

$$LB_{VFL}^1 = n \delta_1 + A_1(\sigma_{SPT}) + B_1(\sigma_{ATSP})$$

(18)

It is needed to solve a traveling salesman problem to compute this lower bound; however, the procedures for solving that problem are rather effective on medium size instances.

This lower bound is similar to the one presented by Della Croce et al. (Della Croce et al., 1996) for two machines, but it works with $m$ machines and more constraints.

### 4.1.3 Existence of Removal Times

If there are removal times, the date of liberation of machine $M_m$ by job $J_i (D_{im})$ is equal to the sum of completion time $C_i$ of job $J_i$ and removal time of the last operation of $O_{im}$ of $J_i$. Thus, the total sum of $D_{im} (1 \leq i \leq n)$ is equal to the total completion time plus a constant term which is equal to the sum of removal times of all last operations. Therefore, to obtain a lower bound of the total completion time we only have to subtract this constant from $LB_{VFL}^1$.

### 4.2 Additional Similar Lower Bounds

A similar approach to the construction of the first lower bound can be developed to achieve the $\ell^{th}$ lower bound ($\forall \ell \in \{1, \ldots, m\}$). Using iteratively lemma 2, we obtain:

$$C_{\alpha(1)} \geq \delta_1 |T_{\alpha(1)}|_m$$

$$C_{\alpha(2)} \geq \delta_1 |T_{\alpha(1)}|T_{\alpha(2)}|_m$$

$$\forall i > 2 : C_{\alpha(i)} \geq \delta_1 \left( \bigotimes_{j=1}^{i-2} t_{11}^{(j)} \right) |T_{\alpha(i-1)}T_{\alpha(i)}|_m$$

(19)

Defining $A_\ell(\sigma)$ and $B_\ell(\sigma)$:

$$A_\ell(\sigma) = \prod_{j=1}^{n-1} t_{11}^{(j)} n-j$$

(20)

$$B_\ell(\sigma) = |T_{\alpha(1)}|_m \left( \bigotimes_{j=2}^{n} T_{\alpha(j-1)}T_{\alpha(j)} \right) t_{11}^{(j-1)}$$

(21)

we have

$$\bigotimes_{i=1}^{n} C_{\alpha(i)} \geq \delta_1^{\ell} A_\ell(\sigma) B_\ell(\sigma)$$

(22)

Similarly to propositions 5 and 6, we can find $\sigma_{SPT}^\ell$ to minimize $A_\ell(\sigma)$ and $\sigma_{ATSP}^\ell$ to minimize $B_\ell(\sigma)$. The $\ell^{th}$ lower bound of the total completion time of the initial flowshop problem is then:

$$LB_{VFL}^\ell = (\delta_1 n A_\ell(\sigma_{SPT}) B_\ell(\sigma_{ATSP}))$$

(23)
5 BRANCH-AND-BOUND ALGORITHM

In order to validate the lower bounds we proposed, we have incorporated them in a branch-and-bound procedure. A branch-and-bound procedure is an enumeration method that builds dynamically a search tree. Lower bounds or other criteria like dominance relations are used to cut some useless branches. We have used the separation scheme introduced by Ignall and Schrage (Ignall and Schrage, 1965): a partial sequence is progressively built as we go deeper in the search tree. A node corresponds to a partial sequence and each node consists in adding a free job at the end of the sequence. A node has as many children as its free jobs. The branching strategy is Depth-First-Search (DFS). An upper bound is computed at the root node and updated at each node. For this purpose, we have used heuristic \( PR4(15) \) presented by Pan and Ruiz (Pan and Ruiz, 2013).

The branch-and-bound procedure is detailed in Algorithm 1 and numerical results are presented in section 6. In this algorithm, \( L \) is the list of nodes that have not yet been separated and \( LC \) the list of child nodes built after separation of a node.

6 EXPERIMENTAL RESULTS

There are few studies on exact resolution of flowshop scheduling problems with criterion of total completion times. We decided to compare our branch-and-bound procedure to the one developed by Allahverdi and Al-Anzi (Allahverdi and Al-Anzi, 2006) for problem \( F3|\text{perm}, \text{sum} \sum C_i \). According to the approach proposed by Allahverdi and Al-Anzi, the processing and setup times values were randomly generated respectively from the uniform distribution on the interval \([1, 100]\) and on the interval \([0, 100k]\). We considered problems of \( n \)-jobs \( (n = 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 20) \). A class of thirty instances was generated for each number of jobs and each \( k \) value. The \( k \) value for each data set was assigned to 0.3, 0.5 and 0.8. It was assumed that all machines were available from the time zero \( (\delta_k = 0, 1 \leq k \leq m) \). To compute lower bounds \( LB_{FLS} \), we used the ATSP solving procedure developed by G. Carpaneto, M. Dell’Amico and P. Toth (Carpaneto et al., 1995). The used machine is based on an Intel Duocore 2.6GHz 4GB RAM.

In their study, Allahverdi and Al-Anzi did not in-
dicate computation times, they prefer computing the percentage of visited nodes to solve an instance relatively to the total number of nodes of the whole search tree. Therefore, we did the same in order to perform a comparison. We have reported in table 1 the mean percentage of visited nodes over the thirty instances of each class (n, k) of problems for our branch-and-bound (columns $PV_{N_{VL}}(k=0.3)$, $PV_{N_{VL}}(k=0.5)$ and $PV_{N_{VL}}(k=0.8)$) and for Allahverdi and Al-Anzi’s branch-and-bound (columns $PV_{N_{AA}}(k=0.3)$, $PV_{N_{AA}}(k=0.5)$, $PV_{N_{AA}}(k=0.8)$).

Furthermore, we have indicated in table 2 the mean computation times (in second) of our branch-and-bound, while for instances of 18 jobs we use only $LB_{VL}$ and $LB_{VL}$, and for instances of 20 jobs, we use $LB_{VL}$ and $LB_{VL}$.

In the other hand, we developed another version of this branch-and-bound procedure using only a very simple lower bound $SLB$: Lower bound $SLB$ of a node is equal to the total completion time of its corresponding partial sequence. We limited the computation times to 1500 (3000 and 9000, respectively) seconds in case of $n \in \{7, 8, 9, 10, 11, 12, 13, 14, 15\}$ ($n \in \{16, 17\}$ and $n \in \{18, 20\}$, respectively). The mean computation times of that branch-and-bound have been also reported in table 2. When it appears something like $"_{i}^{k}$ 1500”, it means that the branch-and-bound has never found the optimal solution within time limit of 1500 seconds over the thirty instances of the class. This version allowed us to evaluate the effectiveness of our proposed lower bounds.

Table 2 shows that with a small number of jobs, it takes a very short time to achieve the optimum. As the number of jobs is increasing, the version using $SLB$ proves that an effective lower bound like $LB_{VL}$ is very important to achieve the optimum within a time limit. In other words, $LB_{VL}$ are effective to eliminate unworthy branches. However, as the number of jobs is large, we need to have also a strategy in order to shorten the computation time. This strategy is under investigation.

7 CONCLUSIONS

We proposed a MaxPlus approach to tackle a $m$-machine flowshop problem with several additional constraints. The MaxPlus approach enables the transformation of a general flowshop problem into a matrix problem. Then some computations over these matrices allow us to highlight new lower bounds for the total completion time criterion, based on the resolution of a one-machine problem and an asymmetric traveling salesman problem. Despite the necessity of solving an NP-hard problem, experimental results and comparison to a previously published research have shown the effectiveness of these lower bounds.

Our further research will aim at improving these lower bounds $LB_{VL}$ as well as improving a branch-and-bound algorithm. In some cases as the number of jobs is large, a strategy in order to improve the quality of lower bounds and to shorten the computation time of the whole branch-and-bound algorithm will be also studied. Moreover, more specific constraints for a flowshop problem can be studied such as no -- wait, $min$ -- $max$ delay, $S_{min}, R_{max}$, limited stocks between machines or blocking constraints by modifying only matrix $T_{i}$ associated to job $J_{i}$. The study can be also extended to the weighted total completion time criterion $\sum_{i=1}^{m}w_{i}C_{i}$.

REFERENCES


