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

Nhat-Vinh Vo[1], Pauline Fouillet[2] and Christophe Lenté[1]
[1]*Université François-Rabelais de Tours, CNRS, LI EA 6300, OC ERL CNRS 6305, Tours, France*
[2]*Université François-Rabelais de Tours, Polytech, Tours, France*

Keywords: Scheduling, Flowshop, Total Completion Time, Lower Bound.

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 Vairaktarakis, 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 $F_3|perm,S_{nsd}|\sum C_i$ 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 and scheduling flowshop problems with minimal delays, setup and removal times (Lenté, 2001; Bouquard et al., 2006). It is also applied to flowshop problems with minimal-

Vo N., Fouillet P. and Lenté C..
General Lower Bounds for the Total Completion Time in a Flowshop Scheduling Problem - MaxPlus Approach.
DOI: 10.5220/0004833403820389
In *Proceedings of the 3rd International Conference on Operations Research and Enterprise Systems* (ICORES-2014), pages 382-389
ISBN: 978-989-758-017-8

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 traveling 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 $F3|perm;S_{nsd}|\sum C_i$ and problem $F_m|perm|\sum C_i$ 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 ($-\infty$) denoted by $\mathbb{0}$. The latter, $\otimes$, is associative, distributive on $\oplus$ and has a neutral element (0) denoted by $\mathbb{1}$. The null element, $\mathbb{0}$, is an absorbing element for $\otimes$. These properties can be summarized by stating that $\mathbb{R}_{max} = (\mathbb{R} \cup \{-\infty\}, \oplus, \otimes)$ is a dioid. It is important to note that in MaxPlus algebra in particular, and in dioids in general, the first operator $\oplus$ can not be simplified, that is $a \oplus b = a \oplus c \not\Rightarrow b = c$. Furthermore, in $\mathbb{R}_{max}$, the second operator $\otimes$ is commutative, and except $\mathbb{0}$, every element is invertible: the inverse of x is denoted by x^{-1} or $\mathbb{1}/x$. 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 $\mathbb{R}_{max}$. Let A and B be two matrices of size $m \times m$, operators $\oplus$ and $\otimes$ are defined by

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

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

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 $\mathbb{R}_{max}$ 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. $\forall j \in \{1,\dots,m\}$:

$$[A \otimes B]_{1j} = \bigoplus_{k=1}^{m} [A]_{1k} \otimes [B]_{kj} \geq [A]_{11} \otimes [B]_{1j} \tag{1}$$

Lemma 2. $\forall \ell, j \in \{1,\dots,m\}$:

$$[A \otimes B]_{1j} \geq [A]_{1\ell} \otimes [B]_{\ell j} \tag{2}$$

$$[A \otimes B]_{\ell j} \geq [A]_{\ell\ell} \otimes [B]_{\ell j} \tag{3}$$

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 $\mathcal{J} = \{J_1,\dots,J_n\}$ and another set of m-machines $\{M_1,\dots,M_m\}$. 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) σ 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_{ik} ($1 \leq k \leq m$): one per machine. An operation is at least described by a processing time p_{ik}: 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_{nsd}) and/or removal times (R_{nsd}) 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 RSb, RCb and RCb^* 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 β that is a subset of $\{min-max\ delay,\ no-wait,\ S_{nsd},\ R_{nsd},\ RSb, RCb, RCb^*\}$. Criterion γ 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 δ_k of availability of machine M_k (before execution of operation O_{ik}), starting time St_{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}. Figure 1 shows an example of flowshop problem $F_m|perm, min-max\ delay, S_{nsd}, R_{nsd}|\sum C_i$ 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_{ik}) $(1 \le k \le m), (1 \le i \le n)$ solutions of the system (S).

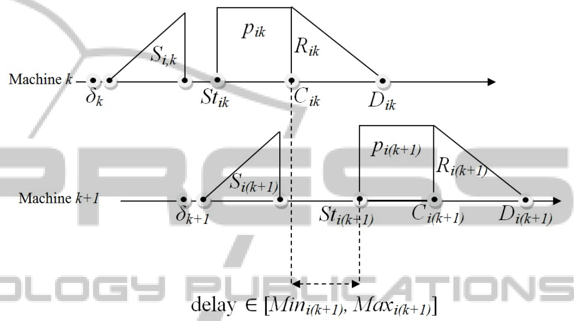

Figure 1: Example of flowshop problem: $F_m|perm, min-max\ delay, S_{nsd}, R_{nsd}|\sum C_i$.

Whatever the set of constraints β is, these calculations lead to a MaxPlus linear relation between dates of liberation D_{ik} and dates of availability δ_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. $\vec{D}_i$) be the line vector of the m dates δ_k (resp. D_{ik}): it exists a $m \times m$ MaxPlus matrix T_i computed from data of job J_i such that*

$$\vec{D}_i = \vec{\delta} \otimes T_i \qquad (4)$$

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 $t^i_{\ell c}$, in other words, $t^i_{\ell c} = [T_i]_{\ell c}$. This matrix sums up the job data (processing times, setup times, delays and so on) and the flowshop constraints.

$$T_i = \begin{pmatrix} t^i_{11} & t^i_{12} & \dots & t^i_{1m} \\ t^i_{21} & t^i_{22} & \dots & t^i_{2m} \\ \dots & \dots & \dots & \dots \\ t^i_{m1} & t^i_{m2} & \dots & t^i_{mm} \end{pmatrix} \qquad (5)$$

These results can be generalized to a sequence of jobs (Lenté, 2011), (Bouquard et al., 2006).

Definition 1 (Matrix associated to a sequence)**.** *Let σ be a sequence of ν jobs: its associated matrix is matrix T_σ defined by*

$$T_\sigma = \bigotimes_{i=1}^{\nu} T_{\sigma(i)} \tag{6}$$

Proposition 2. *If $\vec{\delta}$ is the vector of dates of availability of machines and $\vec{D}_\sigma$ the vector of dates of liberation of machines, after the execution of sequence σ, we have the relation*

$$\vec{D}_\sigma = \vec{\delta} \otimes T_\sigma \tag{7}$$

4 PROPOSED LOWER BOUNDS

This section presents lower bounds for problem $Fm|perm, \beta|\sum C_i$, with $\beta \subset \{min-max\ delay,\ no-wait,\ S_{nsd},\ R_{nsd},\ RSb, RCb, RCb^*\}$. To develop the calculations, we assume that $C_i = D_{im}$ $(1 \leq i \leq n)$. It is true unless there exists 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 σ a sequence of jobs and $\vec{\delta}$ the line vector of dates of availability of the machines ($\vec{\delta}$ = $(\delta_1, \delta_2, \ldots, \delta_m)$). The completion time of the job in k^{th} position in the sequence verifies relation:*

$$\begin{aligned}
if\ k=1: &\quad C_{\sigma(1)} \geq \delta_1 \left[T_{\sigma(1)}\right]_{1m} \\
if\ k=2: &\quad C_{\sigma(2)} \geq \delta_1 \left[T_{\sigma(1)} T_{\sigma(2)}\right]_{1m} \\
if\ k>2: &\quad C_{\sigma(k)} \geq \delta_1 \bigotimes_{j=1}^{k-2} (t_{11}^{\sigma(j)}) \left[T_{\sigma(k-1)} T_{\sigma(k)}\right]_{1m}
\end{aligned}$$

Proof. Let τ be the sub-sequence composed of the first k jobs of sequence σ. Proposition 2 and definition 1 result in:

$$\vec{D}_\tau = \vec{\delta} \otimes T_\tau = \vec{\delta} \otimes \bigotimes_{i=1}^{k} T_{\tau(i)} = \vec{\delta} \otimes \bigotimes_{i=1}^{k} T_{\sigma(i)} \tag{8}$$

Moreover $\vec{D}_{\sigma(k)} = \vec{D}_\tau$ and by assumption $C_{\sigma(k)} = D_{\sigma(k)m}$, which is the last element of vector $\vec{D}_{\sigma(k)}$, we have,

$$C_{\sigma(k)} = \left[\vec{D}_\tau\right]_{1m} = \left[\vec{\delta} \otimes \bigotimes_{i=1}^{k} T_{\sigma(i)}\right]_{1m} \tag{9}$$

If $k=1$, the application of lemma 1 results in:

$$C_{\sigma(1)} \geq \left[\vec{\delta}\right]_{11} \left[T_{\sigma(1)}\right]_{1m} = \delta_1 \left[T_{\sigma(1)}\right]_{1m} \tag{10}$$

If $k \geq 2$, by iteratively applying this lemma into equation (9), we obtain:

$$C_{\sigma(k)} \geq \left[\vec{\delta}\right]_{11} \otimes \bigotimes_{j=1}^{k-2} \left[T_{\sigma(j)}\right]_{11} \left[T_{\sigma(k-1)} T_{\sigma(k)}\right]_{1m} \tag{11}$$

Inequality (11) can be rewritten as

$$C_{\sigma(k)} \geq \delta_1 \bigotimes_{j=1}^{k-2} (t_{11}^{\sigma(j)}) \left[T_{\sigma(k-1)} T_{\sigma(k)}\right]_{1m} \tag{12}$$

$\square$

4.1.2 Lower bound of the Total Completion Time

Definition 2 (Lower Bound LB_{VFL}^1.)**.**
Given a sequence σ of n-jobs, we define:

$$\begin{aligned}
A_1(\sigma) = &\quad \bigotimes_{j=1}^{n-1} (t_{11}^{\sigma(j)})^{n-j} \\
B_1(\sigma) = &\quad \left[T_{\sigma(1)}\right]_{1m} \left(\bigotimes_{j=2}^{n} \frac{\left[T_{\sigma(j-1)} T_{\sigma(j)}\right]_{1m}}{t_{11}^{\sigma(j-1)}} \right)
\end{aligned}$$

Proposition 4.

$$\forall \sigma\ sequence: \bigotimes_{i=1}^{n} C_{\sigma(i)} \geq \delta_1^n \otimes A_1(\sigma) \otimes B_1(\sigma)$$

Proof. Considering proposition 3, we have:

$$\bigotimes_{i=1}^{n} C_{\sigma(i)} \geq \delta_1 \left[T_{\sigma(1)}\right]_{1m} \otimes \bigotimes_{i=2}^{n} \left(\delta_1 \bigotimes_{j=1}^{i-2} (t_{11}^{\sigma(j)}) \left[T_{\sigma(i-1)} T_{\sigma(i)}\right]_{1m} \right) \tag{13}$$

Rearranging the factors on the right side of inequality (13):

$$\bigotimes_{i=1}^{n} C_{\sigma(i)} \geq (\delta_1)^n \otimes \bigotimes_{i=1}^{n-2} \left(t_{11}^{\sigma(i)}\right)^{n-i-1} \otimes \left[T_{\sigma(1)}\right]_{1m} \bigotimes_{i=2}^{n} \left[T_{\sigma(i-1)} T_{\sigma(i)}\right]_{1m} \tag{14}$$

and then multiplying the inequality (14) by $\frac{\bigotimes_{i=1}^{n-1} t_{11}^{i}}{\bigotimes_{i=1}^{n-1} t_{11}^{i}}$, 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.

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

Proof.

$$A_1(\sigma) = \bigotimes_{j=1}^{n-1} (t_{11}^{\sigma(j)})^{n-j} = \bigotimes_{j=1}^{n} (t_{11}^{\sigma(j)})^{n-j+1} . \frac{\mathbb{1}}{\bigotimes_{j=1}^{n} (t_{11}^{\sigma(j)})} \tag{15}$$

The second factor is a constant, so we have to minimize $\bigotimes_{j=1}^{n} (t_{11}^{\sigma(j)})^{n-j+1}$. It is similar to the total completion time criterion in a one-machine problem $(1||\sum C_i)$ where processing times are the t_{11}^{i}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:

$$\begin{cases} \forall i \in \{1,\dots,n\}\text{: } d(0,i) = [T_i]_{1m} \\ \forall i \in \{1,\dots,n\}\text{: } d(i,0) = \mathbb{1}\ (=0) \\ \forall (i,j) \in \{1,\dots,n\}^2\text{: } d(i,j) = \dfrac{[T_i T_j]_{1m}}{t_{11}^{i}} \end{cases} \tag{16}$$

Let sequence σ_{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:

$$B_1(\sigma) = d(0,\sigma(1)) \left(\bigotimes_{i=1}^{n-1} d(\sigma(i),\sigma(i+1)) \right) d(\sigma(n),0) \tag{17}$$

□

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:

$$LB_{VFL}^{1} = n\delta_1 + A_1(\sigma_{SPT}^{1}) + B_1(\sigma_{ATSP}^{1}) \tag{18}$$

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} ($1 \le i \le n$) 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 ℓ^{th} lower bound ($\forall \ell \in \{1,\dots,m\}$). Using iteratively lemma 2, we obtain:

$$\begin{aligned} C_{\sigma(1)} &\ge \delta_\ell \left[T_{\sigma(1)}\right]_{\ell m} \\ C_{\sigma(2)} &\ge \delta_\ell \left[T_{\sigma(1)} T_{\sigma(2)}\right]_{\ell m} \\ \forall i > 2\ : C_{\sigma(i)} &\ge \delta_\ell \left(\bigotimes_{j=1}^{i-2} t_{\ell\ell}^{\sigma(j)} \right) \left[T_{\sigma(i-1)} T_{\sigma(i)}\right]_{\ell m} \end{aligned} \tag{19}$$

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

$$A_\ell(\sigma) = \bigotimes_{j=1}^{n-1} (t_{\ell\ell}^{\sigma(j)})^{n-j} \tag{20}$$

$$B_\ell(\sigma) = \left[T_{\sigma(1)}\right]_{\ell m} \left(\bigotimes_{j=2}^{n} \frac{\left[T_{\sigma(j-1)} T_{\sigma(j)}\right]_{\ell m}}{t_{\ell\ell}^{\sigma(j-1)}} \right) \tag{21}$$

we have

$$\bigotimes_{i=1}^{n} C_{\sigma(i)} \ge \delta_\ell^n A_\ell(\sigma) B_\ell(\sigma) \tag{22}$$

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

$$LB_{VFL}^{\ell} = (\delta_\ell)^n A_\ell(\sigma_{SPT}^{\ell}) B_\ell(\sigma_{ATSP}^{\ell}) \tag{23}$$

Table 1: The performance of branch-and-bound procedure for different k values.

Jobs	$PVN_{AA}(k=0.3)$	$PVN_{VFL}(k=0.3)$	$PVN_{AA}(k=0.5)$	$PVN_{VFL}(k=0.5)$	$PVN_{AA}(k=0.8)$	$PVN_{VFL}(k=0.8)$
7	5.12×10^{-1}	6.86×10^{-1}	4.39×10^{-1}	7.01×10^{-1}	5.12×10^{-1}	10.36×10^{-1}
8	2.20×10^{-1}	2.60×10^{-1}	2.74×10^{-1}	2.66×10^{-1}	2.92×10^{-1}	3.08×10^{-1}
9	5.11×10^{-2}	7.77×10^{-2}	6.46×10^{-2}	6.82×10^{-2}	6.05×10^{-2}	10.03×10^{-2}
10	3.56×10^{-2}	1.63×10^{-2}	2.79×10^{-2}	1.94×10^{-2}	2.40×10^{-2}	2.64×10^{-2}
11	1.63×10^{-2}	0.49×10^{-2}	1.28×10^{-2}	0.58×10^{-2}	1.43×10^{-2}	0.74×10^{-2}
12	3.08×10^{-3}	0.71×10^{-3}	2.74×10^{-3}	1.12×10^{-3}	3.12×10^{-3}	1.44×10^{-3}
13	6.31×10^{-4}	1.87×10^{-4}	6.30×10^{-4}	3.48×10^{-4}	4.71×10^{-4}	3.87×10^{-4}
14	5.02×10^{-5}	4.79×10^{-5}	1.22×10^{-4}	0.52×10^{-4}	1.37×10^{-4}	1.30×10^{-4}
15	2.38×10^{-5}	1.74×10^{-5}	2.19×10^{-5}	1.48×10^{-5}	2.46×10^{-5}	2.13×10^{-5}
16	1.20×10^{-5}	0.18×10^{-5}	1.21×10^{-5}	0.17×10^{-5}	1.24×10^{-5}	0.50×10^{-5}
17	5.70×10^{-6}	0.34×10^{-6}	5.30×10^{-6}	1.03×10^{-6}	5.60×10^{-6}	1.33×10^{-6}
18	5.40×10^{-7}	5.52×10^{-7}	5.00×10^{-7}	4.75×10^{-7}	5.20×10^{-7}	1.62×10^{-7}
20	-	7.04×10^{-9}	-	0.67×10^{-9}	-	3.78×10^{-9}

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 a set of free jobs. The separation of a 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.

Algorithm 1: Branch-and-Bound.

```
procedure BRANCH-AND-BOUND
    BestUB ← ∞
    Generate Root tree NRoot    // an empty node
    Compute LB(NRoot) and UB(NRoot)
    Add NRoot to list L
    while L is not empty do
        N ← top(L) // move the first node of list L
        BestUB ← min{BestUB, UB(N)}
        if LB(N) < BestUB then
            Generate children list LC of N
            for each NChild in LC do
                Compute LB(NChild)
                Compute UB(NChild)
            end for
            Sort LC
        // in non-increasing order of LB(NChild)
            for each NChild in LC do
                if LB(NChild) < BestUB then
                    top(L) ← NChild
                else
                    Delete NChild
                end if
            end for
        else
            Delete N
        end if
    end while
end procedure
```

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 $F3|perm;S_{nsd}|\sum C_i$. 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_{VFL}s, 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 indicate 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 $PVN_{VFL}(k=0.3)$, $PVN_{VFL}(k=0.5)$ and $PVN_{VFL}(k=0.8)$) and for Allahverdi and Al-Anzi's branch-and-bound (columns $PVN_{AA}(k=0.3)$, $PVN_{AA}(k=0.5)$, $PVN_{AA}(k=0.8)$).

Furthermore, we have indicated in table 2 the mean computation times (in second) of our branch-and-bound procedure for each class of instances. For instances with strictly less than 18 jobs, we used the three lower bounds LB^1_{VFL}, LB^2_{VFL} and LB^3_{VFL} in the branch-and-bound, while for instances of 18 jobs we use only LB^1_{VFL} and LB^3_{VFL} and for instances of 20 jobs, we use LB^2_{VFL} and LB^3_{VFL} .

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 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 "¿ 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: The mean computation time for each class.

Jobs	k=0.3	k=0.5	k=0.8	LB
7	0.048	0.049	0.058	$LB^1_{VFL},LB^2_{VFL},LB^3_{VFL}$
	0.030	0.039	0.041	*SLB*
8	0.100	0.098	0.176	$LB^1_{VFL},LB^2_{VFL},LB^3_{VFL}$
	0.099	0.102	0.121	*SLB*
9	0.212	0.165	0.193	$LB^1_{VFL},LB^2_{VFL},LB^3_{VFL}$
	0.683	0.602	0.709	*SLB*
10	0.291	0.285	0.353	$LB^1_{VFL},LB^2_{VFL},LB^3_{VFL}$
	6.654	5.629	6.748	*SLB*
11	0.730	0.838	0.932	$LB^1_{VFL},LB^2_{VFL},LB^3_{VFL}$
	73.674	66.924	74.569	*SLB*
12	1.313	1.779	2.33	$LB^1_{VFL},LB^2_{VFL},LB^3_{VFL}$
	906.030	1173.178	905.417	*SLB*
13	4.589	7.593	7.995	$LB^1_{VFL},LB^2_{VFL},LB^3_{VFL}$
	¿ 1500	¿ 1500	¿ 1500	*SLB*
14	16.317	15.306	34.569	$LB^1_{VFL},LB^2_{VFL},LB^3_{VFL}$
	¿ 1500	¿ 1500	¿ 1500	*SLB*
15	68.559	50.609	68.467	$LB^1_{VFL},LB^2_{VFL},LB^3_{VFL}$
	¿ 1500	¿ 1500	¿ 1500	*SLB*
16	152.996	155.756	404.959	$LB^1_{VFL},LB^2_{VFL},LB^3_{VFL}$
	¿ 3000	¿ 3000	¿ 3000	*SLB*
17	419.347	1478.93	1894.091	$LB^1_{VFL},LB^2_{VFL},LB^3_{VFL}$
	¿ 3000	¿ 3000	¿ 3000	*SLB*
18	7984.230	7638.810	2694.640	LB^1_{VFL},LB^3_{VFL}
	¿ 9000	¿ 9000	¿ 9000	*SLB*
20	2850.320	346.510	1640.350	LB^2_{VFL},LB^3_{VFL}
	¿ 9000	¿ 9000	¿ 9000	*SLB*

Table 1 shows that lower bounds LB^1_{VFL}, LB^2_{VFL} and LB^3_{VFL} are useful in reducing the number of visited nodes. They are really effective when compared to the performance in the study of Allahverdi and Al-Anzi (Allahverdi and Al-Anzi, 2006) for instances with more than nine jobs.

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_{VFL} is very important to achieve the optimum within a time limit. In other words, LB_{VFL}s 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_{VFL} 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_{nsd}, R_{nsd}, 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 $\sum_{i=1}^{n} w_i C_i$.

REFERENCES

Allahverdi, A. and Al-Anzi, F. S. (2006). A branch-and-bound algorithm for three-machine flowshop scheduling problem to minimize total completion time with separate setup times. *European Journal of Operational Research*, 169(3):767–780.

Augusto, V., Lenté, C., and Bouquard, J.-L. (2006). Résolution d'un flowshop avec delais minimaux et maximaux. In *MOSIM*.

Aydilek, H. and Allahverdi, A. (2010). Two-machine flowshop scheduling problem with bounded processing times to minimize total completion time. *Computers & Mathematics with Applications*, 59(2):684–693.

Blazewicz, J., Ecker, K.-H., Pesch, E., Schmidt, G., and Weglarz, J. (1996). *Scheduling in computer and manufacturing processes*. Springer Verlag, Berlin.

Bouquard, J.-L. and Lenté, C. (2006). Two-machine flow shop scheduling problems with minimal and maximal delays. *4or*, 4(1):15–28.

Bouquard, J.-L., Lenté, C., and Billaut, J.-C. (2006). Application of an optimization problem in Max-Plus algebra to scheduling problems. *Discrete Applied Mathematics*, 154(15):2064–2079.

Brucker, P. (2006). *Scheduling Algorithms*. Springer, 5 edition.

Carpaneto, G., Dell'amico, M., and Toth, P. (1995). Exact solution of large asymmetric traveling salesman problems. *ACM Transactions on Methematical Software*, 21(4):394–409.

Cohen, G., Dubois, D., Quadrat, J.-P., and Viot, M. (1985). A linear system-theoretic view of discret-event processes and its use for performance evaluation in manufacturing. *IEEE Trans. Automatic Control*, 30:210–220.

Della Croce, F., Narayan, V., and Tadei, R. (1996). The two-machine total completion time flow shop problem. *European Journal Of Operational Research*, 90:227–237.

Emmons, H. and Vairaktarakis, G. (2013). *Flow Shop Scheduling*. Springer US, New York, 182 edition.

Gaubert, S. (1992). *Théorie des systèmes linéaires dans les dioïdes*. PhD thesis.

Gaubert, S. and Maisresse, J. (1999). Modeling and analysis of timed Petri nets using heaps of pieces. *IEEE Trans. Automatic Control*, 44(4):683–698.

Giffler, B. (1963). Schedule algebras and their use in formulating general systems simulations. In *Industrial schduling*. Prentice Hall, New Jersey.

Graham, R. L., Lawler, E. L., Lenstra, J. K., and Rinnooy Kan, A. H. (1979). Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics*, 5(2):287–326.

Gunawardena, J. (1998). *Idempotency*. Publications of the Newton Institute.

Hanen, C. and Munier, A. (1995). *Cyclic scheduling on parallel processors: an overview*. John Wiley.

Ignall, E. and Schrage, L. (1965). Application of branch-and-bound technique to some flow shop problems. *Operations Research*, 13(3):400–412.

Johnson, S. M. (1954). Optimal two- and three-stage production schedules with setup times included. *Naval Research Logistics*, 1:61–68.

Lenté, C. (2001). *Analyse Max-Plus de problèmes d'ordonnancement de type Flowshop*. PhD thesis, Université François Rabelais de Tours.

Lenté, C. (2011). *Mathématiques, Ordonnancement et Santé*. Habilitation à diriger des recherches, Université François Rabelais de Tours.

Pan, Q.-K. and Ruiz, R. (2013). A comprehensive review and evaluation of permutation flowshop heuristics to minimize flowtime. *Computers & Operations Research*, 40(1):117–128.

Smith, W. E. (1956). Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, 3(1-2):59–66.

Su, L.-H. and Lee, Y.-Y. (2008). The two-machine flowshop no-wait scheduling problem with a single server to minimize the total completion time. *Computers & Operations Research*, 35(9):2952–2963.

Trabelsi, W., Sauvey, C., and Sauer, N. (2012). Heuristics and metaheuristics for mixed blocking constraints flowshop scheduling problems. *Computers & Operations Research*, 39(11):2520–2527.

Vo, N. V. and Lenté, C. (2013). Equivalence between Two Flowshop Problems - MaxPlus Approach. In *Proceedings of the 2nd International Conference on Operations Research and Enterprise Systems*, pages 174–177, Barcelona. SciTePress - Science and and Technology Publications.