Closed Shop Scheduling Optimisation using Max-Plus Automata

Dian Ayu Merdekawati\textsuperscript{1,a}, Subiono\textsuperscript{2,b}

\textsuperscript{1,2}Faculty of Mathematics, Computing and Data Science, Sepuluh Nopember Institute of Technology, Surabaya, Indonesia

E-mail: \textsuperscript{a}merdekawatidianayu@gmail.com, \textsuperscript{b}subiono2008@matematika.its.ac.id

Abstract. An appropriate scheduling system is needed in manufacturing activities to minimize production time and costs. Flow shop is one type of production flow that widely used in manufacturing. Based on the demand criteria, it has been known as a scheduling problem called closed shop. This scheduling emphasizes the make-to-stock (MTS) which produces a stock. We take the closed shop scheduling case studies from an existing literature and optimize scheduling using max-plus automata optimum. We compare the result with regular scheduling and the previous result in the literature. We show that the result of max-plus automata optimum scheduling are optimum compared than the existing regular scheduling and the existing previous result of the literature.

1. Introduction

Generally, the manufacturing working process is to convert raw materials into a product that have a selling value by applying machinery, labor or equipment to them. Flow shop is one of production flow type in manufacturing that has been applied in many factories. In contrast to the job shop type that has a high variety of products, flow shop productions give a low variety of products. To maximize the operating and improve the performance, besides doing the improving working methods, it is necessary to arrange the order of job that will be processed. Scheduling is a process of sequencing the whole production making on several machines. \cite{1}. Scheduling production problems can be classified according to criteria such as demand, process complexity, scheduling criteria, parameter changes, and environmental scheduling. \cite{2}. Based on the demand criteria, there are two conditions are given i.e the production of the open shop and closed shop. Closed shop production has unique tasks, no multi-functional components, and produces stock \cite{3}. The difference between the closed shop and the open shop is about Make-to-Order (MTO) and Make-to-Stock (MTS) \cite{4}. The characteristic of make-to-stock production is generally a feature of a flow shop. The challenge that arises from this production is in determining the time and quantity of the product in replenishing the stock.

Max-plus Algebra has been applied in many problems of Discrete Event System (DES) as in the measuring problem, optimizing and managing production in manufacturing systems. In 2009, Subiono and Nur Shofianah discusses the notion of applying max-plus algebra to dynamic behavior analysis of a flow shop production system model. The solution to this problem is an optimal sequence of work and routine scheduling.\cite{5}. Then in 2011 Laurent Houssin discussed the using of max-plus in the cyclic job shop scheduling problem production models. This study
show the linear max-plus model which can minimize the cycle time of the sequence. Moreover an evaluating of scheduling performance are obtained in this study. [6]. In 1995, Gaubert coined the ”max-plus automata” method as a result of max-plus algebra and automata theory development. As a product of algebraic modeling, he obtain several characterizations for completion time in cases such as worst case, optimal case, and mean case. [7]. The developing of max-plus automata theory is continuing by the research of Boukra in 2012 [8]. This study provides a new presentation of max-plus automata for minimum and maximum time execution at the specified circuit length, as well as the maximum number of events that appear up to the shortest time, can be evaluated or at least limited. Various scheduling problems in the production model have been discussed previously in several studies. Then Freeman, et al in 2005 discussed scheduling factories with closed shop scheduling optimization problems that involves both the quantity and details of logic using mixed-integer linear and sophisticated math programming. From this research, an optimal solution for the completion of 27 hours of production was made to 26.5 hours [9]. Jen-Shiang Chen in 2010 conducted research which discussed the resolution of flow shop scheduling problems using the Binary Integer Programming (BIP) method on two scheduling problem [10].

In this paper we aim to optimize flow shop production line with specification of closed shop production model using max-plus automata. By comparing the optimal scheduling from [9] and the existing regular scheduling result, we show that the optimization scheduling of max-plus automata results are more optimum.

In section II, III, and IV we recall the basic theory of language and automata, max-plus algebra, and max-plus automata, respectively. In Section V, we construct a model of max-plus automata from a study case. In Section VI, we give an analysis to discuss the results in more detail. Then In Section VII, we present the conclusion of this study.

2. Language and Automata

2.1. Word

A finite row of symbols or letters selected from a finite set alphabet (Σ) called word (or string), which denoted by w. The example of alphabet is generally in the phrase, plus word spaces, punctuation, and value of 0 and 1 bits. A word that has one length is symbolized by only a symbol. An empty word (or null word) is a special word, which has no symbol and denoted by γ. The set of word is denoted by Σ* and Σ+ is a non-empty word. In this case we denote each order letter (a0, a1, ..., an) accordingly : a0 a1 ... an, for n ∈ N.

The basic operation of words is concatenation, that is writing words as a compound. The concatenation of the words w1 and w2 is denoted simply by w1 w2. The concatenation is usually not commutative, like the rule: w1 w2 ≠ w2 w1, but in unary case, the alphabet is obviously commutative. The mth (concatenation) power of the word w is w^m = w w ... w (as many m copies). Especially we can write w^1 = w and w^0 = γ and always γ^m = γ.

2.2. Language

Suppose that Σ is a alphabet. A subset of free monoid Σ* is called language. A word can be interpreted as the singleton language, written with w. Customary notations in set theory are also used in languages such as: inclusion, proper inclusion, union, intersection, difference and complement. Belonging of a word w in the language L is denoted by w ∈ L. Concatenation product from two languages L and L’ is languages:

\[ LL’ = \{ uu’ | u ∈ L dan u’ ∈ L’ \} \] (1)

Language to Σ* from a semiring with a union is interpreted as an addition, and the product concatenation as multiplication. Therefore the symbol ∪ can be replaced with +.

\[ L_1 ∪ L_2 = L_1 + L_2 \] (2)
The \( n^{th} \) (concatenation) power of the language \( L \) is \( L^n = \{ w_1w_2...w_n | w_1, w_2, ..., w_n \in L \} \).

A word or language is accepted to automata \( C \) if it is a label from at least one successful path. The recognized language by automata \( C \) is a set, denoted by \( L(C) \), of all word accepted by \( C \). Any language \( L \subseteq \Sigma^* \) is recognized by finite automata, if there is a finite automata \( C \) hence \( L = L(C) \)

2.3. Finite Automata

**Definition 1** [7] A Finite automata is 5-tuple \( (Q, \Sigma, T, I, F) \) where : \( Q \) is finite set of state, \( \Sigma \) is alphabet, \( T \) is subset from \( Q \times \Sigma \times Q \) as transition set, \( I \) is set of initial state, and \( F \) is set of final state.

The automata model is distinguished by transition information i.e is deterministic and non-deterministic.

**Definition 2** [7] An automata non-deterministic conventional is \( C = Q, Q_i, \delta, Q_f \) with \( Q \) is set of state, \( Q_i \) is initial state, \( \delta \) : transition map \( \delta : Q \times \Sigma \rightarrow 2^Q \), and \( Q_f : \) final state, as base of max-plus automata \( C = Q, \alpha, T, \beta \) i.e \( \delta(q,a) = \{ q | T(q,a,q') \neq \varepsilon \} \), \( Q_i = \{ q | \alpha(q) \neq -\infty \} \), \( Q_f = \{ q | \beta(q) \neq -\infty \} \).

**Definition 3** [7] An automata is deterministic if the conventional automata is deterministic, mainly having a single state i.e \( Q_i = q_0 \), and if \( \forall q,a, \delta(q,a) \) at most have one element.

3. Max-plus Algebra

**Definition 4** Given \( \mathbb{R}_e \overset{\text{def}}{=} \mathbb{R} \cup \varepsilon \) with \( \mathbb{R} \) as set of real number and \( \varepsilon \overset{\text{def}}{=} -\infty \). Defined operation :

\[
\begin{align*}
\text{x} \oplus y & \overset{\text{def}}{=} \max\{x, y\} \quad \text{and} \quad \text{x} \odot y \overset{\text{def}}{=} x + y
\end{align*}
\]

for all \( x, y, \in \mathbb{R}_e \). The \( (\mathbb{R}_e, \oplus, \odot) \) is a semiring with neutral element \( \varepsilon \) and unit element \( e = 0 \). Because for all \( x, y, z, \in \mathbb{R}_e \), then it applies commutative, associative, identity, idempotent to addition, distributive property and having null element, unit element, and absorbent element. Operation \( \oplus \) and \( \odot \) can be extended to matrix \( C, D \) where

\[
(C \oplus D)_{ij} = c_{ij} \oplus d_{ij} = \max (c_{ij}, d_{ij}),
\]

for \( C, D \in \mathbb{R}^{m \times n}_\text{max}, i = 1, 2, ..., m \) and \( j = 1, 2, ..., n \).

\[
(C \odot D)_{ij} = \bigoplus_{k=1}^{p} (c_{ik} \odot d_{kj}) = \max_{k=1,..,p} (c_{ik} + d_{kj}),
\]

for \( C \in \mathbb{R}_\text{max}^{m \times p} \), \( D \in \mathbb{R}^{p \times n}_\text{max} \) and \( i = 1, 2, ..., m \), \( j = 1, 2, ..., n \).

Power operation in max-plus algebra are usually introduced using associative properties. For \( x \in \mathbb{R}_\text{max} \) and for all \( n \in \mathbb{N} \), with \( \mathbb{N} \) as the natural number and \( n \neq 0 \) is defined by \( x \odot^n \overset{\text{def}}{=} x \odot x \odot ... \odot x \) (as many \( n \) copies). It also can be written as \( x \times n \). For \( n = 0 \) is defined \( x \odot^0 \overset{\text{def}}{=} \varepsilon \), with \( e = 0 \).

4. Max-plus Automata

Firstly, before discussing the definition of max-plus automata it is very important to understand the function of dater which will have a close connection with the development of the theory of max-plus automata.
Definition 5 [7] A dater is \( y : \Sigma^* \rightarrow \mathbb{R} \cup \{-\infty\} \). For the value of \( y \) on word \( w \) can be written as \((y|w)\). Notation \((y|w)\) is defined as the completion time of the event sequence \( w \) with condition \((y|w) = -\infty\) if \( w \) does not occur.

Time \((y|w)\) can be interpreted as the duration of a successful sequence of completion task that suitable with the information sequence \( w \). Max-plus automata count the worst case from non-deterministic automata with transition time. A special case of dater Boolean (with the value \( \{-\infty, 0\} \)) obtained by the Ramadge - Wonham model, where this model connects between language with several finite devices (such as finite automata). The Ramadge - Wonham model is more clearly discussed in [13]. Likewise in the theory of max-plus, dater satisfies some limited dimensions of recurrent linear systems in the max-plus algebra. Here we consider the class function dater which is with max-plus automata.

Automata with multiplication at \( \mathbb{R}_{\max} \) semiring are called max-plus automata. The definition of max-plus automata is explained more fully as follows:

Definition 6 [7] A max-plus finite automata over an alphabet \( \Sigma \) is 4-tuple \( \mathcal{C} = (Q, \alpha, T, \beta) \) where

- \( Q \) is a finite set of states;
- \( \alpha : Q \rightarrow \mathbb{R} \cup \{-\infty\} \) is the initial state;
- \( T : Q \times \Sigma \times Q \rightarrow \mathbb{R} \cup \{-\infty\} \) is the transition time.
- \( \beta : Q \rightarrow \mathbb{R} \cup \{-\infty\} \) is the final state;

A word is called accepted if it leads from the initial state to the final state. A path with \( n \) length is a state sequence \( p = (q_1, ..., q_{n+1}) \in Q^{n+1} \). The total weight from the labeled path obtained by:

\[
\text{weight}(p, w) = \text{weight} \left( q_1 \xrightarrow{a_1} q_2 \xrightarrow{a_2} q_3 \rightarrow \cdots q_n \xrightarrow{a_n} q_{n+1} \right)
\]

where the total weight is the sum of all transition with labeled paths given. We can get the maximum weight of the path that have accepted \( word(w) \) by multiplying the \( word \) \( w = a_1...a_n \)

\[
(C|w) \overset{\text{def}}{=} \max_{p \in Q^{n+1}} \text{weight}(p, w) = \max_{q...q_{n+1}} [\alpha(q_1) + T(q_1, a_1, q_2) + \cdots + T(q_n, a_n, q_{n+1}) + \beta(q_{n+1})]
\]

A mapping is defined as:

\[
\mu : \Sigma \rightarrow \mathbb{R}^{Q \times Q}_{\max}, \mu(a)_{qq'} \overset{\text{def}}{=} T(q, a, q').
\]

Then, indentifying \( \alpha \) and \( \beta \) as a row and column vector, respectively, so that:

\[
(C|a_1...a_n) = \alpha \mu(a_1)...\mu(a_n) \beta
\]

Since \( \mu \) is a mapping that maps alphabet \( \Sigma \) so it can be extended in a unique way to a morphism \( \Sigma^* \rightarrow \mathbb{R}^{Q \times Q}_{\max} \) by setting \( \mu(a_1...a_n) = \mu(a_1)...\mu(a_n) \), then obtained

\[
(C|w) = \alpha \mu(w) \beta \tag{3}
\]

Therefore, a max-plus automata is defined as equivalent to triple \((\alpha, \mu, \beta)\) where \( \alpha \in \mathbb{R}^{1 \times Q}_{\max}, \beta \in \mathbb{R}^{Q \times 1}_{\max} \), and \( \mu \) is a morphism \( \Sigma^* \rightarrow \mathbb{R}^{Q \times Q}_{\max} \). The function dater \( y \) can also be written as:

\[y = \bigoplus_{w \in \Sigma^*} (y|w)\]
The completion time of the first $k$ events by the word $w = w_1 ... w_k$ is

$$y_k \overset{\text{def}}{=} (y|w) \cap \Sigma^k$$

To optimize the max-plus automata, S. Gaubert coined a new way to optimized the max-plus automata with Markov induction. It can be seen in Theorem 1. Then the complete construction of this model is clearly discussed in [7].

**Theorem 1** [7] There is min-plus semiring

$$l^\text{opt}_k = \alpha' (\kappa')^k \beta'$$

where

$$\kappa'_q = \inf_{a \in \Sigma, \delta(q,a)=q'} \delta(q,a)$$

$$\alpha'_q = \begin{cases} 
\varepsilon' = +\infty, & \text{if } q \neq q_0 \\
0 & \text{if } q = q_0
\end{cases}$$

$$\beta'_q = \begin{cases} 
\phi(q), & \text{if } \phi(q) \neq -\infty \\
+\infty & \text{others}
\end{cases}$$

Matrix $\kappa' \in R_{\text{min}}^{Q \times Q}$ can be seen as transition matrices analogous to Markov chain.

5. **Construction model**

The manufacturing case that used in this paper is taken from literature [9]. Model construction begins with determining the tasks and consider the low of production model. The production model consists of two machines i.e machine 1 ($M_1$) and machine 2 ($M_2$) that process 3 types of alloygroups with 2 different sizes, i.e Size1 and Size2. Therefore, the type variations that will be produced can be determined as :

- Type1 : Containing AlloyGroup1 - Size2.
- Type2 : Containing AlloyGroup2 - Size2.
- Type3 : Containing AlloyGroup3 - Size2.
- Type4 : Containing AlloyGroup1 - Size1.
- Type5 : Containing AlloyGroup2 - Size1.
- Type6 : Containing AlloyGroup3 - Size1.

Each alloygroup is processed on different path. Given three processing path, it is called "Furnace line". Alloygroup1 are worked at the Furnace line1, alloygroup2 at Furnace line2, and Alloygroup3 at Furnace line3. In production order, Size2 is worked first then resume with Size1.

There is an interval time between these Size production that must be considered. The interval time between the type process i.e 30600 on Furnace line1, 12600 in Furnace line2, but nothing (0) for Furnace line3. Information of processing time for each task is given in Table 1. All of completion time units in this paper are presented in seconds, unless the unit is given (e.g 35 'hours' or Monday, 05:00:00). Besides the processing time on each machine, it should be noted
Table 1: Table Processing Time

| Size2 | Task | Time  | Size1 | Task | Time  |
|-------|------|-------|-------|------|-------|
| M₁    | a₂   | 30600 | M₁    | a₁   | 7200  |
|       | b₂   | 19800 |       | b₁   | 25200 |
|       | c₂   | 7200  |       | c₁   | 37800 |
| M₂    | f    | 10800 | M₂    | f    | 10800 |
|       | g    | 10800 |       | g    | 10800 |
|       | h    | 10800 |       | h    | 10800 |

that there is a transition time between machines, which applies to all types i.e 1800.

The processing occurs from M₁ to M₂. But there is another processing occurs from M₁ to M₂, which is wash cast operation process. In this case, it was assumed that the wash cast process did not occur during the processing range since the presence of the wash cast process had unsettled scheduling. As a conditional machine, wash cast is only used when cleaning some defective or unwanted products is needed. Then the raw material is assumed to be in the appropriate pollutant level. A complete description of the process has shown in the Figure 1.

Figure 1: Automata model for single production, where a, b, c, f, g, h is alphabet that represent task of production and p index indicates the size of product.

Given Σ = {a, b, c} as the task that will do job in machine 1, and Σ = {f, g, h} as the task in machine 2. To distinguish assignments for production of Size1 and Size2, then given an index p to the alphabet become Σ = {aₚ, bₚ, cₚ}. The tasks f, g, h are not differentiated since the processing time for Size1 and Size2 is equal. These alphabet is arranged into a schedule or word w. Given word aw = a₁w, bw = b₂w; cw = c₃w; dw = a₄w; ew = b₅w; fw = c₆w for scheduling of Type1, Type2, Type3, Type4, Type5, and Type6 respectively, where l is the period of occurrence of an alphabet. These schedules explain the performed work of task sequence in a production flow which explained as:

- Firstly, alloygroup is processing in M₁, then it transfered to M₂
- When M₂ receive the alloygroup, then it will be processed in M₃.
- As mentioned earlier that from M₂ to M₁ there is a wash cast process, but it does not occur
during the processing range and schedule. Therefore the product that has been processed is taken out from the system.

As a closed-shop problem, the production process also considers the capacity on each machine. This capacity will affect the given scheduling. \( M_1 \) can process 30 – 50 metric tons per 1/2 hour (1800 second). But the capacity for \( M_2 \), in this case, is not limited. The initial length of process in Table 1 can be divided by the amount of capacity to determine \( l \) number for each scheduling, e.g : 30600 is the length of processing time in \( M_1 \) for Type1. With the capacity limit on \( M_1 \), 30600 divided by 1800 then get 17. The word of scheduling Type1 production becomes \( aw = a_2^{17} a_2 h \), where 17 periods of task \( a_2 \) to process in \( M_1 \) and 1 period of task \( h \) to transfer the alloygroup from \( M_1 \) to \( M_2 \), then 1 period of task \( h \). The matrices for task \( a_2 \) and \( h \) are obtained by Figure 2, i.e :

\[
\alpha = \begin{bmatrix} 1800 & \varepsilon \end{bmatrix}; \quad \mu(a) = \begin{bmatrix} 1800 & 3600 \\ \varepsilon & \varepsilon \end{bmatrix}; \quad \mu(h) = \begin{bmatrix} e & e \\ e & 10800 \end{bmatrix}; \quad \beta = \begin{bmatrix} \varepsilon \\ \varepsilon \end{bmatrix}
\] (5)

The schedule \( w = w_1...w_k \) for \( k \) events happen in one period of production, \( u \). This schedule meaning is Type1 can produce 17 product in one period of production \((u = 1)\). This amount number will continue as much as \( u \) for \( u \in \mathbb{N} \). This also applied to other schedules, become \( bw = b_2^{11} b_2 g; cw = c_2^{17} c_2 f; dw = a_1^{17} a_1 h; ew = b_1^{14} b_2 g; \) and \( fw = c_1^{21} c_1 f \) for Type2, Type3, Type4, Type5, and Type6 respectively. Hence we can obtain the amount of stocks which produced in each production. As we look on Figure 1, the automata model is accepted since the schedule \( w \) begins on machine 1 and ends on machine 2.

![Figure 2](image)

**Figure 2:** Production model of Type1 with Automata. It processed in two machines with task \( a_2 \) and task \( h \).

### 5.1. Completion Time

As mentioned in (3), we can obtained the completion time. Here is the completion of time for Type1:

\[
(y|aw) = \alpha a_2^{17} a_2 h \beta
\]

\[
= 46800
\] (6)

Then the completion time for other types is obtained \((y|bw) = 36000; (y|cw) = 23400; (y|dw) = 23400; (y|ew) = 41400; (y|fw) = 54000\). From model on Figure 2, we can obtained a critical trajectory i.e \( \lambda = \max(\frac{1800}{1+\varepsilon}, 10800) \).

By the union concept in the language automata (2), we can obtain the total production process. Thus we have:
8

• Total completion time in Furnace line1

\[ L(aw \cup \gamma \cup dw) = (y|aw) + \gamma + (y|dw) \]
\[ = 100800 \text{ or 28 hours} \]  
(7)

• Total completion time in Furnace line2

\[ L(bw \cup \gamma \cup ew) = (y|bw) + \gamma + (y|ew) \]
\[ = 90000 \text{ or 25 hours} \]  
(8)

• Total completion time in Furnace line3

\[ L(cw \cup \gamma \cup fw) = (y|cw) + \gamma + (y|fw) \]
\[ = 77400 \text{ or 21.5 hours} \]  
(9)

5.2. Optimisation of Max-plus Automata

The scheduling process is obtained by determining the deterministic representation of the previous automata model (Figure 1). Previously we already have matrices in (5) by getting the time information from Table 1 with considering the capacity in \( M_1 \). We know that \( l = 17 \) for Type1, therefore for \( w = w_1...w_k \) from \( aw = a_1^{17}a_2h \), we can get easily that \( k = 19 \).

Firstly, by setting a state \( \mu(e) = [0 \ 0] \), and using matrix in (5), we can obtained:

\[
\begin{align*}
\alpha \mu(e) &= [0 \ 0], \\
\alpha \mu(a_2^2) &= [3600 \ 5400], \\
\alpha \mu(h) &= [0 \ 10800], \\
\alpha \mu(ha_2) &= [10800 \ 10800]
\end{align*}
\]

From these vectors we get \( Q = \{e, a_2^2, h\} \). Since the concept of optimization is using Markov induction, the transition possibilities that can arise include: \( a_2^2, a_2h, ha_2, \) and \( h^2 \).

![Figure 3: Deterministic Representation of Automata for Type1, with e, a_2 and h as new state.](image-url)
By this, we can get some of the following equations:

\[
\alpha \mu(a_2^2) = 1800 \mu(a_2), \quad \alpha \mu(a_2 h) = 3600 \mu(h) \\
\alpha \mu(h a_2) = 10800 \mu(e), \quad \alpha \mu(h^2) = 10800 \mu(h)
\]  

(10)

From (10), we can construct a new automata model where it is called representation of deterministic automata that shown in Figure 3, and it is used to construct matrices in (11)

\[
\alpha' = \begin{bmatrix} 0 & \varepsilon & \varepsilon \end{bmatrix}; \quad K' = \begin{bmatrix} \varepsilon & 1800 & 1800 \\
1800 & 1800 & 3600 \\
10800 & \varepsilon & 10800 \end{bmatrix}; \quad \beta' = \begin{bmatrix} 0 & 3600 \\
3600 & 10800 \end{bmatrix}
\]

(11)

We can obtained an optimal performance for Type1 by calculating these matrices in min-plus multiplication as in (1)

\[
l_{19}^{opt} = \alpha'(K')^{19} \beta' \\
= 34200
\]

This is the optimal completion time for first production \((u = 1)\) of Type1. This also being applied to the other types. With the same way as in the (7) - (9) we can obtain the total completion time for max-plus automata optimal in each Line furnaces. The results of the whole production are presented in the Table 3 for \(u = 10\), and Table 4 and Table 5 for all results have been converted to day names and hour in range four weeks.

| Furnace Line1 | Production Process in Three Furnace |
|---------------|-------------------------------------|
| **Production** | **Scheduling** |
| u             | RS | MAS | MAOS |
| 1             | 97200 | 100800 | 82800 |
| 2             | 194400 | 201600 | 165600 |
| 3             | 291600 | 302400 | 248400 |
| 4             | 388800 | 403200 | 331200 |
| 5             | 486000 | 504000 | 414000 |
| 6             | 583200 | 604800 | 496800 |
| 7             | 680400 | 705600 | 579600 |
| 8             | 777600 | 806400 | 662400 |
| 9             | 874800 | 907200 | 745200 |
| 10            | 972000 | 1008000 | 828000 |
Table 3: Comparison Table of Completiion Time Production Process in Three Furnace (Cont.)

| Furnace Line2 | Production | Scheduling |
|--------------|------------|------------|
|              | u          | RS         | MAS        | MAOS       |
| 1            | 86400      | 90000      | 64800      |
| 2            | 172800     | 180000     | 129600     |
| 3            | 259200     | 270000     | 194400     |
| 4            | 345600     | 360000     | 259200     |
| 5            | 432000     | 450000     | 324000     |
| 6            | 518400     | 540000     | 388800     |
| 7            | 604800     | 630000     | 453600     |
| 8            | 691200     | 720000     | 518400     |
| 9            | 777600     | 810000     | 583200     |
| 10           | 864000     | 900000     | 648000     |

| Furnace Line3 | Production | Scheduling |
|--------------|------------|------------|
|              | u          | RS         | MAS        | MAOS       |
| 1            | 75600      | 77400      | 52200      |
| 2            | 151200     | 154800     | 104400     |
| 3            | 226800     | 232200     | 156600     |
| 4            | 302400     | 309600     | 208800     |
| 5            | 378000     | 387000     | 261000     |
| 6            | 453600     | 464400     | 313200     |
| 7            | 529200     | 541800     | 365400     |
| 8            | 604800     | 619200     | 417600     |
| 9            | 680400     | 696600     | 469800     |
| 10           | 756000     | 774000     | 522000     |

6. Results Analysis

The comparison result of Regular Scheduling (RS), Max-plus Automata Scheduling (MAS), and Max-plus Automata Optimum Scheduling (MAOS) found in Table 3 and Table 2. For the first production in Furnace line1, RS gives a general processing time, which is 97200 seconds or 27 hours. While MAS give a greatest completion time i.e 100800 or 28 hours, MAOS give a smallest result than other scheduling, i.e 75600 or 21 hours. Consequently, the MAOS result on the 10th production are smallest than other scheduling. This also applies at Furnace line2 and Furnace line3 results. MAOS give completion time i.e 64800 or 18 hours in Furnace line2, while MAS give 90000 or 25 hours in Furnace line2. Then in Furnace line3, MAOS give 52200 or 14,5 hours in Furnace line3, while MAS give 77400 or 21,5 hours. The results given by MAS are always greater than the RS results, this is because the max-plus non-deterministic automata (like production model in our case) always calculates the worst case. We do this comparison until u-production, then we get the result as in Figure 4 for all Furnaces line in this production.

It shows the black line as representation of MAS. The red line as representation of RS, and the blue line represents the MAOS results. The black line is always at the top than others lines.
Figure 4: Comparison graphic of production scheduling in three Furnace lines. Black line as representation of Max-plus Automata Scheduling, red line as representation of Regular Scheduling, and blue line as representation of Max-plus Automata Optimum Scheduling.

Figure 5: Comparison graphic of production scheduling in three Furnace lines with zoomed. Line with dot is representation of Furnace line1 production. Line with cross is representation of Furnace line2 production. The dashed line is representation of Furnace line3 production.

The more quantities produced, the higher the time’s graph. The red line always in the middle, between the black and blue lines. Then the blue line always at the bottom of the other lines. It means the MAOS result are far more optimal than others. As the number of production increases, the time value also increases but the blue lines does not exceed the red line.

The range production that given is for 4 weeks. Table 4 and Table 5 shows scheduling result for MAOS that has been converted into day names and hour. The production process at Furnace line1 and Furnace line2 starts on Monday at 06:00:00, while production at Furnace line3 starts on Monday at 06:30:00. From this table, Furnace line1 get \( u = 31 \) (it can produce 31 times of production), Furnace line2 \( u = 36 \), and Furnace line3 \( u = 45 \). It means for once time production in Furnace line1, it produces 17 product for Type1 (for \( l = 17 \) as discussed in (6)), and 4 product for Type4. Furnace line get 11 product of Type2 and 14 product of Type5 for once production. Then in Furnace line3 get 4 product of Type3 and 21 product of Type6 for once production.

| \( u \) | Furnace Line1 | Furnace Line2 | Furnace Line3 |
|-------|---------------|---------------|---------------|
| 0     | Mon, 06:00:00 | Mon, 06:00:00 | Mon, 06:30:00 |
| 1     | Tues, 03:00:00 | Tues, 00:00:00 | Mon, 21:00:00 |
| 2     | Wed, 00:00:00  | Tues, 18:00:00 | Tues, 11:30:00 |
| 3     | Wed, 21:00:00  | Wed, 12:00:00  | Wed, 02:00:00  |
| 4     | Thurs, 18:00:00 | Thurs, 06:00:00 | Wed, 16:30:00  |
| 5     | Fri, 15:00:00  | Fri, 00:00:00  | Thurs, 07:00:00 |
| 6     | Sat, 12:00:00  | Fri, 18:00:00  | Thurs, 21:30:00 |
We obtained the total number of stocks by multiplying each period of production \((l)\) and production times \((u)\). As a result, the product of Type1 is 527 products, Type2 is 396 products,
Type3 is 180 products, Type4 is 124, Type5 is 504 product, and Type6 is 945 product. The make span of the MAOS is in 21 hours for once production. The production process in four weeks finished on Sunday, 11:00:00. If we continue the MAS results in Table 3 for four weeks range and convert it to day names and hour, then we obtained that the last production during four weeks is finish on Sunday, 08:00:00 with production capabilities 22 production in Furnace line1, 26 productions in Furnace line2, and 30 productions in Furnace line3. Then in RS, the entire productions ends on Sunday, 14:30:00 with production capabilities 24 productions in Furnace line1, 27 productions in Furnace line2, and 32 productions in Furnace line3. As mentioned before, the number of stock that had produced by the scheduling obtained by multiplying each period of production \((l)\) and production times \((u)\). The result for all types has shown in Table 6. It compare the number of stock by RS, MAS, and MAOS.

| Production | RS   | MAS  | MAOS |
|------------|------|------|------|
| Type1      | 408  | 374  | 527  |
| Type2      | 297  | 286  | 393  |
| Type3      | 128  | 120  | 180  |
| Type4      | 96   | 88   | 124  |
| Type5      | 378  | 364  | 504  |
| Type6      | 672  | 630  | 945  |
| **Total**  | 1979 | 1862 | 2676 |

It can be seen that the number of products for each type in MAS is smaller than the product results which given by RS. Then for MAOS results, it can be seen that the results are greater than RS. The total number of products that obtained by RS is 1979 product, 1862 product by MAS, and 2676 product by MAOS. Therefore, the products number which given by MAOS are greater than other scheduling. In factory production, MAOS gives an benefit with minimum processing time and great number of products.

7. Conclusion
Through this comparison we show that max-plus automata optimum scheduling have an optimal result for the scheduling than the others. The production process starts on Monday 06:00:00 and finish on Sunday, 11:00:00 (on the 4th week), with the total production results are 527 products of Type1, 393 products of Type2 180 products of Type3, 124 products of Type4, 504 products of Type5, and 945 products of Type6. This amount is excessively than the stock results of Regular Scheduling and Max-plus Automata Scheduling. Moreover, we obtained the make span time for the max-plus automata optimum scheduling is 21 hours. It is more optimum than the previous result on literature that obtained 26.5 hours. It also more optimum than the Regular Scheduling i.e 27 hours. Max-plus automata method is very suitable for the calculation process in manufacturing because it can calculate recurring time, with complicated assignment and flow patterns.
References

[1] Conway R W, Maxwell W L, and Miller L W 1967 *Theory of Scheduling* (Massachusetts : Addison Wesley)

[2] Jones A W, Rabelo L C 1999 *Survey of Job Shop Scheduling Techniques* (John Wiley & Sons, Inc. All)

[3] Jayakumar S 2000 *A heuristic approach for solving open shop scheduling problems, minimizing makespan and resource idleness*, *Thesis Doctor of philosophy, Departement of Management studies Anna University* (Chennai, India)

[4] Soman C A, Donk P v, Gaalman G 2002 *Combined Make-to-Order and Make-to-Stock in a Food Production System* (Holland : University of Groningen, Research Institute SOM)

[5] Subiono, Shofianah N 2009 Using Max-Plus Algebra in The Flow Shop Scheduling. *IPTEK, The Journal of Technology and Science* **20**, 83

[6] Houssin L 2011 Cyclic Jocshop Problem and (Max,plus) Algebra. *Proc. of the 18th World Congress, The International Federation of Automatic Control* (Milano : IFAC-PapersOnline) **18**, 2717-21

[7] Gaubert S 1995 Performance evaluation of Max Plus Automata. *IEEE Trans. On Automatic Control* **40**, 2014-25

[8] Boukra R, Lahaye S, Boimond J L 2012 New representations for (max,+) automata with applications to the performance evaluation of discrete event systems. *IFAC Proc. Volumes*

[9] Freeman N, Kelly J D, Zhang W, Mann J L 2005 Scheduling Optimisation for Aluminium Smelter Casthouses. *TMS (The Minerals, Metals, & Material Society) Light Metals, 843-7*

[10] Chen J S 2010 Optimization Models for the Flow-Shop Scheduling Problem with Multiple Orders Per Job. *The 40th International Conf. on Computers & Industrial Engineering* Japan.

[11] Pin J E 2015, *Mathematical Foundations of Automata Theory*. (Springer Birkhuser)

[12] Ruohon K 2009, *Formal Language* (Online Lecture Material)

[13] Ramadge P J G, Wonham W M 1989 The Control of Discrete Event Systems. *IEEE Proc. Special Issue on Discrete Event Systems, 11*, 81-97.

[14] Subiono 2015, *Aljabar Min-Max Plus dan Terapannya [Min-Max Algebra and Its Application]*. (Surabaya : Institut Teknologi Sepuluh Nopember)