# A Mathematical Model for Cyclic Scheduling With Limited Work-In-Process and Cycle Time Minimization

Mohamed Amin Ben Amar National School Of Computer Sciences, ENSI, University of Manouba, Tunisia *aminbenamar@yahoo.fr*  Hervé Camus LAGIS, LSO team, EC Lille Villeneuve d'Ascq, France herve.camus@ec-lille.fr Ouajdi Korbaa ISITCom, Hammam Sousse, University of Sousse, Tunisia ouajdi.korbaa@Centraliens-Lille.org

In this paper, we will address the cyclic scheduling problem with limited WIP (Work-in-process). In these systems, we have to minimize the cycle time of the production while working with a limited level of WIP. We consider here production systems with linear jobs (every operation was followed and preceded by only one operation). Many methods have been proposed to solve the cyclic scheduling problem. Among them, we focus on the exact approach, and more precisely the mathematical programming. In this paper, we will propose a *Linear Program Model* for cyclic scheduling with limited WIP and cycle time minimization.

Cyclic Scheduling, Mathematical Modeling, Rational Linear Program, limited Work-In-Process, Cycle Time

#### Minimization.

## 1. INTRODUCTION

The job-shop problem is one of the most popular scheduling problems. The popularity is based on its wide range of applications. Similarly, cyclic scheduling problems take place in different applications areas such as compiler design, automated manufacturing systems, digital signal processing, railway scheduling, timetabling, etc. We will focus here on the cyclic job-job problem. In this domain, the production consists of cyclic jobs assigned to machines.

The use of cyclic behavior is justified by the complexity of this problem. The Scheduling problems are well known to be highly combinatorial. It has been shown that project planning problems are of polynomial complexity and that cyclic scheduling problems are NP-complete Serafini (1989). Taking into account transformation tasks, makes the first problem NP-hard in most cases and keeps the second one in the NP-complete class. Hence, the use of heuristics is generally recommended. The cyclic scheduling allows to avoid the scheduling of the whole tasks and to handle the combinatorial explosion of the problem by considering only a small pattern (cycle). It can be a possible solution to global scheduling. The answer to the total demand will be given by the repetition of a sequence known as cyclic scheduling. However, the optimal scheduling of a cycle does not guarantee the optimality of the total production, since "the sum of optimal sub-paths is not necessarily an optimal path" Bellmann (1957). That's why the cyclic behavior is still a heuristic.

We suppose that a production is made by jobs and a job consists of a set of tasks. These tasks have to be processed in a given order (precedence constraints) on the machines and this order may differ among jobs. Every tasks have one successor and one predecessor (linear jobs). Parts processed by these tasks are handled by a transport system which consists of physical supports called pallets. These parts, which represent unfinished products, represent the Work-in-process (WIP). In factories, WIP levels between machines have a limited capacity. This is mainly due to the limited physical space available to store the parts temporarily and the limited capacity of transport system. The processing times are deterministic and fixed from the planing step. No pre-emption is allowed, i.e., once an operation is started, it must be completed before another operation can be started on that machine.

In the cyclic context, many works aim to minimize the "Cycle Time" criteria, which is the time difference between two succeeding occurrences of one task for a given set of constraints. The optimal cycle time is equal to the workload of the bottleneck machine(s)<sup>1</sup>. However, to reach this optimal value, the bottleneck machine(s) need(s) to be fed immediately after performing each tasks. This means that we have to use enough WIP (parts) in order to avoid waiting on machines. In the case of a few number of WIP (one for example), the machines will be pending for parts (particularly, the bottleneck machine(s)) and the throughput will not be optimized. On the other hand, the WIP represents the intermediate stock and is an economic criterion. Indeed, for each new WIP, we have to associate a transport pallet. Thus, we have to minimize, also, the WIP in the system. A classical result in cyclic scheduling theory is that: considering that the WIP is always available (with a sufficient number of WIP) at the input of the bottleneck machine(s) allows reaching the optimal cycle time. However, few works in the literature have addressed the problem of minimizing the cycle time with a limited level of WIP Seo (2002). Indeed, Seo considers job shops with overtaking degrees. In these shops, some operations of a specific job instance are processed prior to some operations of the previous instance of the same job. This means that, if there is an operation x from a job J with an overtaking degree equals to 1, then there must be two instances of the job J on each cycle of the repetitive production command. As a consequence, two WIP are needed to allow this production case. By introducing the overtaking notion, Seo looks for CT minimization. However, he considers that there must be a trade-off between the WIP and the cycle time. Hence, he proposes to work with a limited level of WIP (by fixing the overtaking degree for each operation).

The main purpose to consider limited WIP in production systems is always for economical purposes. Indeed some production systems need sophisticated pallets in order to handle their WIP. These pallets can be so expensive or even unavailable (especially for systems that produce pieces that are relatively large or huge). For example, because of the parts are expensive, they are usually processed when there is a need. In the airplane case, the engine and the electronic parts are very expensive and are, therefore, manufactured only when needed. Thus it is of great concern to these companies to reduce their WIP. Consequently, the production have to be adapted to this new constraint, and this by fixing the WIP level as a hard constraint and then looking for optimizing the cycle time.

Producing with a limited WIP is frequently used by systems that use the Just-In-Time production (JIT) (produce only if there is a customer demand Monden (1983), Shingo (1981)), like in car production systems. In JIT production systems, the WIP level is controlled by the *Constant WIP* method (ConWIP) Hopp (1996). It consists in a control strategy that limits the total number of parts allowed into the system at the same time. The ConWip can be viewed as a variant of the Kanban systems Hall (1981). Note that the ConWIP and the Kanban systems are applied within JIT production system and we are interested here on the predictive context since we deal with deterministic scheduling problems. This means that all the production data; parts, number of parts to produce and production sequences are previously set so that there is no possible random interaction with the environment during this process. However, it is always interesting for the JIT production to have an estimated value on the maximal throughput of the system while using a fixed level of WIP.

Another field that deals with the limited WIP is the *Hoist Scheduling Problem* (HSP) Che (2010), Manier (2003). Indeed the HSP deals with the scheduling of handling devices in electroplating facilities. The parts are carried and handled by one (in general) hoist. Due to tight timing process constraints - minimal and maximal processing time in tanks, it is difficult to deal with several WIP using only a single hoist. Thus, the production must be with limited WIP in order to avoid the loose of parts.

In this work, we will consider that the WIP is limited and we look for optimizing the cycle time. For the resolution approach, we have chosen the exact one. We will model this cyclic scheduling problem with a mathematical model. In fact, this paper can be viewed as an extension of the work proposed in Ben Amar (2009). Obviously, the main objective is not the same, but both works deals with cyclic scheduling problems that involves WIP. The objective function in the mathematical model proposed in this work consists on minimizing the cycle time. However, the major difficulty in this mathematical model is that, while we try to optimize the cycle time, the starting time of operations is not fixed yet. This involves ambiguity to compare the current WIP with the fixed limited level.

The remainder of this paper is organized as follows. In section 2, we will expose two mathematical models representing the cyclic scheduling problem with limited WIP and cycle time minimization. The first one is non-linear and the second one is a consequence of the linearization of the nonlinear version. The linearization process will be

 $<sup>^1\</sup>mathrm{Machine}$  with the maximum workload, i.e. machine m for which the sum of the processing time of tasks performed on m is the greatest

also presented in the same section. In section 3, we illustrate this approach with an example from literature. To conclude, we propose several perspectives to extend this work.

## 2. MATHEMATICAL MODELS

### 2.1. Job Shop Notations

In this section, we will de present the notations used to define a job shop *J*:

- *Machines:* The set  $\mathbb{M} = \{m_1, m_2, \dots, m_{|\mathbb{M}|}\}$ defines the set of machines of *J*. We denote by  $\mathbb{M} = |\mathbb{M}|$  the cardinal number of  $\mathbb{M}$ . These machines are renewable and not shared by operations. Which means that, they are reusable once they have finished the execution of a task and can only process one task at a time.
- *Production:* We define a production as a set of type of pieces that have to be produced  $\mathbb{G} = \{g_1, g_2, \ldots, g_{|\mathbb{G}|}\}$ . We denote by G the cardinal number of  $\mathbb{G}$ , and we order the type of pieces of the production by the formal parameter  $i \in [\![1, G]\!]$ .
- Operating Sequence: We define an operating sequence (type of piece) g<sub>i</sub> of the production system J as a sequence of operations. K<sub>i</sub> denotes the number of operations of g<sub>i</sub>.
- Operation: We define an operation (or a task) of J as  $o_{ij}, i \in [\![1, G]\!]$  represents the associated operating sequence  $g_i$  and  $j \in [\![1, K_i]\!]$  is the index of the operation in  $g_i$ . We denote by  $m_{ij} \in \mathbb{M}$  the machine used to perform  $o_{ij}$ .  $d_{ij}$  is called the duration of the corresponding operation.
- In the following, we will denote by  $\mathbb{O}^{\mathbb{G}} = \left\{ (o_{i,j})_{j \in [\![1,\mathsf{K}_i]\!]} \right\}_{i \in [\![1,\mathsf{G}]\!]} \equiv \left\{ \left( (m_{ij}, d_{ij}) \right)_{j \in [\![1,\mathsf{K}_i]\!]} \right\}_{i \in [\![1,\mathsf{G}]\!]}$  the set of all operations of  $\mathbb{G}$ . We denote by  $O^{\mathsf{G}}$  the cardinal number of  $\mathbb{O}^{\mathbb{G}}$ . Then we have:  $\sum_{i=1}^{\mathsf{G}} \mathsf{K}_i = O^{\mathsf{G}}$ .
- $\mathbb{O}_m^{\mathbb{G}}$  will denote the operations of  $\mathbb{O}^{\mathbb{G}}$  that have to be performed on the machine  $m \in \mathbb{M}$ ,  $\mathbb{O}_m^{\mathbb{G}} = \{o_{i,j} \in \mathbb{O}^{\mathbb{G}}, s.t., m = m_{i,j}\}.$

#### 2.2. Mixed Integer Non-Linear Program

In this section, we will start by presenting a *Mathematical Programming Model* with nonlinear constraints for the cyclic scheduling problem with limited WIP and cycle time minimization ("Figure.1"). This model is based on works developed earlier (Ben Amar (2009) and Bourdeaud'huy (2006)), which propose a Mixed Integer Linear Program (MILP) modeling the Cyclic Scheduling problems with WIP minimization. Indeed, this MILP has been adapted in order to model Cyclic scheduling problem with limited WIP. Therefore, the objective function changed from minimizing the WIP to minimizing the CT and inequality 9 has been added to constraint the total number of WIP in the system to be less or equal to a specific value. In the following, the variables and the constraints used in "Figure.1" will be explained:

Minimize: CT s.t.			(1)
• $\forall i \in \llbracket 1, G \rrbracket, \forall j \in \llbracket 1, K_i \rrbracket,$			
$t_{ij}$	$\in$	$[0, C^*_{max} - 1]$	](2)
$lpha_{ij}$	$\in$	$\{0,1\}$	(3)
$eta_{ij}$	$\in$	$\{0,1\}$	(4)
$t_{ij} - t_{i,(j\%K_i)+1} - B\cdot\alpha_{ij}$	$\geq$	$1 - B - d_{ij}$	(5)
$t_{ij} - t_{i,(j\%K_i)+1} - B\cdot\alpha_{ij}$	$\leq$	$-d_{ij}$	(6)
$t_{ij} - t_{i,(j\%K_i)+1} - B \cdot \beta_{ij} - CT$	$\geq$	$1 - B - d_{ij}$	(7)
$t_{ij} - t_{i,(j\%K_i)+1} - B \cdot \beta_{ij} - CT$	$\leq$	$-d_{ij}$	(8)
$\sum_{\alpha_{ij} \in \mathcal{O}^{\mathbb{G}}} (\alpha_{ij} + \beta_{ij})$	$\leq$	M	(9)
$o_{ij} \in 0^{\circ}$			

- $\forall m \in \mathbb{M}, \forall o_{ij}, o_{kl} \in \mathbb{O}_m^{\mathbb{G}}$ 
  - $\delta_{ij}^{kl} \in \{0,1\}$  (10)

$$t_{ij} - t_{kl} + CT.\delta_{ij}^{kl} - CT \leq -d_{ij} \tag{11}$$

$$t_{kl} - t_{ij} - CT.\delta_{ij}^{kl} \leq -d_{kl} \tag{12}$$

**Figure 1:** Non-Linear Program Model for the Cyclic Job Shop with Limited WIP and Cycle Time Minimization

- CT represents the Cycle Time variable,  $CT \in \mathbb{N}$ . The lower bound of CT is  $C_{max}^* = \max_{m \in \mathbb{M}} (\sum_{o_{i,j} \in \mathbb{O}_m^{\mathbb{G}}} (d_{i,j}))$ . This bound is presented in the mathematical model by the constraint 2. In fact, if there is sufficiently WIP in the system, the cycle time will be fixed by the bottleneck machine(s). In this case, the throughput will be at his maximum level, since the critical machine work at 100% of its capacity.
- Variables  $t_{i,j} \in [0, CT 1]$  correspond to the activation date of the operation  $o_{i,j}$  within the considered cycle. We consider here discrete systems and we suppose that  $t_{i,j} \in \mathbb{N}$  and  $d_{ij} \in \mathbb{N}^*$ .
- The variables  $\delta_{ij}^{kl} \in \{0,1\}$  are the binary variables corresponding to the order between

operations performed on the same machine. Let  $o_{i,j}$  and  $o_{k,l}$ , with  $(i,j) \neq (k,l)$ , two operations that must be performed on the same machine. If  $o_{k,l}$  starts after the end of  $o_{i,j}$ , then  $\delta_{i,j}^{k,l} = 1$ , else, if  $o_{i,j}$  starts after the end of  $o_{k,l}$  then  $\delta_{i,j}^{k,l} = 0$ . Formally:

$$\delta_{i,j}^{k,l} = \begin{cases} 1 \text{ if } t_{i,j} + d_{i,j} \leqslant t_{k,l} \\ 0 \text{ if } t_{k,l} + d_{k,l} \leqslant t_{i,j} \end{cases}$$

- The  $\alpha_{i,j}^{i,s,J}$  and  $\beta_{i,j}^{i,s,J}$  correspond to binary variables used to compute the WIP. Within a production system, jobs consist of operations which have to be processed in a given order (*precedence constraints*) on the machines and this order may differ among jobs. If the precedence constraints between two operations are not respected in a schedule, this means that the system will need extra WIP. In this case, the decision variable  $\alpha$  related to these two operations is fixed to 1. Formally:
  - $\alpha_{i,j} = 1$  if  $o_{i,(j \% K_i)+1}$  is executed before the completion time of  $o_{i,j}$ , where  $o_{i,(j \% K_i)+1}$  stands for the successor of operation  $o_{i,j}$  within the job;
  - $\beta_{i,j} = 1$  if  $o_{i,j}$  overlaps two cycles and completes after the activation time of  $o_{i,(j \ \% \ K_i)+1}$  on the next cycle;

More explanations for  $\alpha$  and  $\beta$  can be found in Ben Amar (2009), since these two variables keep the same meaning for systems with or without assembly/disassembly tasks.

•  $B \in \mathbb{N}$  is a constant used to constrain the discrimination variables  $\alpha_{i,j}$  and  $\beta_{i,j}$  in a linear way. It has to be "big enough" (lower bound:  $2 \cdot CT - 1$ ) in order to make the inequalities (5) to (8) valid. This lower bound was computed as follow:

In order to consider (5) as a valid inequality, we must have:  $t_{i,j} - t_{s(i,j)} - B \cdot \alpha_{i,j} \leq -d_{i,j}$ , and if we consider  $\alpha_{i,j} = 1$  then:  $t_{i,j} - t_{s(i,j)} + d_{i,j} \leq B$ . In addition, we know that  $t_{i,j} \leq CT - 1$  and  $d_{i,j} \leq CT$  then B must respect the following inequality:

$$B \ge 2 \cdot CT - 1 \tag{13}$$

- The objective function of this mathematical model is to minimize the CT (1). Note that inequalities (5) and (6), constraint the decision variables  $\alpha$  according to their meanings presented previously. Similarly, inequalities (7), (8) constraint the variables  $\beta$  and (11), (12) constraint the variables  $\delta$ .
- Inequality (9) constraints the total number of WIP in the system to be less or equal to

M, which represents the maximum number of WIP allowed in the production system. Let suppose that  $WIP_{opt}$  represents the minimal WIP computed in the case of  $CT = C^*_{max}$ .  $WIP_{opt}$  represents the minimal WIP level which allows to the cycle time to be equal to the workload of the critical machine  $(C^*_{max})$ . This miminimal level of WIP can be computed using the Mixed Integer Linear Program proposed in Ben Amar (2007) and Bourdeaud'huy (2006). After defining M and  $WIP_{opt}$ , two case can be considered:

- If  $M < WIP_{opt}$ : this means that the WIPs available for the production system does not allow working with the speed of the critical machine. In this case, CT will be necessary greater than  $C^*_{max}$ . In fact, due to the lack of WIP, the critical machine will be pending for input pieces, and this delay make  $CT > C^*_{max}$ . As a consequence:

if 
$$M < WIP_{opt}$$
 then  $CT > C^*_{max}$  (14)

In this case, our objective will be to minimize the  ${\cal CT}.$ 

- If  $M \ge WIP_{opt}$ : this means that the WIPs are sufficiently available in order to work at the speed of bottleneck machine. In this case:

$$CT = C_{max}^* \tag{15}$$

• Note that the inequalities (11) and (12), are used to guarantee that machines will not perform more than one operation, simultaneously, within the cycle time. These two inequalities have a non-linear term. Indeed,  $(CT.\delta_{ij}^{kl})$  represents the multiplication of two variables that have to be computed by the resolution of the mathematical model. This made the mathematical program (proposed in "Figure.1" a non-linear one.

In the following, we will present how we have linearized these constraints and highlight the related changes that we have made to guarantee that variables will preserve their meaning and nature.

# 2.3. Linearization of the Non-Linear Program

The mean idea here is to change variables by introducing CTi = 1/CT since  $CT \neq 0$ . As a consequence, all other variables will be changed and divided by CT. The resulting mathematical model is a *Rational Linear Program* presented in "Figure.2".

• Within the first model, we look for minimizing the Cycle Time (CT). In the second model, after using CTi, our objective will be to

Maximize: 
$$CTi \ s.t.$$
 (16)

• 
$$\forall i \in \llbracket 1, \mathsf{G} \rrbracket, \forall j \in \llbracket 1, \mathsf{K}_i \rrbracket$$
,

$$t'_{ij} \in [0,1[ \tag{17})$$

$$\alpha'_{ij} \in \left[0, \frac{1}{C_{\max}^*}\right]$$
(18)
  
 $\beta'_{ij} \in \left[0, \frac{1}{C_{\max}^*}\right]$ 
(19)

$$t'_{ij} - t'_{i,(j \ \% \ \kappa_i)+1} - B \cdot \alpha'_{ij} - (1 - B - d_{ij}) \cdot CTi \geq 0$$
(20)  
$$t'_{ij} - t'_{i,(j \ \% \ \kappa_i)+1} - B \cdot \alpha'_{ij} + d_{ij} \cdot CTi \leq 0$$
(21)  
$$t'_{ij} - t'_{i,(j \ \% \ \kappa_i)+1} - B \cdot \beta'_{ij} - (1 - B - d_{ij}) \cdot CTi \geq 1$$
(22)  
$$t'_{ij} - t'_{i,(j \ \% \ \kappa_i)+1} - B \cdot \beta'_{ij} + d_{ij} \cdot CTi \leq 1$$
(23)  
$$\sum (\alpha'_{ij} + \beta'_{ij}) - M \cdot CTi \leq 0$$
(24)

$$o_{ij} \in \mathbb{O}^{\mathbb{G}}$$
 $CTi - \alpha'_{ij} \geq 0$  (25)
 $CTi - \beta'_{ij} \geq 0$  (26)

$$\begin{array}{rcl}
\alpha'_{ij} - C \cdot y_{ij} &\leq 0 & (\mathbf{27}) \\
CTi - \alpha'_{ij} - C \cdot (1 - y_{ij}) &\leq 0 & (\mathbf{28}) \\
\beta'_{ij} - C \cdot z_{ij} &\leq 0 & (\mathbf{29}) \\
CTi - \beta'_{ij} - C \cdot (1 - z_{ij}) &\leq 0 & (\mathbf{30}) \\
\sum & (y_{ij} + z_{ij}) &\leq M (\mathbf{31})
\end{array}$$

$$\sum_{o_{ij}\in\mathbb{O}^{\mathbb{G}}} (g_{ij} + \sim_{ij})$$

•  $\forall m \in \mathbb{M}, \forall o_{ij}, o_{kl} \in \mathbb{O}_m^{\mathbb{G}}$ 

$$\delta_{ij}^{kl} \in \{0,1\}$$
(32)  
$$t'_{ij} - t'_{kl} + \delta_{ij}^{kl} + d_{ij} \cdot CTi \leq 1$$
(33)  
$$t'_{kl} - t'_{ij} - \delta_{ij}^{kl} + d_{ij} \cdot CTi \leq 0$$
(34)

Figure 2: Linear Program Model for the Cyclic Job Shop with Limited WIP and Cycle Time Minimization

maximize this last variable (16). We have to mention that:

$$0 < CTi \le \frac{1}{C_{\max}^*} \tag{35}$$

- $t'_{ij} = \frac{t_{ij}}{CT} \in [0, 1[$ . Indeed,  $0 \le t_{ij} \le CT 1$ , then, after dividing by CT > 0, we will have  $0 \le \frac{t_{ij}}{CT} \le 1 \frac{1}{CT}$ . However,  $\frac{1}{CT} > 0$ , then we have the inequality (17).
- $\alpha'_{ij} = \frac{\alpha_{ij}}{CT} \in \left[0, \frac{1}{C_{\max}^*}\right]$ . Indeed,  $\alpha_{ij}$  is a binary variable, then  $\alpha'_{ij}$  can take only two values: 0 or  $\frac{1}{CT}$ . However,  $\frac{1}{CT} \leq \frac{1}{C_{\max}^*}$ , then the higher bound of  $\alpha'_{ij}$  is  $\frac{1}{C_{\max}^*}$ .
- $\beta'_{ij} = \frac{\beta_{ij}}{CT} \in \left[0, \frac{1}{C_{\max}^*}\right]$ . Similarly to  $\alpha'_{ij}$ , the higher bound  $\beta'_{ij}$  is  $\frac{1}{C_{\max}^*}$ .

- Constraints (20) to (24) and (33), (34) keep the same meaning cited in the previous section. The only difference is that all variables are divided by *CT*.
- Note that, for the linear program we have added more constraints (from (25) to (31)) compared to the non-linear one. These additional constraints are used to guarantee that:

If 
$$\alpha'_{ij}$$
 is not null, then  $\alpha'_{ij} = \frac{1}{CT}$  (36)  
If  $\beta'_{ij}$  is not null, then  $\beta'_{ij} = \frac{1}{CT}$  (37)

Indeed, without the added constraints,  $\alpha'_{ij}$  and  $\beta'_{ij}$  could be non null and their values are not equal to  $\frac{1}{CT} = CTi$ . That's why, we have to take into account the two conditions cited below (36) and (37).

In fact, when we have to test if  $\alpha'_{ij}$  is not null, then this means that we must verify if  $(\alpha'_{ij} > 0)$  or  $(\alpha'_{ij} < 0)$ . However, since we have  $\alpha'_{ij} \in \left[0, \frac{1}{C_{\max}^*}\right]$ , then, the case  $(\alpha'_{ij} < 0)$  will not be taking into account as a valid constraint. Thus, in our case,  $(\alpha'_{ij} \neq 0)$  is equivalent to  $(\alpha'_{ij} > 0)$ . In addition, the statement  $(\alpha'_{ij} = CTi)$  means that  $(\alpha'_{ij} \geq CTi)$  and  $(\alpha'_{ij} \leq CTi)$ . This last statement is always since true since  $\alpha'_{ij} \in \left[0, \frac{1}{C_{\max}^*}\right]$ . That's why we have added this property to the model through constraints (25). This entails that,  $(\alpha'_{ij} = CTi)$  is equivalent, in our case, to  $(\alpha'_{ij} \geq CTi)$ . Finally, the condition cited in (36) is equivalent to:

If 
$$(\alpha'_{ij} > 0)$$
 then  $(\alpha'_{ij} - \frac{1}{CT}) \ge 0$  (38)

Note that we have used in (38) the "*lf/then*" constraint: *If A then B*. This last constraint is equivalent to the logical statement ( $\sim A \cup B$ ). The ( $\sim A$ ) represents the negation of *A*.

In the context of linear programming, a constraint can be viewed as a statement. Specifically, constraints  $(\alpha'_{ij} > 0)$  and  $(CTi - \alpha'_{ij} \le 0)$  are viewed as statements A and B, respectively.

The negation of  $(\alpha'_{ij} > 0)$  is  $(\alpha'_{ij} \le 0)$ . Therefore, (38) is equivalent to:

Either 
$$(\alpha'_{ij} \leq 0)$$
 or  $(CTi - \alpha'_{ij} \leq 0)$  (39)

*Either/or* constraints can be replaced by two simultaneous constraints (Bajalinov (2003), Chen (2010)):

$$\alpha'_{ij} \leq C \cdot y_{ij} \tag{40}$$

$$CTi - \alpha'_{ij} \leq C \cdot (1 - y_{ij})$$
 (41)

 $y_{ij}$  is a binary variable and C is a very big number such that  $C \ge max(\alpha'_{ij}, CTi - \alpha'_{ij})$ . In fact, if  $y_{ij} = 0$ , then  $\alpha'_{ij} = 0$ . Otherwise, if  $y_{ij} = 1$ , then  $\alpha'_{ij} = CTi$ .

As a conclusion, through inequalities (25), (27) and (28), we have used new binary variables to ensure that:

If 
$$\alpha'_{ij} \neq 0$$
, then  $\alpha'_{ij} = CTi$  (42)

• Similarly, through inequalities (26), (29) and (30),we have used new binary variables  $z_{ij}$  to ensure that:

If 
$$\beta'_{ij} \neq 0$$
, then  $\beta'_{ij} = CTi$  (43)

In the next section, we will use an illustrative example and we will resolve it using our approach.

## 3. ILLUSTRATIVE EXAMPLE

In this section we will assess the relevance of our approach using an illustrative example cited in Seo (2002). We will resolve this example by using the linear program proposed in the previous section.

We will use the Petri net formalism Murata (1989) in order to represent the illustrative example. This formalism allows us to model, in the same time, the precedence constraints between operations, the resource needed to perform these operations and the limited level of WIP.

The illustrative example ("Figure.3") consists of three machining centers  $m_1$ ,  $m_2$ ,  $m_3$  and three set-up stations  $s_1$ ,  $s_2$ ,  $s_3$  (that can be also considered as machines). Three types of parts are produced: A, B and C.

Seo (2002) considers that  $o_{1,1}$  and  $o_{2,1}$  have an overtaking degree equals to 1. This means that he fixes the WIP of the system to:

- 2 WIP for the job A.
- 2 WIP for the job B.
- 1 WIP for the job C.

As a result, the production system must works with a limited WIP equals to 5. Under this fixed level of WIP, Seo founds a cyclic schedule with a cycle time equals to 16 t.u. With our approach, we begin by fixing the value of the different constants (M,  $WIP_{opt}$  and,  $C^*_{max}$ . As explained in the previous section (using the condition 14) we have to compare M and  $WIP_{opt}$  to have an idea about the value of CT. However,  $WIP_{opt} = 6$  (computed using the mathematical model proposed in Bourdeaud'huy (2006)), M = 5 and  $C^*_{max} = 14$  (the workload of the



Figure 3: Illustrative example Seo (2002)

critical machine wich is  $m_3$ ).

Since  $M > WIP_{opt}$  then  $CT > C_{max}^* = 14$ . Thus, with a WIP fixed to 5, the cycle time must be greater than 14. In order to resolve this example with our approach, we have, first, modeled the example as a mathematical linear program. This latter has been solved by CPLEX solver (version 9.0, ILOG (2003)) on an *Intel Pentium 4* at *2.8 GHz* and *1Go RAM*, under *Windows XP*. The computed schedule is presented in ("Figure.4"). In this illustration, there is two Gantt diagrams, the first one



Figure 4: Scheduling of the illustrative example

Note that the cycle time found by our approach is equal to the computed by *Seo* (which is 16 t.u.). This value is optimal if we take into account that the WIP

is fixed to 5. Indeed, under this fixed level of WIP, the job C have to be processed with only one WIP. As a consequence, the cycle time will be fixed by this job.

If we consider, now, that the WIP level will be fixed at 3 (one WIP for each each job). In this case, the cycle time will be fixed by the higher processing time among the jobs, which is in our case equals to 22 t.u.).

Note that we have resolved also this case (M = 3) with our linear program and the corresponding cycle time is equal to 22 t.u., which is also the same value found by *Seo* approach's when he resolves the same example without allowing any overtaking for operations.

# 4. CONCLUSION

This paper deals with cyclic scheduling problem with limited WIP and Cycle Time minimization. The main contribution here is to propose a mathematical model for such production environnement. First, we have presented systems with limited WIP and we have discussed and justified the use of this production constraint. Secondly, we have proposed a non linear mathematical program which model systems with limited WIP and cycle time minimization. Afterwards, we have presented the method used to linearize this non-linear program. The last part was reserved to validate our approach by comparing it with an illustrative example from the literature.

This study show that, for a production with a fixed level of WIP, one can find the optimal cycle time. In addition, we have shown that we can found the same results as those found by Seo (2002). Further works will consider another strategy of pallets allocations. In fact, we suppose, actually that pallets are used only by one job and are not allowed to be reused by other jobs within the same cycle time. This last constraint will be relaxed in the upcoming works by allowing pallets to be used if they are available. Moreover, we aim to substitute the CPLEX solver by an algorithm especially fitted to the mathematical model, in order to improve the resolution time.

# 5. REFERENCES

Bajalinov E., (2003), *Linear-fractional programming: theory, methods, applications and software*, "Kluwer Academic, Dordrechts, the Netherlands", 423 pages. Bellmann R., (1957), *Dynamic Programming*, "Princeton University Press", Princeton, New Jersey. Ben Amar M.A., Bourdeaud'huy T. and Korbaa O., (2007), *Cyclic Scheduling MIP Implementation: Cutting Tetchniques*, "International Conference on Production Research ICPR'07", Valparaiso, Chili, 29 July - 2 August.

Ben Amar M.A., Camus H. and Korbaa O., (2009), *A Mathematical Model for Cyclic Scheduling With Assembly Tasks and Work-In-Process Minimization*, "IEEE International Symposium on Assembly and Manufacturing, ISAM 2009", Suwon, Korea, November 17 - 20, pp. 358-363.

Bourdeaud'huy T. and Korbaa O., (2006), *A Mathematical Model For Cyclic Scheduling With Work-In-Progress Minimization*, "IFAC Symposium on Information Control Problems in Manufacturing, INCOM'06", Saint Etienne, France, 17-19 May.

Che A., Kats V., Levner E., (2010), *Cyclic Scheduling in Robotic Flowshops with Bounded Work-in-Process Levels*, "Naval Research Logistics", John Wiley & Sons, Ltd. [Online], Available: *http://onlinelibrary.wiley.com/doi/10.1002/nav.20435/abstract* 

Chen D.-S., Batson R.-G., Dang Y., (2010), *Applied Integer Programming: Modeling and Solution*, "John Wiley and Sons Ltd", 468 pages.

ILOG Inc., ILOG CPLEX 9.0, (2003), User's manual, octobre 2003.

Hall R., (1981), *Driving the Productivity Machine: Production Planning and Control in Japan*, "American Production and Inventory Control Society", Falls Church, Virginia.

Hillion H.P., Proth J.M., (1989), *Performance evaluation of job-shop systems using timed event-graphs*, "IEEE Trans. on Automatic Control", Vol. 34, n. 1, pp. 3-9.

Hopp W. J., Spearman M. L., (1996), *Factory physics foundations of manufacturing management*, "Boston: Irwin", 668 pages.

Seo J.-W., Lee T.-E., (2002), *Steady-state analysis and scheduling of cycle job shops with overtaking*, "The International Journal of Flexible Manufacturing Systems", Vol. 14, pp. 291-318.

Manier M.A., Bloch C., (2003), *A classification for Hoist Scheduling Problems*, "International Journal of Flexible Manufacturing Systems", 15, Vol. 1, pp. 37-55.

Monden Y., (1983), *Toyota Production System: Practical Approach to Production Management.* "Industrial Engineering and Management Press", Norcross, Georgia.

Murata T., (1989) *Petri nets: properties, analysis and application*, "IEEE Conference", vol. 77, n. 4, pp. 541-580.

Ramamoorthy C.V., Ho G.S., (1980), *Performance evaluation of asynchronous concurrent systems using Petri nets*, "IEEE Transactions on Software Engineering4, Vol.SE-6, n. 5, pp. 440-449. Serafini P. and Ukovich W., (1989). *A mathematical model for periodic scheduling problems*, "SIAM Journal Discrete Mathematics", Vol. 2, n.4, pp. 550-581.

Shingo S., (1981), *Study of Toyota Production System from Industrial Engineering View-Point*, "Japan Management Association", Business & Economics, 257 pages.