Some Precedence Relations in Single Machine Sequencing

Kristina Šorić

Faculty of Economics Zagreb, Zagreb, Croatia

A model for real-time control of flexible manufacturing systems is considered. In this model, a machine can process a finite number of part types at specified rates, but only one part type can be processed at any given time. Each switch from one type to another requires different setup times.

A better upper bound on the total work backlog than those available in literature is derived by introducing the *clear-the-largest-work-after- setup (CLWS)* heuristic policy which stabilizes the system in the sense that, in the long run, the required demand is met.

Studying whether the CLWS scheduling policy induces convergence to a stationary state, some precedence relations between part types are reported with the computational results.

Keywords: single machine sequencing, CLWS heuristic, upper bound, precedence relations

1. Introduction

In a paper [6], Perkins and Kumar (1989) proposed a model for a flexible manufacturing system and made a number of important observations about the stability and performance of feedback scheduling policies implementable in a real time. The problem is to stabilize the system in the sense that, in the long run, the required demand is met. This is equivalent to saying that the total work backlog in the system remains bounded. They also derived a finite upper bound on the buffer level for given scheduling policy and a lower bound on the average buffer levels for any s cheduling polic y.

After that, Lou, Sethi and Sorger (1991) showed how these bounds can be improved by taking into account the dynamic evolution of the system, when controlled by a feedback policy.

In this paper a better upper bound on the total work backlog than those available in literature is derived by introducing the *clear-the-largest-work-after-setup (CLWS)* heuristic policy which stabilizes the system in the sense that, in the long run, the required demand is met.

The paper is organized as follows.The next section introduces the problem (called PKM) and defines the CLWS heuristic policy. The subsequent sections derive an upper bound on the total work backlog, compare it to the upper bound from [4] and give some precedence relations between part types. These relations could be useful in studying whether the CLWS scheduling policy induces convergence to a stationary state. The final section supposes the finite horizont and reports some computational results indicating that the stationary state exists.

2. The PKM Problem

This model assumes a single machine processing different part types at specified rates, but only one part type can be processed at any given time. Each switch from one type to another requires different setup times and once a part type has been chosen for the production, it has to be produced until the moment at which its buffer level hits zero.

In the same way as in [6], we suppose that a machine processes I part types. Each part of type i needs to be produced at a prespecified demand rate d_i, $i = 1 \dots I$. The machine can process only one part at a time. Let τ_i, $i = 1 \dots I$, be the time required to produce one unit of part i and

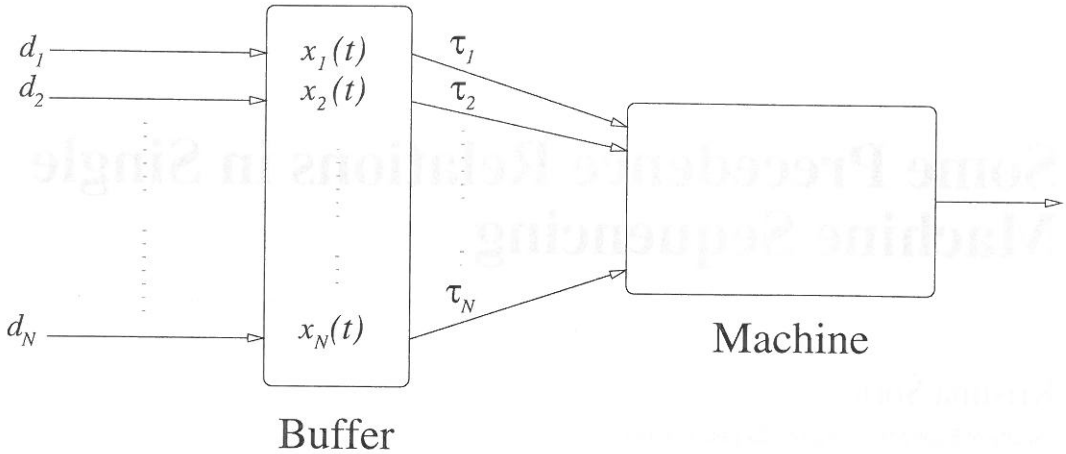

Fig. 1. The flexible single machine manufacturing system.

let us suppose that a setup from any part type to part type i, requires δ_i, $i = 1 \ldots I$, units of time.

It is assumed that there is an unlimited buffer in front of the machine (Figure 1).

Let $x_i(t)$ denotes the *buffer level* of part type i at time t. We define the $w_i(t) = x_i(t)\tau_i$ and $\rho_i = d_i\tau_i$ as the *work backlog* of part type i at time t and the *machine load or input* due to part type i, respectively.

We can then define

$$X(t) = \sum_{i=1}^{I} x_i(t)$$

as the *total buffer level* at time t,

$$W(t) = \sum_{i=1}^{I} w_i(t)$$

as the *total work backlog* at time t and

$$\rho = \sum_{i=1}^{I} \rho_i$$

as the *total machine load*.

We could also say that ρ is the average proportion of time the machine spends in production.

In the same way as in [6], we say that the system is stable if

$$\sup_{0 \le t < \infty} W(t) \le M < \infty$$

holds for some constant M.

Clearly, the necessary condition for stability of any policy is

$$\rho < 1$$

what we assume throughout this note.

Remark 1. *Note that $\rho_i = d_i\tau_i$, that is, this is the time needed for the production of the quantity of part type i arrived into the buffer in one unit of time. So, the condition $\rho < 1$ means that we are able to produce faster than the part types arrive and in this way satisfy the demand. Otherwise, the buffer level would increase without any limits and we would not be able to keep the system stable.*

The problem is to find a policy which gives an upper bound on the buffer level as small as possible. This policy should at a certain moment T_n choose a part type for the production. When a part type, say i, has been chosen, the machine actually begins its production at time $T_n + \delta_i$ and continues until the moment T_{n+1} at which the buffer level $x_i(t)$ hits zero (Figure 2).Because of this property, the considered policy will be called *clearing policy*.

From [7], we have the following definition.

Definition 1. *A scheduling policy is called a clear-the-largest-work-after-setup (CLWS) policy if it is a clearing policy and if, at time T_n, the machine chooses for the next production run a part type i_n which satisfies*

$$w_{i_n}(T_n + \delta_{i_n}) \ge w_j(T_n + \delta_{i_n}) \quad \forall j = 1 \ldots I,\ j \ne i_n \quad (1)$$

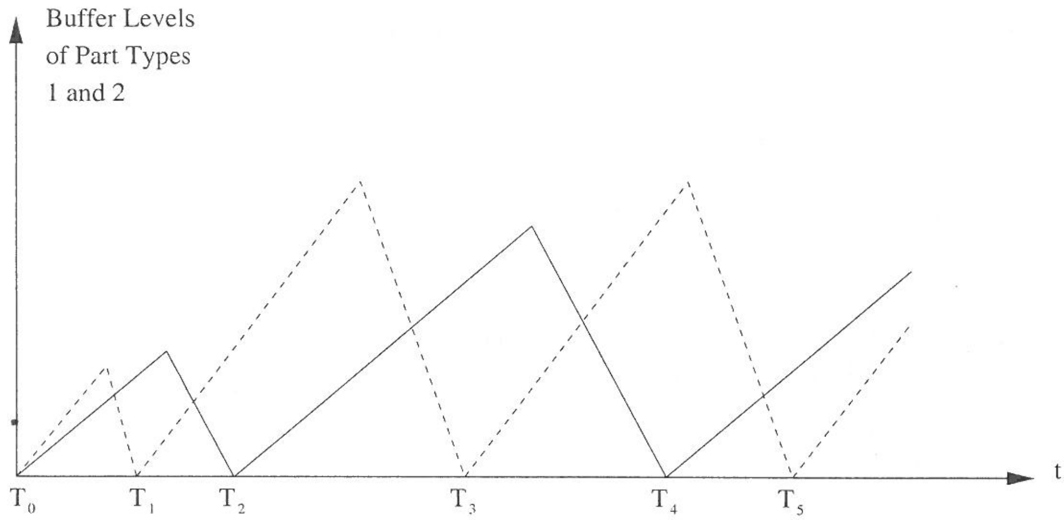

Fig. 2. Performances of buffer levels.

Thus, the *CLWS* policy, at a moment T_n, chooses the part type i_n which would have the largest work backlog after the setup time.

Obviously, *CLWS* clearing policy generates a sequence of setups and production runs (Figure 2).

Since, during the setup, the machine does not produce anything, we have that $w_j(T_n + \delta_{i_n}) = w_j(T_n) + \rho_j\delta_{i_n}$, $\forall j = 1 \dots I$ and summing the inequalities (1) for all $j = 1 \dots I$, we obtain

$$w_{i_n}(T_n) + \delta_{i_n}\rho_{i_n} \geq \frac{1}{I}\sum_{j=1}^{I} w_j(T_n) + \frac{1}{I}\delta_{i_n}\sum_{j=1}^{I}\rho_j$$

that is,

$$w_{i_n}(T_n) + \delta_{i_n}\rho_{i_n} \geq \frac{1}{I}W(T_n) + \frac{1}{I}\delta_{i_n}\rho \qquad (2)$$

3. Upper Bound on the Total Work Backlog

Once the part type i_n has been chosen, its production run can begin at the time $T_n + \delta_{i_n}$ and continue until the time T_{n+1} at which the buffer level $x_{i_n}(t)$ hits zero.

At the time T_{n+1}, as in [4] and [6], we have

$$x_{i_n}(T_{n+1}) = x_{i_n}(T_n) + d_{i_n}\delta_{i_n} + (d_{i_n} - \frac{1}{\tau_{i_n}})(T_{n+1} - T_n - \delta_{i_n})$$

and, clearly, multiplying by τ_{i_n},

$$w_{i_n}(T_{n+1}) = w_{i_n}(T_n) + \rho_{i_n}\delta_{i_n} + (\rho_{i_n} - 1)(T_{n+1} - T_n - \delta_{i_n})$$

Since, $x_{i_n}(T_{n+1}) = 0$, it has to be $w_{i_n}(T_{n+1}) = 0$ and we obtain the production interval

$$T_{n+1} - T_n = \frac{w_{i_n}(T_n) + \delta_{i_n}}{1 - \rho_{i_n}} \qquad (3)$$

For the part types that were not chosen for the production run $[T_n, T_{n+1}]$, it is

$$w_j(T_{n+1}) = w_j(T_n) + \rho_j(T_{n+1} - T_n), \quad j = 1 \dots I, \ j \neq i_n$$

Thus, the total work backlog at the moment T_{n+1} is

$$\begin{aligned} W(T_{n+1}) &= \sum_{j\neq i_n} w_j(T_{n+1}) + w_{i_n}(T_{n+1}) \\ &= \sum_{j\neq i_n} w_j(T_n) + \sum_{j\neq i_n}\rho_j(T_{n+1} - T_n) \\ &\quad + w_{i_n}(T_n) + \rho_{i_n}\delta_{i_n} + \rho_{i_n}(T_{n+1} - T_n) \\ &\quad - \rho_{i_n}\delta_{i_n} - (T_{n+1} - T_n) + \delta_{i_n} \\ &= W(T_n) - (1-\rho)(T_{n+1} - T_n) + \delta_{i_n} \end{aligned}$$

Using (3), we obtain

$$W(T_{n+1}) = W(T_n) - w_{i_n}(T_n)\frac{1-\rho}{1-\rho_{i_n}} + \delta_{i_n}\frac{\rho-\rho_{i_n}}{1-\rho_{i_n}}$$

From (2), we have that

$$-w_{i_n}(T_n) \leq -\frac{1}{I}W(T_n) + \delta_{i_n}\rho_{i_n} - \frac{1}{I}\delta_{i_n}\rho$$

So, it follows

$$W(T_{n+1}) \leq W(T_n) - \frac{1}{I}\frac{1-\rho}{1-\rho_{i_n}}W(T_n) + \delta_{i_n}\frac{1-\rho}{1-\rho_{i_n}}(\rho_{i_n} - \frac{1}{I}\rho) + \delta_{i_n}\frac{\rho-\rho_{i_n}}{1-\rho_{i_n}}$$

that is

$$W(T_{n+1}) \leq W(T_n)[1-\frac{1}{I}\frac{1-\rho}{1-\rho_{i_n}}] - \frac{1}{I}\frac{1-\rho}{1-\rho_{i_n}}\delta_{i_n}\rho + \frac{\delta_{i_n}}{1-\rho_{i_n}}(\rho-\rho_{i_n}+\rho_{i_n}-\rho\rho_{i_n})$$

This implies

$$W(T_{n+1}) \leq W(T_n)[1 - \frac{1}{I}\frac{1-\rho}{1-\rho_{i_n}}] + \rho\delta_{i_n} - \frac{1}{I}\rho\delta_{i_n}\frac{1-\rho}{1-\rho_{i_n}}$$

and since $1 - \frac{1}{I}\frac{1-\rho}{1-\rho_{i_n}} > 0$

$$W(T_{n+1}) \leq W(T_n)[1 - \frac{1}{I}\frac{1-\rho}{1-\rho_m}] + \rho\delta_M(1 - \frac{1}{I}\frac{1-\rho}{1-\rho_m})$$

where $\rho_m = min_{i=1...I}\rho_i$ and $\delta_M = max_{i=1...I}\delta_i$.

The theory of difference equations has proposed as follows (see Lou, Sethi and Sorger, [4]).

Proposition 1. *Let (W_n) be a sequence of real numbers satisfying*

$$A^- W_n + B^- \leq W_{n+1} \leq A^+ W_n + B^+$$

with $0 \leq A^+ < 1$, $0 \leq A^- < 1$ and $\Delta W_n = W_{n+1} - W_n$. Then,

$$W_n \leq \frac{B^-}{1-A^-} \Rightarrow \Delta W_n \geq 0$$
$$W_n \geq \frac{B^+}{1-A^+} \Rightarrow \Delta W_n \leq 0$$

and

$$\frac{B^-}{1-A^-} \leq \liminf_{n\to\infty} W_n \leq \limsup_{n\to\infty} W_n \leq \frac{B^+}{1-A^+}$$

As in [4], we have the following lemma.

Lemma 1. *For the CLWS policy, it holds*

$$\limsup_{n\to\infty} W(T_n) \leq K - \rho\delta_M$$

where

$$K = \frac{I\rho\delta_M(1-\rho_m)}{1-\rho}$$

Proof. From the previous proposition and from the fact that

$$0 < 1 - \frac{1}{I}\frac{1-\rho}{1-\rho_m} < 1$$

QED.

Now, we can determine the following upper bound on the total work backlog.

Theorem 1. *For the CLWS policy, it holds*

$$\limsup_{t\to\infty} W(t) \leq K$$

Proof. We know that for $t \in [T_n, T_{n+1}]$, for every n, it holds that

$$\begin{aligned} W(t) &\leq W(T_n + \delta_M) \\ &= W(T_n) + \rho\delta_M \end{aligned}$$

From this fact and from the previous lemma, we obtain the proof.

QED.

Remark 2. *Observe that we considered* $\limsup_{t\to\infty}$ *$W(t)$. This means that at the beginning it may happen that the level of the total workbacklog is higher than K. But, once the level of workbacklog falls bellow the upper bound K, it will not increase any more.*

Remark 3. *Let us analyse the upper bound K a little bit. If $\delta_M = 0$, it follows that $\delta_i = 0$, $\forall i = 1 \ldots I$, that is, there is no setup time between any two part types. And, in this case, the upper bound K becomes equal to zero. This means that without the setup time, it is enough to keep the machine busy, that is, to produce as the part types come into the buffer. Also, we can notice that, if the total machine load ρ is great enough, that is, almost equal to 1, then the value $1-\rho$ is almost equal to 0 and the upper bound K becomes very high.*

4. Comparison to the SW Upper Bound

In the work [4], assuming the fixed setup time, Lou, Sethi and Sorger used the *clear-the-largest-work (CLW)* policy which, at a certain moment T_n, chooses a part type i_n such that

$$w_{i_n}(T_n) \geq \frac{1}{I} W(T_n)$$

With the assumption of different setup times, it holds that

$$\limsup_{t \to \infty} W(t) \leq SW$$

where

$$SW = \frac{I\delta_M(\rho - \rho_m)}{1-\rho} + \delta_M \rho$$

Comparing these two upper bounds, we get the following theorem.

Theorem 2. *The upper bound SW is greater than the upper bound K if and only if there is at least one input different from the others.*

Proof. Let us consider the difference

$$\begin{aligned} SW-K &= \frac{I\delta_M(\rho-\rho_m)}{1-\rho} + \delta_M\rho - \frac{I\rho\delta_M(1-\rho_m)}{1-\rho} \\ &= \frac{I\delta_M}{1-\rho}[\rho-\rho_m+\frac{(1-\rho)\rho}{I}-\rho(1-\rho_m)] \\ &= \frac{I\delta_M}{1-\rho}(1-\rho)(\frac{\rho}{I}-\rho_m) \\ &= \delta_M(\rho-I\rho_m) \\ &= \delta_M(\sum_{j=1}^{I}\rho_j-\sum_{j=1}^{I}\rho_m) \\ &= \delta_M\sum_{j=1}^{I}(\rho_j-\rho_m) \end{aligned}$$

Thus, $SW - K > 0$ if and only if there is at least one input ρ_i different from the others.

QED.

Remark 4. *Note that if $\rho_i = \rho_m$, $\forall i = 1 \ldots I$, the upper bounds SW and K are equal. Consequently, the policies CLW and CLWS have the same rules for choosing the part types for the production runs because, assuming that $\rho_i = \rho_m$, $\forall i = 1 \ldots I$, the part type which has the largest workbacklog in a certain moment T_n, will also have the largest workbacklog at the moment $T_n + \delta_i$, for any $i = 1 \ldots I$.*

The next section gives some precedence relations between part types, that could be useful in searching for the way of obtaining convergence to a stationary state (that is, to a solution consisting of cycles of identical lengths)

5. Some Precedence Relations

From Figure 2, we can observe that there is a certain cyclic behavior of the system we are considering. So, we can set our problem in the framework of the sequence dependent cyclic lot scheduling problems (see, for example Dobson [1]).

Dobson's goal is to minimize the average holding cost (or workbacklog) over all possible cycle lengths and over all frequences of production (that is, over all numbers of runs per cycle) for each part type. The problem was to compute the production frequencies that were used then as the dates in minimizing the average holding costs, but now, they are used with all sequences of production having the given frequences.

In [8] the CLWS heuristic is compared to the algorithm branch and bound. The reported computational results show that the CLWS heuristic policy is very effective, giving an optimality gap that is, the difference between the optimal value of the objective function and the value obtained by the heuristic, of approximately 2.5-3.0 %.

Using the results from [8], we suggest that one way of computing production frequencies in sequence dependent cyclic lot scheduling problems could be by using the CLWS policy. So, it is interesting to study the cycles, their lengths and the number of runs of certain part type per one cycle (that is, their frequences per cycle).

In this section, we will see how a sequence and frequencies of setups and production runs depend on inputs and setup times.

Let us consider a sequence $S_i = (i, \ldots i)$ generated by the *CLWS* policy which contains two runs of part type i (one at the beginning and one at the end of the sequence), say, $[T_n, T_{n+1}]$ and $[T_k, T_{k+1}]$ for $k \geq n+1$. From this follows that $w_i(T_{n+1}) = w_i(T_{k+1}) = 0$.

Observe that for *CLWS* policy it may not be true that two consecutive runs do not process the same part type. According to the *CLWS* policy it can happen that, in order to keep the upper bound as small as possible, it is better to process a part type, say j, for a time interval during which $x_j(t)$ is equal to zero, rather than switch up to another part type.

Let us now make these observation more concise and state the following theorems.

Theorem 3. *If $\rho_i \leq \rho_j$, for some $j \neq i$, there is at least one run for part type j in the sequence S_i.*

Proof. During the period when the part type j is not processed, for some moment t, $T_{n+1} \leq t < T_k$, it is

$$\begin{aligned} w_j(t) &= w_j(T_{n+1}) + \rho_j(t - T_{n+1}) \\ &> \rho_j(t - T_{n+1}) \\ &\geq \rho_i(t - T_{n+1}) \\ &= w_i(t) \end{aligned}$$

Since $\delta_j\rho_i \leq \delta_j\rho_j$, we have

$$w_j(t) + \delta_j\rho_j > w_i(t) + \delta_j\rho_i$$

that is,

$$w_j(t + \delta_j) > w_i(t + \delta_j)$$

Thus, all the part types j such that $\rho_j \geq \rho_i$, have to be processed at least one time before the moment T_k.

QED.

Corollary 1. *If $\rho_j = \rho_i$, for some $j \neq i$, there is only one run of part type j in the sequence S_i.*

Proof. From the previous theorem, we know that there is at least one run of part type j in the sequence S_i. Let T_l, $T_{n+1} \leq T_l < T_k$, be the moment when the part type j was chosen for the production for the first time and $[T_l, T_{l+1}]$, the interval of its production. By the *CLWS* policy, at the moment T_{l+1}, we have $w_j(T_{l+1}) = 0$ and $w_i(T_{l+1}) = \rho_i(T_{l+1} - T_{n+1})$. Thus, $w_j(T_{l+1}) < w_i(T_{l+1})$. Also, $\forall t,\ T_{l+1} \leq t \leq T_k$, since $\rho_j = \rho_i$, it is $\rho_j(t - T_{l+1}) = \rho_i(t - T_{l+1})$. Combining these, we obtain

$$w_j(T_{l+1}) + \rho_j(t - T_{l+1}) < w_i(T_{l+1}) + \rho_i(t - T_{l+1}),$$

$$\forall t,\ T_{l+1} \leq t \leq T_k$$

and since $\rho_i\delta_i = \rho_j\delta_i$, it follows

$$\begin{aligned} w_j(T_{l+1}) + \rho_j(t - T_{l+1}) + \rho_j\delta_i < w_i(T_{l+1}) \\ + \rho_i(t - T_{l+1}) + \rho_i\delta_i,\ \forall t,\ T_{l+1} \leq t \leq T_k \end{aligned}$$

that is,

$$w_j(t + \delta_i) < w_i(t + \delta_i),\quad \forall t,\ T_{l+1} \leq t \leq T_k$$

Thus, by the *CLWS* policy, in the sequence S_i the part type j will not be produced any more after the moment T_{l+1}.

QED.

Theorem 4. *For all $j \neq i$, such that $\rho_j \leq \rho_i$, there is not more than one run of part type j in the sequence S_i.*

Proof. Let us suppose the contrary, that is, that there are at least two runs of part type j in the sequence S_i.

Let $[T_l, T_{l+1}]$ and $[T_m, T_{m+1}]$, $(T_{n+1} \leq T_l < T_{l+1} < T_m < T_{m+1} \leq T_k)$, be two consecutive intervals of the production of part type j in the sequence S_i. By the *CLWS* strategy, it follows that

$$w_j(T_l + \delta_j) \geq w_i(T_l + \delta_j)$$

and

$$w_j(T_m + \delta_j) \geq w_i(T_m + \delta_j) \qquad (4)$$

Also, at the moment T_{l+1}, it is $w_j(T_{l+1}) = 0$ and $w_i(T_{l+1}) = \rho_i(T_{l+1} - T_{n+1})$. Since $\rho_j \leq \rho_i$, it is

$$\begin{aligned} w_j(T_m + \delta_j) &= w_j(T_{l+1}) + \rho_j(T_m - T_{l+1} + \delta_j) \\ &< w_i(T_{l+1}) + \rho_i(T_m - T_{l+1} + \delta_j) \\ &= w_i(T_m + \delta_j) \end{aligned}$$

which is in contradiction with the (4).

QED.

6. Computational Results

Based on the results from the previous sections we implemented the CLWS heuristic policy and assuming the finite horizont, executed it until all the part types were scheduled.

Our computational experiments have objective to estimate empirically whether the CLWS heuristic policy induces convergence to a stationary state that is, to a solution consisting of cycles of identical lengths.

We performed all the computations on a SPARC Server 20 mod. S20 (2 processors SuperSPARC 50 MHz, 1 Mb SuperCache, RAM 128 MB, operational system UNIX SunOS 5.5 (Solaris 2.5). The CLWS heuristic is coded in programming language C.

The problems solved were randomly generated. We specify for each machine load of the problem a range of values ($\rho_i \in [0.01, 0.6]$, $i = 1 \ldots I$) and each machine load was randomly generated within this range. The setup cost was chosen so that the sum of the machine loads is less than 1. We generate set of problems with 2,3 and 4 part types and horizont equal to 50, 100 and 400.

All the problem instances assume that the machine is off at the begining of the production and that there is no setup at the start of the first production run.

Let denote the part types by $J_1, J_2, \ldots, J_I$.

Example 1. I=2, T=50

(a) $\rho_1 = 0.1$, $\rho_2 = 0.2$, $\delta_1 = 0.5$, $\delta_2 = 1.0$

The production cycles are:

$J_2J_1J_2J_2J_1J_2J_2J_1J_2J_2J_1J_2J_2J_1J_2J_2 \ldots$

(b) $\rho_1 = 0.1$, $\rho_2 = 0.2$, $\delta_1 = 0.5$, $\delta_2 = 0.5$

The production cycles are:

$J_2J_1J_2J_1J_2J_1J_2J_1J_2J_1J_2J_1 \ldots$

(c) $\rho_1 = 0.1$, $\rho_2 = 0.2$, $\delta_1 = 1.0$, $\delta_2 = 0.5$

The production cycles are:

$J_2J_1J_2J_1J_2J_1J_2J_1J_2J_1J_2J_1 \ldots$

Example 2. I=3, T=400

(a) $\rho_1 = 0.1$, $\rho_2 = 0.2$,$\rho_3 = 0.6$, $\delta_1 = 0.5$, $\delta_2 = 0.5$, $\delta_3 = 1.0$

The production cycles are:

$J_3J_2J_3J_1J_3J_2J_3J_1J_2J_3J_2J_3J_1J_3J_2J_3J_1J_2J_3$

$J_2J_3J_1J_3J_2J_3J_1J_2J_3J_2J_3J_1J_3J_2J_3J_1J_2 \ldots$

(b) $\rho_1 = 0.1$, $\rho_2 = 0.2$,$\rho_3 = 0.6$, $\delta_1 = 0.5$, $\delta_2 = 0.5$, $\delta_3 = 0.5$

The production cycles are:

$J_3J_2J_3J_1J_3J_2J_3J_1J_2J_3J_2J_3J_1J_3J_2J_3J_1J_2J_3$

$J_2J_3J_1J_3J_2J_3J_1J_2J_3J_2J_3J_1J_3J_2J_3J_1J_2 \ldots$

(c) $\rho_1 = 0.1$, $\rho_2 = 0.2$,$\rho_3 = 0.6$, $\delta_1 = 1.0$, $\delta_2 = 0.5$, $\delta_3 = 0.5$

The production cycles are:

$J_3J_2J_3J_1J_3J_2J_3J_1J_3J_2J_3J_1J_2J_3J_2J_3J_1J_3J_2J_3$

$J_1J_2J_3J_2J_3J_1J_3J_2J_3J_1J_2J_3J_2J_3J_1J_3J_2J_3J_1J_2 \ldots$

(d) $\rho_1 = 0.1$, $\rho_2 = 0.2$,$\rho_3 = 0.6$, $\delta_1 = 1.0$, $\delta_2 = 1.0$, $\delta_3 = 0.5$

The production cycles are:

$J_3J_2J_3J_1J_3J_2J_3J_1J_2J_3J_2J_3J_1J_3J_2J_3J_1J_2J_3J_2$

$J_3J_1J_3J_2J_3J_1J_2J_3J_2J_3J_1J_3J_2J_3J_1J_2 \ldots$

Example 3. I=5, T=100

(a) $\rho_1 = 0.05$, $\rho_2 = 0.2$,$\rho_3 = 0.2$, $\rho_4 = 0.25$, $\rho_5 = 0.3$, $\delta_1 = 0.5$, $\delta_2 = 0.5$, $\delta_3 = 1.0$,$\delta_4 = 1.3$, $\delta_5 = 1.5$

The production cycles are:

$J_5J_4J_3J_5J_2J_4J_5J_3J_4J_2J_5J_3J_4J_5J_2J_1J_3J_4J_5J_2$

$J_3J_4J_5J_2J_3J_4J_5J_2J_3J_4J_5J_1J_2J_3J_4J_5J_2J_3J_4J_5$

$J_2J_3J_4J_5J_2J_3J_4J_5J_1J_2J_3J_4 \ldots$

(b) $\rho_1 = 0.05$, $\rho_2 = 0.2$,$\rho_3 = 0.2$, $\rho_4 = 0.25$, $\rho_5 = 0.3$, $\delta_1 = 0.5$, $\delta_2 = 0.5$, $\delta_3 = 0.5$,$\delta_4 = 0.5$, $\delta_5 = 0.5$

The production cycles are:

$J_5J_4J_2J_3J_5J_4J_2J_5J_3J_4J_5J_2J_3J_4J_5J_1J_2J_3J_4J_5$

$J_2J_3J_4J_5J_2J_3J_4J_5J_2J_3J_4J_5J_1J_2J_3J_4J_5J_2J_3J_4$

$J_5J_2J_3J_4J_5J_2J_3J_4J_5J_1J_2J_3J_4 \ldots$

Considering these results, we can observe that after some production runs at the begining, the part types are produced with a certain regularity depending on the date (the machine loads and setup times). In *Example 1* (a) and (b), we can notice that the regularity changed when we decreased the value of the setup δ_2, but in the majority of examples this regularity did not even depend on the values of setup times. Yet, it surely depended on the values of machine loads, in the sense of the theorems from the previous section.

So, the results verify these theorems and make them useful in resolving the sequence dependent cyclic lot scheduling problem, because we can assume a rotation schedule and use the production frequences obtained from the CLWS policy as the dates in its modelling (for example, in the Dobson's model from [1]).

7. Conclusions

The objective of this work was to study the single machine sequencing problem PKM proposed by Perkins and Kumar [6]. In this paper we proposed on new heuristic strategy (called CLWS) giving better upper bound on the work backlog than those available in the literature.

Introducing different setup times for different part types we gave some precedence relations between the part types in order to investigate whereas the stationary state exists. The computational results reported in the last section indicate this possibility.

For the future research it would be very interesting to examine this problem in the context of periodic scheduling problems.

Acknowledgement

I am very grateful to the Department of Mathematics and Computer Sciences, University of Udine, Italy for the hospitality and the facilities it has provided to me to complete this paper.

References

[1] G. DOBSON, 'The Cyclic Lot Scheduling Problem with Sequence Dependent Setups', *Operations Research*, Vol. 40, No. 4, pp. 736–749, 1992.

[2] G. DOBSON, C.A. YANO, 'Cyclic Scheduling to Minimize Inventory in a Batch Flow Line', *European Journal of Operational Research*, Vol. 75, pp. 441–461, 1994.

[3] MARTIN E. DYER, LAURENCE A. WOLSEY, 'Formulating the Single Machine Sequencing Problem with Release Dates as a Mixed Integer Program', *Discrete Applied Mathematics* 26 (1990), p. 255–270

[4] S. LOU, S. SETHI AND G. SORGER, 'Analysis of a Class of Real-Time Multiproduct Lot Scheduling Policies', *IEEE Transactions on Automatic Control*, vol. 36, p. 243–248, 1991.

[5] GEORGE L. NEMHAUSER, LAURENCE A. WOLSEY, 'Integer and Combinatorial Optimisation', John Wiley & Sons, 1988.

[6] J.R. PERKINS AND P.R. KUMAR, 'Stable Distributed, Real-Time Scheduling of Flexible Manufacturing/ Assembly/ Disassembly Systems', *IEEE Transaction on Automatic Control*, vol34, p. 139–148, 1989.

[7] KRISTINA ŠORIĆ, 'Analysis of Buffer Level', *Proceedings of 4th Conference of Operational Research*, Rab, Croatia, October 4–6, 1994, p.p. 244–251.

[8] KRISTINA ŠORIĆ, 'The CLWS Heuristic for Single Machine Sequencing Problem' (*submited*), 1997.

Received: April, 1997
Accepted: December, 1997

Contact address:

Kristina Šorić
Faculty of Economics Zagreb
Kennedyev trg 6
10 000 Zagreb
Croatia
e-mail: ksoric@oliver.efzg.hr

KRISTINA ŠORIĆ born on December 13, 1965, is a PhD student of computational mathematics at the Department of Pure and Applied Mathematics at the University of Padua, Italy. She also works as an assistant in mathematics at the Faculty of Economics, University of Zagreb. Her research interests belong to combinatorial optimization and machine scheduling problems.