On-line Part Scheduling in a Surface Treatment System

Fabrice Chauvet, Eugene Levner, Leonid K. Meyzin, Jean-Marie Proth

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

On-line Part Scheduling in a Surface Treatment System

Fabrice Chauvet - Eugene Levner
Leonid K. Meyzin - Jean-Marie Proth

N° 3318
Décembre 1997

THÈME 4

Rapport de recherche

Les rapports de recherche de l'INRIA
sont disponibles en format postscript sous
ftp.inria.fr (192.93.2.54)

si vous n'avez pas d'accès ftp
la forme papier peut être commandée par mail :
e-mail : dif.gesdif@inria.fr
(n'oubliez pas de mentionner votre adresse postale).

par courrier :
Centre de Diffusion
INRIA
BP 105 - 78153 Le Chesnay Cedex (FRANCE)

INRIA research reports
are available in postscript format
ftp.inria.fr (192.93.2.54)

if you haven't access by ftp
we recommend ordering them by e-mail :
e-mail : dif.gesdif@inria.fr
(don't forget to mention your postal address).

by mail :
Centre de Diffusion
INRIA
BP 105 - 78153 Le Chesnay Cedex (FRANCE)

INRIA LORRAINE

On-line Part Scheduling in a Surface Treatment System

Fabrice CHAUVET*, Eugene LEVNER**, Leonid K. MEYZIN**
and Jean-Marie PROTH* and ***

Rapport de recherche n° 3318 — Décembre 1997 — 13 pages

ABSTRACT

A real-time scheduling algorithm which guarantees an optimal makespan to each part which arrives in a line of chemical baths for surface treatment purpose is proposed. We first consider the case when the treatment periods are much greater than the transportation times, which allows us to neglect these times. We then extend our approach to the case when transportation times cannot be neglected. Some numerical examples are provided to illustrate this approach.

KEYWORDS

Makespan optimization, Real-time scheduling, Surface treatment.

* INRIA Lorraine, Technopôle Metz 2000, 4 rue Marconi, 57070 Metz, France
** Center for Technological Education Holon, 52 Golomb St., 58102 Holon, Israël
*** Institute for Systems Research, University of Maryland, College Park, MD 20742, USA

Ce travail a été financé par le programme franco-israélien "Arc-en-Ciel/Keshet" n° 64.

Ordonnancement en temps réel d'un produit dans un système de traitement de surface

Fabrice CHAUVET*, Eugene LEVNER**, Leonid K. MEYZIN**
et Jean-Marie PROTH* et ***

RESUME

Nous proposons ici un algorithme en temps réel qui fournit l'ordonnancement minimisant le temps d'achèvement de la dernière opération pour chaque produit devant subir une suite de traitements de surface par bains chimiques. Nous considérons tout d'abord le cas où les durées de traitement sont beaucoup plus importantes que les temps de déplacement entre bains, ce qui permet de négliger ces temps. Puis nous étendons notre approche au cas où les temps de déplacement ne sont plus négligés. Des exemples numériques sont fournis pour illustrer ces approches.

MOTS-CLEFS

Optimisation du Makespan. Ordonnancement en temps réel, Traitement de surface.

* INRIA Lorraine, Technopôle Metz 2000, 4 rue Marconi, 57070 Metz, France
** Center for Technological Education Holon, 52 Golomb St., 58102 Holon, Israël
*** Institute for Systems Research, University of Maryland, College Park, MD 20742, USA

Ce travail a été financé par le programme franco-israélien "Arc-en-Ciel/Keshet" n° 64.

1. INTRODUCTION

Industrial robots have an increasing role in production systems due to their enhanced capacities. They have been applied to a great variety of production situations, such as material handling, loading and unloading operations, especially in engineering, electronics and electrochemical industries. The operational problem of how to use and schedule the robots in an electroplating line is the concern of this paper.

A new real-time approach is introduced to schedule transport activities of the robots in the line containing sequential baths of several types. The parts (jobs) of several types enter the production system in arbitrary moments of time and are to be processed in a job-shop mode, that is, each job has its own route through the machines in the line. Obviously, an efficient robotic transport schedule may attain a maximum production rate for the line.

The problem considered is different from the traditional job-shop scheduling problems, due to three factors. First, loading, unloading and transportation operations of the robot are assumed to be non-negligible and must be taken into account. Second, processing operations durations are interval-valued, that is given in the form of so-called "time windows". Third, and the most important, the jobs may enter the system at any time, and the control system must react in real-time, quickly and effectively providing an on-line schedule for each part entering the line.

Due to the interval-valued processing times, the shortest processing time (SPT) on-line rule used for traditional scheduling and sequencing [1] turns out inapplicable in our problem. Song et al. [2] have considered a similar problem of scheduling identical parts in a chemical processing tank line. These authors suggest an on-line heuristic called "earliest start time" (EST) rule according to which the start of each job entering the system is assigned to each machine as early as possible. This rule is shown to give better results if combined with a combinatorial enumeration of variants in the case of conflicts. Generalizing this approach, we will study the case of non identical part schedule and consider, instead of the classical SPT and EST rules, a new "forward-backward earliest start time" (FBEST) rule which is proven to provide an exact optimum (the minimum makespan) for each entering job.

Several studies of real-time heuristics for sequencing robot activities in automatic line applications have appeared in previous literature. These easy-to-implement heuristic algorithms were recently produced by Lin et al. [3] for the minimization of makespan. Two heuristics for the deterministic single-operator multiple-machine multiple-run cyclic scheduling problem, based on the longest processing time rule, have been developed by Aronson [4] for the minimization of the total costs of the operator (robot) and machines. An extended survey of more than a hundred on-line rules for scheduling manufacturing lines (not necessarily with robots) can be found in [5-7].

In the second section of this paper, we describe the problem at hand, assuming that the times spent by the parts in the baths are much greater than the transportation times; in other words, the robot is supposed to be available at any time and to operate simultaneously. In Section 3, we introduce a real-time algorithm which guarantees that a part which arrives in the system will be completed as soon as possible. Section 4 extends the above algorithm to the case when loading, transportation and unloading tasks of the robot should be taken into account, but the empty moves of the robot are supposed to be dependent only on the baths. Section 5 is the conclusion.

2. PROBLEM SETTING

m different chemical baths are available. For $i = 1,\ldots,m$, n_i is the number of tanks containing the same chemical mixture i. The problem arises at time 0. At this time, we know the idle periods (also called windows) in each bath.

For $i = 1,\ldots,m$ and $j = 1,\ldots,n_i$, $a_{i,j}^k$ is the beginning of the k-th idle period in tank j containing the chemical mixture i. We assume that $k = 1,\ldots,f_{i,j}$. Similarly, $b_{i,j}^k$ is the end of this k-th period.

We denote by θ_i^g and $\theta_i^g + \delta_i^g$ respectively the lower bound and the upper bound of the time a part which is manufactured following the manufacturing process g should spend in the chemical mixture i. Indeed, $\theta_i^g > 0$. In the remaining of this paper, we do not use the notation g for the shake of simplicity, since we consider only one part at a time and no confusion is possible.

Furthermore, the m different chemical baths mentioned are those visited by the part under consideration, in the order of their index, and the other baths are neglected. Indeed, the next part arriving in the system may visit another sequence of baths.

We denote by x_i the instant the part starts its stay in bath i. Since a part is not allowed to stay outside a bath before its completion, x_i is also the completion time of the previous operation, if any. The x_i values are the solution of the problem. We assume that the part visits successively the bath 1, 2, …, m. x_{m+1} is the completion time of the m-th operation.

In the following part of this paper, $F_{i,j}^k = \left(a_{i,j}^k, b_{i,j}^k\right)$ is the k-th window concerning tank j containing the chemical mixture (i.e. the bath) i.

For the tanks containing the same chemical mixture i, the windows $F_{i,j}^k$ are ordered in the increasing order of the $a_{i,j}^k$ and, if two $a_{i,j}^k$ are identical, in the decreasing order of the $b_{i,j}^k$. After these ordering processes, we have the ordered sets:

$$S_i = \left\{F_{i,j}^k\right\} \quad \text{for } i = 1,\ldots,m$$

We denote by $\alpha_{i,r}$ (resp. $\beta_{i,r}$), for $r = 1,\ldots,q_i$, the lower bound (resp. the upper bound) of the idle periods of the tanks containing the same chemical mixture i, ordered in the increasing order of the lower bounds and, in case of equality of the lower bounds, in the decreasing order of the upper bounds. According to the above notations, a period $[\alpha_{i,r}, \beta_{i,r}]$ corresponds to a period $[a_{i,j}^k, b_{i,j}^k]$ where j is the j-th tank containing the chemical mixture i, and $[a_{i,j}^k, b_{i,j}^k]$ is the k-th idle period in this tank, the periods being ordered as mentioned above.

Thus:

$S_i = \{[\alpha_{i,r}, \beta_{i,r}]\}_{r=1,\ldots,q_i}$

where q_i is the number of idle windows associated with bath i. Note that, in this kind of problem, β_{i,q_i} is always equal to infinity. Thus, a feasible solution always exists.

Let us first describe the problem in the case when the robot is available at any time, i.e. when the times associated with the robot can be neglected. It can be set as follows:

Find $x_1 < x_2 < \ldots < x_m < x_{m+1}$ and a sequence of intervals $SQ = \{[\alpha_{i,r_i}, \beta_{i,r_i}]\}_{i=1,\ldots,m}$ which minimizes x_{m+1} and satisfies:

$\theta_i \leq x_{i+1} - x_i \leq \theta_i + \delta_i$

$x_i \geq \alpha_{i,r_i}$

$x_{i+1} \leq \beta_{i,r_i}$

for $i = 1,\ldots,m$

This set of constraints can be replaced by:

$x_{i+1} \geq \theta_i + x_i$ (1)

$x_i \geq \alpha_{i,r_i}$ (2)

$x_{i+1} \leq \theta_i + \delta_i + x_i$ (3)

$x_{i+1} \leq \beta_{i,r_i}$ (4)

for $i = 1\ldots,m$.

A sequence $x_1 < x_2 < \ldots < x_m < x_{m+1}$ is said to be feasible for the sequence SQ of windows if it verifies constraints (1) to (4).

3. AN FBEST ALGORITHM

3.1. Sequence building

In this section, we remain in the case when the operation times of the robot are negligible.

Let us build the sequence $X = \{x_1, x_2, \ldots, x_{m+1}\}$ as follows, after choosing a sequence $[\alpha_{i,r_i}, \beta_{i,r_i}]$ of idle windows:

$t_1 = \alpha_{1,r_1}$ (5-1)

$t_i = \text{Max}\,(\alpha_{i,r_i}, \theta_{i-1} + t_{i-1})$, for $i = 2,3,\ldots,m$ (5-2)

$t_{m+1} = \theta_m + t_m$ (5-3)

$x_{m+1} = t_{m+1}$ (5-4)

$x_i = \text{Max}\,(t_i, x_{i+1} - \theta_i - \delta_i)$, for $i = m, m-1, \ldots, 1$ (5-5)

In Result 1, we show that the elements of the sequence $x_1, x_2, \ldots, x_{m+1}$ verify inequalities (1), (2) and (3).

3.2. Feasibility

Result 1

The sequence X defined by (5-1) to (5-5) verifies inequalities (1), (2) and (3) for the given sequence $[\alpha_{i,r_i}, \beta_{i,r_i}]$, $i = 1,\ldots,m$, of idle windows which have been selected.

Proof

a. X verifies inequality (1) i.e. $x_{i+1} \geq \theta_i + x_i$ for $i = 1,\ldots,m$

From inequality (5-5), we derive:

$$\theta_i + x_i = \mathrm{Max}\,(t_i + \theta_i,\ x_{i+1} - \delta_i) \leq \mathrm{Max}\,(t_i + \theta_i,\ x_{i+1}), \quad \text{for } i = 1,\ldots,m \tag{6}$$

But, according to (5-2) and (5-3):

$$t_i + \theta_i \leq t_{i+1}, \quad \text{for } i = 1,\ldots,m \tag{7}$$

Thus, using inequality (7) in (6):

$$\theta_i + x_i \leq \mathrm{Max}\,(t_{i+1},\ x_{i+1}), \quad \text{for } i = 1,\ldots,m \tag{8}$$

But (see (5-4) and (5-5)):

$$t_i \leq x_i, \quad \text{for } i = 1,\ldots,m+1 \tag{9}$$

Then using (8) and (9), we obtain:

$$\theta_i + x_i \leq x_{i+1}$$

and X verifies inequality (1) for the sequence of idle windows into consideration.

b. X verifies inequality (2), i.e. $x_i \geq \alpha_{i,r_i}$ for $i = 1,\ldots,m$

According to (5-1) and (5-2):

$$t_i \geq \alpha_{i,r_i}, \quad \text{for } i = 1,\ldots,m \tag{10}$$

Considering (9) and (10), we obtain:

$$x_i \geq \alpha_{i,r_i}, \quad \text{for } i = 1,\ldots,m$$

and X verifies inequality (2) for the sequence of idle windows into consideration.

c. X verifies inequality (3), i.e. $x_{i+1} \leq \theta_i + \delta_i + x_i$ for $i = 1,\ldots,m$

According to inequality (5-5):

$$x_{i+1} - \theta_i - \delta_i \leq x_i, \quad \text{for } i = 1,\ldots,m$$

and thus:

$$x_{i+1} \leq x_i + \theta_i + \delta_i, \quad \text{for } i = 1,\ldots,m$$

X verifies inequality (3) for the sequence of idle windows into consideration.

Result 1 does not mean that a sequence X defined by applying (5-1) to (5-5) starting from a given sequence of idle windows is feasible. It only shows that X verifies three (out of four) conditions to be feasible. We still have to verify condition (4).

The following definition is used in Result 2. A sequence $S^* = \{\alpha_{i,r_i}, \beta_{i,r_i}\}$ is a subsequent sequence of $S = \{\alpha_{i,v_i}, \beta_{i,v_i}\}$ if $v_i \geq r_i$ for $i = 1,2\ldots,m$ and $v_i \neq r_i$ for at least one $i \in \{1,2,\ldots,m\}$

Result 2

Let us assume that, in a given sequence X defined by applying (5-1) to (5-5) starting from a given sequence S of idle windows, there exists $w \in \{1,\ldots,m\}$ such that $x_{w+1} > \beta_{w,r_w}$, where $[\alpha_{w,r_w}, \beta_{w,r_w}]$ is the w-th window in the sequence of idle windows under consideration, then no subsequent sequence S* of idle windows containing $[\alpha_{w,r_w}, \beta_{w,r_w}]$ will lead to a sequence X* which is a feasible solution to our problem. Indeed, X is not feasible either, since $x_{w+1} > \beta_{w,r_w}$ means that relation (4) does not hold.

Proof

Applying (5-1) to (5-3) to S* leads to values t_i^* which are greater than or equal to the values t_i obtained by applying the same relations to S since $\alpha_{i,r_i}^* \geq \alpha_{i,r_i}$.

Considering (5-4), (5-5) and the fact that $t_i^* \geq t_i$ for $i = 1,\ldots,m$, we see that $x_i^* \geq x_i$ for $i = 1,\ldots,m$.

As a consequence, $x_{w+1}^* \geq x_{w+1} > \beta_{w,r_w}$ and X* is not feasible since (4) does not hold.

3.3. Real-time algorithm

In this section, we propose a real-time algorithm which leads to a feasible solution. We have proven that this feasible solution is optimal in the sense that it guarantees that the completion time of the product at hand is minimal.

Algorithm 1

1. For each type i of chemical mixture, we order the idle windows as explained in Section 2. The resulting sequence of idle windows is denoted by $[\alpha_{i,r}, \beta_{i,r}]_{r=1,\ldots,q_i}$ for $i = 1,\ldots,m$.
2. We set $r_i = 1$ for $i = 1,\ldots,m$.
3. We build the sequence $X = \{x_1, x_2, \ldots, x_{m+1}\}$ by applying the following sequence of equalities:

   $t_1 = \alpha_{1,r_1}$

   $t_i = \text{Max}\ (\alpha_{i,r_i}, \theta_{i-1} + t_{i-1})$, for $i = 2,3,\ldots,m$

   $t_{m+1} = \theta_m + t_m$

   $x_{m+1} = t_{m+1}$

   $x_i = \text{Max}\ (t_i, x_{i+1} - \theta_i - \delta_i)$, for $i = m, m-1, \ldots, 1$
4. If, whatever $i \in \{1,\ldots,m\}$, inequality $x_{i+1} > \beta_{i,r_i}$ does not hold, we stop the algorithm.
5. For all the $i \in \{1,\ldots,m\}$ which verify $x_{i+1} > \beta_{i,r_i}$, we set $r_i = r_i + 1$, and return in 3.

Since the sequence of equalities applied at the third step of the algorithm is the sequence (5-1) to (5-5), we know that the sequences X which are obtained verify inequalities (1), (2) and (3) (see Result 1). According to the fourth step of the algorithm, the sequence X obtained when the algorithm stops verifies inequality (4), and X is feasible. Considering step five of the algorithm and Result 2, the sequence X obtained when the algorithm stops corresponds to a sequence of idle windows which is not a subsequent sequence of a sequence of idle windows leading to a feasible solution. The result provided in the next subsection is based on this remark. Note that step 3 is executed at most Q times, where $Q = \sum_{i=1}^{m} q_i$ is the number of idle windows. Thus, the complexity of the algorithm is $o\left(Q\left(m + \log_2(Q)\right)\right)$.

Remark

In step 5, if the idle period is included in the current one, that is $[\alpha_{i,r_i+1}, \beta_{i,r_i+1}] \subset [\alpha_{i,r_i}, \beta_{i,r_i}]$, it is not necessary to test a sequence which contains this period. We have to consider the next one, that is $[\alpha_{i,r_i+2}, \beta_{i,r_i+2}]$, assuming that this period is not included in $[\alpha_{i,r_i}, \beta_{i,r_i}]$.

3.4. Optimality

Result 3

The first feasible solution obtained by applying the algorithm is optimal.

Proof

As mentioned in the proof of Result 2, if S* is a subsequent sequence of windows of S, then $x_i^* \geq x_i$ for $i = 1,\ldots,m$, where x_i^* is obtained starting from S* while x_i is obtained starting from S. In particular, $x_{m+1}^* \geq x_{m+1}$, which ends the proof.

3.5. Example

Let us consider the following example. A part should be plunged in three chemical mixtures denoted by 1, 2 and 3. The first chemical mixture is available in three different tanks, the second one in two tanks and the third one in three tanks. The part to be processed arrives at time 0. At this time, the idle periods are the following ones:

Chemical mixture 1

Tank 1: [0,2]; [4,9]; [13,15]; [25,+∞)

Tank 2: [0,4]; [8,17]; [20,+∞)

Tank 3: [4,8]; [13,16]; [20,35]; [40,+∞)

Chemical mixture 2

Tank 1: [1,9]; [11,17]; [21,+∞)

Tank 2: [2,7]; [11,13]; [22,+∞)

Chemical mixture 3

Tank 1: [8,15]; [22,33]; [35,+∞)

Tank 2: [7,11]; [22,28]; [30,35]; [40,+∞)

Tank 3: [4,15]; [18,22]; [30,+∞)

For each chemical mixture, we order the periods as show in Section 2.

Chemical mixture 1

$S_1 = \{[0,4]; [0,2]; [4,9]; [4,8]; [8,17]; [13,16]; [13,15]; [20,+\infty); [20,35]; [25,+\infty); [40,+\infty)\}$

Chemical mixture 2

$S_2 = \{[1,9]; [2,7]; [11,17]; [11,13]; [21,+\infty); [22,+\infty)\}$

Chemical mixture 3

$S_3 = \{[4,15]; [7,11]; [8,15]; [18,22]; [22,33]; [22,28]; [30,+\infty); [30,35]; [35,+\infty); [40,+\infty)\}$

Let us assume that the part should remain:

- in the chemical mixture 1 between 2 and 3 units of time,
- in the chemical mixture 2 between 3 and 4 units of time,
- in the chemical mixture 3 between 2 and 4 units of time.

We develop the different steps of the algorithm (the first step is the order of the idle periods performed above):

2. We set $r_1 = r_2 = r_3 = 1$
3. Sequence building:

$t_1 = 0$

$t_2 = \text{Max}\ (1, 2+0) = 2$

$t_3 = \text{Max}\ (4, 3+2) = 5$

$t_4 = 2 + 5 = 7$

$x_4 = 7$

$x_3 = \text{Max}\ (5, 7-4) = 5$

$x_2 = \text{Max}\ (2, 5-4) = 2$

$x_1 = \text{Max}\ (0, 2-3) = 0$

4. Inequality checking:

$x_2 = 2 < \beta_{1,1} = 4$

$x_3 = 5 < \beta_{2,1} = 9$

$x_4 = 7 < \beta_{3,1} = 15$

As we can see, none of the $i \in \{1,2,3\}$ satisfies $x_{i+1} > \beta_{i,r_i}$. Thus, the algorithm stops and the solution is as follows:

- the part is plunged in the second tank of the chemical mixture 1 at time $x_1 = 0$ and is taken out at time $x_2 = 2$,
- the part is plunged in the first tank of the chemical mixture 2 at time $x_2 = 2$ and is taken out at time $x_3 = 5$,
- finally, the part is plunged in the third tank of the chemical mixture 3 at time $x_3 = 5$ and is taken out at time $x_4 = 7$.

$x_4 = 7$ is the minimal makespan for this part.

Only one iteration was necessary in this particular case. If, for instance, the first idle window of the chemical mixture 2 had been [1,4] instead of [1,9], we would have had $x_3 = 5 > \beta_{2,1} = 4$, and we would have had to restart the computation with the sequence [0,4]; [2,7] and [4,15] of idle windows, and so on.

4. GENERALIZATION

4.1. Notations and remarks

Let $w_{i,i+1}$, be the time required by the robot to transport a part from a tank containing chemical mixture i to a tank containing a chemical mixture i+1. We also denote by e_i the time needed by the robot to arrive in a tank containing a bath i whatever its former position, assuming that the robot is empty.

Three hypotheses are made at this level. The first one concerns the values $w_{i,i+1}$, since it is assumed that the time required by the robot to move from a chemical mixture to the next one does not depend on the tanks which contain these mixtures. This hypothesis is not very restrictive in practice, since the tanks containing the same mixture are usually close to each other. The second hypothesis concerns the empty moves, that is the moves made by the robot to arrive in the tank where it is required. The time required by an empty move is supposed to depend only on upon the destination. This hypothesis is more restrictive than the previous one, but not as restrictive as it could appear, since the empty moves are usually much faster than the loaded moves. The third hypothesis is $e_i \leq \theta_i$, which means that the so-called empty moves need much less time than the minimal time a part remains in a bath.

Nevertheless, time e_i must be chosen as the maximal time required to join a tank containing the chemical mixture e_i from any tank of the system, and this may slightly degrade the efficiency of the proposed management.

4.2. Extension of the algorithm

We denote by n_0 the number of robots available to move the parts from tank to tank. We will show successively how to introduce the loaded moves and the empty moves.

a. Introduction of the loaded moves

We know that the part has to visit successively bath 1,...,m. Between each bath, a moving operation is needed. Thus, the process required to complete the surface treatment can be represented as:

1, (1,2), 2, (2,3), 3, ..., m-1, (m-1,m), m

The time spent by the part in bath i = 1,...,m is included in $[\theta_i, \theta_i+\delta_i]$, while the time to process transportation (i,i+1), i = 1...,m-1 is $w_{i,i+1}$. As we can see, there is no flexibility for the transportation time, unlike what happens for the time spent in the baths (i.e. the chemical mixtures).

When the part arrives in the system, idle periods are available for each tank and each robot. Each robot and each tank are resources. The first difference between these two types of resource is that a tank is used at most once for a given part while a robot can be used several times. The second difference is that no flexibility is allowed when using a robot. But these two remarks have no effect on the following approach.

The introduction of the loaded moves does not modify the basis of the approach used in Algorithm 1. Algorithm 2 leads to a minimal makespan when the loaded moves are taken in to account and the empty moves are neglected or, in other words, the speed of the robot when it moves unloaded is much higher than its speed when it moves loaded and than the minimal period a part should remain in a chemical mixture.

Algorithm 2

1. For each type i of chemical mixture and for the set of available robots, we order the idle windows as explained in Section 2. The resulting sequence of idle windows is denoted by $[\alpha_{i,r}, \beta_{i,r}]_{r=1,\dots,q_i}$ for the chemical mixture $i = 1,\dots,m$ and by $[\alpha_{0,r}, \beta_{0,r}]_{r=1,\dots,q_0}$ for the set of available robots.
2. We set $r_i = 1$ for $i = 1,\dots,m$ and $z(i-1,i) = 1$ for $i = 2,\dots,m$.
3. We build the sequence $X = \{x_1, y_{1,2}, x_2, \dots, x_{m-1}, y_{m-1,m}, x_m\}$ by applying the following sequence of equalities:

   $t_1 = \alpha_{1,r_1}$

   $u_{i-1,i} = \mathrm{Max}\ (\alpha_{0,z(i-1,i)}, \theta_{i-1} + t_{i-1}),\quad \text{for } i = 2,3,\dots,m$

   $t_i = \mathrm{Max}\ (\alpha_{i,r_i}, w_{i-1,i} + u_{i-1,i}),\quad \text{for } i = 2,3,\dots,m$

   $t_{m+1} = \theta_m + t_m$

   $x_{m+1} = t_{m+1}$

   $x_m = \mathrm{Max}\ (t_m, x_{m+1} - \theta_m - \delta_m)$

   $y_{i-1,i} = \mathrm{Max}\ (u_{i-1,i}, x_i - w_{i-1,i}),\quad \text{for } i = m, m-1, \dots, 2$

   $x_{i-1} = \mathrm{Max}\ (t_{i-1}, y_{i-1,i} - \theta_{i-1} - \delta_{i-1}),\quad \text{for } i = m, m-1, \dots, 2$

4. If:

   (i) $x_{m+1} \leq \beta_{m,r_m}$, and

   (ii) $x_i \leq \beta_{0,z(i-1,i)}$, for $i = 2,3,\dots,m$, and

   (iii) $y_{i-1,i} \leq \beta_{i-1,r_{i-1}}$, for $i = 2,3,\dots,m$,

   then the algorithm stops and X is the optimal solution of the problem (i.e. the components for X are the starting time of the operations).
5. If:

   (i) $x_{m+1} > \beta_{m,r_m}$, we set $r_m = r_m + 1$,

   (ii) $x_i > \beta_{0,z(i-1,i)}$ $(i \in \{2,3,\dots,m\})$, we set $z(i-1,i) = z(i-1,i) + 1$,

   (iii) $y_{i-1,i} > \beta_{i-1,r_{i-1}}$ $(i \in \{2,3,\dots,m\})$, we set $r_{i-1} = r_{i-1} + 1$,

   and we return in 3.

b. Introduction of the unloaded moves

A loaded move algorithms follows an unloaded move. We made the assumption that an unloaded move which arrives in a set of tanks containing the chemical mixture i needs a time e_i to be completed.

As a consequence, Algorithm 2 applies by replacing $\alpha_{0,z(i-1,i)}$ by $\alpha_{0,z(i-1,i)} + e_i$.

4.3. Example

We consider an example with three chemical mixtures, each of them being contained in one tank.

The idle periods are as follows:

- chemical mixture 1: [0,2]; [4,10]; [12,+∞)
- chemical mixture 2: [4,8]; [10,+∞)
- chemical mixture 3: [10,30]; [40,+∞)
- robot: [6,17]; [19,+∞)

We assume that:

- each empty move requires 1 unit of time to be performed,
- each loaded move requires 2 units of time to be performed.

The times the part should remain in the baths are given hereafter:

$\theta_1 = 2; \delta_1 = 2$

$\theta_2 = 4; \delta_2 = 1$

$\theta_3 = 5; \delta_3 = 1$

The first sequence of idle periods to be considered is:

[0,2]; [6,17]; [4,8]; [6,17]; [10,30]

The second and the third periods of the sequence concern the robot. According to Algorithm 2, we obtain:

$t_1 = 0$

$u_{1,2} = \text{Max}\ (6+1, 2+0) = 7$

$t_2 = \text{Max}\ (4, 2+7) = 9$

$u_{2,3} = \text{Max}\ (6+1, 4+9) = 13$

$t_3 = \text{Max}\ (10, 2+13) = 15$

$t_4 = 5+15 = 20$

$x_4 = 20$

$x_3 = \text{Max}\ (15, 20-6) = 15$

$y_{2,3} = \text{Max}\ (13, 15-2) = 13$

$x_2 = \text{Max}\ (9, 13-5) = 9$

$y_{1,2} = \text{Max}\ (7, 9-2) = 7$

$x_1 = \text{Max}\ (0, 7-4) = 3$

We can see that:

$x_4 = 20 < \beta_{3,1} = 30$

$x_3 = 15 < \beta_{0,z(2,3)} = 17$

$x_2 = 9 < \beta_{0,z(1,2)} = 17$

$y_{2,3} = 13 > \beta_{2,1} = 8$

$y_{1,2} = 7 > \beta_{1,1} = 2$

According to Algorithm 2, the next sequence of idle periods to be considered is:

[4,10]; [6,17]; [10,+∞); [6,17]; [10,30]

Applying Algorithm 2, we obtain:

$t_1 = 4$

$u_{1,2} = \text{Max}\ (6+1,\ 2+4) = 7$

$t_2 = \text{Max}\ (10,\ 2+7) = 10$

$u_{2,3} = \text{Max}\ (6+1,\ 4+10) = 14$

$t_3 = \text{Max}\ (10,\ 2+14) = 16$

$t_4 = 5+16 = 21$

$x_4 = 21$

$x_3 = \text{Max}\ (16,\ 21-6) = 16$

$y_{2,3} = \text{Max}\ (14,\ 16-2) = 14$

$x_2 = \text{Max}\ (10,\ 14-5) = 10$

$y_{1,2} = \text{Max}\ (7,\ 10-2) = 8$

$x_1 = \text{Max}\ (4,\ 8-4) = 4$

We see that:

$x_4 = 21 < \beta_{3,1} = 30$

$x_3 = 16 < \beta_{0,z(2,3)} = 17$

$x_2 = 10 < \beta_{0,z(21,2)} = 17$

$y_{2,3} = 14 < \beta_{2,2} = +\infty$

$y_{1,2} = 8 < \beta_{1,2} = 10$

These inequalities show that the optimal solution is obtained.

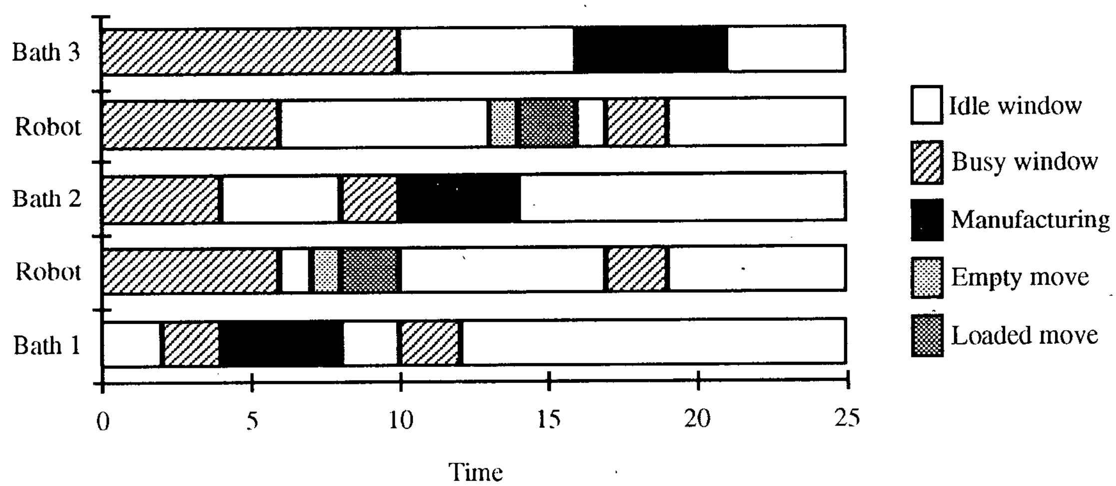

Fig. 1: Optimal solution

5. CONCLUSION

Many algorithms have been proposed in the past to solve the hoist scheduling problem. As mentioned in the introduction, the problem we solved in this paper is different, mainly since the goal is to make decisions in real time, as soon as each part arrives in the system. Since the goal is to minimize the makespan of each part, considered individually, the tendency is to fill the closest idle period first. As a consequence, the resources (tanks and robot(s)) are well utilized in most of the cases which, in turn, leads to a high productivity.

We will explore the application of the FBEST approach to schedule job-shop systems with multiple (identical or not) resources, when the order parts are launched in the system is known.

REFERENCES

[1] Baker K., *Introduction of Sequencing and Scheduling*, Wiley, 1974.

[2] Song W., Zabinsky Z.B and Storch R.L., "An algorithm for scheduling a chemical processing tank line", *Production Planning & Control* 4(4), 323-332, 1993.

[3] Lin J.T, Wen U.P. and Hu S.J., "Operational scheduling for the robot centered parallel processor work cell, *Journal of Chin. Inst. Eng.* 14, 311-318, 1991.

[4] Aronson J.E., "Two heuristics for the deterministic single-operator multiple-machine multiple-run cyclic scheduling problem", *Journal of Operations Management* 4(2), 159-173, 1984.

[5] Panwalkar S. and Isakander W., "A survey of scheduling rules", *Operations Research* 25, 45-61, 1977.

[6] Rachamadugu R. and Stecke K., "Classification and review of FMS scheduling procedures", *Production Planning and Control* 5(1), 2-20, 1994.

[7] Morton T.E. and Pentico D.W., "Heuristic scheduling systems" in *Applications to Production Systems and Project Management*, Wiley, New York, 1993.

Unité de recherche INRIA Lorraine
Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 Villers lès Nancy Cedex (France)

Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot St Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - B.P. 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, B.P. 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399

RR. 3318