Single machine scheduling with chain structured precedence constraints and minimal and maximal separation times

Chengbin Chu, Jean-Marie Proth

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Single Machine Scheduling with Chain Structured Precedence Constraints and Minimal and Maximal Separation Times

Chengbin CHU
Jean-Marie PROTH

N° 2268
Mai 1994

PROGRAMME 5

Rapport de recherche

SINGLE MACHINE SCHEDULING WITH CHAIN STRUCTURED PRECEDENCE CONSTRAINTS AND MINIMAL AND MAXIMAL SEPARATION TIMES

Chengbin Chu[1] and **Jean-Marie Proth**[1,2]

1. Projet SAGEP, INRIA-Lorraine, CESCOM Technopôle Metz 2000, 4 rue Marconi, 57070 Metz, France.
2. Institute for Systems Research, University of Maryland, College Park, MD 20742, USA.

Abstract. We consider a single machine scheduling problem. This problem has been solved for a medical laboratory. It comprises not only chain-structured precedence constraints, but also minimal and maximal times separating successive jobs in the same chain. The criterion to be minimized is the makespan. This problem arises particularly in systems where chemical processes are involved. Consider a chemical plant in which every chemical process is a sequence of chemical reactions the duration of which is upper and lower bounded. Between two successive chemical reactions which are performed in different locations, a transportation system such as robot is used to carry the product from place to place. In this kind of plant, jobs are transportation operations and production processes can be modeled as chains. Therefore the problem studied in this paper is of practical importance. We first prove that the problem is NP-complete. As a consequence, we propose heuristics for large size problems and a branch and bound based algorithm for small size problems. Computational results are reported.

Keywords. Single Machine Scheduling, Makespan, Chain Structured Precedence Constraints, Minimal and Maximal Separation Times.

ORDONNANCEMENT A UNE MACHINE AVEC CONTRAINTES DE PRECEDENCE EN STRUCTURE DE CHAINES ET TEMPS DE SEPARATION MINIMAUX ET MAXIMAUX

Chengbin Chu[1] and **Jean-Marie Proth**[1,2]

1. Projet SAGEP, INRIA-Lorraine, CESCOM Technopôle Metz 2000, 4 rue Marconi, 57070 Metz, France.
2. Institute for Systems Research, University of Maryland, College Park, MD 20742, USA.

Résumé : Nous considérons un problème d'ordonnancement à une machine que nous avons résolu pour un laboratoire d'analyses médicales. Dans ce problème, il y a non seulement des contraintes de précédence en structure de chaines, mais aussi des temps minimaux et maximaux séparant deux tâches successives d'une même chaine. Le critère est de minimiser la durée de l'ordonnancement ou "makespan". Des problèmes de ce type se posent en particulier dans des systèmes où des processus chimiques interviennent. Considérons une usine chimique dans laquelle chaque processus est une séquence de réactions chimiques dont la durée est bornée inférieurement et supérieurement. Entre deux réactions chimiques successives qui ont lieu dans différentes cuves, le système de transport est utilisé pour transférer le produit d'une cuve à l'autre. Dans ce type de systèmes, les tâches sont les opérations de manutention des produits alors que les processus chimiques peuvent être modélisés par des chaines. Par conséquent, le problème étudié dans ce papier a une importance pratique. Nous démontrons tout d'abord que le problème est NP-complet. De ce fait, nous proposons des méthodes heuristiques pour résoudre les problèmes de grande taille, et une méthode basée sur la procédure par séparation et évaluation pour résoudre les problèmes de petite taille. Nous présentons également des applications numériques.

Mots Clés : Ordonnancement à une machine, Makespan, Contraintes de précédence, Chaines, Temps de séparation minimaux et maximaux.

1. INTRODUCTION

In this paper, we consider a single machine scheduling problem which was solved for a medical laboratory. The problem formulation can be summarized as follows. A set J of N jobs available at time 0 should be scheduled on a single machine that can process at most one job at a time. There are chain-structured precedence constraints between jobs. Let n be the number of chains and m_i the number of jobs in the chain i. The total number of jobs is then $N=\sum_{i=1}^{n} m_i$. We denote by (i, j) the jth job in the chain i. Its processing requires $p_{i,j}$ units of time. $p_{i,j}$ is called processing time. The processing of a job cannot be interrupted, that is, there is no preemption of jobs. Besides the precedence constraints, the completion of job (i, j) and the beginning of job $(i, j+1)$ should be separated by a time belonging to $[l_{i,j}, L_{i,j}]$. $l_{i,j}$ and $L_{i,j}$ are respectively the minimal and the maximal separation time between jobs (i, j) and $(i, j+1)$. We are looking for a schedule which minimizes the makespan. If $C_{i,j}(s)$ denotes the completion time of job (i, j) in a schedule s (when there is no ambiguity, we write simply $C_{i,j}$), then the problem can be formulated as follows:

$$\underset{\{C_{i,j}\}}{\text{Min}} \; \max_i C_{i,m_i} \tag{1}$$

subject to

$$C_{i,j+1} - C_{i,j} \geq p_{i,j+1} + l_{i,j}\,, \qquad \forall j < m_i,\ \forall i, \tag{2}$$

$$C_{i,j} - C_{i,j+1} \geq -p_{i,j+1} - L_{i,j}\,, \qquad \forall j < m_i,\ \forall i, \tag{3}$$

$$C_{i,j} \geq p_{i,j}\,, \qquad \forall j < m_i,\ \forall i, \tag{4}$$

$$C_{i,j} - C_{i',j'} \geq p_{i,j} \quad \text{or} \quad C_{i',j'} - C_{i,j} \geq p_{i',j'}\,, \qquad \forall (i,j) \neq (i',j'). \tag{5}$$

Constraints (2) and (3) are introduced to ensure that the separation time between two successive jobs in the same chain is bounded. Constraints (4) say that the jobs cannot be processed before time 0. Constraints (5) express the fact that the machine can perform at most one job at a time. The fact that no preemption is allowed is implicitly taken into account in constraints (2) — (5).

The problem studied here arises particularly in systems where chemical processes are involved. Consider a chemical plant in which every chemical process is a sequence of chemical reactions the duration of which is upper and lower bounded. Between two successive chemical reactions which are performed in different locations, a transportation system is used to carry the product from place to place. In this kind of plant, jobs are transportation operations and production processes can be modeled as chains. Therefore the problem studied in this paper is of practical importance.

Single machine scheduling problems have been extensively studied under various assumptions and criteria. Gupta and Kyparisis (1987) and Lawler et al. (1989) survey hundreds of papers. After these surveys, a lot of new papers were published. However, to our knowledge, no work has been published on the problem studied in this paper.

Some authors consider hoist scheduling problems in electroplating systems. In these systems, products have to visit a number of tanks in a given order. The transportation from one tank tank is carried out by hoists. In each tank, a product has to stay for a time which is upper and lower bounded. The problems studied so far concerns mono-product and cyclic production. Lei and Wang (1989a, 1989b), Phillips and Unger (1976), and Shapiro and Nuttle (1988) examine the problem with single hoist. The problem with two hoists is studied by Lei and Wang (1991) using a minimum common cycle algorithm and the results obtained for single hoist scheduling problems.

Our problem can be considered as a non cyclic multi-product hoist scheduling problem in which the capacity of each tank is infinite. From this remark, one can see that the results published in hoist scheduling problem cannot be used to solve our problem.

We show in Section 2 that the problem is NP-complete. Therefore, the existence of a polynomially bounded algorithm is unlikely. For large size problems, only heuristics can be used. We propose three heuristics based on priority rules in Section 3. For small size problems, we develop in Section 4 a branch and bound based algorithm.

2. NP-COMPLETENESS

In this section, we show that the problem formulated in Section 1 is NP-complete (for the definition of NP-completeness, see Garey and Johnson 1979). We show that the knapsack problem is reducible to our problem. The knapsack problem is well known to be NP-complete.

Theorem 1. The single machine scheduling problem with chain-structured precedence constraints and minimal and maximal separation times between successive jobs in the same chain to minimize the makespan is NP-complete.

Proof. Given integers $a_1, a_2, \ldots, a_t, b, A \triangleq \sum_{i=1}^{t} a_i$, we define a corresponding instance of the considered problem by

$$\begin{aligned} & n = t + 1, \\ & m_i = 1, \quad p_{i,1} = a_i, \qquad\qquad \forall\ 1 \le i < n, \\ & m_n = 2, \quad p_{n,1} = p_{n,2} = 1, \quad l_{n,1} = L_{n,1} = b. \end{aligned}$$

If $\sum_{i \in S} a_i = b$ for some $S \subset T = \{1, 2, \ldots, t\}$, *i.e.* if the knapsack problem has a solution, then a schedule with a makespan less than or equal to $A + 2$ is obtained by scheduling the jobs in the following order $(n, 1)$, $\{(i, 1) | i \in S\}$, $(n, 2)$, $\{(i, 1) | i \in T - S\}$.

For any feasible processing order, T can be decomposed into three subsets S_1, S_2, and $T - S_1 - S_2$, such that jobs of S_1, S_2, and $T - S_1 - S_2$ are respectively processed before $(n, 1)$, between $(n, 1)$ and $(n, 2)$, and after $(n, 2)$. For the order to be feasible, it is necessary that $\sum_{i \in S_2} a_i \le b$. If the knapsack problem has no solution, we have $\sum_{i \in S_2} a_i < b$. The makespan of this schedule is then $\sum_{i \in S_1} a_i + 2 + b + \sum_{i \in T - S_1 - S_2} a_i > \sum_{i \in S_1} a_i + 2 + \sum_{i \in S_2} a_i + \sum_{i \in T - S_1 - S_2} a_i = \sum_{i \in T} a_i + 2 = A + 2$.

This means that the considered problem has a solution such that the makespan is less than or equal to $A + 2$ if and only if the knapsack problem has a solution. Q.E.D.

Since the problem considered is NP-complete, it is necessary to use an enumerative algorithm to obtain an optimal solution, which is irrealistic for large size problems. Therefore, we propose heuristics to cope with the complexity of large size problems.

3. HEURISTICS

In this section, we present some heuristic algorithms. They are based on priority rules. First of all, we show how to calculate the completion times for a given sequence of jobs s, or to detect the unfeasibility of this sequence. Since the separation time between two successive jobs in the same chain is upper bounded, this calculation is not straightforward. Let $c_k(s)$ and $w_k(s)$ be respectively the chain and the job number in this chain of the job processed at the kth position in the sequence s. We are looking for completion times $C_{c_k(s),w_k(s)}$, $k = 1, 2, \ldots, N$ such that the makespan is minimized. This problem, denoted by P1 can be formulated as follows by replacing disjunctive constraints (5) by conjunctive ones.

Problem P1. $$\min_{\{C_{i,j}\}} C_{c_N(s),w_N(s)} \tag{6}$$

subject to

$$C_{c_k(s),w_k(s)} - C_{c_k(s),w_k(s)-1} \geq p_{c_k(s),w_k(s)} + l_{c_k(s),w_k(s)-1}, \qquad \forall\ k \text{ such that } w_k(s) > 1, \tag{7}$$

$$C_{c_k(s),w_k(s)-1} - C_{c_k(s),w_k(s)} \geq -p_{c_k(s),w_k(s)} - L_{c_k(s),w_k(s)-1}, \qquad \forall\ k \text{ such that } w_k(s) > 1, \tag{8}$$

$$C_{c_1(s),w_1(s)} \geq p_{c_1(s),w_1(s)}, \tag{9}$$

$$C_{c_k(s),w_k(s)} - C_{c_{k-1}(s),w_{k-1}(s)} \geq p_{c_k(s),w_k(s)}, \qquad \forall\ 1 < k \leq N. \tag{10}$$

Clearly, this is a linear programming problem. However, due to the particular structure of the problem, that is, in each constraint except one (i.e. (9)) one coefficient of the variables is 1 and the other one is -1, this problem can be represented by a directed graph with $N + 1$ vertices $\{0, 1, 2, \ldots, N\}$. If we denote by $\pi_{i,j}(s)$ the position taken by the job (i, j) in sequence s then, in this graph, for any k such that $w_k(s) > 1$, there are, an arc with length $p_{c_k(s),w_k(s)} + l_{c_k(s),w_k(s)-1}$ from vertex $\pi_{c_k(s),w_k(s)-1}(s)$ to vertex k, and an arc with length $-p_{c_k(s),w_k(s)} - L_{c_k(s),w_k(s)-1}$ from vertex k to vertex $\pi_{c_k(s),w_k(s)-1}(s)$. For $1 \leq k \leq N$, there is an arc with length $p_{c_k(s),w_k(s)}$ from vertex k - 1 to vertex k. According to Roy (1970), this problem has a feasible solution if and only if there is no circuit with positive length. In that case, one of the optimal solutions is such that for any k ($1 \leq k \leq N$), $C_{c_k(s),w_k(s)}$ is the length of the longest path from vertex 0 to vertex k. The computation of these lengths and detection of existence of circuits with positive length can be solved by the algorithm due to Bellman (Gondran and Minoux 1979). Applied to our problem, this algorithm, called TIMING, is as follows:

Procedure TIMING

1. $i := 1$; $C^{0}_{c_k(s),w_k(s)} := 0$, $\forall\ k = 0, 1, \ldots, N$.

2. $C^{i}_{c_0(s),w_0(s)} := 0$;

   For k from 1 *to N do*

   $$C^{i}_{c_k(s),w_k(s)} := \max\left\{C^{i-1}_{c_{k-1}(s),w_{k-1}(s)} + p_{c_k(s),w_k(s)}\ ,\ X\ ,\ Y\right\};$$

   where

   $$X = \begin{cases} 0 & \text{if } w_k(s) = 1\,, \\ C^{i-1}_{c_k(s),w_k(s)-1} + p_{c_k(s),w_k(s)} + l_{c_k(s),w_k(s)-1} & \text{otherwise.} \end{cases}$$

   $$Y = \begin{cases} 0 & \text{if } w_k(s) = m_{c_k(s)}\,, \\ C^{i-1}_{c_k(s),w_k(s)+1} - p_{c_k(s),w_k(s)+1} - L_{c_k(s),w_k(s)} & \text{otherwise.} \end{cases}$$

   done

3. *If* $C^{i}_{c_k(s),w_k(s)} = C^{i-1}_{c_k(s),w_k(s)}$ for any $0 \le k \le N$ *then* STOP, *otherwise if* $i \le N$ *then* $i := i +$ 1 and *go to* 2, *otherwise* there is no feasible solution.

Let $\{C^{*}_{i,j}\}$ and $\{C^{'}_{i,j}\}$ be respectively the solution obtained by Procedure TIMING and a feasible solution of P1. As a consequence from the process used to compute $\{C^{*}_{i,j}\}$, the following property holds, which is important to the computation of the lower bound in the next section.

Property 1. $C^{*}_{i,j} \le C^{'}_{i,j}\ ,\ \ \forall\ 1 \le j \le m_i,\ \forall\ i.$

The Procedure TIMING will be used in the heuristics presented hereafter and the branch and bound algorithm developed in the next section.

In all the heuristics, at each iteration, a chain is scheduled. After having scheduled a subset of chains, we have to decide which chain to be scheduled next and how to schedule it. The selected chain is the one the first job of which can start the earliest. If more than one chain lead to this minimum, the ties are broken according to three policies presented afterwards. These three policies lead to three heuristics.

Let s_0 denote a partial schedule, and s_i the schedule obtained by adding the chain i to s_0.

The general scheme of the heuristic is given below. We denote by I the set of unscheduled chains and by x_j the first position in the schedule that the jth job of a chain of I can occupy.

General scheme

1. $I := \{1, 2, \ldots, n\}$, $x_1 := 1$, $s := \emptyset$.
2. *While* $I \neq \emptyset$ *do*

    $z := +\infty$;

    For any $i \in I$ *do*

    $j := 1$, $\sigma_0 = s$;

    While $j \leq m_i$ *do*

    2.1. Find a position y_j for job (i, j), construct a schedule σ_j by inserting (i, j) at the position y_j and calculate the completion times using Procedure TIMING.

    2.2. *If* σ_j is not feasible *then* reschedule job $(i, j-1)$ at the next subiteration: $x_j := y_j$, $x_{j-1} := y_{j-1} + 1$, $j := j - 1$ *otherwise* consider job $(i, j + 1)$ at the next subiteration: $x_{j+1} := y_j + 1$, $j := j + 1$.

    done

    $s_i := \sigma_{m_i}$.

    If $C_{i,1}(s_i) < z$ then $z := C_{i,1}(s_i)$, $i^* := i$.

    done

    $s := s_{i^*}$, $x_1 := \pi_{i^*,1}(s) + 1$, $I := I - \{i^*\}$.

    done

When we schedule job (i, j), we try to construct a schedule σ_j based on schedule σ_{j-1} in which the job $(i, j - 1)$ has already been scheduled. In our algorithm, the scheduling of job (i, j) is not fixed. It may change if the scheduling of $(i, j + 1)$ fails. However, it is easy to see that the completion time cannot decrease from one scheduling to the next.

While scheduling job (i, j), we compute the "earliest" and the "latest" starting times $r_{u,v}$ and $d_{u,v}$ as follows for each scheduled job and job (i, j).

$$r_{i,j} = \begin{cases} 0 & \text{if} \quad j = 1, \\ C_{i,j-1}(\sigma_{j-1}) + l_{i,j-1} & \text{otherwise.} \end{cases}$$

$$r_{u,v} = C_{u,v}(\sigma_{j-1}) - p_{u,v}, \qquad \forall\ (u, v) \neq (i, j).$$

$$d_{u,v} = \begin{cases} +\infty & \text{if} \quad v = 1, \\ C_{u,v-1}(\sigma_{j-1}) + L_{u,v-1} & \text{otherwise}, \end{cases} \qquad \forall\ (u, v).$$

We then determine the position y_j to put the job (i, j), that is the first y such that

$$y \geq x_j$$
and
$$\left[\ r_{c_y(\sigma_{j-1}),w_y(\sigma_{j-1})} > \rho_y \quad \text{or} \quad \left(\ r_{c_y(\sigma_{j-1}),w_y(\sigma_{j-1})} = \rho_y \quad \text{and} \quad d_{c_y(\sigma_{j-1}),w_y(\sigma_{j-1})} > d_{i,j}\ \right)\ \right]$$

where $\rho_y = \max\left\{C_{c_{y-1}(\sigma_{j-1}),w_{y-1}(\sigma_{j-1})},\ r_{i,j}\right\}$.

In fact ρ_y is the earliest time that (i, j) can start if it is the yth job in the schedule. If the job $(c_y(\sigma_{j-1}),\ w_y(\sigma_{j-1}))$ can start earlier, then it takes the position y, otherwise the most urgent takes this position, the most urgent being the one with the smallest latest starting times. The idea behind this choice is from the well known Schrage's algorithm which is used to minimize the maximum tardiness.

As one can see, when two chains lead to the same starting time of their first job, we have to select one of them. This is done using one of the following three policies.

1. A chain with a larger λ value has a higher priority, where

$$\lambda_i = \sum_{j=1}^{m_i} p_{i,j} + \sum_{j=1}^{m_i-1} l_{i,j}\,.$$

2. A chain with a larger sum of job processing times has a higher priority.
3. A chain whose first job has a longer processing time has higher priority. If two chains give the same value, the chain whose second job has a longer processing time has higher priority, and so on.

It should be noticed that a policy plays an important part in the performance of the algorithms, since choices should be done very often especially at the beginning of the scheduling and the scheduling of the first chains has a very important impact on that of subsequent chains.

4. BRANCH AND BOUND ALGORITHM

In practice, there may be situations where the number of jobs to be scheduled is small. In this case, an enumerative algorithm applies. For this purpose, we first show how to calculate lower bounds and to identify unfeasible solutions. Based on these results, we develop a branch and bound algorithm. The computational experiences are reported in Section 5.

Let s be a partial schedule in which, if a job is scheduled, then all its predecessors in the same chain are scheduled, or equivalently, its immediate predecessor, if any, is scheduled. Let S denote the set of scheduled jobs in s. For any chain i, let z_i be the number of scheduled jobs in this chain. The set of unscheduled jobs is then J - S. The problem of scheduling the jobs belong to J - S after scheduling the jobs of S in order to minimize the makespan can be formulated as follows.

Problem P2.

$$\min_{\{C_{i,j}\}} \max_{i/z_i<m_i} C_{i,m_i} \tag{11}$$

subject to (2), (3) and

$$C_{c_1(s),w_1(s)} \geq p_{c_1(s),w_1(s)}, \tag{12}$$

$$C_{c_k(s),w_k(s)} - C_{c_{k-1}(s),w_{k-1}(s)} \geq p_{c_k(s),w_k(s)}, \qquad \forall\ 1 < k \leq |S|, \tag{13}$$

$$C_{i,j} - C_{c_{|S|}(s),c_{|S|}(s)} \geq p_{i,j}, \qquad \forall\ (i,j) \in J - S, \tag{14}$$

$$C_{i,j} - C_{i',j'} \geq p_{i,j} \quad \text{or} \quad C_{i',j'} - C_{i,j} \geq p_{i',j'},$$
$$\forall\ (i,j) \in J-S,\ \forall\ (i',j') \in J-S \text{ and } (i,j) \neq (i',j'). \tag{15}$$

Constraints (2) and (3) can be rewritten as

$$C_{c_k(s),w_k(s)} - C_{c_k(s),w_k(s)-1} \geq p_{c_k(s),w_k(s)} + l_{c_k(s),w_k(s)-1}, \qquad \forall\ k \leq |S| \text{ and } w_k(s) > 1, \tag{16}$$

$$C_{c_k(s),w_k(s)-1} - C_{c_k(s),w_k(s)} \geq -p_{c_k(s),w_k(s)} - L_{c_k(s),w_k(s)-1}, \qquad \forall\ k \leq |S| \text{ and } w_k(s) > 1, \tag{17}$$

$$C_{i,j} - C_{i,j-1} \geq p_{i,j} + l_{i,j-1}, \qquad \forall j > 1 \text{ and } (i,j) \in J - S, \tag{18}$$

$$C_{i,j-1} - C_{i,j} \geq -p_{i,j} - L_{i,j-1}\,, \qquad \forall j > 1 \text{ and } (i,j) \in J - S. \quad (19)$$

Constraints (14) and (18) lead to

$$C_{i,j} \geq \max\left\{C_{i,z_i} + l_{i,z_i}\,, C_{c_{|S|}(s),w_{|S|}(s)}\right\} + p_{i,j} + \sum_{j'=z_i+1}^{j-1}\left(l_{i,j'} + p_{i,j'}\right),$$
$$\forall\ z_i < j < m_i,\ \forall\ i \text{ such that } z_i \geq 1, \quad (20)$$

$$C_{i,j} \geq C_{c_{|S|}(s),w_{|S|}(s)} + p_{i,j} + \sum_{j'=1}^{j-1}\left(l_{i,j'} + p_{i,j'}\right), \qquad \forall\ 1 \leq j < m_i,\ \forall i \text{ such that } z_i = 0, \quad (21)$$

$$C_{i,m_i} \geq C_{i,j} + \sum_{j'=j+1}^{m_i}\left(l_{i,j'-1} + p_{i,j'}\right), \qquad \forall\ (i,j) \in J - S,\ \forall\ i \text{ such that } z_i < m_i. \quad (22)$$

Therefore, the criterion value of the following problem is a lower bound of Problem P.

Problem P3.

$$\underset{\{C_{i,j}\}}{\text{Min}}\ \max_{(i,j)\in J-S}\left\{C_{i,j} + \sum_{j'=j+1}^{m_i}\left(l_{i,j'-1} + p_{i,j'}\right)\right\} \quad (23)$$

subject to (12), (13), (15) — (17), (20) and (21).

One can notice that the constraints (12), (13), (16) and (17) are the same as (7) - (10), except that N is replaced by $|S|$. If these constraints lead to a feasible solution then according to Property 1, when we replace $C_{c_k(s),w_k(s)}$ by $C'_{c_k(s),w_k(s)}$ in (20) and (21), where $C'_{c_k(s),w_k(s)}$ is the completion time obtained using Procedure TIMING on the jobs of S, the objective function of the resulting problem gives a lower bound to Problem P3, and consequently to Problem P2. This problem, denoted by P4, can be formulated as follows.

Problem P4.

$$\underset{\{C_{i,j}/(i,j)\in J-S\}}{\text{Min}}\ \max_{(i,j)\in J-S}\left\{C_{i,j} + \sum_{j'=j+1}^{m_i}\left(l_{i,j'-1} + p_{i,j'}\right)\right\} \quad (24)$$

subject to (15) and

$$C_{i,j} \geq \max\left\{C'_{i,z_i} + l_{i,z_i}\,, C'_{c_{|S|}(s),w_{|S|}(s)}\right\} + p_{i,j} + \sum_{j'=z_i+1}^{j-1}\left(l_{i,j'} + p_{i,j'}\right),$$

$$\forall\ z_i < j < m_i,\ \forall\ i \text{ such that } z_i \geq 1, \quad (25)$$

$$C_{i,j} \geq C'_{c_{|S|}(s),w_{|S|}(s)} + p_{i,j} + \sum_{j'=1}^{j-1}\left(l_{i,j'} + p_{i,j'}\right), \qquad \forall\ 1 \leq j < m_i,\ \forall i \text{ such that } z_i = 0. \quad (26)$$

Problem P4 is exactly the single machine scheduling problem to minimize makespan with release dates and delivery times, where the release dates $r_{i,j}$ and delivery times $q_{i,j}$ are respectively,

$$r_{i,j} = \max\left\{C'_{i,z_i} + l_{i,z_i},\ C'_{c_{|S|}(s),w_{|S|}(s)}\right\} + \sum_{j'=z_i+1}^{j-1}\left(l_{i,j'} + p_{i,j'}\right),$$

$$\forall\ z_i < j < m_i,\ \forall\ i \text{ such that } z_i \geq 1, \quad (27)$$

$$r_{i,j} = C'_{c_{|S|}(s),w_{|S|}(s)} + \sum_{j'=1}^{j-1}\left(l_{i,j'} + p_{i,j'}\right), \qquad \forall 0 < j < m_i,\ \forall i \text{ such that } z_i = 0, \quad (28)$$

$$q_{i,j} = \sum_{j'=j+1}^{m_i}\left(l_{i,j'-1} + p_{i,j'}\right), \qquad \forall (i,j) \in J - S. \quad (29)$$

This problem has been extensively studied in the literature (Bratley et al. 1973, Baker and Su 1974, McMahon and Florian 1975, Carlier 1982, Grabowski et al. 1986, Dessouky and Margenthaler 1972, Hall and Rhee 1986, Lageweg et al. 1976, Larson et al. 1985). Unfortunately, the problem is still NP-hard. But if we further relax the non preemption constraints, the problem can be polynomially solved by the following algorithm (Baker and Su 1974): When we have to select a job among the available unscheduled ones, the one with the largest delivery time is chosen. Furthermore, an arriving job preempts the job being processed if the delivery time of the new arrival is strictly smaller than that of the job in process.

Therefore the following result holds.

Theorem 2. If there is a feasible solution to constraints (12), (13), (16) and (17), the criterion value of Problem P4 without non preemption constraints is a lower bound of Problem P2.

In addition, we have the following theorem.

Theorem 3. If there is no feasible solution to constraints (12), (13), (16) and (17), any schedule σ such that $(c_k(\sigma), w_k(\sigma)) = (c_k(s), w_k(s))$, $\forall\ 1 \le k < |S|$ is not feasible.

Proof. Assume that there is such a feasible schedule σ. In this case, the completion times of the jobs S in σ satisfy the constraints (12), (13), (16) and (17). This is in contradiction with the assumption that there is no feasible solution to constraints (12), (13), (16) and (17). Q.E.D.

We develop a branch and bound algorithm using Theorems 2 and 3. In this algorithm, each node is defined by a partial sequence s. Initially, s is empty. Each descendant node is obtained by adding, behind the partial sequence s, a new job (i, j) chosen among the unscheduled jobs. Procedure TIMINIG is then used to compute the completion times of the jobs in s completed by job (i, j). If this new partial sequence is unfeasible, all the descendant nodes of s are eliminated according to Theorem 3. Otherwise, a new node is created with a lower bound computed using Theorem 2 and we continue by considering other descendant nodes with other unscheduled jobs.

Heuristics presented in Section 3 are used to provide an initial upper bound. The upper bound is updated each time a better complete schedule is found. The selection of the node to branch next is performed with the depth first policy, because with this policy, the application of Theorem 3 can be easily implemented and requires a reasonable storage core.

We can improve the Theorem 3 taking into account the following remark. We know from (14) and (19) that for a partial schedule s in which S is the set of scheduled jobs,

$$C_{c_{|S|}(s), w_{|S|}(s)} \le C_{i,j} + L_{i,j}, \qquad \forall\ (i, j) \in S \text{ and } (i, j+1) \in J - S. \tag{29}$$

This means that in the directed graph defined in Section 3 there is an arc with length $-L_{i,j}$ from vertex $|S|$ to vertex $\pi_{i,j}(s)$. If this graph contains circuits with positive length, there is no feasible complete sequence starting by the partial sequence s, thus the node represented by the partial sequence s can be eliminated.

5. COMPUTATIONAL RESULTS

In order to evaluate the performance of the algorithms, we randomly generate 50 test examples. For each example $n = 10$. The m_i values are randomly generated from 1 to 3. The $p_{i,j}$ and $l_{i,j}$ values are randomly generated from 10 to 40 and from 100 to 400, respectively. $L_{i,j}$ is set to $1.1l_{i,j}$.

The algorithms are implemented in C codes on a SUN SPARC 1+ workstation. For 9 examples, the branch and bound algorithm is stopped because the optimal solution is not found after 10 000 branchings. Except for these examples, the mean number of nodes considered is 124.24, the mean number of branchings is 653.37 and the mean computation time is 725.51 CPU milliseconds. It should be noticed that for 23 problems out of 50, the heuristics give the optimal solutions.

The mean computation time of the heuristics is 6.42 CPU milliseconds. The deviation of the sum of criterion values obtained by the heuristics from the sum of criterion values by the branch and bound algorithm is 1.86%. That is, $\left(\sum_{i=1}^{50} h_i - \sum_{i=1}^{50} o_i\right) \big/ \sum_{i=1}^{50} o_i = 0.0186$, where h_i and o_i are the criterion values obtained respectively with the heuristics and the branch and bound algorithm (If it is stopped before the optimal solution is found, o_i is taken as the minimal lower bounds of the nodes in the list. Therefore o_i is smaller than the optimal solution). The mean deviation from the optimum is 1.92%. That is, $\left(\sum_{i=1}^{50} (h_i - o_i)/o_i\right) \big/ 50 = 0.0192$.

When the heuristics are applied to real life problems of the company we worked for, the service ratio (ratio between the total processing time and the makespan) is between 75%—95%. The size of these problems is very large: The number of jobs is between 100 and 400 and each chain contains 3 or 4 jobs. It is difficult to evaluate the performance of the heuristics for large size problems. Of course, it is always possible to compare the heuristic solution with the lower bound. However, the lower bound could be quite far away from the optimum. One can imagine to improve the lower bound using Lagrangian relaxation technique as done by Luh and Hoitomt. (1993). Unfortunately, the computation time and storage core used by this method increase with the considered time horizon which should be larger than the makespan and thus than the total processing time. For the real life problem we studied, total processing time could be 40 000 seconds. It is also impossible to change time unit, because the processing time of each job is between 19 and 53 seconds.

6. CONCLUSION

We study a real life sequencing problem with chain structured constraints and minimal and maximal separation times between jobs. Heuristic and branch and bound algorithms are developed to solve respectively large and small size problems. Computational results show that the heuristics are quite effective.

Compared to classical scheduling problems issued from manufacturing systems, our problem arises in systems where chemical processes are involved and the transportation times are not negligible with respect to operation processing times. We think that this kind of scheduling problems need more attention from researchers, because transportation facilities are expensive and thus are used to service several machines. In that case, the assumption with negligible transportation times does not hold any more.

REFERENCES

BAKER, K.R., AND Z. SU. 1974. Sequencing with Due-Dates and Early Start Times to Minimize Maximum Tardiness. *Nav. Res. Logist. Quart.* **21,** 171-176.

BRATLEY, P., M. FLORIAN AND P. ROBILLARD. 1973. On Scheduling with Earliest Start and Due Dates with Application to Computing Bounds for the (*n*/*m*/*G*/*F*max) Problem. *Nav. Res. Logist. Quart.* **20,** 57-67.

CARLIER, J. 1982. The One Machine Sequencing Problem. *Europ. J. Oper. Res.* **11,** 42-47.

DESSOUKY, M.I., AND C.R. MARGENTHALER. 1972. The One-Machine Sequencing Problem with Early Starts and Due Dates. *AIIE Transactions* **4,** 214-222.

GAREY, M.R., AND D.S. JOHNSON. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman and Company, New York.

GONDRAN, M, AND M. MINOUX. 1979. *Graphes et algorithmes.* Editions Eyrolles, Paris.

GRABOSKI J., E. NOWICKI AND S. ZDRZALKA. 1986. A Block Approach for Single-Machine Scheduling with Release Dates and Due Dates. *Europ. J. Oper. Res.* **26,** 278-285.

GUPTA, S.K., AND J. KYPARISIS. 1987. Single Machine Scheduling Research. *OMEGA Int. J. Mgmt. Sci.*, **15,** 207-227.

HALL, N.G., AND W.T. RHEE. 1986. Average and Worst-Case Analysis of Heuristics for the Maximum Tardiness Problem. *Europ. J. Oper. Res.* **26,** 272-277.

LAGEWEG, B.J., J.K. LENSTRA AND A.H.G. RINNOOY KAN. 1976. Minimizing Maximum Lateness on One Machine: Computational Experience and Some Applications. *Statist. Neer.* **30,** 25-41.

LARSON, R.E., M.I. DESSOUKY AND R.E. DEVOR. 1985. A Forward-Backward Procedure for the Single Machine Problem to Minimize Maximum Lateness. *IIE Transactions* **17,** 252-260.

LAWLER, E.L., J.K. LENSTRA J.K., A.H.G. RINNOOY KAN AND S.D.B. SHMOY. 1989.

Sequencing and Scheduling: Algorithms and Complexity. *Report* **BS-R8909,** Center for Mathematics and Computer Science, Amsterdam.

LEI, L. AND T.J. WANG. 1989a. On the Optimal Cyclic Scheduling of Single Hoist Electroplating Processes. *Working Paper* **89-0006,** Rutgers University.

LEI, L. AND T.J. WANG. 1989b. A Proof: The Cyclic Hoist Scheduling Problem is NP-Complete. *Working Paper* **89-0016,** Rutgers University.

LEI, L. AND T.J. WANG. 1991. On the Optimal Cyclic Scheduling of Single Hoist Electroplating Processes. *Man. Sci.* **37,** 1629-1639.

LUH, P.B., AND D.J. HOITOMT. 1993. Scheduling of Manufacturing Systems Using the Lagrangian Relaxation technique. IEEE Trans. Aut. Cont. 1066-1079.

MCMAHON, G., AND M. FLORIAN. 1975. On Scheduling with Ready Times and Due Dates to Minimize Maximum Lateness. *Opns. Res.* **23,** 475-482.

PHILLIPS, L.W. AND P.S. UNGER. 1976. Mathematical Programming Solution of a Hoist Scheduling Program. *AIIE Trans.* **13,** 304-310.

ROY, B. 1970. *Algèbre moderne et théorie des graphes.* Dunod, Paris.

SHAPIRO, G.W. AND H.W. NUTTLE. 1988. Hoist Scheduling for a PCB Electroplating Facility. *IIE Trans.* **20,** 157-167.

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 - 46, avenue Félix Viallet - 38031 Grenoble Cedex 1 (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)

ISSN 0249 - 6399

RR_2268