RAIRO. RECHERCHE OPÉRATIONNELLE

CHENGBIN CHU

A new class of scheduling criteria and their optimization

RAIRO. Recherche opérationnelle, tome 30, nº 2 (1996), p. 171-189

<http://www.numdam.org/item?id=RO_1996__30_2_171_0>

Recherche opérationnelle/Operations Research
(vol. 30, n° 2, 1996, pp. 171-189)

A NEW CLASS OF SCHEDULING CRITERIA AND THEIR OPTIMIZATION (*)

by Chengbin CHU ([1])

Communicated by Jacques CARLIER

Abstract. – *In this paper, we define a new class of scheduling criteria called over-regular criteria. This class is a subclass of regular criteria. We analyze the properties of scheduling problems with over-regular criteria. In the literature a lot of effort has been made to solve specific problems. Our properties, however, are applicable to several scheduling problems. Using these properties and additional specific ones, we propose a procedure to optimize an over-regular criterion in single machine case: minimization of total tardiness with generalized due dates. This problem particularly arises in maintenance departments where maintained parts are interchangeable. Computational results about this procedure are also reported.*

Keywords: Scheduling, over-regular criteria, properties, optimization.

Résumé. – *Dans cet article, nous définissons une nouvelle classe de critères d'ordonnancement appelés critères plus que réguliers. Cette classe est une sous classe des critères réguliers. Nous analysons les propriétés des problèmes avec critères plus que réguliers. Dans la littérature, beaucoup d'efforts se sont portés sur la résolution de problèmes spécifiques. Nos propriétés, en revanche, sont applicables à plusieurs problèmes. En utilisant ces propriétés et en développant des propriétés supplémentaires spécifiques, nous proposons une procédure pour optimiser un des critères plus que réguliers dans le cas d'une seule machine : la minimisation de la somme des retards avec délais généralisés. Ce dernier problème se pose en particulier dans des services de maintenance où les pièces réparées sont interchangeables. Nous présentons également des expériences numériques sur cette procédure.*

Mots clés : Ordonnancement, critères plus que réguliers, propriétés, optimisation.

1. INTRODUCTION

In the scheduling literature, a lot of effort has been made to solve specific problems. If some context changes, the results become inapplicable. General results are limited to the definition of regular criteria and the dominance of semi-active and active schedules (Conway, Maxwell and Miller, 1967; Baker, 1974). In this paper, we define a new class of scheduling criteria called

(*) Received February 1994.

([1]) INRIA-Lorraine, CESCOM Technopôle Metz 2000, 4, rue Marconi, 57070 Metz, France. Email: chu@lapin.loria.fr.

over-regular criteria. We give a number of dominance properties applicable for any one of the criteria in the case of one-machine scheduling. We also show that these criteria can be minimized by scheduling the jobs according to the shortest remaining processing time (SRPT) priority rule if the jobs are preemptive. Using these properties and supplementary dominance properties, we propose a branch and bound algorithm to solve a particular over-regular criterion scheduling problem: minimizing total tardiness with generalized due dates.

The concept of generalized due dates has been introduced by Hall (1986). Traditionnally, the due dates are job specific, that is, with each job i is associated a due date d_i. If job i is completed after its due date d_i, it is said to be tardy. With generalized due dates, however, the due dates are independent of jobs. If we denote by $d_{[j]}$ the jth smallest due date (*i.e.* $d_{[1]} \leq d_{[2]} \leq \ldots \leq d_{[n]}$, where n is the number of jobs), the job completed at the jth position will be considered to be tardy if its completion time $C_{[j]}(S)$ in a schedule S is greater than $d_{[j]}$. In the same paper, Hall also described a number of applications in which the generalized due date definition is appropriate. These applications include publicity planning problems, survey design, and scheduling problems in some manufacturing environments. Hall, Sethi and Sriskandarajah (1991) cited a particular instance in the petrochemical industry, where a number of interchangeable heat exchangers must be maintained. The repair time of a heat exchanger is estimated with respect to the extent of corrision and erosion on it. Since a number of heat exchangers must be serviced by a certain date, the problem can be formulated as a generalized due date scheduling problem.

This paper considers scheduling problems with over-regular criteria minimization involving generalized due dates. Section 2 is devoted to the definition of over-regular criteria and related analysis. Section 3 considers the total tardiness minimization with generalized due dates.

2. OVER-REGULAR CRITERIA AND THEIR PROPERTIES

In the literature, a regular criterion is defined as follows (Conway, Maxwell and Miller, 1967; Baker, 1974).

DEFINITION 1: *Let $C_i(S)$ and $G(S)$ denote, respectively, the completion time of job i and the value of criterion G for a schedule S. If $C_i(\sigma) \leq C_i(\sigma')$ $\forall\, i = 1, 2, \ldots, n$ implies $G(\sigma) \leq G(\sigma')$ for two schedules σ and σ', the criterion G is said to be a* regular criterion.

According to the notations, it is not difficult to see that the series $(C_{[1]}(S), C_{[2]}(S), \ldots, C_{[n]}(S))$ is obtained by reordering in nondecreasing order the series $(C_1(S), C_2(S), \ldots, C_n(S))$.

Other notions already defined in the literature (Baker, 1974 for example) are dominant subset and active schedules. A *dominant subset* is a subset of solutions containing at least one optimal solution. An *active schedule* is such that no job can be completed earlier without delaying another. It has been shown that the subset of active schedules is dominant for regular criteria (Baker, 1974).

Now we define a new class of criteria called over-regular criteria as follows.

DEFINITION 2: *If, for any two schedules σ and σ' such that*

$$C_{[j]}(\sigma) \leq C_{[j]}(\sigma'), \qquad \forall\ j = 1, 2, \ldots, n,$$

we have $G(\sigma) \leq G(\sigma')$, G is then said to be an over-regular criterion.

In fact, if two schedules σ and σ' are such that $C_i(\sigma) \leq C_i(\sigma')$, $\forall i = 1, 2, \ldots, n$ it is easy to see that $C_{[j]}(\sigma) \leq C_{[j]}(\sigma')$, $\forall j = 1, 2, \ldots, n$. If G is an over-regular criterion, then $G(\sigma) \leq G(\sigma')$. This means that G is a regular criterion as well. Consequently, the following property holds.

PROPERTY 1: *Any over-regular criterion also is a regular criterion.*

It is from this property that we choose the name "over-regular" criterion. Let us look at some scheduling criteria.

(*a*) Makespan $C_{\max}(S) = \max\limits_{i=1, 2, \ldots, n} \{C_i(S)\} = C_{[n]}(S)$.

(*b*) Total completion time $C(S) = \sum\limits_{i=1}^{n} C_i(S) = \sum\limits_{j=1}^{n} C_{[j]}(S)$.

(*c*) Total tardiness with generalized due date $T(S) = \sum\limits_{j=1}^{n} T_{[j]}(S)$, where $T_{[j]}(S) = \max\{C_{[j]}(S) - d_{[j]}, 0\}$.

(*d*) Maximum lateness with generalized due dates

$$L_{\max}(S) = \max_{j=1, 2, \ldots, n} \{L_{[j]}(S)\},$$

where $L_{[j]}(S) = C_{[j]}(S) - d_{[j]}$.

(e) Total number of tardy jobs with generalized due dates

$$U(S) = \sum_{j=1}^{n} U_{[j]}(S),$$

where $U_{[j]}(S) = \begin{cases} 1 & \text{if} \quad C_{[j]}(S) > d_{[j]}, \\ 0 & \text{otherwise}. \end{cases}$

It is easy to see that all these criteria are over-regular. Since the total completion time minimization is equivalent to total (mean) flow time, total (mean) waiting time and total (mean) lateness minimization (Conway, Maxwell and Miller, 1967), these latter criteria are over-regular as well. This problem has been extensively studied in single machine case with unequal release dates (Ahmadi and Bagchi, 1990; Liu and MacCarthy, 1991; Belouadah, Posner and Potts, 1992; Bianco and Ricciardelli, 1982; Chandra, 1979; Chu, 1992*b*, *c*; Deogun, 1983; Dessouky and Deogun, 1981; Hariri and Potts, 1983). Among the optimal algorithms, the one proposed by Chu (1992*b*) capable of solving problems with up to 100 jobs seems to be the most effective.

With specific due dates, the single machine scheduling problem to minimize maximum lateness has also been extensively studied. When the jobs are available at the same time, the problem can be polynomially solved using Jackson's algorithm (Jackson 1955). If the release dates are unequal, the problem has been considered by Dessouky and Margenthaler (1972), Shwimer (1972), Bratley, Florian and Robillard (1973), Baker and Su (1974), McMahon and Florian (1975), Lageweg, Lenstra and Rinnooy Kan (1976), Potts (1980), Carlier (1982), Larson, Dessouky and Devor (1985), Grabowski, Nowicki and Zdrzalka (1986), Hall and Rhee (1986). The algorithm proposed by Carlier (1982) is able to handle problems with up to 10000 jobs.

In the remainder of this section, we establish some properties for over-regular criteria in the case of single machine scheduling. In this problem, a set of jobs $N = \{1, 2, \ldots, n\}$ have to be scheduled on a single machine assumed to be able to process at most one job at a time. Each job i has a release date $r_i \geq 0$ at which it becomes available to be processed, and a processing time p_i. Once a job starts being processed, the processing cannot be interrupted.

The following theorem shows that if the jobs are preemptive, the over-regular criteria minimization problem can be polynomially solved by applying the shortest remaining processing time (SRPT) priority rule. According to this rule, jobs are scheduled portion after portion from time 0. When a job is completed, a waiting job with the shortest remaining processing time is chosen to be processed by the machine. When a new job arises, it takes possession of the machine if its processing time is strictly smaller than the remaining processing time of the job in process.

THEOREM 1: *The single machine scheduling problem to minimize over-regular criteria with preemptive jobs is polynomially solvable by applying the SRPT rule. The complexity of the problem is* $O(n \log n)$.

Proof: Let σ be a schedule obtained by applying the SRPT rule and σ' *any* feasible schedule. We proved (Chu, 1992*a*) that $C_{[j]}(\sigma) \leq C_{[j]}(\sigma')$, $\forall j = 1, 2, \ldots, n$. From the definition of over-regular criterion G, we have $G(\sigma) \leq G(\sigma')$. This means that σ is an optimal schedule. In the same paper, we also showed that the SRPT rule can be implemented in $O(n \log n)$. ■

According to Theorem 1, all over-regular criteria can be minimized by the shortest processing time (SPT) rule if the release dates are identical, as the problems with and without preemption are then equivalent (French 1982). The SRPT rule also reduces to the SPT (shortest processing time) rule in this case.

From now on, we present some dominance properties for over-regular criteria minimization. Before doing this, we define what "dominance" and "dominance property" mean.

DEFINITION 3: *Consider a scheduling problem where the criterion G is to be minimized. If two schedules σ and σ' are such that $G(\sigma) \leq G(\sigma')$, we say that schedule σ* dominates *schedule σ'.* A dominance property *indicates the conditions under which a schedule dominates another.*

According to the definition, we can see that dominance properties are very useful to reduce the search for an optimal solution and hence to reduce computational effort because we do not need to consider dominated solutions.

From the definition of over-regular criteria, in order to prove that σ dominates σ', it is sufficient to prove that $C_{[j]}(\sigma) \leq C_{[j]}(\sigma')$, $\forall j = 1, 2, \ldots, n$. Another way to prove this is to use the following property.

PROPERTY 2: *If two schedules σ and σ' and two jobs i and k are such that*

(i) $\min\{C_i(\sigma), C_k(\sigma)\} \leq \min\{C_i(\sigma'), C_k(\sigma')\}$,

(ii) $\max\{C_i(\sigma), C_k(\sigma)\} \leq \max\{C_i(\sigma'), C_k(\sigma')\}$

and

(iii) $C_l(\sigma) \leq C_l(\sigma')$, $\forall l$, $l \neq i$ *and* $l \neq k$,

then σ dominates σ' for any over-regular criterion.

Proof: The conditions described in Property 2 imply that

$$C_{[j]}(\sigma) \leq C_{[j]}(\sigma'), \qquad \forall j = 1, 2, \ldots, n. \quad \blacksquare$$

In the following we give dominance properties that can be proved using Property 2 or directly the definition. We only give the proof of Theorem 5. The proofs of other theorems are quite straightforward using each time an interchange argument. Some theorems have been proved for total completion time minimization. Interested reader can find in Chu (1992*d*) the proofs of these theorems for general over-regular criteria.

Henceforth, let $R_i(t) = \max(t, r_i)$ and $F_i(t) = R_i(t) + p_i$ denote respectively the earliest start time and the earliest finish time of job i at time t. Let K denote a partial schedule, $J(K)$ the set of jobs in K, $q(K)$ the number of jobs in K and $E(K)$ the completion time of the last job in K. K/i is the new partial schedule obtained by adding job i behind the given partial schedule K. $P(K/i)$ is the schedule composed of K/i, completed by the partial optimal schedule of jobs belonging to $N - J(K/i)$ starting from the moment $E(K/i) = F_i(E(K))$.

THEOREM 2: *If* $\max_{i \in N - J(K)} (r_i) \leq \min_{i \in N - J(K)} \{F_i(E(K))\}$ (i.e. *if we schedule any one of the jobs, all the other jobs become available before its completion) and j and k are two jobs belonging to $N - J(K)$ such that $R_j(E(K)) \leq R_k(E(K))$ and $F_j(E(K)) \leq F_k(E(K))$ (this means that j is available earlier than k and can also be completed earlier than k) then $P(K, k)$ is dominated.*

THEOREM 3: *Given two jobs, $i, j \in N - J(K)$, if $p_i \geq p_j$ and $F_i(E(K)) \leq F_j(E(K))$, then $P(K, j)$ is dominated.*

THEOREM 4: *There is an optimal schedule such that no pair of adjacent jobs i and j (i followed by j) are such that $R_j(t_i) \leq R_i(t_i)$ and $F_j(t_i) \leq F_i(t_i)$, with at least one inequality being strict, where t_i denotes the completion time of the job preceding immediately job i in the schedule. If i is the first job in the schedule, t_i is, by convention, set to 0.*

THEOREM 5: *Given two jobs $i, j \in N - J(K)$, if*

$$F_i(E(K)) \leq R_j(E(K)) + \min_{k \in N - J(K)} \{p_k\},$$

then $P(K, j)$ is dominated.

Proof: In this proof, let $\delta = E(K)$ and $\sigma = P(K, j)$ to simplify the notations. Consider the schedule σ. Construct another schedule S by scheduling job i before job j and by delaying by $F_i(\delta) - R_j(\delta)$ all the jobs between jobs i and j, and the positions of all the jobs after job i remaining the same (see *Fig.* 1).

Considering the assumption of Theorem 5, we have:

$$R_i(\delta) = F_i(\delta) - p_i \leq F_i(\delta) - \min_{k \in N - J(K)} \{p_k\} \leq R_j(\delta).$$

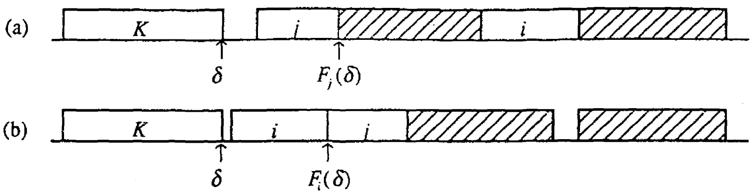

Figure 1. – Illustration of Theorem 5. (a) Schedule σ. (b) Schedule S.

This means that the schedule S is feasible. Suppose that job i is at the πth position in σ. From the construction of schedule S, we have:

(i) $C_{[q(K)+1]}(S) = F_i(\delta)$,

$$C_{[q(K)+1]}(\sigma) = F_j(\delta) \qquad \text{and} \qquad C_{[\pi]}(\sigma) = C_{[\pi-1]}(\sigma) + p_i;$$

(ii) $C_{[l]}(S) = C_{[l-1]}(\sigma) + F_i(\delta) - R_j(\delta)$, $\forall\, l = q(K) + 2,\, q(K) + 3, \ldots, \pi$.

Relations (ii) come from the fact that the jobs between jobs i and j in σ are delayed of 1 position in S and their completion is delayed of $F_i(\delta) - R_j(\delta)$. From the assumption of Theorem 5 and (i), we have

$$F_i(\delta) \leq R_j(\delta) + \min_{k \in N - J(K)} \{p_k\} \leq R_j(\delta) + p_j = F_j(\delta),$$

i.e. $C_{[q(K)+1]}(S) \leq C_{[q(K)+1]}(\sigma)$.

As $C_{[l-1]}(\sigma) \leq C_{[l]}(\sigma) - p_{[l]}$, from (ii) and the assumption of Theorem 5, we have, for any $l = q(K) + 2,\, q(K) + 3, \ldots, \pi$,

$$\begin{aligned} C_{[l]}(S) &\leq C_{[l]}(\sigma) - p_{[l]} + F_i(\delta) - R_j(\delta) \\ &\leq C_{[l]}(\sigma) - \min_{k \in N - J(K)} \{p_k\} + F_i(\delta) - R_j(\delta) \\ &\leq C_{[l]}(\sigma). \end{aligned}$$

(Here is the key point, where the full assumption of the theorem is necessary to its proof.) Therefore $C_{[l]}(S) \leq C_{[l]}(\sigma)$, $\forall\, l = q(K) + 1$, $q(K) + 2, \ldots, \pi$. ■

Since over-regular criteria are also regular ones (Property 1), the set of active schedules is dominant. However, Theorem 5 makes the dominance of active schedules redundant. This means that all non active schedules (hence dominated) can also be identified as dominated by Theorem 5. In a non active schedule σ, there are two jobs i and j such that i follows j and $R_j(t_j) \geq F_i(t_j)$. Let K be the partial schedule before job j. Then we have $t_j = E(K)$ and $i \in N - J(K)$. Therefore $R_j(t_j) \geq F_i(t_j)$ implies that $F_i(E(K)) \leq R_j(E(K)) < R_j(E(K)) + \min_{k \in N - J(K)} \{p_k\}$. Theorem 5 also tells us that σ is dominated.

3. MINIMIZING TOTAL TARDINESS WITH GENERALIZED DUE DATES

This section considers the problem of minimizing total tardiness with release dates and generalized due dates. It has been shown to be NP-hard (Hall, Sethi and Sriskandarajah, 1991).

In the literature, the minimization of total tardiness with respect to specific due dates has received a lot of attention, especially with identical release dates (Baker and Bertrand, 1982; Du and Leung, 1990; Emmons, 1969; Fisher, 1976; Lawler, 1977, 1982; Potts and Van Wassenhove, 1982; Srinivasan, 1971; Wilkerson and Irwin, 1971), among which the algorithm proposed by Potts and Van Wassenhove can solve problems with up to 100 jobs using a decomposition approach and the algorithm of Baker and Schrage (1978) for the problem with precedence constraints between jobs. With different release dates but still with *specific* due dates, we proved a sufficient condition for local optimality which can also be considered as a dynamic priority rule. On the basis of this priority rule, we proposed some efficient heuristics (Chu and Portmann, 1992). We also developed a branch-and-bound algorithm capable of optimally solving hard problems with up to 30 jobs and relatively "easy" problems with up to 230 jobs (Chu, 1992*a*).

Since the problem discussed in this paper is NP-hard, only branch-and-bound approaches or dynamic programming approaches seem to be available to build exact methods. With unequal release dates, idle times may be inserted in the optimal schedules (Conway, Maxwell and Miller, 1967). The presence of these idle times in optimal solutions destroys the usual scheme of dynamic programming approach. Therefore we use a branch-and-bound approach. For this purpose, we give additional dominance properties which are specific to the problem at hand.

3.1. Additional dominance properties

THEOREM 6: *If there are two jobs $j \in N - J(K)$ and $i \in J(K)$ in the kth position such that $F_i(t_i) \geq F_j(t_i)$ and*

$$\{n - q(K) - 1\} \times (p_i - p_j) \leq T_{i,j} - T_{j,i},$$

then $P(K, j)$ is dominated, where

$$T_{u,v} = \max[F_u(t_i) - d_{[k]}, 0] + \max[F_v(E(K)) - d_{[q(K+1)]}, 0].$$

The left term is in fact an upper bound of the increase of total tardiness of remaining jobs by interchanging jobs i and j. The right term is the decrease of total tardiness of jobs i and j by the same interchange.

Proof: Consider the schedule $P(K, j)$. We construct another schedule S by interchanging the positions of jobs i and j (see *Fig.* 2).

In the case where $p_i \leq p_j$, considering that $F_i(t_i) \geq F_j(t_i)$, it is clear that $C_i(S) \leq C_j(P(K, j))$ and $C_j(S) \leq C_i(P(K, j))$. The other jobs after job j can be scheduled earlier. The schedule $P(K, j)$ then is dominated according to Property 2.

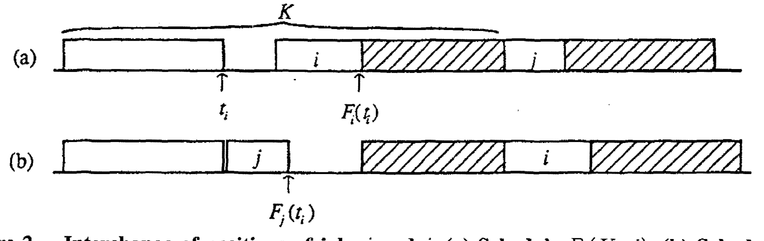

Figure 2. – Interchange of positions of jobs i and j. (a) Schedule $P(K, j)$. (b) Schedule S.

In the case where $p_i > p_j$, the jobs after j in $P(K, j)$ are delayed in S (see *Fig.* 2). This increases the completion time and consequently the tardiness of each of these jobs at most by $p_i - p_j$. The number of these jobs is $n - q(K) - 1$. On the other hand, the interchange of positions can reduce the tardiness of jobs i and j by $T_{ij} - T_{ji}$. From the assumption of Theorem 6, we have

$$T(S) - T(P(K, j)) \leq \{n - q(K) - 1\} \times (p_i - p_j) - \{T_{i,j} - T_{j,i}\} \leq 0.$$

As a consequence $P(K, j)$ is dominated. ■

THEOREM 7: *The schedule $P(K, i)$ is dominated if there is another partial solution K' such that $J(K') = J(K) \cup \{i\}$,*

$$\{n-q(K')\}\times R_j(E(K'))+T(K') \leq \{n-q(K')\}\times R_j(E(K/i))+T(K/i)$$

and $T(K') \leq T(K/i)$, where j is a job of $N - J(K')$ with the smallest release date.

The proof of Theorem 7 needs the following lemma that has to be proved first.

LEMMA 1: *Given a set W of jobs, if we construct two optimal schedules σ and σ' respectively starting from time t and t' $(t' \geq t)$, then*

$$T(\sigma') - T(\sigma) \leq \operatorname{card}(W)(t' - t).$$

Proof: We construct another schedule S starting from t' in which the jobs are scheduled in the same order as in σ, the completion time and consequently the tardiness of each job is increased at most by $t' - t$. We then have $T(S) - T(\sigma) \leq \operatorname{card}(W)(t' - t)$. In addition, from the definition of σ', we have $T(\sigma') \leq T(S)$ which implies $T(\sigma') - T(\sigma) \leq \operatorname{card}(W)(t' - t)$. ■

Proof of Theorem 7: Let $\sigma = P(K, i)$, $\delta = E(K/i)$ and $\delta' = E(K')$ to simplify the notations. Consider schedule σ and schedule S composed of K' and the optimal partial schedule of jobs $N - J(K')$. It is clear that in σ the jobs of $N - J(K')$ are optimally scheduled from time $R_j(\delta)$ and in S these jobs are optimally scheduled from time $R_j(\delta')$.

If $R_j(\delta') \leq R_j(\delta)$, it is obvious that σ is dominated by S considering the assumption $T(K') \leq T(K/i)$.

If $R_j(\delta') > R_j(\delta)$, the difference of total tardiness of the optimal partial schedules of jobs of $N - J(K')$ in S and σ is at most $\{n - q(K')\} \times \{R_j \delta') - R_j(\delta)\}$ according to Lemma 1. This implies that

$$T(S) - T(\sigma) \leq T(K') - T(K/i) + \{n - q(K')\} \times \{R_j(\delta') - R_j(\delta)\}.$$

Considering the assumption of Theorem 7 we have $T(S) \leq T(\sigma)$, which implies that schedule σ is dominated. ■

In a branch and bound algorithm, we also need a lower bound. In general, with unequal release dates and non preemptive jobs, the lower bound is obtained by relaxing the problem to a problem with preemptive jobs. We also use this scheme to compute lower bounds, because the problem with preemptive jobs, as shown in Section 2, is polynomially solvable. Note that this is no longer true for the related problem with specific due dates (*see* Chu 1992*a*).

Usually, a good initial solution can make a branch and bound algorithm more efficient. For this purpose, we use two heuristics called respectively EST (earliest start time) and ECT (earliest completion time) that we present first. These two heuristics have been reported to be efficient for total completion time minimization (Chandra, 1979; Dessouky and Deogun, 1981) which is also an over-regular criterion.

3.2. Heuristic algorithms

All the heuristics construct a new partial schedule K/i by adding an unscheduled job i behind the partial schedule K constructed before. K then becomes the new partial schedule. This continues untill all the jobs are scheduled. Initially, the partial schedule K is set to $\varnothing$. We now explain the choice of a job i.

Earliest Start Time (EST) Priority Rule

Select job $i \in N - J(K)$ with the smallest earliest start time $R_i(E(K)) \leq R_j(E(K))$ for all $j \in N - J(K)$. Break ties by choosing i with min (p_i), and further ties by choosing i with min (i). Schedule i and update K.

Earliest Completion Time (ECT) Priority Rule

Select job $i \in N - J(K)$ with the smallest earliest completion time $F_i(E(K)) \leq F_j(E(K))$ for all $j \in N - J(K)$. Break ties by choosing i with $\min\{R_i(E(K))\}$, and further ties by choosing i with min (i). Schedule job i and update K.

We can see that the ECT heuristic may insert idle times on the machine and the EST heuristic does not if some job is available at time $E(K)$. It should be noticed that both heuristics can be efficiently implemented in $O(n \log n)$.

3.3. The branch-and-bound algorithm

To construct an optimal schedule, we use a branch-and-bound technique. We first outline the algorithm and then present it in detail.

The algorithm uses a forward sequencing branching rule. Each node is defined by a partial sequence K of jobs scheduled from time zero. Initially, K is an empty set.

Each descendant node is obtained by adding, next to the partial sequence K, a new job i chosen among the unscheduled jobs. The node is really created only if $P(K, i)$ is not dominated according to the dominance theorems presented in Section 2 and Subsection 3.1.

If the release dates of *all* the unscheduled jobs are smaller than the completion time of K, the remaining problem becomes equivalent to a problem with identical release dates and can be solved by ordering the unscheduled jobs in the SPT order.

The lower bound for each descendant node is computed by relaxing the subproblem composed of unscheduled jobs into a problem with preemptive jobs and applying the SRPT rule to it. The initial feasible complete solution is obtained by taking the best one given by the EST and ECT heuristics. When a new feasible complete solution is obtained, and is better than the previous one, it is retained as the new best solution found so far. A descendant node is eliminated if its lower bound is greater than or equal to the total tardiness of the best solution found so far. We use a branch-and-bound with best first search policy, which means that we always select the node with the smallest lower bound for branching.

In order to present the algorithm in detail, we first introduce some notations. With each node h, is associated the following information:

K_h: Partial schedule;

J_h: Set of scheduled jobs in K_h;

q_h: Number of scheduled jobs in K_h;

δ_h: Completion time of the last job in K_h;

T_h: Total tardiness of scheduled jobs;

A_h: Set of unscheduled jobs ($A_h = N - J_h$);

b_h: Lower bound.

Other useful notations are the following:

$S*$: Best complete schedule found from the beginning of the algorithm;

Q: Queue of not yet considered nodes in non decreasing order of their lower bounds.

The detailed branch and bound algorithm is as follows.

Step 1. $Q := \emptyset$ and set S^* to the best solution given by the EST and ECT heuristics.

Step 2. Create the root node o with $K_o := \emptyset$, $J_o := \emptyset$, $A_o := N$, $T_o := 0$, $q_o := 0$, $\delta_o := 0$.

Step 3. Compute b_0 by relaxing the non preemption constraints and using the SRPT rule on jobs in A_0.

If $b_o = T(S^*)$, S^* is an optimal solution, STOP. Otherwise put o into the queue $Q := \{o\}$.

Step 4. If $Q = \emptyset$, S^* is an optimal solution, STOP. Otherwise set h to the head of the queue Q, and eliminate h from the queue, $Q := Q - \{h\}$.

Step 5. Create descendant nodes of h by performing following substeps.

5.1. Initialize the set of candidate jobs to be scheduled immediately after K_h, $W := A_h$.

5.2. If $W = \emptyset$, go to Step 4.

5.3. Choose a job $i \in W$ and eliminate it from W, $W := W - \{i\}$. Create a node h_i such that $K_{h_i} := K_h \circ i$, $J_{h_i} := J_h \cup \{i\}$, $A_{h_i} := A_h - \{i\}$, $q_{h_i} := q_h + 1$, $\delta_{h_i} := F_i(\delta_h)$, and $T_{h_i} := T_h + \max(\delta_{h_i} - d_{[q_{h_i}]}, 0)$.

5.4. Try to eliminate node h_i using Theorems 2-7. If node h_i is dominated, eliminate it and go to 5.2.

5.3. If $r_j \leq \delta_{h_i}$, for any job $j \in A_{h_i}$, go to 5.5.

5.4. Compute b_{h_i} by relaxing the non preemption constraints and using the SRPT rule on jobs in A_{h_i}.

If $b_{h_i} < T(S^*)$ put h_i into the queue, $Q := Q \cup \{h_i\}$, in non decreasing order of the lower bounds. Otherwise eliminate node h_i. Go to 5.2.

5.5. Schedule jobs in A_{h_i} in the SPT order, we obtain then a complete solution S. If $T(S) < T(S^*)$, set $S^* = S$. Go to 5.2.

The computational experiments about this branch and bound algorithm are reported in the next subsection.

3.4. Computational results

In order to evaluate the performance of the algorithm, we randomly generated a great number of examples on which the algorithm was applied. The generation of examples was carried out based on uniform probability distributions. For the generation of processing times and release dates of jobs, we used the same scheme proposed by Hariri and Potts (1983) for the minimization of total weighted completion time. The processing times were generated between 1 and 100, and the release dates were generated between 0 and $50.5 \times n \times \alpha$, α being a generation parameter. We propose to generate the generalized due dates in the following way. After having generated the release dates and processing times, we constructed a schedule σ by using the EST heuristic. The generalized due dates were generated between 0 and $\beta \times C_{[n]}(\sigma)$, β being another generation parameter. The motivation to generate the generalized due dates in this way is the fact that $C_{[n]}(\sigma)$ is a lower bound of the makespan in all feasible schedules (Rinnooy Kan, 1976, Theorem 4.6, p. 59). We carried out two series of experiments respectively with $n=50$ and $n=70$. For each series, the combination of 10 values of α (0.2, 0.4, 0.6, 0.8, 1.0, 1.25, 1.50, 1.75, 2.0, 3.0) and 4 values of β (0.5, 0.75, 1.0, 1.5) gave 40 sets of examples, each of which contained 20 randomly generated examples. In total 1 600 examples were generated. The experiments were carried out on a SUN Sparc 1+workstation.

Tables I and II respectively report the average number of nodes considered and the computation time (in CPU milliseconds) consumed by the algorithm for the series with $n=50$.

TABLE I
Average number of nodes considered ($n = 50$).

	α									
β	0.2	0.4	0.6	0.8	1.0	1.25	1.50	1.75	2.0	3.0
0.5	107.30	443.85	1 040.50	1 537.90	1 255.20	336.55	158.45	97.60	77.20	55.10
0.75	18.60	270.60	1 767.15	1 346.80	1 069.30	258.10	222.65	113.50	83.20	57.95
1.0	36.40	13.90	1 134.05	1 226.90	1 061.35	431.95	201.00	98.55	80.15	48.60
1.5	0.35	0.85	20.15	9.95	13.85	7.05	8.90	7.80	7.50	2.35

TABLE II
Average computation time ($n = 50$).

	α									
β	0.2	0.4	0.6	0.8	1.0	1.25	1.50	1.75	2.0	3.0
0.5	7 030	19 649	26 668	22 871	11 240	1 996	663	358	267	175
0.75	1 375	10 426	43 274	19 245	9 383	1 327	1 128	418	289	184
1.0	3 142	506	55 390	15 600	8 750	2 642	774	356	274	161
1.5	51	58	621	193	154	76	80	72	68	43

Tables III and IV are analogue to Tables I and II but for the series with $n=70$. From the tables, we can see that problems are considerably less difficult for extreme values of α. In fact, when α is small, the problem easily becomes a problem without release dates after having scheduled a few jobs, which is polynomially solvable. When α is large, the number of active schedules to be considered is very small.

TABLE III
Average number of nodes considered (n = 70).

	α									
β	0.2	0.4	0.6	0.8	1.0	1.25	1.50	1.75	2.0	3.0
0.5	377.20	1 802.80	4 636.60	4 444.65	4 405.75	1 337.35	520.35	254.35	157.55	90.65
0.75	14.80	304.95	4 947.30	4 226.65	3 133.10	863.65	380.60	523.30	164.95	91.65
1.0	3.15	24.40	188.65	2 126.80	1 895.55	1 208.00	881.60	369.40	157.60	84.75
1.5	0.70	2.40	5.50	19.54	24.40	14.90	10.60	10.50	7.30	4.10

TABLE IV
Average computation time (n = 70).

	α									
β	0.2	0.4	0.6	0.8	1.0	1.25	1.50	1.75	2.0	3.0
0.5	54 238	136 662	229 638	119 256	76 905	15 441	3 754	1 295	697	348
0.75	1 418	24 539	233 284	124 378	40 434	7 312	2 313	4 391	741	352
1.0	220	1 299	6 304	59 359	18 955	10 471	5 566	1 941	689	333
1.5	86	138	195	463	350	186	126	119	97	74

There is no simple relation between the hardness of the problem and the variation of β, except when β is very large. In this case the problem becomes very easy because of the large due dates and the large number of feasible solutions whose total tardiness is zero. Among the tested examples, some were so simple that very few branchings took place (only a small number of nodes were considered). The most difficult case seems to be the one with α=0.6 and β=0.75. We can also remark that the problem difficulty increases very quickly with n.

When n=50 and α=0.6, the average number of nodes considered in the case β=1.0 is less than in the case β=0.75, but the mean computation time is larger. This is because there is an example in the case β=1.0 for which the search tree is very large (18 458 nodes generated and the maximal length in the queue is 6 256 nodes), the algorithm had to take a lot of time to compare the lower bound of a new created node with the lower bound of existing nodes in the queue in order to keep the nodes in nondecreasing order of their lower bounds.

Table V gives general evaluation of the usefulness of different dominance properties. This evaluation is given in ratio between the total number of dominated nodes identified by each property and the total number of considered nodes. It should be noticed that some dominated nodes are identified by several properties, that is why we also give in Table VI the ratio between total number of dominated nodes identified *properly* by each property and the total number of considered nodes.

TABLE V
Performance of dominance theorems.

n	Th. 2	Th. 3	Th. 4	Th. 5	Th. 6	Th. 7
50	0.0039	22.4665	1.5345	23.3517	0.0075	2.1933
70	0.0025	34.3612	1.4760	34.8375	0.0071	1.9514

TABLE VI
Proper performance of dominance theorems.

n	Th. 2	Th. 3	Th. 4	Th. 5	Th. 6	Th. 7
50	0.0011	0.4375	0.1842	1.3652	0.0002	0.8434
70	0.0012	0.6912	0.2681	1.2453	0.0013	0.7420

It can be seen that Theorems 3 and 5 allow to eliminate a lot of dominated solutions. However, many dominated solutions are identified by both of them. This explains the important difference between Tables V and VI for these two theorems. It should be pointed out that Theorems 3, 5 and 7 often identify dominated solutions at the very beginnning of the search tree. This leads to a considerable reduction of the search tree.

One can also see that we can solve problems with larger size than for problems with specific due dates. The reason is twofold. Firstly, the problem becomes polynomially solvable when all unscheduled jobs become available, while this is not true for the problem with specific due dates. Another reason is that when the non preemption contraints are relaxed, the problem with generalized due dates can be solved polynomially while it remains NP-hard if the due dates are job specific. This gives rise to tighter lower bounds for problems with generalized due dates.

4. CONCLUDING REMARKS

In this paper, we defined a new class of criteria called over-regular criteria and established properties for these criteria. These properties are applicable for several scheduling problems and not specific for a particular problem as most work in the scheduling literature.

We also considered one of the over-regular criteria scheduling problems: minimizing total tardiness with generalized due dates. This problem is relevant in real life, especially in maintenance departments where maintained parts are interchangeable. For this problem, we proposed supplementary dominance properties and developed a branch-and-bound algorithm able to optimally solve problems with 70 jobs for the hardest case in our experiments.

ACKNOWLEDGEMENTS

The author is grateful to anonymous referees for their helpful comments on an earlier version of this paper.

REFERENCES

R. H. Ahmadi and U. Bagchi, Lower Bounds for Single-Machine Scheduling Problems, *Nav. Res. Logist.*, 1990, *37*, pp. 967-979.

K. R. Baker, *Introduction to Sequencing and Scheduling*, John Wiley & Sons, New York, 1974.

K. R. Baker and J. W. M. Bertrand, A Dynamic Priority Rule for Scheduling Against Due-Dates, *J. Opns. Man.*, 1982, *3*, pp. 37-42.

K. R. Baker and L. E. Schrage, Finding an Optimal Sequence by Dynamic Programming: An Extension to Precedence-Related Tasks, *Opns. Res.*, 1978, *26*, pp. 111-120.

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

H. Belouadah, M. E. Posner and C. N. Potts, Scheduling with Release Dates on a Single Machine to Minimize Total Weighted Completion Time, *Disc. Applied Math.*, 1992, *36*, pp. 213-232.

L. Bianco and S. Ricciardelli, Scheduling of a Single Machine to Minimize Total Weighted Completion Time Subject to Release Dates, *Nav. Res. Logist. Quart.*, 1982, *26*, pp. 111-126.

P. Bratley, M. Florian and P. Robillard, 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.*, 1973, *20*, pp. 57-67.

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

R. Chandra, On $n/1/\bar{F}$ Dynamic Deterministic Systems, *Nav. Res. Logist. Quart.*, 1979, *39*, pp. 265-283.

C. CHU, A Branch-and-Bound Algorithm to Minimize Total Tardiness with Different Release Dates, *Nav. Res. Logist.*, 1992*a*, *39*, pp. 265-283.

C. CHU, A Branch-and-Bound Algorithm to Minimize Total Flow Time with Unequal Release Dates, *Nav. Res. Logist.*, 1992*b*, *39*, pp. 859-875.

C. CHU, Efficient Heuristics to Minimize Total Flow Time with Release Dates, *Operations Research Letters*, 1992*c*, *12*, pp. 321-330.

C. CHU, *Scheduling to Minimize Over-Regular Criteria*, Rapport de Recherche No. 1704, INRIA, Rocquencourt, France, 1992*d*.

C. CHU and M. C. PORTMANN, Some New Efficient Methods to Solve the $n/1/r_i/\sum T_i$ Scheduling Problem, *Europ. J. Opnl. Res.*, 1992, *58*, pp. 404-413.

R. W. CONWAY, W. L. MAXWELL and L. W. MILLER, *Theory of Scheduling*, Addison-Wesley, Reading, MA, 1967.

J. S. DEOGUN, On Scheduling with Ready Times to Minimize Mean Flow Time, *The Computer Journal*, 1983, *26*, pp. 320-328.

M. I. DESSOUKY and J. S. DEOGUN, Scheduling Jobs with Unequal Ready Times to Minimize Mean Flow Time, *SIAM Journal of Computing*, 1981, *10*, pp. 192-202.

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

J. DU and J. Y.-T. LEUNG, Minimizing Total Tardiness on One Machine is NP-hard. *Math. Opns. Res.*, 1990, *15*, pp. 483-495.

H. EMMONS, One-Machine Sequencing to Minimize Certain Functions of Job Tardiness, *Opns. Res.*, 1969, *17*, pp. 701-715.

M. L. FISHER, A Dual Algorithm for the One-Machine Scheduling Problem, *Math. Prog.*, 1976, *11*, pp. 229-251.

S. FRENCH, *Sequencing and Scheduling: An introduction to the Mathematics of the Job-Shop*, John Wiley & Sons, New York, 1982.

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

N. G. HALL, Scheduling Problems with Generalized Due Dates, *IIE Transactions*, 1986, *18*, pp. 220-222.

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

N. G. HALL, S. P. SETHI and C. SRISKANDARAJAH, On the Complexity of Generalized Due Date Scheduling Problems, *Europ. J. Opnl. Res.*, 1991, *51*, pp. 100-109.

A. M. A. HARIRI and C. N. POTTS, An Algorithm for Single Machine Sequencing with Release Dates to Minimize Total Weighted Completion Time, *Discr. Appl. Math.*, 1983, *5*, pp. 99-109.

J. R. JACKSON, Scheduling a Production Line to Minimize Maximum Tardiness, *Research Report*, 1955, *43*, Management Science Research Project, University of California, Los Angeles.

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

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

E. L. LAWLER, A "Pseudopolynomial" Algorithm for Sequencing Jobs to Minimize Total Tardiness, *Ann. Disc. Math.*, 1977, *1*, pp. 331-342.

E. L. LAWLER, A Fully Polynomial Approximation Scheme for the Total Tardiness Problem, *Opns. Res. Lett.*, 1982, *1*, pp. 207-208.

J. LIU and B. L. MACCARTHY, Effective Heuristics for the Single Machine Sequencing Problem with Ready Times, *Int. J. Prod. Res.*, 1991, *29*, pp. 1521-1533.

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

C. N. POTTS, Analysis of a Heuristic for One Machine Sequencing with Release Dates and Delivery Times, *Opns. Res.*, 1980, *28*, pp. 1436-1441.

C. N. POTTS and L. N. VAN WASSENHOVE, A Decomposition Algorithm for the Single Machine Total Tardiness Problem, *Opns. Res. Lett.*, 1982, *1*, pp. 177-181.

A. H. G. RINNOOY KAN, *Machine Sequencing Problems: Classification, Complexity and Computation*, Nijhoff, The Hague, 1976.

J. SHWIMER, On the n-job, One-Machine, Sequence-Independent Scheduling Problem with Tardiness Penalties: a Branch-and-Bound Solution, *Man. Sci.*, 1972, *18*, pp. B301-B313.

V. SRINIVASAN, A Hybrid Algorithm for the One Machine Sequencing Problem to Minimize Total Tardiness, *Nav. Res. Logist. Quart.*, 1971, *18*, pp. 317-327.

L. J. WILKERSON and J. D. IRWIN, An Improved Algorithm for Scheduling Independent Tasks, *AIIE Transactions*, 1971, *3*, pp. 239-245.