IIE Transactions (2000) **32**, 1091–1101

TECHNICAL NOTE

An exact approach to minimizing total weighted tardiness with release dates

M. SELIM AKTURK* and DENIZ OZDEMIR

Department of Industrial Engineering, Bilkent University, 06533 Bilkent, Ankara, Turkey
E-mail: akturk@bilkent.edu.tr

Received July 1999 and accepted November 1999

The study deals with scheduling a set of independent jobs with unequal release dates to minimize total weighted tardiness on a single machine. We propose new dominance properties that are incorporated in a branch and bound algorithm. The proposed algorithm is tested on a set of randomly generated problems with 10, 15 and 20 jobs. To the best of our knowledge, this is the first exact approach that attempts to solve the $1|r_j|\sum w_jT_j$ problem.

1. Introduction

This study deals with scheduling a set of jobs with unequal release dates, r_j, on a single machine to minimize the total weighted tardiness, $(1|r_j|\sum w_jT_j)$. There are n independent jobs $1,\ldots,n$, each of which has an integer processing time p_j, a release date r_j, a due date d_j and a positive weight w_j. Jobs are processed without interruption on a single machine that can handle only one job at a time. The machine may be left idle while there are available jobs in the queue. A tardiness penalty T_j is incurred for each time unit that job j exceeds its due date, i.e., $T_j = \max\{0, (C_j - d_j)\}$, where C_j and T_j are the completion time and the tardiness of job j, respectively. The objective is to find a schedule that minimizes the total weighted tardiness criterion of all jobs given that no job can start processing before its release date.

Rinnooy Kan (1976) shows that the total tardiness problem with unequal release dates, $1|r_j|\sum T_j$ is NP-hard. Lawler (1977) shows that the total weighted tardiness problem, $1||\sum w_jT_j$, is strongly NP-hard, implying that $1|r_j|\sum w_jT_j$ is also strongly NP-hard. Enumerative solution methods have been proposed for both weighted and unweighted cases when all jobs are simultaneously available. Emmons (1969) derives several dominance rules for $1||\sum T_j$. Rinnooy Kan *et al.* (1975) and Rachamadugu (1987) extended these results to $1||\sum w_jT_j$. The Branch and Bound (BB) algorithm of Potts and van Wassenhove (1985) can solve problem instances with up to 40 jobs. Vairaktarakis and Lee (1995) present a BB algorithm to minimize the total tardiness subject to a minimum number of tardy jobs. Szwarc (1993) proves the existence of a special ordering for the single machine Earliness-Tardiness (E/T) problem with job-independent penalties where the arrangement of two adjacent jobs in an optimal schedule depends on their start time. Recently, Akturk and Yildirim (1998) proposed a new dominance rule and a lower bounding scheme for the $1||\sum w_jT_j$ problem which can be used to reduce the number of alternatives in any exact approach.

All the optimizing approaches discussed above assume that the jobs have equal release dates. To the best of our knowledge, we know of no exact algorithm for the $1|r_j|\sum w_jT_j$ problem. Unequal release dates have been considered for other optimality criteria, by Chu (1992a) and Chand *et al.* (1996) for $1|r_j|\sum F_j$, by Hariri and Potts (1983) and Beloudah *et al.* (1992) for $1|r_j|\sum w_jC_j$, and Potts and van Wassenhove (1988) for $1|r_j|\sum w_jU_j$. Chu (1992b) proves some dominance properties and provides a lower bound for the $1|r_j|\sum T_j$ problem. A BB algorithm is then constructed using the previous results of Chu and Portmann (1992) and problems with up to 30 jobs can be solved for certain problem instances, even though computational requirements for larger problems tend to become prohibitive.

In the following section, we discuss the underlying assumptions and present the proposed dominance rules. Lower bounds for the problem are developed in Section 3. We present a BB algorithm along with a numerical example in Section 4. Computational analysis of the BB

*Corresponding author

algorithm is reported in Section 5, and some concluding remarks are provided in Section 6.

2. Dominance properties

In this section we present new dominance properties to eliminate a number of dominated solutions in any exact algorithm. We show that the arrangement of adjacent jobs in an optimal schedule depends on their start times. For each pair of jobs i and j that are adjacent in an optimal schedule, there can be a critical value t_{ij} such that i precedes j if processing of this pair starts earlier than t_{ij} and j precedes i if processing of this pair starts after t_{ij}. For convenience the jobs are indexed in EDD order such that if $d_i < d_j$, or $d_i = d_j$ then $p_i < p_j$, or if $d_i = d_j$ and $p_i = p_j$ then $w_i > w_j$ or $d_i = d_j$ and $p_i = p_j$ and $w_i = w_j$ then $r_i \le r_j$ for all i and j such that $i < j$.

To introduce the dominance rule, consider schedules $S_1 = Q_1 ij Q_2$ and $S_2 = Q_1 ji Q_2$ where Q_1 and Q_2 are two disjoint subsequences of the remaining $n-2$ jobs. Let t be the completion time of the jobs in Q_1 and jobs i and j are available at t, such that $r_i \le t$ and $r_j \le t$.

The interchange function $\Delta_{ij}(t)$ gives the cost of interchanging adjacent jobs i and j whose processing starts at time t, where

$$\Delta_{ij}(t) = f_{ij}(t) - f_{ji}(t),$$

$$f_{ij} = \begin{cases} 0 & \max\{r_i, r_j, t\} \le d_i - (p_i + p_j), \\ w_i(t + p_i + p_j - d_i) & r_j \le t \text{ and } \\ & \quad d_i - (p_i + p_j) < t \le d_i - p_i, \\ w_i(r_j + p_j - t) & d_i - p_i \le t < r_j, \\ w_i(r_j + p_i + p_j - d_i) & t \le d_i - p_i \text{ and } t < r_j, \\ w_i p_j & \max\{r_j, d_i - p_i\} \le t. \end{cases}$$

$\Delta_{ij}(t)$ does not depend on how the jobs are arranged in Q_1 and Q_2 but on the start time t of the pair,

- if $\Delta_{ij}(t) < 0$ then, j should precede i at time t;
- if $\Delta_{ij}(t) > 0$ then, i should precede j at time t;
- if $\Delta_{ij}(t) = 0$ then, it is indifferent to whether i or j is scheduled first.

There are five conditions for the computation of f_{ij}. For the first condition, both jobs i and j finish on time, so it is indifferent on whether i or j is scheduled first. In the second condition, job i will become tardy if it is not scheduled first. In the third condition, job j arrives after time t and job i will be tardy if it is scheduled after job j (there is also an idle time on the machine before the beginning of job j). In the fourth condition, if job i is scheduled before job j then it can be finished on time, otherwise it will be tardy. In the last condition, job j arrives before time t, and job i will be tardy even if it is scheduled before job j.

The time dependent dominance properties of the $1|r_j|\sum w_j T_j$ problem can be determined by looking at points where the piecewise linear and continuous functions $f_{ij}(t)$ and $f_{ji}(t)$ intersect. When all possible cases are studied, it can be seen that there are at most seven possible points where functions $f_{ij}(t)$ and $f_{ji}(t)$ intersect. These cases and the following propositions are included here without proof and are described in detail in Akturk and Ozdemir (1998).

$$t_{ij}^1 = [(w_i d_i - w_j d_j)/(w_i - w_j)] - (p_i + p_j), \quad (1)$$

$$t_{ij}^2 = d_j - p_i - p_j(1 - w_i/w_j), \quad (2)$$

$$t_{ij}^3 = d_i - p_j - p_i(1 - w_j/w_i), \quad (3)$$

$$t_{ij}^4 = w_j/w_i(r_i + p_i + p_j - d_j) - (p_i + p_j - d_i), \quad (4)$$

$$t_{ij}^5 = [(w_j - w_i)p_i + w_j r_i + w_i(d_i - p_j)]/(w_i + w_j), \quad (5)$$

$$t_{ij}^6 = w_i/w_j(r_j + p_i + p_j - d_i) - (p_i + p_j - d_j), \quad (6)$$

$$t_{ij}^7 = [(w_i - w_j)p_j + w_i r_j + w_j(d_j - p_i)]/(w_i + w_j). \quad (7)$$

As a result, we can state a general rule that provides a sufficient condition for schedules that cannot be improved by adjacent job interchanges. We show that if any sequence violates the proposed dominance rule, then switching these jobs either lowers the total weighted tardiness or leaves it unchanged as stated below in Proposition 1. In this rule, there are two possibilities for each pair of jobs. Either there is at least one breakpoint or an unconditional ordering. A *breakpoint* is a critical start time for each pair of adjacent jobs after which the ordering changes direction such that if $t \le$ *breakpoint*, i precedes j, denoted by $i \prec j$, (or j precedes i) and then j precedes i, denoted by $j \prec i$, (or i precedes j). If i *unconditionally* precedes j, denoted by $i \to j$, then the ordering does not change, i.e., i always precedes j when they are adjacent, but this does not imply that an optimal sequence exists in which i precedes j.

Before defining the new dominance properties, we will present some definitions. Let J be the set of all jobs to be scheduled, $S(t)$ the set of jobs scheduled before time t, $A(t)$ the set of available unscheduled jobs at time t, i.e., $A(t) = \{i | r_i \le t\} - S(t)$, $B(t)$ the set of unavailable and unscheduled jobs at time t, i.e., $B(t) = \{k | r_k > t\}$, and $U(t)$ the set of unscheduled jobs at time t, i.e., $U(t) = (A(t) \cup B(t))$.

Proposition 1. *Let job k be the last scheduled job in the sequence at time t given that processing of job k starts at time $t - p_k$. For all unscheduled jobs $i \in U(t)$, if scheduling job i at time t violates the proposed dominance rule, i.e., $i \prec k$ at time $t - p_k$, then scheduling job i right after job k at time t will not lead to an optimal schedule.*

It is well-known that the Shortest Weighted Processing Time (SWPT) rule gives an optimal sequence for the $1||\sum w_j T_j$ problem when either all due dates are zero or all jobs are tardy, i.e., $t > \max_{i \in J}\{d_i - p_i\}$. Under this situation the problem reduces to the total weighted