7. Single machine problems

7.1 Polynomially solvable problems

7.1.1 Some $1|d_i|\overline{C}, f_{\max}$ problems

In this section we provide various results for single machine problems involving criteria $\overline{C}$ and f_{max}, where f_{max} refers to a maximum increasing function of completion times. We first focus on the $1|d_i|\epsilon(\overline{C}/L_{\max})$ problem since criterion L_{max} is a particular case of the function f_{max}. Next we briefly review the results available for the general $1|d_i|\epsilon(\overline{C}/f_{\max})$ problem.

The $1|d_i|\overline{C}, L_{\max}$ problem

The early paper met in the literature is due to [Smith, 1956], and deals with a particular case of the $1|d_i|\epsilon(\overline{C}/L_{\max})$ problem where $L_{max} = 0$ is imposed. The algorithm of [Smith, 1956] is extended to the $1|d_i|\epsilon(\overline{C}/L_{\max})$ problem by [Heck and Roberts, 1972], who propose an *a priori* algorithm. An *a posteriori* algorithm for this problem is provided by [VanWassenhove and Gelders, 1980]. This algorithm represents a major step in multicriteria scheduling and it has led to numerous similar, exact or heuristic, algorithms. Its principle is as follows. For a fixed value ϵ, a strict Pareto optimum is determined using a greedy algorithm which combines the rules SPT and EDD. The constraint $L_{max} \leq \epsilon$ is equivalent to imposing deadlines on jobs, and so, the algorithm proceeds backward, starting from the last position. At each position, a list of eligible jobs is calculated and among this one the SPT/EDD priority rule is applied to select the job to schedule. The next value ϵ is deduced from the built schedule. The *a posteriori* algorithm, denoted by EWG1, is presented in figure 7.1 and requires, as shown by Van Wassenhove and Gelders, $O(n^2 \log(n)\overline{p})$ time with $\overline{p} = \sum_{i=1}^{n} p_i$.

Example.
We consider a problem for which $n = 5$.

i	1	2	3	4	5
p_i	3	5	6	7	9
d_i	23	22	24	22	18

```
ALGORITHM EWG1
/* T is the set of jobs to schedule */
/* We assume that p_1 <= p_2 <= ... <= p_n */
Step 1:  /* Initialisation of the algorithm */
         ε = Σ_{i=1}^{n} p_i;
         /* Initialisation of the deadlines */
         ~d_i = d_i + ε, ∀i = 1,...,n;
         End=FALSE; E = ∅;
Step 2:  /* Computation of the set E */
         While (End=FALSE) Do
         | L = T; S = ∅;
         | /* We use a modified version of the rule SPT */
         | While ((End=FALSE) and (L ≠ ∅)) Do
         | | F = {J_i ∈ L / ~d_i >= Σ_{J_k ∈ L} p_k};
         | | If (F = ∅) Then End=TRUE;
         | | Else
         | | | Let J_i ∈ F be such that p_i = max_{J_k ∈ F}(p_k);
         | | | /* Break ties by choosing the job */
         | | | /* with the greatest due date */
         | | | S = {J_i}//S;
         | | | L = L - {J_i};
         | | End if;
         | End While;
         | If (L = ∅) Then
         | | E = E + {S};
         | | ε = L_max(S) - 1;
         | | ~d_i = d_i + ε, ∀i = 1,...,n;
         | | End=FALSE;
         | End If;
         End While;
Step 3:  Print E;
[VanWassenhove and Gelders, 1980]
```

Fig. 7.1. An a posteriori algorithm for the $1|d_i|\epsilon(\overline{C}/L_{max})$ problem

(i) $\epsilon = 30$, $\widetilde{d_i} = [53; 52; 54; 52; 48]^T$, End=FALSE and $E = \emptyset$.
(ii) $L = \{J_1, J_2, J_3, J_4, J_5\}$, $S_1 = \emptyset$.
$F = \{J_1, J_2, J_3, J_4, J_5\}$, $i = 5$, $S_1 = (J_5)$.
$F = \{J_1, J_2, J_3, J_4\}$, $i = 4$, $S_1 = (J_4, J_5)$.
$F = \{J_1, J_2, J_3\}$, $i = 3$, $S_1 = (J_3, J_4, J_5)$.
$F = \{J_1, J_2\}$, $i = 2$, $S_1 = (J_2, J_3, J_4, J_5)$.
$F = \{J_1\}$, $i = 1$, $S_1 = (J_1, J_2, J_3, J_4, J_5)$, End=TRUE, $L_{max}(S) = 12$ and $\overline{C}(S_1) = 76$.
$E = \{(J_1, J_2, J_3, J_4, J_5)\}$.
End=FALSE, $\epsilon = 11$, $\widetilde{d_i} = [34; 33; 35; 33; 29]^T$.