Division of Research
Graduate School of Business Administration
The University of Michigan

April 1985

TWO-STAGE APPROXIMATION HEURISTIC FOR THE SINGLE MACHINE SCHEDULING PROBLEM WHERE THE OBJECTIVE IS TO MINIMIZE WEIGHTED EARLINESS

Working Paper No. 424

Suresh Chand
Purdue University
and
Hans Schneeberger
The University of Michigan

ABSTRACT

In this paper we develop a two-stage approximation algorithm for the scheduling problem where the objective is to minimize weighted earliness subject to no tardy jobs. The research was motivated by the fact that the optimization problem is NP-hard and the only existing heuristic to this problem can yield very poor objective function values. The computational study revealed very promising results, both with respect to the quality of the solutions obtained and with respect to the computational burden.

1. Introduction

This paper develops a two-stage approximation algorithm for the single machine scheduling problem where the objective is to minimize weighted earliness. As we pointed out in an earlier paper [2], such problems arise in capacitated "just-in-time" environments.

Assume that the due dates for the jobs to be processed on a single processor are known and that the customers accept delivery of their jobs exactly at the specified due dates. The shop incurs a holding cost for early jobs; jobs are not allowed to be tardy. An ideal solution for such a scheduling problem would be to complete the jobs exactly at their due dates. However, if the shop capacity is tight and the due dates for several jobs are clustered together, then it may not be possible to schedule the jobs to finish exactly at their due dates. As a result, the shop may need to schedule some jobs early and incur a holding cost for earliness. Thus, the scheduling problem under consideration requires minimizing the total holding cost for early jobs (or the total weighted earliness) subject to no tardy jobs.

Let P_i, U_i, d_i, and C_i denote the processing time, weight, due date, and completion time, respectively, for job i. Let N be the number of jobs. The problem of finding the optimal values of $C_1, C_2, \ldots, C_N$ can be formulated as follows:

$$\text{Minimize} \quad \sum_{i=1}^{N} (d_i - C_i)U_i \tag{1}$$

$$\text{subject to} \quad C_i - d_i \leq 0 \quad \text{for } i=1,2,\ldots,N. \tag{2}$$

We will call this problem the Weighted Earliness Problem (WE-Problem). The problem with the added constraint that no machine idle time is allowed, that is:

$$C_i \leq \sum_{j=1}^{N} P_j$$

for i=1,2,...,N, will be referred to as the Constrained Weighted Earliness Problem (CWE-Problem). In both these problems and the rest of the paper, we assume that jobs are not allowed to be preempted.

Note that the scheduling criteria introduced above are not regular measures of performance (see Page 13 of Baker [1] for a definition of regular measures). Most scheduling research to date has considered only the problems with regular measures. For example, Baker [1] claims that nearly all important scheduling criteria use regular measures of performance. Thus, this paper develops a heuristic solution procedure for an important class of scheduling problems which does not use regular measures.

In the next section, we briefly review some of the results developed for the CWE- and the WE-Problem. In section three we develop a two-stage approximation algorithm for the WE-Problem and section four states the computational results. The paper concludes in section five.

2. Review of the CWE- and WE-Problem

Both the CWE- and the WE-Problem have been shown to be NP-hard; even for the special case where all the weights are equal, that is $U_1=U_2=U_3=....=U_N$, [4]. This implies that it is unlikely that these problems can ever be solved without explicitly or implicitly enumerating all possible feasible solutions. It is relatively easy to see that the CWE-Problem can be solved with any standard dynamic-programming algorithm. The most effective DP algorithm that can be used for this problem is probably the algorithm developed by Potts and van Wassenhoven to solve the single machine scheduling problem to minimize the total weighted completion time subject to no tardy jobs [3].

The WE-Problem is considerably more difficult. Chand and Schneeberger [2] developed a dynamic goal programming algorithm to solve this problem. Such an algorithm, however, is inherently very slow and the computational overhead required prohibits solving problems which consist of more than 10 to 15 jobs.

The heuristic developed by Smith to solve the weighted earliness problem subject to no tardy jobs [5], can be modified to solve the CWE-and the WE-Problem [2]. The performance of this heuristic varies considerably from possible best to possible worst cases. We developed several results to identify problem instances where the modified Smith-heuristic is guaranteed to provide optimal solutions to either the CWE- or the WE-Problem [2].

The computational study revealed that for some problem instances the modified Smith-heuristic can be more than 50 percent away from the optimal solution. In addition, the dynamic goal programming algorithm developed to solve the WE-Problem cannot find solutions to most problems with more than 15 jobs. Therefore an improved heuristic is needed which provides adequate objective function values and which is such that even large problems can be solved.

It should also be noted that an algorithm which solves the WE-Problem will also solve the CWE-Problem but not vice versa.

3. Two Stage Approximation Algorithm for the WE-Problem

In this section, we develop a two-stage approximation algorithm for the WE-Problem. The algorithm starts by scheduling the job in the last position first, then the job in the second-to-the-last position gets scheduled, and so on, until all the jobs are scheduled.

We will introduce the algorithm by first presenting a heuristic that solves the following restricted WE-Problem P1:

P1: Given a collection of jobs, B, a job $a\varepsilon B$, a job $b\varepsilon B$ and a value T, find a sequence that minimizes

$$\sum_{i\varepsilon B} (d_i - C_i)U_i$$

such that $C_i \leq T$, $C_i - d_i \leq 0$ for all $i\varepsilon B$; job b is in the last position, and job a is in the second-to-the-last position of the sequence.

Clearly problem P1 can be reduced to a WE-Problem where only the jobs in {B-a-b} are considered. The following heuristic for P1 draws upon ideas developed by Smith. Let a, b, B, and T be the parameters for the job in the second-to-last position, the job in the last position, the set of jobs to be scheduled, and the maximum completion time, respectively.

```
Heuristic for P1:  HP1(a,b,B,T)
INITIALIZATION:  Let S be the set of scheduled jobs:  $S = \phi$
                 Further let $T' = T$
      Schedule job b:  Set $C_b = \min\{d_b, T'\}$ and $T' = C_b - P_b$
      Schedule job a:  Set $C_a = \min\{d_a, T'\}$ and $T' = C_a - P_a$
STEP 1:  Schedule job j:  Set $C_j = \min\{d_j, T'\}$ and $T' = C_j - P_j$
    where j is such that
        $P_j/U_j = \min_{i|d_i \geq T',\ i\notin S} \{P_i/U_i\}$    if $\max_{i\notin S}\{d_i\} \geq T'$
        $P_j/U_j = \min_{i|d_i = \max_{k\notin S}\{d_k\}} \{P_i/U_i\}$    otherwise
```

STEP 2: $S=S\cup\{j\}$, Stop if $S=B$, otherwise go to Step 1.

Denote the objective function value for HP1(a,b,B,T) by VHP1(a,b,B,T). Next consider Problem P2:

P2: Given a collection of jobs B, a job $a\varepsilon B$ and a value T, find a sequence that minimizes

$$\sum_{i\varepsilon B} (d_i - C_i)U_i$$

such that $C_i \leqq T$, $C_i - d_i \leqq 0$, $C_i - p_i \geqq 0$ for all jobs $i\varepsilon B$; and job a is in the last or the second-to-the-last position.

Let VHP2(a,B,T) be the objective function value for an approximated optimal solution to P2 with parameters identifying the job in the-second-to-the last or the last position, the set of jobs to be scheduled, and the maximum completion time, respectively. VHP2(a,B,T) can be obtained by solving:

$$\mathrm{VHP2}(a,B,T) \quad = \quad \min_{i\varepsilon\{B-a\}\cup\phi}\{\overline{\mathrm{VHP1}}(a,i,B,T)\} \tag{3}$$

$$\text{where } \overline{\mathrm{VHP1}}(a,i,B,T) = \begin{cases} \mathrm{VHP1}(a,i,B,T) & \text{if } \min_{i\varepsilon B}\{C_i - P_i\} \geqq 0 \\ \\ +\infty & \text{otherwise.} \end{cases} \tag{4}$$

Hence, VHP2(a,B,T) can be obtained by computing $\overline{\mathrm{VHP1}}(a,i,B,T)$ for all $i\varepsilon\{B-a\}\cup\phi$ and finding the lowest of all these values. However, the search can be reduced considerably by making the following observation:

Let $X = i|d_i \leq \min\{T-P_a-\Delta,\ d_a-P_a\}$, and $i\varepsilon\{B-a\}\}$,

where $\Delta = \min_{i|d_i > T-P_a}\{P_i\}$.

Then: $\min_{i\varepsilon B} \overline{VHP1}(a,i,B,T > \min_{i\varepsilon\{B-a-X\}\cup\phi}\{\overline{VHP1}(a,i,B,T)\}$

which implies: $VHP2(a,B,T) = \min_{i\varepsilon\{B-a-X\}\cup\phi}\{\overline{VHP1}(a,i,B,T)\}$.

Note that due to (4), a solution with time zero infeasibility, that is,

$$\min_i\{C_i-P_i\} < 0$$

will always have an objective function value of $+\infty$.

We can now present the main routine of the two-stage approximation algorithm. The algorithm is based on the results of Chand and Schneeberger [2]. From all the jobs feasible to be scheduled in the last position we schedule the one with the lowest P_i/U_i ratio first, in the very last position of the sequence. Next, from all the jobs feasible to be scheduled in the second-to-the-last position we schedule the one with the lowest P_i/U_i ratio in this position. In this way we are guaranteed an optimal subsequence as long as

$$P_{[i]}/U_{[i]} \geq P_{[i+1]}/U_{[i+1]} \tag{5}$$

where $P_{[i]}$ denotes the processing time of the job in position i.

Let $S' = \{i,S\}$ and $S'' = \{j,i,S\}$ be two optimal subsequences, which satisfy condition (5). Consider the case where the next step in our algorithm is to schedule job h and we obtain the subsequence $S''' = \{h,j,i,S\}$ which violates the condition for an optimal subsequence, that is, we are not guaranteed optimality any longer. At this point we back off to the sequence S' and approximate the optimal decision concerning which job to schedule in front of S' by solving $P2(h,\{K-S'\},C_i-P_i)$, where K is the set of all jobs to the N-job WE-Problem. Let the optimal sequence to P2 be {a,..., x, y}. If y=h we schedule y such that we obtain the subsequence {y, i, S}. If x=h we schedule x and y such that we obtain the subsequence {x, y, i, S}. Then we proceed by scheduling the job with the lowest P_i/U_i ratio until the optimality condition mentioned earlier is violated again, at which time we will again solve P2. The steps of the algorithm are given below.

STEPS FOR THE TWO-STAGE APPROXIMATION ALGORITHM

INITIALIZATION Let K be the set of all jobs to the N-job WE-Problem.

Let S be the set of all scheduled jobs:

$S=\phi$

Let $T = \max_{i \notin K} \{d_i\}$

Let $l = N+1$

Let $X_{[j]}$ be the parameter X of the job in position j.

STEP 1: Schedule job j: $C_j=\min\{d_j,T\}$, $l=l-1$, $T=C_j-P_j$

$$\text{where j is such that} \begin{cases} P_j/U_j=\min\limits_{i|di \geq T \text{ and } i \notin S}\{P_i/U_i\} & \text{if } \max\limits_{i \notin S}\{d_i\} \geq T \\ \\ P_j/U_j=\min\limits_{i|d_i=\max\limits_{k \notin S}\{d_k\}}\{P_i/U_i\} & \text{otherwise} \end{cases}$$

```
STEP 2: If $S = K$ stop
        if $P_{[l]}/U_{[l]} \geq P_{[l+1]}/U_{[l+1]}$   go to Step 1.
        otherwise $S = \{S-[l]-[l+1]\}$
            $l = 1 + 2$
            $T = C_{[l]}-P_{[l]}$
            solve $P2(j,\{K-S\},t)$
            where j is the job found in the first step of the algorithm.
            Denote the job in the last position of the sequence that solves
            P2 by b and the job in the second-to-the-last position of this
            sequence by a.
        Schedule job b:
            set $C_b=\min\{d_b,T\}$, $l=l-1$, $T=C_b-P_b$
        if b=j go to Step 1.
        otherwise schedule job a:
            set $C_a=\min\{d_a,T\}$, $l=l-1$, $T=C_a-P_a$
            go to Step 1.
```

EXAMPLE

Consider the following five-job WE-Problem.

i	P_i	U_i	d_i
1	2	6	11
2	4	2	7
3	2	3	12
4	5	2	17
5	2	4	18

Solution

Step 1: Schedule job 5: $C_5 = 18$, $p_5 = 2$, $T = 16$

Step 2: go to Step 1.

Step 1: Schedule job 4: $C_4 = 16$, $p_4 = 5$, $T = 11$

Step 2: go to Step 1.

Step 1: Schedule job 1: $C_1 = 11$, $p_1 = 2$, $T = 9$

Step 2: Solve P2 with $a = 1$, $B = \{1,2,3,4\}$, $t = 16$

find VHP1(a,i,B,T) for $i = 4,3,\emptyset$.

The following sequence solves P2: {2-3-1-4}

Schedule job 4: $C_4 = 16$, $p_4 = 5$, $T = 11$

Schedule job 1: $C_1 = 11$, $p_1 = 2$, $T = 9$

go to Step 1.

Step 1: Schedule job 3: $C_3 = 9$, $p_3 = 2$, $T = 7$

Step 2: Stop

Note that for this example, the two-stage approximation algorithm found the optimal solution.

A simple single pass heuristic identical to the modified Smith-heuristic [2] can be constructed by simply restricting the algorithm to iterate between the first and the second step without ever solving the second-stage problem P2. We will include such a single pass heuristic in our computational tests.

4. Computational Study

The objective of the computational study in this section is to analyze and compare the performance of the developed Two-Stage Algorithm (TSA) with the Modified Smith-heuristic (MSH) and the Dynamic Goal Programming (DGP) algorithm [2].

The design of the experiment is identical to the one used in [2]. Hence the results obtained in this section can be compared to results obtained in the paper mentioned above. As in [2], the processing for all jobs is kept constant (P_i=5). The weights and due dates were generated by using three control parameters (α, β, and D) as described below. These parameters are allowed to assume the following values:

α = 0.0, 0.50, 1.0

β = 0.0, 0.2, 0.4, 0.6, 0.8, 1.0

D = 0, 25, 50, 75

α : Control parameter for the variance in the P_i/U_i - ratios. This parameter is used to control the variance in the Pi/Ui ratios. The variance increases with increasing α. The lower limit (α = 0) implies that the P_i/U_i is constant for all jobs. The modified Smith-heuristic will give an optimal sequence for α = 0 [2].

β : Control parameter for the correlation of P_i/U_i and d_i. This parameter is used to control the correlation between P_i/U_i-ratios and the due dates. If β is at its lower limit (β = 0) then $R_i < R_j \rightarrow d_i > d_j$. The modified Smith-heuristic will give an optimal sequence for β = 0 [2].

D : Control parameter for the due dates. The parameter D is used to control the due dates; a small value of D implies that the due dates are uniformly distributed over certain intervals of time and a large value of D means the due dates are clustered towards the end of the scheduling horizon. The upper limit for D will give identical due dates for all jobs. The modified Smith-heuristic will give an optimal sequence for D = 100 [2].

The due dates and weights were generated as follows. Note that N denotes the number of jobs.

i) $d_i = 5N + U(D, 100)$,

where U(a,b) denotes the uniform distribution between a and b. The term 5N has been added to guarantee feasibility. It is easy to see that all due dates are identical (equal to 5N + 100) for D = 100. For D = 0, the due dates are uniformly distributed between 5N and 5N + 100.

ii) $$\frac{P_i}{U_i} = 1 + \alpha \begin{Bmatrix} -1 \text{ if } x \leq 1.0 \\ 1 \text{ if } x < 1.0 \end{Bmatrix} \frac{d_i}{100 + 5N},$$

$$x = U(\beta, 1+\beta).$$

By using formula ii), we can generate weights (U_i's) while controlling the variance of P_i/U_i-ratios and also the correlation between P_i/U_i and d_i. For $\alpha = 0$, $p_i/U_i = 1$ for all jobs; the variance of P_i/U_i-ratios increases with increasing α. For $\beta = 0$, $x \leq 1$, and the correlation between P_i/U_i and d_i will be negative. Note that because $d_i \leq 5N + 100$, $P_i/U_i \geq 0$ for all jobs.

We used N = 10 and 50 jobs in this experiment. For each combination of parameter values, we generated 10 problems and solved each problem by using the DGP, MSH, and TSA for N = 10 and MSH and TSA for N = 50. In total we used 3x(5 x 6 x 7 x 10) = 1440 test problems.

Table 1.1 and Table 2.1 summarize the results with respect to the objective function value and Table 1.2 and Table 2.2 summarize the result with respect to the CPU-time.

It should be noted that the results obtained with TSA show a dramatic improvement over MSH with respect to objective function value. In addition the CPU-time for this procedure, although significantly larger than for MSH, still allows solving scheduling problems with 50 and more jobs.

5. Conclusion

In this paper we developed a two-stage approximation algorithm for the scheduling problem where the objective is to minimize weighted earliness subject to no tardy jobs. The research was motivated by the fact that the optimization problem is NP-hard and the only existing heuristic to this problem can yield very poor results. The computational study revealed very promising results, both with respect to the quality of the solutions obtained and with respect to the computational burden.

It should also be noted that the approach taken in the development of the two-stage approximation algorithm can be generalized to solve a variety of difficult optimization problems.

Table 1.1

Objective function value for N=10

D	α	β=0.0	β=0.2	β=0.4	β=0.6	β=0.8	β=1.0	Comments
average optimal solution value								
0	0.0	78.7	78.7	78.7	78.7	78.7	78.7	DGP
	0.5	60.7	62.9	70.8	77.8	85.4	117.2	
	1.0	49.8	55.8	83.9	100.6	132.7	762.6	
25	0.0	112.8	112.8	112.8	112.8	112.8	112.8	
	0.5	83.4	92.9	106.7	120.9	131.9	177.1	
	1.0	66.4	93.4	145.6	184.6	239.6	1146.4	
50	0.0	246.2	246.2	246.2	246.2	246.2	246.2	
	0.5	174.4	199.6	220.9	252.5	292.1	413.0	
	1.0	135.1	201.7	325.7	487.6	665.3	2498.7	
75	0.0	580.3	580.3	580.3	580.3	580.3	580.3	
	0.5	396.6	435.4	487.1	560.4	719.8	1052.8	
	1.0	301.3	451.8	901.8	1700.0	2913.8	8821.4	
average deviation from optimal [in %]								
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	MSH
	0.5	0.0	0.0	2.1	0.0	0.0	0.0	
	1.0	0.0	0.0	5.1	0.6	0.0	0.0	
25	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
	0.5	0.0	0.0	0.0	0.0	0.0	0.0	
	1.0	0.0	3.7	3.8	7.6	3.9	0.0	
50	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
	0.5	0.0	0.0	4.2	1.3	0.0	0.0	
	1.0	0.0	19.2	24.9	11.3	13.8	0.0	
75	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
	0.5	0.1	1.5	0.7	0.5	0.0	0.0	
	1.0	0.1	43.8	51.2	6.7	1.7	0.0	
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	TSA
	0.5	0.0	0.0	0.0	0.0	0.0	0.0	
	1.0	0.0	0.0	0.0	0.0	0.0	0.0	
25	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
	0.5	0.0	0.0	0.0	0.0	0.0	0.0	
	1.0	0.0	0.0	0.0	1.8	0.0	0.0	
50	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
	0.5	0.0	0.0	0.0	0.0	0.0	0.0	
	1.0	0.0	5.1	1.9	0.3	2.9	0.0	
75	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
	0.5	0.0	0.0	0.0	0.0	0.0	0.0	
	1.0	0.0	0.6	0.0	0.4	0.0	0.0	

Table 1.2

CPU-time for N=10 (in seconds)

D	α	β=0.0	β=0.2	β=0.4	β=0.6	β=0.8	β=1.0	Comments
	0.0	0.072	0.071	0.072	0.072	0.072	0.072	
0	0.5	0.055	0.058	0.061	0.063	0.063	0.067	
	1.0	0.056	0.058	0.062	0.063	0.063	0.067	
	0.0	0.171	0.172	0.171	0.173	0.173	0.172	
25	0.0	0.122	0.126	0.138	0.142	0.148	0.171	
	1.0	0.121	0.132	0.144	0.151	0.152	0.171	DGP
	0.0	0.729	0.746	0.737	0.739	0.742	0.737	
50	0.5	0.655	0.678	0.739	0.726	0.741	0.832	
	1.0	0.661	0.782	1.020	1.170	1.179	0.818	
	0.0	4.312	4.309	4.299	4.293	4.287	4.295	
75	0.5	3.998	5.310	5.437	5.541	5.263	4.178	
	1.0	4.026	6.202	7.041	7.433	6.626	4.163	
	0.0	0.006	0.006	0.006	0.006	0.006	0.006	
0	0.5	0.006	0.006	0.006	0.006	0.006	0.006	
	1.0	0.006	0.006	0.006	0.006	0.006	0.006	
	0.0	0.006	0.006	0.006	0.006	0.006	0.006	
25	0.5	0.006	0.006	0.006	0.006	0.006	0.006	
	1.0	0.006	0.006	0.006	0.006	0.006	0.006	MSH
	0.0	0.005	0.005	0.004	0.007	0.006	0.009	
50	0.5	0.005	0.005	0.004	0.008	0.007	0.005	
	1.0	0.006	0.005	0.005	0.007	0 008	0.005	
	0.0	0.005	0.005	0.005	0.005	0.005	0.005	
75	0.5	0.005	0.005	0.005	0.005	0.005	0.005	
	1.0	0.005	0.005	0.005	0.005	0.005	0.005	
	0.0	0.039	0.039	0.039	0.040	0.039	0.039	
0	0.5	0.084	0.073	0.066	0.058	0.052	0.039	
	1.0	0.084	0.072	0.065	0.058	0.052	0.039	
	0.0	0.030	0.030	0.030	0.030	0.031	0.030	
25	0.5	0.092	0.081	0.065	0.049	0.046	0.030	TSA
	1.0	0.093	0.082	0.065	0.049	0.045	0.030	
	0.0	0.017	0.018	0.018	0.014	0.020	0.015	
50	0.5	0.097	0.090	0.073	0.053	0.036	0.018	
	1.0	0.094	0.096	0.074	0.057	0.041	0.018	
	0.0	0.014	0.013	0.013	0.013	0.014	0.013	
75	0.5	0.106	0.080	0.056	0.028	0.022	0.014	
	1.0	0.107	0.077	0.059	0.029	0.023	0.013	

Table 2.1

Objective function value for N=50

D	α	β=.00	β=0.2	β=0.4	β=0.6	β=0.8	β=1.0	Comments
average solution value for MSH								
0	0.0	18435	18435	18435	18435	18435	18435	
	0.5	12741	12911	13833	18955	22398	30994	
	1.0	9738	1109	17695	44718	62876	120490	
25	0.0	21482	21482	21482	21482	21482	21482	
	0.5	14706	14899	16525	23276	27321	37594	
	1.0	11182	13103	25806	71673	98692	186813	
50	0.0	24530	24530	24530	24530	24530	24530	
	0.5	16639	17040	19793	28205	32984	44774	
	1.0	12590	18113	48236	129018	174481	318459	
75	0.0	27577	27577	27577	27577	27577	27577	
	0.5	18542	19225	22924	33227	38781	52622	
	1.0	13967	29449	105146	292064	391947	717399	
average improvement by using TSA [in %]								
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
	0.5	0.0	0.0	0.2	0.4	0.0	0.0	
	1.0	0.0	1.4	9.4	6.1	1.6	0.0	
25	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
	0.5	0.0	0.0	0.0	0.4	0.0	0.0	MSH
	1.0	0.0	0.9	3.1	6.0	1.0	0.0	=100%
50	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
	0.5	0.0	0.2	1.0	1.5	0.7	0.0	
	1.0	0.0	6.2	12.0	8.0	2.3	0.0	
75	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
	0.5	0.0	0.0	0.5	1.1	0.0	0.0	
	1.0	0.0	11.1	8.2	6.7	1.1	0.0	

Table 2.2

CPU-time for N=50 (in seconds)

D	α	β=0.0	β=0.2	β=0.4	β=0.6	β=0.8	β=1.0	Comments
	0.0	0.088	0.089	0.089	0.088	0.089	0.890	
0	0.5	0.083	0.083	0.083	0.083	0.084	0.083	
	1.0	0.083	0.084	0.083	0.084	0.083	0.083	
	0.0	0.090	0.090	0.090	0.090	0.090	0.089	
25	0.5	0.084	0.084	0.083	0.083	0.082	0.083	
	1.0	0.084	0.083	0.084	0.085	0.083	0.083	MSH
	0.0	0.089	0.090	0.089	0.089	0.090	0.090	
50	0.5	0.082	0.077	0.078	0.078	0.078	0.077	
	1.0	0.076	0.074	0.076	0.076	0.072	0.076	
	0.0	0.085	0.084	0.088	0.085	0.087	0.084	
75	0.5	0.075	0.076	0.075	0.076	0.074	0.075	
	1.0	0.076	0.078	0.075	0.077	0.076	0.074	
	0.0	0.182	0.183	0.182	0.181	0.183	0.183	
0	0.5	23.450	12.157	2.324	0.718	0.636	0.172	
	1.0	23.419	13.032	2.633	0.762	0.674	0.170	
	0.0	0.185	0.183	0.185	0.184	0.183	0.184	
25	0.5	21.926	9.087	0.770	0.654	0.567	0.170	
	1.0	22.098	9.883	0.798	0.664	0.557	0.172	TSA
	0.0	0.185	0.183	0.183	0.182	0.184	0.184	
50	0.5	17.333	2.682	0.739	0.588	0.510	0.157	
	1.0	15.887	2.701	0.784	0.634	0.556	0.156	
	0.0	0.174	0.173	0.174	0.171	0.172	0.174	
75	0.5	10.158	1.165	0.528	0.454	0.374	0.158	
	1.0	10.073	1.203	0.527	0.453	0.375	0.157	

REFERENCES

Baker, K. R., Introduction to Sequencing and Scheduling, John Wiley and Sons, Inc., 1974.

Chand, S. and H. Schneeberger, "Single Machine Scheduling to Minimize Weighted Earliness Subject to No Tardy Jobs," Working Paper, Purdue University.

Potts, C. N. and L. N. van Wassenhoven, "An Algorithm for Single Machine Sequencing with Deadlines to Minimize Total Weighted Completion Time," European Journal of Operations Research, No. 12 (1983), pp. 379-387.

Schneeberger, H. M., "Job Shop Scheduling in Pull Type Production Environments," Ph.D. Thesis, Krannert Graduate School of Management, Purdue University, West Lafayette, Indiana, May 1984.

Smith, W. E., "Various Optimizers for Single Stage Production," Naval Research Logistics Quarterly, Vol. 3, No. 1 (1956), pp. 56-66.