# RELATIONSHIP BETWEEN THE MODIFIED DUE DATE RULE AND THE HEURISTIC OF WILKERSON AND IRWIN

## J.C. NYIRENDA
**University of Cape Town, Department of Statistical Sciences,
Rondebosch 7701, Cape Town, South Africa**

**ABSTRACT**

In this paper, we consider the problem of scheduling $N$ jobs on a single machine to minimise total tardiness. Both the modified due date (**MDD**) rule and the heuristic of Wilkerson and Irwin (**W-I**) are very effective in reducing total tardiness. We show that in fact the **MDD** rule and the **W-I** heuristic are strongly related in the sense that both are based on the same local optimality condition for a pair of adjacent jobs, so that a sequence generated by these methods cannot be improved by any further adjacent pair-wise interchange.

**Keywords**: Scheduling, heuristic, tardiness

## 1.    INTRODUCTION

There is a plethora of methods, both simple and complex, that have been developed to solve the total tardiness problem. Among the simple rules, the modified due date (**MDD)** rule due to Baker and Bertrand (1982) and the heuristic of Wilkerson and Irwin (**W-I**) (1971) have been found to be very effective in reducing total tardiness and in some cases have been found to produce optimal tardiness schedules (see Baker and Bertrand (1982) and Baker and Kanet (1983)). The W-I heuristic is an example of a pair-wise interchange technique and is based on conditions of local optimality.  Thus, any sequence that satisfies these conditions cannot be improved by adjacent pair-wise interchange. In this paper, we show that the **MDD** rule is in fact *strongly related* to the heuristic of Wilkerson and Irwin (**W-I)** in that it also generates a sequence that satisfies the local optimality conditions. This is a useful result since it demonstrates that these two "classical" techniques are much more similar than has generally been recognized.

Rachamadugu (1987) has presented local pair-wise optimality conditions for the general weighted tardiness problem. He has also shown that the **MDD** rule is a special case of the optimality conditions. By showing the strong relationship that exists between the **W-I** heuristic and the **MDD** rule we note that, as indicated by Rachamadugu's results, at least one optimal sequence will satisfy the **MDD** conditions and thus the **W-I** heuristic conditions.

The balance of the paper is organized as follows. In the next section, we describe the **MDD** rule and the **W-I** heuristic in detail and in section 3 we present the proof of the relationship between them. This is followed in section 4 with the presentation and discussion of some experimental results while section 5 concludes the paper.

## 2.    RELATIONSHIP BETWEEN THE MDD RULE AND THE W-I HEURISTIC

The scheduling problem considered is as follows. There is a set $Z=\{1,2,...,N\}$ of jobs simultaneously available at time zero, to be processed on a single machine under the common assumptions listed in Baker (1974). Job $i$ requires $t_i$ processing time, $\forall i \in Z$, on the machine which can only process one job at a time without pre-emption and inserted idle time. In addition, each job $i$ is given a due date $d_i$, $\forall i \in Z$. If job $i$ is finished late, a penalty equal to its tardiness $T_i$ $=\max(C_i - d_i, 0)$ will be incurred where $C_i$ is the completion time of job $i$. The objective is to find the optimal job sequence $\sigma^*$ that minimises the total tardiness, that is

$$f(\sigma^*) = \min_{\sigma \in \pi}(f(\sigma)) \tag{1}$$

where $f(\sigma) = \sum \max(C_{[i]} - d_{[i]}, 0)$, ($i$=1,2,...,$N$); $\sigma$ is an arbitrary job sequence; the subscript [$i$] denotes the job in position $i$ of $\sigma$; $\pi$ is the set of all possible sequences of which there are $N!$.

**Wilkerson-Irwin Heuristic:**

Wilkerson and Irwin (1971), using dominance conditions produced by Emmons (1969), developed an effective heuristic to solve the total tardiness problem in (1). The **W-I** heuristic is an example of an interchange procedure. The **W-I** heuristic uses a decision rule that employs a pair-wise job comparison in the construction of a sequence. The decision rule can be stated as follows: for any given pair of adjacent jobs $i$ and $j$ that can start after time $T$, schedule the jobs according to the earliest due date (**EDD**) rule if

$$T + \max\{t_i, t_j\} < \max\{d_i, d_j\} \qquad (2)$$

otherwise schedule the jobs according to the shortest processing time (**SPT**) rule if

$$T + \max\{t_i, t_j\} \geq \max\{d_i, d_j\} \qquad (3)$$

The time $T$ depends on the partial job sequence established prior to the start of job $i$ or job $j$. Notice that our form of representing the heuristic is slightly different from that used by others (see Baker (1974)). Thus, instead of using the '$\leq$' sign in (2) and the '$>$' in (3) the '$<$' and the '$\geq$' sign have been applied respectively. Our form ensures that when $T + \max[t_i, t_j] = \max[d_i, d_j]$, a short job is the one that is assigned the next position in the schedule. In the conventional form, a long job may as well be preferred to a short job. In appendix A, we show that scheduling a long job in this way may cause an increase in the tardiness of the schedule.

To implement the **W-I** heuristic two ordered lists are used. The first is a list of jobs that have been scheduled. It is in fact a partially completed job sequence and is subject to possible rescheduling. The second is the unscheduled list which contains the remaining jobs in earliest due date (**EDD**) order so that $d_{[1]} \leq ... \leq d_{[R]}$, and $t_{[i]} \leq t_{[1+i]}$ whenever $d_{[i]} = d_{[1+i]}$ where ($i = 1, 2, ..., R-1$), and $1 < R \leq N$. At each stage, the first job on the unscheduled list is removed to act as a pivot job in order to implement the decision rule. In practice, the heuristic is used as an algorithm the steps of which have been described in Baker (1974). For the sake of completeness we shall repeat the steps of the heuristic below. Following Baker's approach we shall let

$\alpha$=the index of the last job on the scheduled list

$\beta$=the index of the pivot job

$\gamma$=the index of the first job on the unscheduled list

Adopting our form of representing the **W-I** heuristic we can now repeat the steps of the algorithm as follows.

### *Step 1*

(Initialisation) Put all the jobs on the unscheduled list in **EDD** order. Let $a$ and $b$ denote the first two jobs on this list. If $\max\{t_a, t_b\} < \max\{d_a, d_b\}$, then assign the first position in the scheduled sequence to the job with the earliest due date; otherwise assign the first position in the sequence to the shorter job. The assigned job becomes $\alpha$ and the other job becomes $\beta$, the pivot job. $T$ becomes equal to the completion time of job $\alpha$.

*Step 2*

If $T + \max\{t_\beta, t_\gamma\} < \max\{d_\beta, d_\gamma\}$ or if $t_\beta \leq t_\gamma$, then, add job $\beta$ to the scheduled list. Job $\beta$ now becomes job $\alpha$; job $\gamma$ is removed from the unscheduled list and now becomes job $\beta$; and the next job on the unscheduled list now becomes job $\gamma$. Repeat Step 2 unless the unscheduled list is empty, in which case add job $\beta$ to the scheduled list and stop. If, on the other hand, $T + \max\{t_\beta, t_\gamma\} \geq \max\{d_\beta, d_\gamma\}$ and $t_\beta > t_\gamma$, then, return job $\beta$ to the unscheduled list and let job $\gamma$ now become job $\beta$. Proceed to Step 3.

*Step 3*

If $T - t_\alpha + \max\{t_\alpha, t_\beta\} < \max\{d_\alpha, d_\beta\}$ or if $t_\alpha \leq t_\beta$, then, add job $\beta$ to the scheduled list. Job $\beta$ now becomes job $\alpha$; the first unscheduled job now becomes the pivot job; and the next job on the unscheduled list now becomes job $\gamma$. Go to Step 2. If, on the other hand, $T - t_\alpha + \max\{t_\alpha, t_\beta\} \geq \max\{d_\alpha, d_\beta\}$ and $t_\alpha > t_\beta$, then, a jump condition results. Go to Step 4.

*Step 4*

(Jump condition) Remove job $\alpha$ from the scheduled list and return it to the unscheduled list in **EDD** order. If jobs remain on the scheduled list, the last remaining job now becomes job $\alpha$. Return to Step 3. If there are no jobs on the scheduled list, job $\beta$ is assigned first position on the scheduled list and becomes job $\alpha$; the first job on the unscheduled list now becomes the pivot job; and the next job on the unscheduled list now becomes job $\gamma$. Go to Step 2.

This quite complicated definition has a relatively straightforward result, which we call the **W-I** condition: for adjacent jobs $i$ and $j$ in the final sequence, (with $i$ before $j$), if $T + \max\{t_i, t_j\} < \max\{d_i, d_j\}$ then $d_i \leq d_j$ (i.e. **EDD** ordering) and if $T + \max\{t_i, t_j\} \geq \max\{d_i, d_j\}$ then $t_i \leq t_j$ (i.e. **SPT** ordering).

**MDD Rule:**

Baker and Bertrand (1982) have also developed an effective heuristic - the **MDD** rule - to solve the problem in (1). Unlike the heuristic of Wilkerson and Irwin, the **MDD** rule is a construction heuristic. It is interesting to note, though, that Smith (1956) appears to have been the first to formulate the **MDD** rule; Baker and Bertrand do not seem to have been familiar with this work.

Further, it should be noted that while the **W-I** heuristic has been used mainly to solve deterministic static problems, the **MDD** rule is mainly associated with dynamic scheduling problems. As all jobs are available at time $T=0$ in a static deterministic case, the **MDD** rule can be described as follows. For any given pair of jobs $i$ and $j$ which can start processing after time $T$, schedule job $i$ first followed by job $j$ if

$$\max\{d_i, T + t_i\} \le \max\{d_j, T + t_j\} \tag{4}$$

In practice, jobs are selected for scheduling based on a priority index. For the **MDD** rule the priority index for a job $k$, which can start processing at time $T$, $\text{PI}_k$ is given as

$$\text{PI}_k = \max\{d_k, T + t_k\} \tag{5}$$

where $d_k$ and $t_k$ are the original due date and the processing time of job $k$ respectively. The index, $PI_k$, is known as the modified due date of job $k$ and is defined as the larger of the original due date of job $k$ and its earliest completion time. Among all the unscheduled jobs, the job with the smallest value of $\text{PI}_k$ is the one that is selected for scheduling.

## 3. PROOF OF RELATIONSHIP BETWEEN MDD AND W-I HEURISTIC

**Proposition 1.** *For adjacent jobs $i$ and $j$ in a schedule, the relationship (4) of the MDD rule is equivalent to the W-I condition.*

**Proof.** We will first demonstrate that the **W-I** heuristic implies the **MDD** rule and then that the **MDD** rule implies the **W-I** heuristic. We shall use the notation $p \leftarrow q$ to denote that job $p$ immediately precedes job $q$.

The **W-I** heuristic states that if

$$T + \max\{t_i, t_j\} < \max\{d_i, d_j\}$$

then use the **EDD** to sequence jobs $i$ and $j$. Suppose

$T + \max\{t_i, t_j\} < d_i = \max\{d_i, d_j\}$.

So **W-I** implies $j \leftarrow i$ and we know that $d_i$ is greater than each of $T + t_i$, $T + t_j$ and $d_j$. Hence,

$$\max\{d_j, T + t_j\} < \max\{d_i, T + t_i\}.$$

and so **MDD** yields $j \leftarrow i$.

Similarly, if

$$T + \max\{t_i, t_j\} < d_j = \max\{d_i, d_j\}$$

then **W-I** yields $i \leftarrow j$. We also know that $d_j$ is greater than each of $T + t_i$, $T + t_j$ and $d_i$. Hence,

$$\max\{d_i, T + t_i\} < \max\{d_j, T + t_j\}$$

and so **MDD** yields $i \leftarrow j$. Notice that if

$$T + \max\{t_i, t_j\} < D$$

such that $D = d_i = d_j$, then both the **W-I** and **MDD** use **SPT** to sequence jobs $i$ and $j$.

The **W-I** heuristic also states that if

$$T + \max\{t_i, t_j\} \geq \max\{d_i, d_j\}$$

then use **SPT** to sequence jobs $i$ and $j$. Suppose

$$T + \max\{t_i, t_j\} = T + t_i \geq \max\{d_i, d_j\}.$$

**W-I** then implies $j \leftarrow i$ and we know that $T + t_i$ is greater than $T + t_j$, and greater than or equal to each of $d_i$ and $d_j$. Hence,

$$\max\{d_j, T + t_j\} \leq \max\{d_i, T + t_i\}$$

and so **MDD** yields $j \leftarrow i$. Similarly, if

$$T + \max\{t_i, t_j\} = T + t_j \geq \max\{d_i, d_j\}$$

then **W-I** yields $i \leftarrow j$. We know that $T + t_j$ is greater than $T + t_i$, and greater than or equal to each of $d_i$ and $d_j$. Hence,

$$\max\{d_i, T + t_i\} \leq \max\{d_j, T + t_j\}$$

and so **MDD** yields $i \leftarrow j$.

Therefore **W-I** implies **MDD**.

Suppose that

$$\max\{d_i, T + t_i\} < \max\{d_j, T + t_j\} = d_j.$$

So **MDD** implies $i \leftarrow j$ and $d_j$ is greater than each of $d_i$, $T + t_i$ and $T + t_j$. Hence,

$$T + \max\{t_i, t_j\} < d_j = \max\{d_i, d_j\}$$

which implies that **W-I** yields $i \leftarrow j$, the **EDD** sequence for those two jobs.

Now suppose that

$$\max\{d_i, T + t_i\} < \max\{d_j, T + t_j\} = T + t_j.$$

Then **MDD** implies $i \leftarrow j$ and $T + t_j$ is greater than each of $d_i$, $d_j$ and $T + t_i$. Hence,

$$T + \max\{t_i, t_j\} = T + t_j > \max\{d_i, d_j\}$$

which implies that **W-I** yields $i \leftarrow j$, the **SPT** sequence for those two jobs. Notice that if

$$\max\{d_i, T + t_i\} = \max\{d_j, T + t_j\}$$

then both the **MDD** and **W-I** use **SPT** to sequence jobs $i$ and $j$.

Similarly, suppose that

$$\max\{d_i, T + t_i\} = d_i > \max\{d_j, T + t_j\}.$$

So **MDD** implies $j \leftarrow i$ and $d_i$ is greater than each of $T + t_i$, $T + t_j$ and $d_j$. Hence,

$$T + \max\{t_i, t_j\} < d_i = \max\{d_i, d_j\}$$

which implies that **W-I** yields $j \leftarrow i$, the **EDD** sequence for those two jobs.

Now suppose that

$$\max\{d_i, T + t_i\} = T + t_i > \max\{d_j, T + t_j\}.$$

Then **MDD** implies $j \leftarrow i$ and $T + t_i$ is greater than each of $d_i$, $d_j$ and $T + t_j$. Hence,

$$T + \max\{t_i, t_j\} = T + t_i > \max\{d_i, d_j\}$$

which implies that **W-I** yields $j \leftarrow i$, the **SPT** sequence for those two jobs.

Therefore the **MDD** implies the **W-I**. ∎

We wish to point out here that the proof we have given above serves only to show that the two procedures are based on the same local optimality condition. It does not simultaneously prove that the mechanisms of the two procedures for generating a tardiness schedule have exactly the same result so that for the same problem, the two heuristics will always generate the same tardiness schedule. On the other hand, experimental results presented below appear to indicate that the two procedures may in fact be exactly equivalent. However, we have been unable to prove that this is the case.

## 4. EXPERIMENTAL RESULTS

Two experiments each involving 20 problems were conducted in order to compare the performance of the **MDD** rule and the **W-I** heuristic on total tardiness. In the first experiment, each of the 20 problems had 50 jobs, and in the second, each problem had 100 jobs. The 20 problems in each case were generated in the manner suggested by Potts and Van Wassenhove (1987), as follows. First, for each job $i$, ($i=1,2,...,N$), an integer processing time, $t_i$, is generated

from a uniform distribution [1, $N$], and then the total processing time, $P = \sum t_i$, ($i$=1,2,...,$N$) is computed. $N$ is the total number of jobs in the problem. Thus, $N$=50 and $N$=100 for problems in the first and second experiments respectively. The relative range of due dates (RDD) and the average tardiness factor (TF) were selected from the sets {0.2,0.4,0.6,0.8,1.0} and {0.2,0.4,0.6,0.8} respectively, and for each job $i$ a due date, $d_i$, was generated from a uniform distribution as [P(1-TF-RDD/2), P(1-TF+RDD/2)]. Fisher (1976), Schrage and Baker (1978), and Potts and Van Wassenhove (1982) have observed that problems with TF=0.6 and TF=0.8 appear to be hardest to solve particularly when RDD is small.

The results in Table 1 compare the relative performance of the **MDD** rule and the heuristic of Wilkerson and Irwin in reducing total tardiness with respect to problems with 50 and 100 jobs respectively. We can see that the performance of the two procedures in reducing tardiness is same for all combinations of RDD and TF values in both experiments. This suggests that the two methods may in fact be exactly equivalent. We also observed that while the **MDD** rule took the same computation time to solve each of the 20 problems with the same number of jobs, the computation time taken by the **W-I** heuristic was variable. In general, it took less time than the **MDD** rule to solve problems with TF=0.2, and took substantially more time than the **MDD** rule to solve problems with TF between 0.4 and 0.8 inclusive.

**Table 1:**     **Tardiness values obtained for MDD and W-I heuristics for problems with 50 and 100 jobs.**

| | | TOTAL TARDINESS | | | |
|---|---|---|---|---|---|
| RDD | TF | WITH 50 JOBS | | WITH 100 JOBS | |
| | | **MDD** | **W-I** | **MDD** | **W-I** |
| 0.2 | 0.2 | 430 | 430 | 2921 | 2921 |
| 0.2 | 0.4 | 3021 | 3021 | 22195 | 22195 |
| 0.2 | 0.6 | 7770 | 7770 | 57772 | 57772 |
| 0.2 | 0.8 | 13961 | 13961 | 108443 | 108443 |
| 0.4 | 0.2 | 45 | 45 | 28 | 28 |
| 0.4 | 0.4 | 2208 | 2208 | 14195 | 14195 |

| | | | | | |
|---|---|---|---|---|---|
| 0.4 | 0.6 | 6845 | 6845 | 48202 | 48202 |
| 0.4 | 0.8 | 13907 | 13907 | 102808 | 102808 |
| 0.6 | 0.2 | 0 | 0 | 0 | 0 |
| 0.6 | 0.4 | 1361 | 1361 | 7748 | 7748 |
| 0.6 | 0.6 | 6079 | 6079 | 40828 | 40828 |
| 0.6 | 0.8 | 14483 | 14483 | 105367 | 105367 |
| 0.8 | 0.2 | 0 | 0 | 0 | 0 |
| 0.8 | 0.4 | 641 | 641 | 1788 | 1788 |
| 0.8 | 0.6 | 5930 | 5930 | 36806 | 36806 |
| 0.8 | 0.8 | 15179 | 15179 | 109626 | 109626 |
| 1.0 | 0.2 | 0 | 0 | 0 | 0 |
| 1.0 | 0.4 | 133 | 133 | 0 | 0 |
| 1.0 | 0.6 | 6412 | 6412 | 39447 | 39447 |
| 1.0 | 0.8 | 15926 | 15926 | 114495 | 114495 |

## 5    CONCLUSION

In this paper we have studied the single machine total tardiness problem. We have shown that the **W-I** heuristic and the **MDD** rule are strongly related, and experimental evidence appears to suggest that the two methods are exactly equivalent so that for each tardiness problem, they will always arrive at the same result. However, we have been unable to prove  that this is indeed the case. Nevertheless, the above findings partly accounts for the outstanding performance of the **MDD** rule reported in the literature. We have also noted that the **W-I** heuristic is a special case of the Rachamadugu's local optimality condition for the general weighted tardiness problem.

**REFERENCES**
[1]    K.R. BAKER, *Introduction to Sequencing and Scheduling*. Wiley, New York (1974).

[2]    K.R. BAKER, and J.W.M. BERTRAND, , A dynamic priority rule for scheduling against due dates, *Journal of Operations Management*, **3**, 37-42 (1982).

[3]    K.R. BAKER and J.J. KANET, Job shop scheduling with modified due dates, *Journal of Operations Management*, **4**, 11-22 (1983).

[4]   H. EMMONS, One-machine sequencing to minimise certain functions of job tardiness, *Operations Research*, **17**, 701-715 (1969).

[5]   M.L. FISHER, A dual algorithm for the one machine scheduling problem, *Mathematical Programming*, **11**, 229-251 (1976).

[6]   C.N. POTTS and L.N. VAN WASSENHOVE, A decomposition algorithm for the single machine total tardiness problem, *Operations Research Letters*, **1**, 177-181 (1982).

[7]   R.M.V. RACHAMADUGU, A note on the weighted tardiness problem, *Operations Research*, **35**, 450-452 (1987).

[8]   L. SCHRAGE and K.R. BAKER, Dynamic programming solution of sequencing problems with precedence constraints, *Operations Research*, **26**, 444-449 (1978).

[9]   W.E. SMITH, Various optimisers for single stage production, *Naval Research Logistics Quarterly*, **3**, 59-66 (1956).

[10]   L.J. WILKERSON and J.D. IRWIN, An improved algorithm for scheduling independent tasks. *AIIE Transactions*,**3**, 239-245 (1971).

## Appendix A

Consider a schedule in which $N$ jobs are to be scheduled starting at time $T=0$ with the objective of minimizing tardiness. Suppose that the three jobs shown below are the remaining jobs which require scheduling beginning at time $T=10$ (this being the completion time of the $(N-3)^{th}$ job in the schedule formed so far) and assume that we use the conventional form of the **W-I** heuristic to schedule these jobs in order to minimize tardiness.

| Job Number:$i$ | $t_i$ | $d_i$ |
|---|---|---|
| 1 | 41 | 0 |
| 2 | 5 | 51 |
| 3 | 7 | 53 |

Remember that the conventional form states that: for any given pair of adjacent jobs $i$ and $j$ that can start after time $T$, schedule the jobs according to the earliest due date (**EDD**) rule if

$$T + \max\{t_i, t_j\} \leq \max\{d_i, d_j\}$$

otherwise schedule the jobs according to the shortest processing time (**SPT**) rule if

$$T + \max\{t_i, t_j\} > \max\{d_i, d_j\}$$

To implement the **W-I** heuristic jobs are first arranged in **EDD** order. In our case the jobs are already in **EDD** order. Adopting the notation introduced earlier, we find that at time $T=10$, job 1 becomes job $\beta$ and job 2 becomes job $\gamma$. When the decision rule is applied to these jobs we find that

$$T + \max\{t_\beta, t_\gamma\} = \max\{d_\beta, d_\gamma\} = 51$$

since $t_\beta = 41$, $t_\gamma = 5$, $d_\beta = 0$, and $d_\gamma = 51$. Therefore, according to the decision rule, job 1 is added to the schedule and becomes job $\alpha$ whilst job 2 becomes job $\beta$ and job 3 becomes job $\gamma$. $T$ is increased to 51. When the decision rule is applied to jobs $\beta$ and $\gamma$ we find that

$$T + \max\{t_\beta, t_\gamma\} = 58 > \max\{d_\beta, d_\gamma\} = 53$$

since $t_\beta = 5$, $t_\gamma = 7$, $d_\beta = 51$, $d_\gamma = 53$. Hence, according to the decision rule, job 2 is added to the schedule. Finally, job 3 is scheduled in last position. The contribution to tardiness due to these last three jobs is 66.

Similarly, following the same procedure, it can be shown that our form of the **W-I** heuristic schedules the three jobs starting with job 2, followed by job 3 and then job 1 in last position with a tardiness contribution of 63.

This demonstrates that our form is superior to the conventional one in some cases.