A Sequencing Heuristic to Minimize Total Tardiness in the Two-Machine Permutation Flowshop

Eric A. Siy
Department of Industrial Engineering
De La Salle University
email: eric.siy@dlsu.edu.ph

Abstract: The two machine flow shop is a job shop that processes all jobs through a uni-directional sequence of two machines with deterministic processing times per job, and no preemption. For any given sequence of jobs, there exists a set of completion times for each job C_j. Tardiness is defined as $T_j = \max\{0, C_j - d_j\}$. The total tardiness in the schedule is the objective to be minimized. Pinedo (2008) showed that the minimizing total tardiness in the permutation flowshop problem is NP-hard, which justifies the development of heuristic procedures to shorten computational steps. This paper will present a heuristic procedure to search permutation schedules that uses an adaptation of the Duespan concept of Siy (1999) along with Johnson's Rule (1954), to create a set of possible sequence schedule. The paper concludes by showing how the heuristic performs against a complete enumeration of the permutation schedules and demonstrates how it determines the same schedule as Modified Due Date (MDD) of Baker and Kanet (1983) with less processing effort, making it appropriate for human-centric computations without computers.

Key Words: Permutation Flowshop; Two-Machine Flowshop; Total Tardiness; Heuristic

1. INTRODUCTION

The two-machine flowshop is a standard textbook illustration of sequencing jobs in a two-stage production system that utilizes Johnson's Rule (Johnson, 1954) for minimizing makespan (or maximum completion time of all jobs). Johnson's Rule constructs a sequence by determining the first and last jobs, and subsequently inserting the jobs successively in the middle. Johnson's Rule can be formulated thus: for the set of jobs not yet sequenced, find the minimum processing time in either machine; if the job associated with this minimum time is from the first machine, sequence this job first in the sequence; however, if the said minimum processing time is in the second machine then schedule this job last in the sequence. Johnson's Rule does not consider total tardiness as a job sequencing criterion, but can be used when jobs have identical due dates. An identical due date requires that jobs must be finished at the soonest possible time in order that the last job in the sequence would presumably finish the earliest, and hence minimize tardiness for the last job in the sequence.

Total tardiness is the scheduling criterion that gives equal weight to all jobs and seeks to minimize the total delay that jobs are completed beyond their due dates. When jobs are completed early, then they are not tardy. When due dates of jobs are each job are set reasonably to provide adequate time for processing in the two machines, then each job appears reasonably able to meet their due dates. A scheduling problem occurs when multiple jobs vie for processing time on the same pair of machines, but cannot be processed simultaneously (non-interference constraint) but rather sequentially, thereby affecting the completion times of jobs scheduled later than the first job.

The modified due date (MDD) concept created by Baker and Kanet (1982) has been shown to work remarkably well in minimizing total tardiness in the single machine setup (Koulamas, 1994). MDD is a dynamic scheduling rule that chooses the minimum among each job's completion index Πj=max(dj, t+pj) where dj=due date of job j, t= current time available to start next job, and pj=processing time of job j. The completion index Πj essentially determines the modified due date as either the job's due date (d_j) or the expected completion time of a job ($t+p_j$) beyond. When a job is expected to be completed beyond its due date, then the job that is least tardy is chosen for the next job in the sequence, thereby realizing a smallest contribution to total tardiness. Noteworthy is when jobs have some slack between expected time to finish $t+p_j$ and its corresponding due date, the MDD Rule reverts to being the earliest due date (EDD) rule since jobs are scheduled according to increasing due dates.

The MDD rule can be modified to work for two-machine flow shops by applying the same concept of finding the higher value between a job's due date and its completion time. We merely have to update the current time variable t to correspond to either the first machine's earliest time to begin the next job, or else as the second machine's earliest time for a next job.

This paper proposes a heuristic that utilizes the author's original concept of duespan (Siy, 1999) that determines a sequence for jobs in a list of available jobs for operation, and uses Johnson's Rule for breaking ties for the subproblem of common due dates.

2. HEURISTIC DEVELOPMENT

Duespan, denoted by DS_j, is the difference between the theoretical minimum makespan and a job's due date. Theoretical minimum makespan (Cmax) for a two machine flowshop can be determined as the maximum of either (total processing times in the first machine + the minimum processing time of any job in second machine) or (mimimum processing time of any job at first machine + total processing times in the second machine). These relations can be denoted by the following equations:

$$\textit{Theoretical } C_{max} = max \begin{pmatrix} \sum_{j=1}^{n} p_{1j} + \min(p_{2j}) \\ \min(p_{1j}) + \sum_{j=1}^{n} p_{2j} \end{pmatrix} \qquad \text{(Eq. 1)}$$

$$DS_j = \textit{Theoretical } C_{max} - \text{due date } d_j \qquad \text{(Eq. 2)}$$

where:

C_{max} = Maximum completion time of any job
P_{1j} = Processing time on first machine of job j
P_{2j} = Processing time on second machine job j

A higher duespan denotes that a job's due date is earlier than another with a lower duespan. This observation leads to the formulation of the heuristic for minimizing total tardiness in the two machine flowshop.

When jobs have a common duespan, it denotes that they have the same due dates. In this case, we can invoke Johnson's Rule to minimize the makespan for this subset of jobs. This is justified to maintain the earliest possible finish time for the subset of jobs, so that total tardiness for the later jobs in the sequence may be made as small as possible.

The flowshop heuristic can be therefore described hereunder: **Sequence jobs in the flowshop via descending order of duespans; in case of ties in duespan values, use Johnson's rule to minimize total processing times for those jobs.**

3. ILLUSTRATIVE EXAMPLE

Consider the six job two-machine flowshop problem presented in Table 1 that seeks a sequence that minimizes total tardiness.

Table 1 Processing times for four jobs

Jobs	J1	J2	J3	J4	***sum***
First Machine	2	3	8	6	***19***
Second Machine	7	5	4	3	***19***
Due time d_j	10	10	25	25	

3.1 Recommended solution via duespan heuristic

Theoretical makespan is max(19+3, 2+19) = 22

For each job, duespan (Dj) may be determined from (2):

Job 1 = 22-10 =12	Job 3 = 22-25 = -3
Job 2 = 22-10 =12	Job 4 = 22-25 = -3

Scheduling by descending values of duespan, we have only two values, 12 and minus 3. We choose job duespan of 12 to be sequenced first. Since Jobs 1 and 2 have tied duespans of 12, Johnson's Rule can be applied: the minimum processing time in either machine for jobs 1 and 2 is Job 1 at Machine 1 equivalent to 2 time units. Since the minimum is in Machine 1, schedule Job 1 first among the two sequence as prescribed by Johnson's Rule.

Partial job sequence is currently (J1, J2, -- , --).

Descending order of duespan leads us to scheduling Jobs 3 and 4 next, again with a tied duespan of -3. Appyling Johnson's Rule, Job 4 has the minimum processing time in any machine of 3 at machine 2, so schedule Job 4 last in the sequence. Then Job 3 is the last job not yet sequenced, so Job 3 is placed in third place.

Completed job sequence recommended by the heuristic is therefore (J1, J2, J3, J4).

A Gantt Chart (Fig. 1) may be constructed to show the completion times of each job. One can see that only job 2 is tardy by 4 units from its due time of 10.

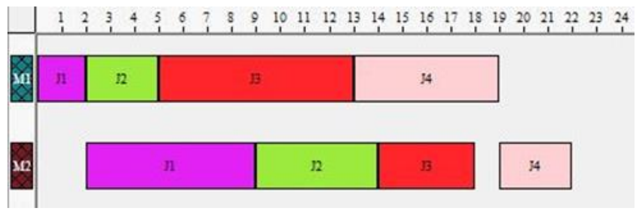

Fig. 1: Completed sequence using Heuristic for Table 1 data

3.2 Recommended solution via Modified Due Date

Table 2 shows the process by which MDD is used to choose the schedule. MDD Rule also gives the same recommended sequence J1-J2-J3-J4.

Table 2: Demonstration of Modified Due Date Procedure for Table 1 data

	J1	J2	J3	J4	
Due time dj	10	10	25	25	
Stage 1: t=0	J1	J2	J3	J4	
completion	9	8	12	9	
MDD	max(10,9)	max(10,8)	max(25,12)	max(25,9)	Chosen job
Minimum	10	10	25	25	J1 or J2
Stage 2: t1=2, t2=9		J2	J3	J4	
completion	--	14	13	12	
MDD	--	max(10,14)	max(25,13)	max(25,12)	Chosen job
	--	14	25	25	J2
Stage 3: t1=5, t2=14			J3	J4	
completion	--	--	17	14	
MDD	--	--	max(25,17)	max(25,12)	Chosen job
	--	--	25	25	J3 or J4
Stage 4: t1=13, t2=18				J4	
completion	--	--	--		
MDD	--	--	--		Chosen job
	--	--	--		Last job J4

3.3 Complete Enumeration of 4! = 24 permutation schedules for four jobs of Table 1

A complete enumeration procedure for finding the permutation schedules for n jobs results in n! possible schedules to be tested. This is an unwieldy process, and justifies the use of heuristics to find short-cuts through the schedule search space. Table 3 is a complete enumeration of the 4!=24 permutation schedules possible for the four job problem given as the illustrative problem. One can see that both the proposed heuristic and the MDD rule both identified the optimal sequence J1-J2-J3-J4 with a total tardiness of 4.

Table 3: Complete Enumeration of 24 permutation schedules of Illustrative example Table

	Completion times				
Sequence	J1	J2	J3	J4	Total Tardiness
1234	9	14	18	22	4
1243	9	14	23	17	4
1324	9	19	14	22	9
1342	9	24	14	19	14
1423	9	17	23	12	7
1432	9	25	20	12	15
2134	15	8	19	22	5
2143	15	8	22	18	5
2314	22	8	22	25	12
2341	22	8	15	20	17
2413	27	8	23	12	9
2431	28	8	21	12	18
3124	19	24	12	27	25
3142	19	27	12	22	26
3214	24	17	12	27	23
3241	27	17	12	20	24
3412	23	28	12	15	31
3421	29	22	12	17	31
4123	16	21	25	9	17
4132	16	25	20	9	21
4213	21	14	25	9	15
4231	28	14	21	9	22
4312	25	30	18	9	35
4321	30	23	18	9	33

1

4. COMPLEXITY OF ALGORITHM

Given n jobs to be sequenced, the modified due date takes up 2(n)+1 computational steps for the first stage, then 2(n-1)+1 computational steps for the second stage, and so on until the last stage of 2 remaining jobs (2(2)+1 steps=5 steps for the last stage. The last stage does not need computing for MDD since it is the last job. This results in 2[n(n+1)/2-1]+n steps to arrive at a sequence. This results in an order of O(n2) required steps for the MDD procedure.

The proposed heuristic requires 3 steps to derive the theoretical makespan, then another n steps to derive the duespans of each job. A final step is necessary to arrange the jobs via descending values of duespan. Therefore, the proposed heuristic takes up 2n+4 steps to arrive at a recommended sequence. This results in an order of complexity that is only linear O(n).

5. CONCLUSIONS

A heuristic was suggested that uses a simpler rule compared to the modified due date (MDD) heuristic rule stated by Baker and Kanet [2]. The linear order O(n) of the proposed heuristic requires less computational effort than the O(n2) quadratic order of MDD. For the tested problems, the heuristic shows promise of determining sequences that can minimize total tardiness more easily than the current acceptable MDD heuristic.

While more and more scheduling problems require computers to process much simulation steps as in the Ant Colony Optimization process [5], this heuristic reverts back to the earlier days of human hand calculation and can give a case for human-centric scheduling procedures that do not need modern computational power to answer simple structured problems.

6. REFERENCES

Baker, Kenneth R and J.W.M. Bertrand (1982) “A dynamic priority rule for scheduling against due-dates”, *Journal of Operations Management* Vol. 3, p 37-42

Johnson, S. M. (1954) “Optimal Two- and Three-Stage Prodution Schedules with Set-up Time Included”. *Naval Research Logistics Quarterly*, 1: 61-68.

Koulamas, C. (1994). The total tardiness problem: Review and extensions. *Operations Research*, 42, 1025–1041.

Pinedo, M. (2008) *Scheduling: theory, algorithms, and systems*. Springer.

Siy, Eric (1999) *Open shop scheduling minimizing bicriteria makespan and total weighted tardiness*. Unpublished MSIE thesis. De La Salle University.

Stutzle T (1998) “An ant approach to the flow shop problem.” In *Proceedings of the Sixth European Congress on Intelligent Techniques & Soft Computing (EUFIT'98)*, volume 3, pages 1560–1564. Verlag Mainz, Wissenschaftsverlag, Aachen, Germany.