Presented at the Research Congress 2013
De La Salle University Manila
March 7-9, 2013

A NEIGHBORHOOD SEARCH SCHEDULING HEURISTIC FOR MINIMIZING TOTAL WEIGHTED TARDINESS IN THETWO-MACHINE FLOWSHOP

Eric A. Siy
Department of Industrial Engineering
De La Salle University

Abstract: Sequencing jobs on the two-machine flowshop minimizing total weighted tardiness is NP-hard (Pinedo, 2002) and hence, warrants the development search heuristics that generate near-optimal but aceptable schedules. The well-known Johnson's Rule is the optimal rule for minimizing makespan in the two-machine flowshop, but the scheduling criterion of minimizing total weighted tardiness has received less attention. This paper proposes divide-and-conquer approach to sequencing using Johnson's Rule and the lower-bound formula proposed by Ignall and Schrage (1965). A neighbourhood-swap final iteration process improves the initial sequences found. Through comparison of worked examples of the problem using the proposed heuristic with the branch-and-bound approach (Bestwick and Hastings 1976, Ignall and Schrage 1965, Lomnicki 1965, McMohon and Burton 1967) and complete enumeration of possible sequences, the performance of the heuristic will be shown to work at par if not better than known B&B approaches and generates schedules that is as close as possible to the optimal schedule as may be found by complete enumeration.

Key Words: Two-machine flowshop; minimizing total weighted tardiness; heuristic methods

1. INTRODUCTION

Consider a set of jobs that has to undergo two machining processes in uniform order, with deterministic processing time for each machine (P_{ij}: i=index for machine 1 or 2, and j=jobs 1,2,..n), with corresponding due dates for each job ending at the second machine, d_j, and priority index as represented by weight w_j.) For any sequence of the n jobs, $\sigma(n)$, there exists a set of completion times for each job, C_j, out of which each job's possible tardiness, T_j, may be defined as the non-negative difference between a job's completion time, and its due date: $T_j=\max(0, C_j-d_j)$.

Minimizing makespan--the time to complete all jobs--in the two-machine flowshop ($F2/ /C_{max}$) is a deterministic time sequencing problem that can be optimally solved by Johnson's Rule (S.M. Johnson, 1954) known virtually in all production operations management textbooks (Heizer and Render, 2001). This paper tackles another scheduling criterion: minimizing total weighted tardiness for all jobs. ($F2//\Sigma W_jT_j$), which has been solved with NP-hard considerations via Branch-and-Bound methods (Bestwick and Hastings

1976, Ignall and Schrage 1965, Lomnicki 1965, McMohon and Burton 1967).

Minimizing total weighted tardiness in the two-machine flowshop (F2// $\Sigma W_j T_j$) was classified as an NP-hard problem (Pinedo, 2002) implying that for a flowshop with n jobs available at time=0, the search through the $(n!)^2$ possible non-preemptive sequences of the n jobs on the two machines faces a considerable computation time that increases exponentially with the number of jobs n, and could not be done within n-polynomially determined time. NP-hard problems discourage solutions that cover complete enumeration. Consider a typical problem as presented in Table 1, which the this author has used in a similar paper (Siy, 2012).

Table 1. Four jobs on two-machine flowshop

Job j	Processing time on first machine P_{1j}	Processing time on second machine P_{2j}	Due date d_j	Penalty Weight W_j
A	5	3	10	1
B	2	5	10	3
C	4	3	15	2
D	1	3	15	1

The next section demonstrates a benchmark procedure for finding the optimal sequence.

2. BENCHMARK PROCEDURE FOR SOLVING (F2//$\Sigma W_j T_j$): Ignall-Schrage Branch-and-Bound method

A Branch-and-Bound method that schedules jobs backwards was proposed by Bestwick and Hastings (1976) which was based Ignall and Schrage (1965). The former authors utilizes a lowerbound (LB) function for a sequence of jobs scheduled last, shown hereunder:

$$LB\ (sequence\ S) = \min \left\{ \begin{array}{l} \sum_{jobs}^{not\ in\ S} P_{1j} + P_{2\ next\ job\ on\ S} \\ \min P_{1\ job\ not\ in\ S} + \sum_{jobs}^{not\ in\ S} P_{2j} \end{array} \right\} \quad \textbf{(Eq. 1)}$$

Where:

LB = lower bound for completion time of a job scheduled in a test sequence S

sequenced

The procedure for Branch-and-Bound search for the optimal sequence of jobs that minimizes total weighted tardiness ($\Sigma W_j T_j$) proceeds as follows: Initially let S={ø}. Begin by testing the sequences with any of the n jobs scheduled last in the sequence S={any of n jobs}. Evaluate the lower bound for the completion time of said job in sequence S, and determine the weighted tardiness. Among the sequences (branches), choose the sequence with the lowest weighted tardiness. Expand the sequence on this branch by prefixing the other jobs not yet sequenced, expanding sequence S with the current last best sequence. Reevaluate the lowerbound completion times and the corresponding total weighted tardiness of the expanding set of test sequences S. Search among the evaluated branches for the lowest value of Total Weighted tardiness and expand the sequences on the lowest objective values found. Continue the cycle of {(1) searching for the current best value of $\Sigma W_j T_j$ among the partial sequences S (2) evaluating the lower bound LB of the sequences that is extended by the set of unsequenced jobs, and their corresponding $\Sigma W_j T_j$'s } until all jobs have been annexed on the test sequences tree. The optimal sequence can be found among the set of sequences with the lowest $\Sigma W_j T_j$.

Figure 1 demonstrates the branch and bound procedure for the problem presented in Table 1. The optimal sequences were BADC and BACD, both with total weighted tardiness of 2.

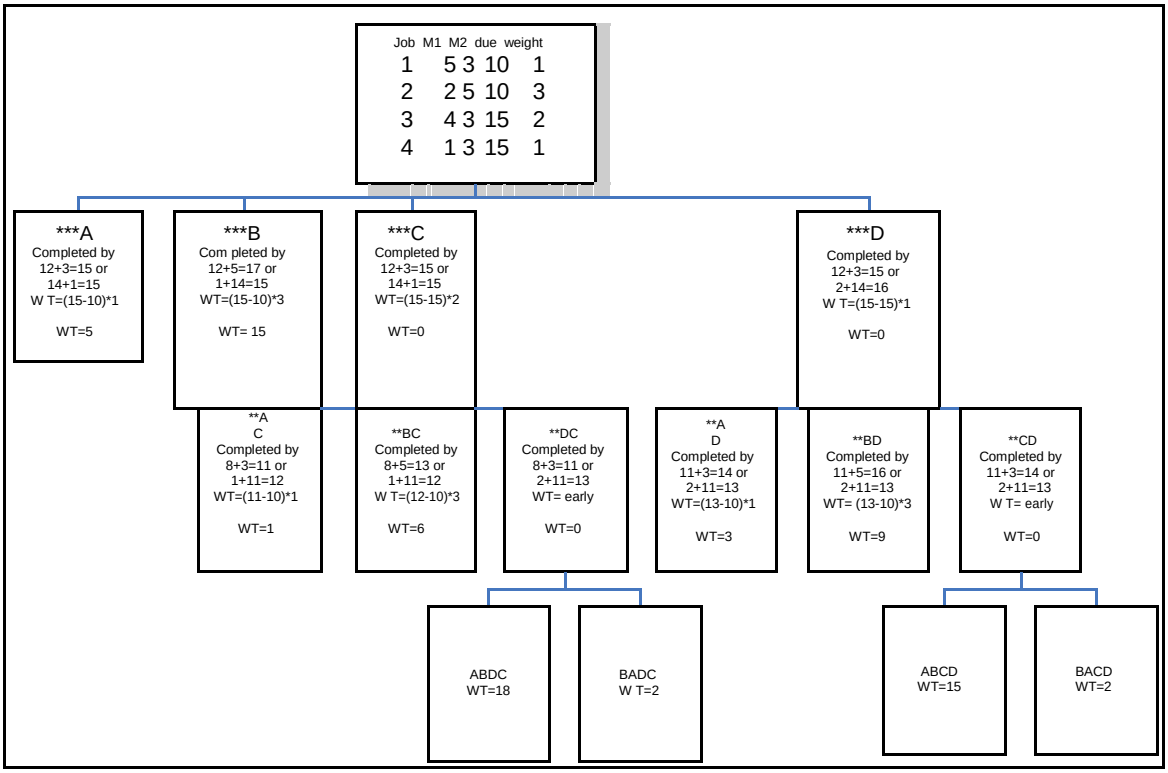

Figure 1. Branch and Bound Search Tree for four job flowshop example

3. PROPOSED PROCEDURE FOR SOLVING ($F2//\Sigma W_jT_j$): JOHNSON'S RULE WITH NEIGHBORHOOD SEARCH

The Branch and Bound procedure guarantees an optimal solution can be bound, but with NP-hard solution effort. This paper proposes an improved method that is less complex computationally:

(1) Arrange jobs via earliest due date. Jobs with common due dates may be considered as a batch of jobs.

(2) For each batch of jobs with common due dates, apply Johnson's Rule to create a partial sequence for each job group. Jobs that do not share common due dates will be sequenced in the order made in step 1. [Johnson's Rule creates a sequence

from the ends converging in the middle. Of the jobs not yet sequenced, find the minimum processing times at machine *i* of job *j (*P_{ij}): If the minimum time

is in Machine 1, sequence the job at the beginning. If the minimum time is in Machine 2, sequence the job at the end. Remove job with minimum time found from the list of unscheduled jobs. Repeat this minimum Pij search and sequencing step for the remaining jobs in the list, until all jobs have been sequenced.**]**

(3) Iterative neighborhood search process that includes any or all of the following improvement steps

a. Pairwise exchange of adjacent jobs that results in partial sequences with non-increasing weights

b. Inclusion of jobs of uncommon due dates in a new Johnson's Rule sequence.

4. RESULTS AND DISCUSSION

Using the same problem in Table 1, we can demonstrate the procedure thus:

(1) Jobs arranged by EDD rule: Batch 1 includes Jobs A and B both with duedate at 10. Batch 2 is Jobs C and D, due at 15.

(2) Johnson's Rule applied to batch 1 results in an ordered sequence (Job B, Job A). For batch 2, resulting order sequence is (Job D, Job C). Therefore, the current suggested sequence for the flowshop is B-A-D-C. Only Job C is late, by 1, resulting in a total weighted tardiness of 2x1=2. Figure 2a shows the sequence as a Gantt Chart.

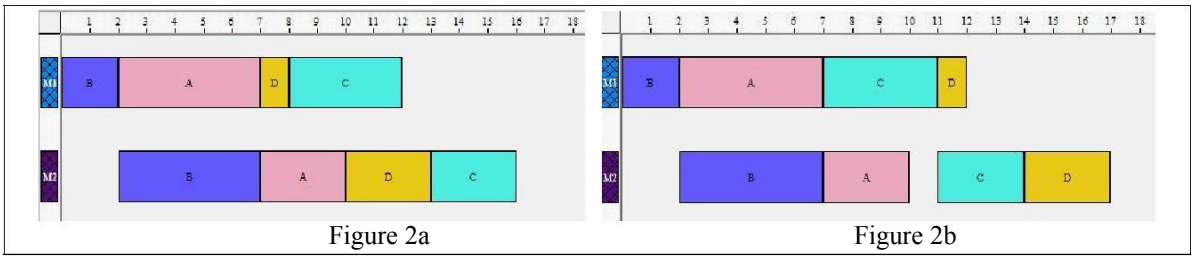

Figure 2a Figure 2b

Figure 2: Gantt Charts for proposed solution to Table 1 flowshop

(3) A. Neighborhood search: Batch 1 Jobs A and B were already completed at their due date of 10, so no improvements are possible (Inverting their order results in higher makespan of 13, job B becoming late by 3.) Batch 2 jobs on the other hand has a lower weighted job 4 preceding job 3. A pairwise swap could be performed between them, resulting in a new test sequence 2-1-3-4, as shown in Figure 2b. Job 3 is early, but Job 4 is now late by 2, resulting in total weighted tardiness of 1x2=2. We now have two sequences which both have $\Sigma W_j T_j = 2$, both sequences shown in figure 2.

B. Neighborhood search: expand Johnson's rule scope to not just batches (BA) and

(CD), but to newly partitioned batches (B)(ADC) and/or (BAD)(C).

B.1 Batching together B+(ACD) results in a Johnson's Rule sequence of B+(DAC) or B+(DCA). BDAC results in a worse total weighted tardiness of 5. BDCA is even not better at 6.

B.2 Batching together (BAD)+C results in (DBA)+C: which is actually another optimal sequence with total weighted tardiness of 2. Job A is late by 2, resulting in $\Sigma W_j T_j = 1x2=2$.

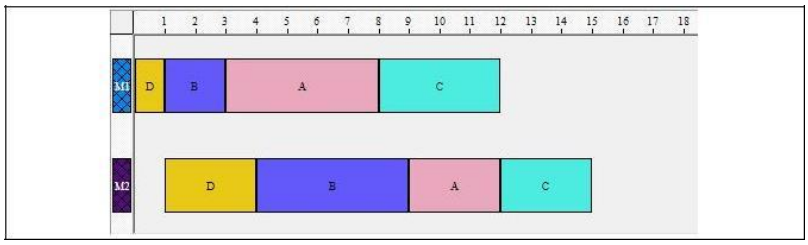

Figure 3: Neighborhood search yields another optimal sequence with WT=2

B.3 Batching all four jobs ABCD together as a Johnson's Rule batch would result in two possible sequences DBAC and DBCA. DBCA was already found to be optimal, but DBCA results in a $\Sigma W_j T_j = 5$. At this point, no other possible partitioning of jobs are possible, therefore, neighborhood search can stop.

Johnson's Rule will guarantee the lowest makespan or completion time of a batch of jobs. When a due-date objective like weighted tardiness is applied, Johnson's rule may not guarantee each job being on time, but the rule will suggest an earlier overall finish, regardless of the jobs' due date. This quality will enable subsequent jobs of a later due date to begin as early as possible, resulting in possibly earlier completion times for the later job batches.

The neighborhood search process considers the presence of tardiness weights, and attempts to include the decoupled jobs connecting one job batch to another, and tries to do pairwise exchanges that could improve the overall weighted tardiness: sequences presented by Johnson's Rule does not consider due date performance, so a local search among adjacent jobs may find better sequences.

A complexity analysis of the Branch and bound method would result in an order of O(n!): Each of the n parent nodes (representing which of the n jobs could be sequenced last) could at worst require (n-1) child nodes per parent node. For each stage k of the schedule job

sequence, there are at worst (n-k+1) nodes to check, until the last stage (n^{th}) job. This permutation schedule search represents the n*(n-1)* …. *2*1 = n! schedules, at worst case, that are searched by a branch and bound method.

In contrast, the proposed method would take, at worst case, a total number of steps $2n^2+n-1$ for n jobs in a flowshop. In terms of complexity, the proposed method has a highest order of 2, i.e. complexity order of $O(n^2)$, which has less predicted calculation steps compared to the O(n!) order for the branch and bound. This means the needed steps are dramatically less to arrive at a feasible set of schedules, from which an optimal sequence may be found. Determinining the algorithmic complexity for the proposed method may be summarized in table 2.

5. CONCLUSIONS

This paper presented a procedure that is takes less computational steps to find a sequence that minimizes total weighted tardiness in the two-machine flowshop. Compared to the branch-and-bound method, which would at worst try to search through the order of O(n!) possible permutations for n jobs, this proposed method would have less steps at $O(n^2)$. While it may be challenging to use branch and bound methods for searching for optimal sequences in $(F2//\Sigma W_jT_j)$, an improved method using a battery of divide-and-conquer Johnson's Rule, neighborhood pairwise exchanges, and repartitioning of jobs and reapplying Johnson's Rule, it turns out that minimizing makespan does contribute to minimizing total weighted tardiness as well.

Table 2. Computational steps needed in Proposed method:

Johnsons rule with Neighborhood search method

Step	Worst case required number of steps for n jobs
1. Arranging jobs via ascending due date	n-1
2. Johnson's Rule applied to each batch	n* (n-1)
3. Neighborhood search via pairwise swap	${}_nC_2 = n(n+1)/2$
4. Neighborhood search via partitioned groups	${}_nC_2 = n(n+1)/2$
5. Search for best sequence in each stage	n^2
Total steps: **$2n^2+n-1$** from	(n+1)(2n-1)

6. REFERENCES

Bestwick, P.F. and Hastings, N.A.J. (1976) A new bound in machine scheduling. *Operational Research Quarterly. 27*, 479-87.

Edward Ignall and Linus Schrage (1965). Application of the Branch and Bound Technique to Some Flow-Shop Scheduling Problems. *Operations Research. May/June 1965 13*, 400-412.

Johnson, S.M. (1954). Optimal Two and Three Stage Production Schedules with Set-up Times Included. *Naval Research Logistics Quarterly, Vol.1, No. 1*, 61-68

Lomnicki, Z.A. (1965) "A branch and bound algorithm for the exact solution of the three-machine scheduling problem*" Operational Research Quarterly. 16*, 89-100.

McMohon, G.B. and Burton, P.G. (1967) "Flowshop scheduling with branch and bound method" *Operations Research 15*, 473-81.

Pinedo, M. *Scheduling: Theory, Algorithms, and Systems* (2nd ed.) New Jersey: Prentice-Hall, 2002. p.92.

Siy, Eric (2012). A Sequencing Heuristic for Minimizing Total Weighted Tardiness in the Two-Machine Flowshop. *Proceedings of 6th National Conference of Operations Research Society of the Philippines.* November 9, 2012. ISSN #2094-6031.