Presented at the DLSU Research Congress 2014
De La Salle University, Manila, Philippines
March 6-8, 2014

Heuristic approach to Non-preemptive Open Shop Scheduling to Minimize Number of Late Jobs n_T

Eric A. Siy

Department of Industrial Engineering
De La Salle University
2401 Taft Avenue, Manila, Philippines
email: eric.siy@dlsu.edu.ph

Abstract: This paper considers how to sequence n jobs through m resources, with all jobs initially available for sequencing through any of the resources, all commenced jobs are committed to completion at a deterministic time with the engaged resource (non-preemption), and with no prescribed sequence of operations through the m resources (Open Shop).Minimizing number of late jobs for the Open Shop has been shown to be NP-hard (Pinedo, 2008) due to the Open shop's relaxed structure in contrast with the Job Shop, where jobs have a prescribed order of resource processing. The development of heuristic procedures that undercut the computational extensiveness of complete implicit enumeration is therefore befitting. Many practical applications for this scheduling model exist especially in testing and maintenance in which the order in which jobs can be processed makes no difference. Teacher-class time-tabling is also another open shop problem instance: teachers have to be assigned to classes but cannot be scheduled to two classes at the same time, but should teach the assigned classes in any order for the day. The paper presents an O(mn) procedure that should generate schedules quite quickly and can approximate optimal sequences. Small cases numerical examples demonstrates the heuristic procedure and results obtained by complete enumeration. Finally, the heuristic procedure can be shown to require simpler computational steps compared to those presented by Blazewicz et al (2003), and would be conducive for non-computer-based human computing.

Key Words: Open Shop; Non-preemptive Scheduling; Number of tardy jobs; Late jobs; Heuristic

1. THE LATE JOBS OPEN SHOP PROBLEM

1.1 Introduction

The tardy jobs open shop scheduling problem can be described as follows: given n jobs and m different machines with each job j=1,2,…n requiring machining operation on resource i=1,2,..m with duration processing time p_{ij}. Each machine cannot process two jobs at the same time, and the job being processed on a machine must not be interrupted once begun (i.e. non-preemption). Open shops can operate on the jobs in any order of machining as long as the job gets processed by machine resource i. Each job is initially available to be processed at time zero, and has an associated due date d_j. A job can be sequenced through the m machines so that it completes all operations at time C_j. When jobs are completed beyond its due date d_j,

the job is considered tardy. The objective of the sequencing process is to minimize the total number of tardy jobs. This sequencing problem is considered NP-hard since its lower complexity case of minimizing maximum tardiness (L_{max}) is already NP-hard (Pinedo, 2008).

1.2 Possible applications

This problem occurs in real-world situations, particularly in inspection, testing, and maintenance (Liaw, 2003), where the order of the operations does not matter. The open shop is different from a job shop in the condition of jobs having a prescribed order of processing. This structured problem environment of a job shop makes the feasible number of schedules already fixed and large, and the open shop increases the schedule possibilities in factorial magnitudes. This NP-hard condition is the motivation to create heuristic approaches to sequencing the tardy jobs open shop problem: complete implicit enumeration of permutation schedules of n jobs on each single machine is already n! at worst case. Open shop scheduling is to search n! sequences for each of the m machines (i.e. O(m*n!)). This paper proposes a practical heuristic procedure that is takes considerably less computational steps, yet still provides sequences that can approximate optimal solutions.

2. HEURISTIC DEVELOPMENT

2.1 Arranging jobs on a machine based on due dates

hen considering due dates, the earliest due date (EDD) rule states that jobs must be sequenced by ascending due dates. In a previous work [3], this writer introduced the concept of duespan. Duespan is the difference between the theoretical minimum makespan of jobs min(Cmax) and a job's due date dj. Theoretical minimum makespan is the highest sum of processing times on either all the machines or all the jobs.

Duespan was developed to simultaneously consider makespan and due dates, but this same concept can be used for considerations on a similar due date criterion. A high value duespan connotes that the job's due date is early relative to another job with a lower duespan. The relationship still applies for negative valued duespans: when job A with a negative 5 duespan is compared with job B with negative 7 duespan, the former job A still has an earlier due date than job B by 2 time units. This gives rationale for scheduling jobs with descending (non-increasing) values of duespan.

2.2 Open shop schedules non-interference pattern

Suppose that a set of n jobs all have uniform processing time on an open shop with m machines. The uniform processing times can be likened to square tiles that have to be laid out on a nxm matrix with m rows and n columns. To ensure that jobs will not be at two different machines at the same time (i.e. non-interference constraint), it is possible to repeat a certain pattern on each row but removed by one tile on the next like in a flowshop schedule, but further imagine having the tiles that do not fit into the grid can "cycle back" to the first row and continue the pattern like in a rotary spinning jenny. Such a pattern ends with tiles of one kind (i.e job denotation) will occupy exactly once in each column and in each row. This pattern is called a Latin square. Figure 1 shows instances of Latin Squares tile designs. The Latin square setup is an initial sequence that may be done for open shops such that jobs occupy a certain time window and may not interfere with others. When jobs are not equal in duration, obviously the Latin square design may not fit as Gantt Charts templates, but they are a starting point for an initial permuation schedule.

A	B	C
C	A	B
B	C	A

3x3

M	N	O	P
N	O	P	M
P	M	N	O

3x4

J	K	L	-	-
L	J	K	-	-
-	L	J	K	-
K	-	-	L	J

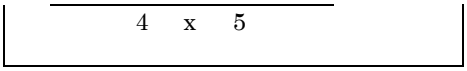

Figure 1: Examples of Latin square tile designs

3. HEURISTIC FOR OPEN SHOP MINIMIZING TOTAL NUMBER OF TARDY JOBS NT

The following steps shows how to create trial permutation schedules to find the least number of late jobs:

1. Determine the theoretical minimum makespan from the processing time matrix.
2. Determine the duespan for each job.
3. On the machine M' with the longest total processing time , sequence the jobs via descending values of duespan. When ties exist on the machine with total processing time, generate sequences on each of the machines; when ties exist in duespan values, generate sequences where each of the tied jobs appear in their possible permuations as a set. This sequence is deemed relatively permanent on machine M'.
4. Following the pattern of jobs on the machine M', create Latin square sequences on the other machines. Determine the completion times of each job to derive the number of tardy jobs and the individual tally of lateness of jobs, as well as the earliness of jobs.
5. Begin a neighborhood search process on moving late jobs earlier in the machines other than machine M' the longest total processing times. This writer called this search process "retrofitting" in a previous work (Siy, 2010).

4. DEMONSTRATION OF HEURISTIC PROCEDURE:

Consider the three-machine three-job open shop problem on Table 1.

The theoretical minimum makespan is 17, based on the maximum total job time for job 2. Duespan on each job j can thus be determined as 17-d_j. [J1,J2,J3]=[7,-1,7] with corresponding possible sequences J1-J3-J2 or J3-J1-J2 on machine M1 as machine M' with longest total processing time. Possible permutation schedules in Latin square format are shown on figure 2.

Three optimal schedules all with no tardy jobs was found by the heuristic: Schedules 1 with no retrofitting, Schedules 2 and 3 with some retrofitting. Schedule 4 did not yield optimal schedule, but gave one job late by one time unit.

One of the optimal schedule was found in Schedule 1. A Gantt chart can be generated using LEKIN (Feldman, 1995), as shown on Figure 3. No retrofitting process was necessary.

Table 1: First Example problem

Machine\Job	J1	J2	J3
M1	4	7	4
M2	2	4	3
M3	3	6	1
Due date d_j	10	18	10

Schedule 1			
M1	1	3	2
M2	2	1	3
M3	3	2	1

Schedule 2		
1	3	2
3	2	1
2	1	3

Schedule 3			
M1	3	1	2

Schedule 4		
3	1	2

M2	2	3	1	1	2	3
M3	1	2	3	2	3	1

Figure 2: Latin Square schedules generated for problem 1 (Table 1)

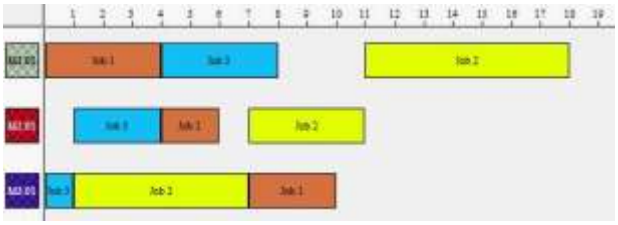

Figure 3: Schedule 1 with no jobs late, Tardy jobs=0 (Optimal)

Schedule 2 has one late job (job 1), seen in figure 4. But one can see that the job 1 was late due to the machine 2 placement. There is a time window between job 3 and job 2 which Job 1 can fit exactly, as shown in figure 5.

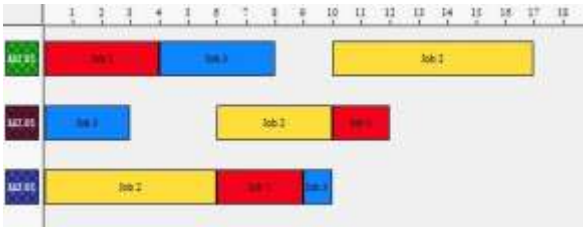

Figure 4: Schedule 2 with Job 1 late, number of tardy jobs=1

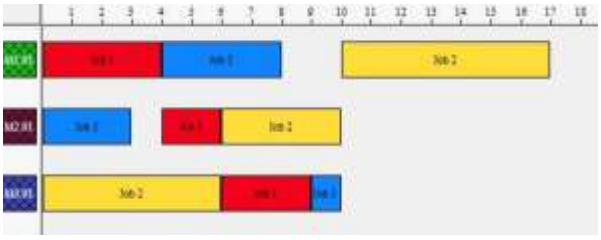

Figure 5: Improved Schedule 2 after moving job 1 earlier in machine 2, n_T=0 (Optimal)

Schedule 3 has one late job (job 3 on Machine 3 on Figure 6). But retrofitting late job 3 prior to job 2 on Machine 3 yields an optimal schedule. Figure 7 gives this optimal schedule based on Schedule 3.

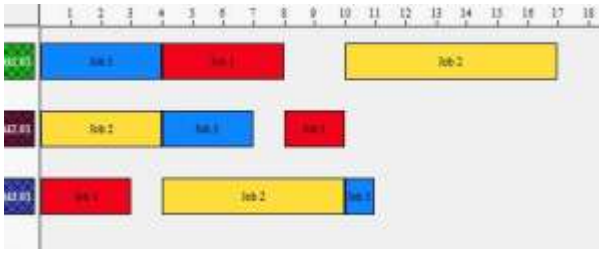

Figure 6: Schedule 3 with one late job 3.

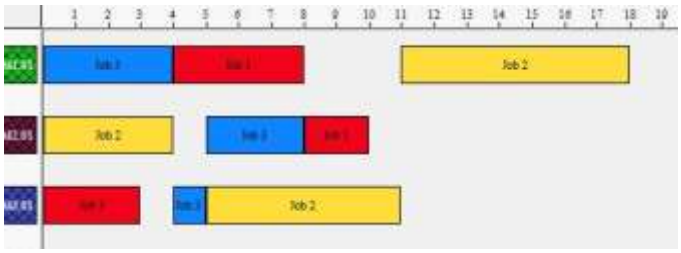

Figure 7: Improved Schedule 3 after moving Job 3 forward on Machine 3. n_T=0.

Schedule 4 could never be optimal answer because scheduling job 2 last on machine 2 will always have a late job. See Figure 8. Further retrofitting as done in Figure 9 did not yield zero late jobs.

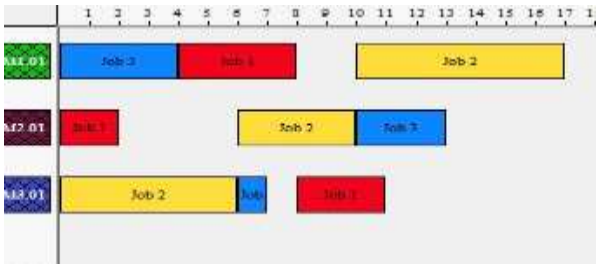

Figure 8: Schedule 4, with two late jobs. (jobs 1 and 3)

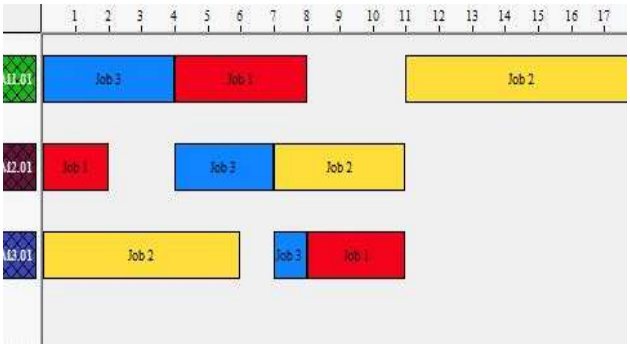

Figure 9: Schedule 4, after retrofit process with one tardy job

5.COMPARISON WITH BENCHMARK ALGORITHM

Blazewicz, Pesch, Sterna and Werner (2004) proposed an algorithm for scheduling the two-machine open shop with common due dates and minimize the number of late jobs. Their solution used a makespan minimizing process first proposed by Gonzalez and Sahni (1976) to construct an initial schedule, then modify it by shifting some jobs in order to minimize the idle time before the common due date and, consequently, to minimize the late work in the system. As this benchmark algorithm is only for two-machines open shops, the proposed heuristic may seem incommensurate for the general purpose application for the latter. But the number of computational steps can be compared to determine the level of complexity between the two.

Blazewicz, et al (2004) reports that with n jobs and d common due dates, their algorithm has a complexity in the order of $O(n^3d^2)$. This paper's presented heuristic can find or approximate optimal schedules for m=2 machines by searching for schedules in the order of O(mn)= O(2n), as shown hereunder Table 2.

Table 2: Complexity of the proposed heuristic

Step	Number of computational steps
1. Sum n Job times across m machines	m steps
2. Sum m machine times across n jobs	n
3. Find the maximum Cmax possible.	1
4. Determine duespan per job	n
5. Sort duespans to indicate sequence on bottleneck machine.	n-1
6. Sequence other non-bottleneck machines with Latin Square pattern	n*(m-1) = nm-n
7. Neighborhood search for retrofit insertion improvement	(m-1)*(n-1)= nm-n-m+1
8. Global comparison for best schedule found among schedules generated.	nm-1
Total possible steps	3mn + n -1
with highest order O(mn)	

6. Conclusion

The proposed heuristic has a lower complexity level compared with the benchmark algorithm proposed by Blazewicz et al [6], despite the specificity of having only two machines in the schedule. Without generalizing, the proposed heuristic is promising in terms of creating a set of testable schedule sequences for any sized open shop, and has made the search through the implicit enumeration of all possible m*n! schedules manageable and computationally practical.

Larger problems may be tested with more than 5 jobs to see if the method can prove useful in practical scheduling. This is an area for further study.

7. REFERENCES

Blazewicz, J., Pesch, Sterna, M. and Werner, F. (2004) "Open shop scheduling problems with late work criteria". *Discrete Applied Mathematics* 134, 1-24.

T. Gonzalez, S. Sahn (1976) Open shop scheduling to minimize finish time, *J. Assoc. Comput. Mach.* ACM 23 65–679.

LEKIN Scheduling System Version 3.0 developed by Andrew Feldman for the book by Michael Pinedo *Scheduling: Theory, algorithms and Systems*. Prentice-Hall, 1995.

Liaw, Ching-Fang (2003). "Scheduling preemptive open shops to minimize total tardiness", *European Journal of Operational Research* 162 (2005) 173-183. Elsevier. Available through sciencedirect.com retrieved 16 October 2013.

Pinedo, Michael (2008). *Scheduling: Theory, Algorithms, and Systems* 3rd edition. Springer Science+Business Media, LLC pp. 23-28.

Siy, Eric (1999). "Minimizing Makespan and Total Weighted Tardiness in the Open Shop". Unpublished MS Thesis. De La Salle University. Manila.

Siy, Eric (2010) "Parallel Machines in the Two-Machine Flowshop with Makespan Minimization *Proceedings of the 13th DLSU-Osaka Academic Workshop*. October 4, 2010.