



Reentrant Flow-Shop Scheduling Problem - A Heuristic Method

Qazi Shoeb Ahmad

Department of Statistics & Operations Research
College of Science, Qassim University
Kingdom of Saudi Arabia

Email: q.shoeb@qu.edu.sa

ABSTRACT

This paper presents a heuristic method for solving a reentrant flow-shop scheduling problem with '2 machines and n jobs', where the processing times for jobs on the 2 machines are specified. We obtained a schedule with a set of n jobs to minimize the makespan. The procedure is illustrated with the help of a numerical example.

Keywords: Flow shop, transportation time, weight of jobs

Mathematics Subject Classification: 90B35, 90B30, 90C59

1. INTRODUCTION

Practical machine scheduling problems are numerous and varied. They arise in diverse areas such as flexible manufacturing systems, production planning, computer design, logistics, communication, etc. A scheduling problem is to find sequences of jobs on given machines with the objective of minimizing some function of the job completion times. In a simpler version of this problem, flow shop scheduling [12], all jobs pass through all machines in the same order. In classical scheduling, it is typically assumed that each job visits a given machine at most once (Baker, 1974).

In reality, though, this assumption doesn't always apply, and many actual scenarios are modeled as reentrant flow shops, where a job might go through some machines multiple times (Danping & Lee, 2011). Graves et al. (1983) first introduced the reentrant flow shop problem, and it has since been studied extensively, as reviewed by Aspern et al. (2025). Semiconductor manufacturing, particularly in wafer fabrication lines, is well-known for its example of RFS. Through this procedure, chemical materials are sequentially deposited onto silicon wafers, with the wafers needing to undergo cleanroom passages repeatedly (Lin et al., 2013).

RFS layouts are also used in signal processing, where signals undergo preliminary processing by a computer, are transmitted and retrieved via a sensing and command system, and then undergo further processing by the computer (Rifai et al., 2021). Other examples encompass textile fabric production, mirror manufacturing systems, and repair processes necessitating rework (Danping & Lee, 2011). What sets the reentrant flow shop problem apart from the classical flow shop problem is its flexibility in permitting jobs to be run multiple times on certain machines. Creating heuristics for reentrant flow shop is crucial because of the NP-hard nature of the problem. Various heuristics have been suggested over the years to accomplish reentrant permutation flow shop solutions (Choi & Kim, 2007, 2008; Jing et al., 2008; Pan & Chen, 2003).

2. LITERATURE REVIEW

Consider a reentrant flow shop problem with a cyclic-reentrant pattern. It consists of n jobs, m machines, and l levels. A level in a cyclic pattern is a pass of a job in all machines in the order 1, 2, ..., m. A level is also referred to as a cycle or a loop (Yu & Pinedo, 2020). "Every job can be decomposed into several layers, each of which starts on M1 and finishes on Mm. In the reentrant flow shop case, if the job ordering is the same on any machine at each layer, then no passing is said to be allowed, since no job is allowed to pass any former job. The reentrant flow shop scheduling problem in which no passing is allowed is called a reentrant permutation flow shop problem" — quote taken from Chen et al. (2008).

An m-machine reentrant flow shop is a complex scheduling problem where jobs must visit machines sequentially $M_1, M_2, \dots, M_m$, then re-enter the shop more than once, looping back to earlier machines (like M_1) for additional processing steps, common in semiconductor & electronics manufacturing. This creates complex precedence constraints (a loop must finish before the next starts).

3. KEY CHARACTERISTICS

- Sequential Processing: Jobs move from M_1 to M_2 etc., like a normal flow shop.



- Re-entry: After the last machine M_m , jobs return to M_1 (or other earlier machines) for more operations, forming loops (e.g., $M_1 - M_2 - M_1 - M_2$)
- Precedence: A job's subsequent loop cannot start on M_1 until its previous loop finishes on M_m .
- Complexity: The looping and precedence make it much typical than standard flow shops.
- Applications: Semiconductor wafer fabrication, PCB manufacturing, complex electronics.

Objective is to minimize makespan (total time to finish all jobs). For example in making a microchip - A wafer goes to M_1 (clean), then M_2 (etch), then M_3 (deposit). It then re-enters M_1 for another cleaning, M_2 for more etching, and so on, repeating steps until all fabrication loops are complete, often with different processing times and requirements for each loop. Consider a simple numerical example for 2 jobs (J_1, J_2) and 2 machines (M_1, M_2). Each job needs to go through two loops, i.e., flow shop with reentrance (where each job follows the sequence $(M_1 - M_2 - M_1 - M_2)$ can be demonstrated to minimize the makespan (total completion time).

This problem in the tableau form can be stated (table-1) as follows:

Table-1

Job J_i	Loop 1		Loop 2	
	Operation 1 (Machine M_1)	Operation 2 (Machine M_2)	Operation 3 (Machine M_1)	Operation 4 (Machine M_2)
	M_{i1}	M_{i2}	M'_{i1}	M'_{i2}
J_1	M_{11}	M_{12}	M'_{11}	M'_{12}
J_2	M_{21}	M_{22}	M'_{21}	M'_{22}

Where

M_{i1} = processing time of job i on machine M_1 for operation 1

M_{i2} = processing time of job i on machine M_2 for operation 2

M'_{i1} = processing time of job i on machine M_1 for operation 3

M'_{i2} = processing time of job i on machine M_2 for operation 4

4. PROPOSED HEURISTIC METHOD (2-JOBS AND 2-MACHINES)

Compute the standard deviation (SD) of processing times for all operations on both machines for jobs 1 and 2. Compare the SDs and prioritize processing the job with the higher SD before the one with the lower SD. In the event of a tie, choose arbitrarily.

5. EXAMPLE

Consider the following reentry flow-shop problem:

Job J_i	Loop 1		Loop 2	
	Operation 1 (Machine M_1)	Operation 2 (Machine M_2)	Operation 3 (Machine M_1)	Operation 4 (Machine M_2)
	M_{i1}	M_{i2}	M'_{i1}	M'_{i2}
J_1	4	3	5	4
J_2	3	4	4	3

Solution:

As the standard deviation for job-1 (0.5) exceeds that of job-2 (0.25), job-1 will be processed first.

Job J_i	Loop 1	Loop 2
	Operation 1 (Machine M_1) M_{i1}	Operation 3 (Machine M_1) M'_{i1}



	Start time	Processing time	Completion time	Start time	Processing time	Completion time
J₁	0	4	4	7	5	12
J₂	4	3	7	12	4	16
	Operation 2 (Machine M ₂) M _{i2}			Operation 4 (Machine M ₂) M' _{i2}		
J₁	4	3	7	12	4	16
J₂	7	4	11	16	3	19

If we were to process job-2 before job-1, then

Job <i>J_i</i>	Loop 1			Loop 2		
	Operation 1 (Machine M ₁) M _{i1}			Operation 3 (Machine M ₁) M' _{i1}		
	Start time	Processing time	Completion time	Start time	Processing time	Completion time
J₂	0	3	3	7	4	11
J₁	3	4	7	11	5	16
	Operation 2 (Machine M ₂) M _{i2}			Operation 4 (Machine M ₂) M' _{i2}		
J₂	3	4	7	11	3	14
J₁	7	3	10	16	4	20

Clearly, the minimum makespan (19 units) is for **J₁ - J₂**.

6. FOR n-JOBS AND 2-MACHINES

We can employ the suggested heuristic approach for 3 or more jobs (n jobs) and 2 machines. Determine the standard deviation (SD) of processing times for every operation conducted on both machines for all n jobs. Compared to their standard deviations, prioritize processing the job in descending order; this means starting with the job having the higher standard deviation and proceeding to the one with the lower standard deviation. If a tie occurs, select arbitrarily.

7. CONCLUSION

The paper dealt with an important problem in scheduling theory with the objectives to minimize the makespan for n-jobs and 2-machines reentrant flow shop problem. A heuristic method has been developed for finding an optimal schedule or a near-optimal solution. A numerical example is given to illustrate the procedure. This work can be extended for more than two machines.

REFERENCES

[1]. Aspern, M.V., Buld, F., Klein, N.,& Pinedo, M. (2025). Flowshops with reentry: The total weighted completion time objective. *Journal of Scheduling*, 1–16.

[2]. Baker, K.R. (1974). Introduction to sequencing and scheduling.

[3]. Campbell, H. G. Dudek, R. A., & Smith, M. L. (1970). A heuristic algorithm for the n job, m machine sequencing problem. *Management Science*, 16(10), 630.

[4]. Chen, J. S., Pan, J. C. H., & Wu, C. K. (2008). Hybrid tabu search for re-entrant permutation flow-shop scheduling problem. *Expert Systems with Applications*, 34(3), 1924–1930.



- [5]. Choi, S. W., & Kim, Y. D. (2007). Minimizing makespan on a two-machine re-entrant flowshop. *Journal of the Operational Research Society*, 58(7), 972–981.
- [6]. Choi, S. W., & Kim, Y. D. (2008). Minimizing makespan on an m-machine re-entrant flowshop. *Computers & Operations Research*, 35(5), 1684–1696.
- [7]. Danping, L. & Lee, C. K. (2011). A review of the research methodology for the re-entrant scheduling problem. *International Journal of Production Research*, 49(8), 2221–2242.
- [8]. Graves, S.C., Meal, H.C., Stefek, D., & Zeghmi, A. H. (1983). Scheduling of re-entrant flow shops. *Journal of Operations Management*, 3(4), 197–207.
- [9]. Jing, C., Tang, G., & Qian, X. (2008). Heuristic algorithms for two machine re-entrant flow shop. *Theoretical Computer Science*, 400(1–3), 137–143.
- [10]. Lin, D., Lee, C. K., & Ho, W. (2013). Multi-level genetic algorithm for the resource-constrained re-entrant scheduling problem in the flow shop. *Engineering Applications of Artificial Intelligence*, 26(4), 1282–1290.
- [11]. Pan, J. H. & Chen, J. S. (2003). Minimizing makespan in re-entrant permutation flow-shops. *Journal of the operational Research Society*, 54, 642–653.
- [12]. Rifai, A. P., Kusumastuti, P. A., Mara, S. T.W., Norcahy, R., & Dawal, S. (2021). Multi-operator hybrid genetic algorithm-simulated annealing for reentrant permutation flow-shop scheduling. *ASEAN Engineering Journal*, 11(3), 109–126.
- [13]. Yu, T. S., & Pinedo, M. (2020). Flow shops with reentry: Reversibility properties and makespan optimal schedules. *European Journal of Operational Research*, 282(2), 478–490.