Advance Algorithm for Scheduling Problems: JOHNSON'S with Heuristic's Procedure

Abstract- *The job-shop problem of scheduling n≥2 jobs in m≥2 machines is a non-polynomial (NP) hard problem with no general method. Multiple machine scheduling problems are challenging due to the complexity arising from finding an optimal sequence of jobs in the face of several and sometimes, conflicting criteria functions. In this work we modified the Johnson's algorithm, creating a new heuristic algorithm, which schedules n-jobs (n≥2) in multiple machines (n≥3) directly without conversion to a two machine problem. The numerical illustration results obtained from the heuristic algorithm for the multiple machine scheduling problems are compared with solution from the Palmer's heuristic and found to produce better results for makes-span and idle times. Other heuristics can be modified and compared with results found in this work.*

Keywords- *Scheduling Problem, Johnson's Algorithm, Palmer's Heuristic, Makespan, Idle Time*

1. Introduction

In real world, scheduling operation is a very difficult task in planning and managing of any production process. Scheduling is an approach through which resources are allocated to complete the tasks execution process in time. Gupta and Chauhan (2015) opined that the toughness and easiness of scheduling tasks depends on the shop environment, process constraints and the performance measures. Chandry (2012) defined scheduling as the allocation of resources over a period of time to execute a number of tasks with the aim of optimizing certain objectives. Scheduling algorithms are employed to determine the optimal sequence of machines to be used for optimal production. The application of the algorithm depends on the number of tasks and machines in a particle problem. The Johnson algorithm (1954) algorithm is one of such procedures to fine optimal solution for scheduling n ≥2 on m-machine (m≥2), but cannot be applied to n-jobs and m-machines problem directly.

According to Tsetimi (2010), scheduling becomes more complex as the numbers of job and machine increases and there are no easy means or general method to solve the problem mathematically. In general, if there are n-jobs to be processed on m-machine, there are $(n!)^m$ possible ways. Stevenson (2007) opined that the Johnson's rule is a technique that managers can use to minimize the makespan for a group of jobs to process on two machines or two consecutive work centers. For the Johnson's rule to be applied to more than two machine problem the problem has to be converted to a virtual two machine problem and this convention process can be highly challenging as the machines increases. To reduce the time and make the process easier to compute, we modified the Johnson's algorithm for the multiple (m≥3) machine problems.

2. Review of Related Literatures

In our day to day activities, we come across situations were one needs to save time needed to process services. Scheduling plays an important role in finding optimal sequence which

minimizes the total time required to complete all tasks (makespan). Breit (2004) considered the problem of scheduling in jobs in two machine flow shop where the second machine is not available for processing during a given time interval. They proposed a fast approximation algorithm with objective to minimise makespan. The best fast approximation algorithm for this problem guarantees relative worst-case error bound. Haq *et al*, (2008) used genetic algorithm for optional allocation and scheduling of jobs is multiple processing line of parallel-line job-shop in order to minimize makespan. Bansal and Singh (2022a), compares natural and optimised algorithms to address the task scheduling problem in cloud computing. Their method effectively handles real and dynamic tasks by combining the bees and grey wolf algorithm to provide a deeper analysis and understanding to the problem.

Chia and Lee (2009) analysed the total completion time problem in a permutation flow shop within a learning effect. The objective is to minimize the sum of completion time. They used the dominance rule and several lower bounds to speed up the search for the optimal solution. Eren and Guner (2008) developed a bi-criteria flow shop scheduling problem with learning effect. They considered learning effect in a two-machine flow shop with objective of finding a sequence that minimized a weighted sum of total complexion time and makespan. Chihaoui *e*t *al*., (2011) studied the two-machine no wait flow shop scheduling problem, when every machines is subject to one non-availability constraint and jobs have different release dates with the aim of minimizing the makespan. They proposed several lower and upper bounds and incorporated in a branch and bound algorithm. Bansal and Singh (2023), proposed an algorithm for resource allocation in cloud environment, their method which uses the workflow sim toolkit was tested using some exiting work requests and the Amazon EC2 pricing model. Yang and Wang (2011) considered the minimisation of total weighted complexion time in a two machine flow shop under simple deterioration. Their objective is to obtain a sequence so that the total weighted completion time is minimised. Jayakumar *et al.*, (2016) developed a heuristic approach for solving 2-machine n-jobs flowshop scheduling problem with the objective of minimising the makespan. They made a comparison with the Johnson algorithm and it was found that the proposed algorithm is superior to Johnson's algorithm. Bansal and Singh (2022b), gave a detail reviews of various work on scheduling, highlighting recent trends and approaches that addresses challenges for task scheduling problem. Although various studies have proposed many approaches, but it is challenging to find the most straight foreword method to determine the optimal sequence for solving the problem of n-jobs on multiple machines.

2. Johnson's Method

The Johnson's algorithm developed by Johnson (1954) is an efficient algorithm for solving the two machine problem. Let A and B be the machines, it is assumed that the jobs must be processed on machine A first and then on machine B. Suppose we j have jobs to be processed on two machines A and B, $j \le 1 \le n$

A_j = Processing time of job j on machine A

B_j = Processing time of job j on machine B

The jobs are processed in such a way that job j precedes job $j+1$,

If,

$$Min(A_j, B_{j-1}) < Min(A_{j+1}, B_j) \qquad 1$$

The problem of scheduling n-jobs on more than two machines is considerably more complex. Johnson tries to reduce the problem to a two machine problem. For the three

machines problem, it can be reduced essentially to a two machine problem provided the following condition is satisfied:

$$\left.\begin{array}{l} Min\ A_j \geq Max\ B_j \\ \quad or \\ Min\ C_j \geq Max\ B_j \end{array}\right\} \qquad 2$$

By defining

$$\left.\begin{array}{l} A_j^1 = A_j + B_j \\ B_j^1 = B_j + C_j \end{array}\right\} \qquad 3$$

where,

A_j = Processing time of job j on machine A

B_j = Processing time of job j on machine B

C_j = Processing time of job j on machine C

Thereafter, the problem is then solved using the new sets A_j^1 and B_j^1 obtained above following the Johnson's algorithm.

Johnson's Algorithm as found in French (1982) is as follows:

Step1: List all jobs and their processing time for each machine

Step2: Select the jobs with the smallest processing time.

i. If it appears in the first machine, schedule that job first.

ii. If it appears in the second machine, schedule that job last.

iii. Break ties arbitrarily.

Step3: Delete the scheduled job and its processing time.

Step5: Repeat step 2 and 3 until all jobs are scheduled.

2.1 Definitions and Notification

The problem considered in this research is described as follow:

For a set of n-jobs (n≥2) which need to be processed on a set of m-machine (m≥3), we wish to identify a set of sequences for the jobs so as to minimize makespan.

That is:

$$MinC_{\max} \quad , i = 1, \quad 2, \quad ..., \quad n \qquad 4$$

where,

$$C_{ij} = \sum_{i=1}^{m}\sum_{j=1}^{n} P_{ij}$$ hi

P_{ij} = The processing time of job j on machine i

C_{ij} = The completion time of job j on machine i

2.2 Assumptions

The following assumptions were adapted for this study, (French 1982):

(i) each job can be processed on one machine at a time and each machine can process only one job at a time.

(ii) there is no preemption. That is the processing time of a job on machine cannot be interrupted.
(iii) there are no precedence constraints
(iv) jobs follow the same processing order on the machines.

3. Modified Johnson's Algorithm

Step1: List all jobs and their processing time for each machine
Step2: Select the job with the smallest processing time
i. If it is on machine1, schedule that job in the first available position
ii. If it is on last machine n, schedule it in the last available position.
iii. If it is on the machine $i : 1 < i < n$, then go to step 3
Step3: find the sum of all processing time of that job above machine i and also
` the sum below machine i
That is:

$$\sum_{m=1}^{i-1} P_{jm} \qquad 5$$

and

$$\sum_{m=i+1}^{n-1} P_{jm} \qquad 6$$

i if the sum above i is smaller than the sum below i
That is:
$\sum_{m=1}^{i-1} P_{jm} < \sum_{m=i+1}^{n-1} P_{jm}$,
schedule that job in the first available position.
ii. If, $\sum_{m=i+1}^{n-1} P_{jm} < \sum_{m=1}^{i-1} P_{jm}$,
schedule the job i in the last available position.
step4: Delete the scheduled job and its processing time.
step5: Repeat step 2 to 4, until all jobs are scheduled

3.1 Numerical Illustration

Example 1 (French, 1982)

Consider 5/4/F/F_{max} problem with the following data

Jobs

Machines	1	2	3	4	5

1	7	11	2	14	18
2	15	18	13	4	11
3	14	18	11	27	32
4	21	6	16	14	16

Solution

Applying algorithm (3.0)

Job 3 is scheduled first		3	-	-	-	-
Job 4 is scheduled second	3	4	-	-	-	
Job 2 is scheduled fifth		3	4	-	-	2
Job 1 is scheduled third		3	4	1	-	2
Job 5 is scheduled fourth		3	4	1	5	2

Hence, the optimal sequence is (3, 4, 1, 5, 2)

Example 2

Consider 4/5/F/F_{max} problem with data

Jobs

Machines	1	2	3	4
1	6	5	4	7
2	4	5	3	2
3	1	3	4	2
4	2	4	5	1
5	8	9	7	5

Solution

We build up the optimal sequence as follows

Job 3 is scheduled first		3	-	-	-
Job 4 is scheduled fourth		3	-	-	4
Job 2 is scheduled third	3	3	-	2	4
Job 1 is scheduled second	3	1	2	4	

Hence, the optimal sequence is (3, 1, 2, 4)

3.2 Comparison between Palmer's Algorithm and Modified Johnson's Algorithm

Palmer's Algorithm								
Job Sequence	**M_1**		**M_2**		**M_3**		**M_4**	
	S_j	**C_j**	**S_j**	**C_j**	**S_j**	**C_j**	**S_j**	**C_j**

1	0	7	7	22	22	36	36	57
5	7	25	25	36	36	68	68	84
4	25	39	39	43	68	95	95	109
3	39	41	43	56	95	106	109	125
2	41	52	56	74	106	124	125	131

Table 1: Palmer's Algorithm for 5-jobs 4-machines problem

Modified Johnson's Algorithm								
Job Sequence	**M_1**		**M_2**		**M_3**		**M_4**	
	S_j	**C_j**	**S_j**	**C_j**	**S_j**	**C_j**	**S_j**	**C_j**
3	0	2	2	15	15	25	25	41
4	2	16	16	20	25	52	52	66
1	16	23	23	38	52	66	66	87
5	23	41	41	52	66	98	98	114
2	41	52	52	70	98	116	116	122

Table 2: Modified Johnson's Algorithm for 5-jobs 4-machines problem

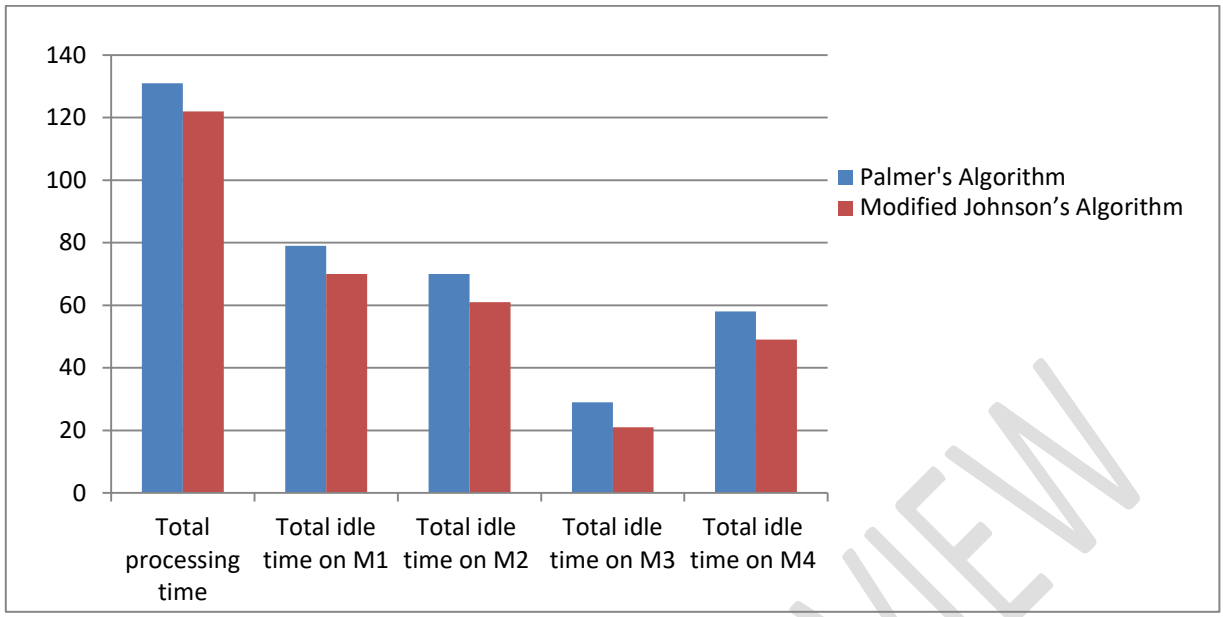

Figure 1: 5-jobs 4-machines problem

Palmer's Algorithm										
Job Sequence	M_1		M_2		M_3		M_4		M_5	
	S_j	C_j	S_j	C_j	S_j	C_j	S_j	C_j	S_j	C_j
2	0	5	5	10	10	13	13	17	17	26
3	5	9	10	13	13	17	17	22	26	33
1	9	15	15	19	19	18	22	24	33	33
4	15	22	22	24	24	26	26	27	41	46

Table 3: Palmer's Algorithm for 4-jobs 5-machines problem

Modified Johnson's Algorithm										
Job Sequence	**M_1**		**M_2**		**M_3**		**M_4**		**M_5**	
	C_j	**S_j**	**C_j**	**S_j**	**C_j**	**S_j**	**C_j**	**S_j**	**C_j**	**S_j**
3	0	4	4	7	7	11	11	16	16	23
1	4	10	10	14	14	15	16	18	23	31
2	10	15	15	20	20	23	23	27	31	40
4	15	22	22	24	24	26	27	28	40	45

Table 4: Modified Johnson's Algorithm for 4-jobs 5-machines problem

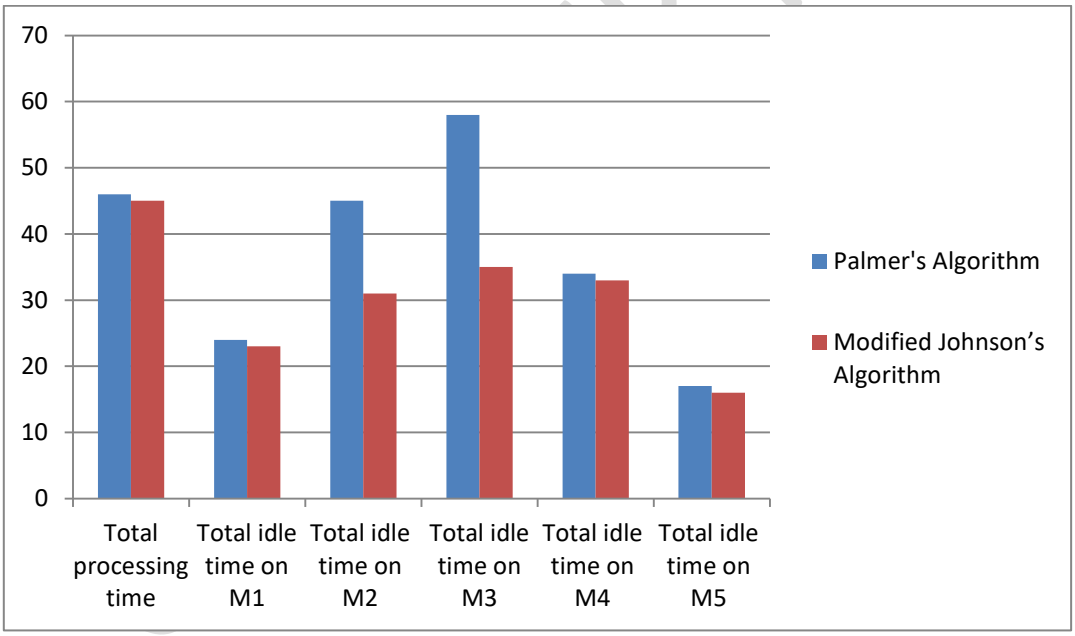

Figure:2 4-jobs 5-machines problem

4. Discussion of Results

From the computational analysis carried out, the following calculations are obtained from **Table 1**, total processing time =131, total idle time on M_1 =79, total idle time on M_2 =70, total idle time on M_3 =29 and total idle time on M_4 =58 and from **Table 2**, we have total processing time to be 122, total idle time on M_1 =70, total idle time on M_2 =61, total idle time on M_3 =21 and total idle time on M_4 =49. **Table 3** and **4** show the results of the Palmer's heuristic and the modified Johnson's algorithm for the 4-jobs,

5-machines problem. In **figure 1** and 2, the graphical comparison of the Palmer's heuristic and modified Johnson's algorithm for the numerical examples was given. It can be seen that the modified Johnson's heuristic yielded a minimum values for makespan and idle time on the machines than the Palmer's heuristic for most of the problems.

5. Conclusion

Scheduling problem typically involve allocating resources to tasks over a given time frame. It is obvious that to remain competitive in today's open market, manufacturing and service providing organizations need to manage their resources in an effective way. There is no general method for scheduling n-jobs ($n \geq 2$) on m-machines ($m \geq 3$). Thus we developed a new heuristic algorithm for the problem based on the Johnson's heuristic. On the basis of our analysis and computation our results have shown that the modified Johnson's algorithm is efficient for scheduling $n \geq 2$ jobs on $m \geq 3$ machines.

Disclaimer (Artificial intelligence)

The authors hereby declare that NO generative AI technologies such as Large Language Models (ChatGPT, COPILOT, etc.) and text-to-image generators have been used during the writing or editing of this manuscript.

REFERENCES

Breit, J. (2004). An Improved Approximation Algorithm for Two Machine Flow Shop Scheduling with Availability Constraints. *Information Processing Letters*, 90(6): 273-278.

Bansal, N. & Singh, A.K. (2023). Development of a New Task Scheduling in Cloud Computing. *International Journal of System Assurance Engineering and Management*, 14:2267-2275.

Bansal, N. & Singh, A.K. (2022a). Effective Task Scheduling Algorithm in Cloud Computing with Quality of Service Alert Bees and Grey Wolf Optimization. *Indonesian Journal of Electrical Engineering and Computer Science*, 25(1): 550-560.

Bansal, N. & Singh, A.K. (2022b). Valuable Survey on Scheduling Algorithms in the Cloud with Various Publications. *International Journal of System Assurance Engineering and Management, 13:1-19*.

Chandry, I.A. (2012). Job Shop Scheduling Problem with Alternative Machine using Genetic Algorithms. *Journal of Central South University of Technology*, 19(5):1322-1333.

Chia & Lee. (2009). Minimizing the Total Completion Time in Permutation Flow Shop. *Computers and Operations Research*, 6:2111-2121.

Chihaoui, F. B. , Kacem, I. , Hadj-Alouane, A. B. , Dridi, N. & Rezg, N. (2011). No Wait Scheduling of a Two Machine Flowshop to Minimize the Makespan Under Non-Availability Constraints and Different Release Dates. *International Journal of Production Research, Taylor Francis*, pp.1.10.1080/007543.2010.531775.hal-00662385.

Eren, T. & Guner, E. (2008). Bicriteria Flow Shop Scheduling. *Applied Mathematical Modeling*, 32(9):1719-1733.

French, S. (1982). *Sequencing and Scheduling: Ellis Horwood Sense in Mathematics and its Application*, (Series Editor).Bell G.M.

Gupta, A. & Chauhan, S.R. (2015). A Heuristic Algorithm for Scheduling in a Floswshop Environment to Minimize Makespan. *International Journal of Industrial Engineering Computations*, 3:185-198.

Haq, A.N., Balasubramanian, K., Sashidharan, B. & Karthick, R.B (2008). Parallel Line Job Shop Scheduling using Genetic Algorithm. *International Journal of Advanced Manufacturing Technology*, 35(9-10):1047-1052.

Hong, T. P., Huang, P. Y. & Horng, G. (2006). Using the LPT and the Palmer's Approaches to Solve Group Flexible Flow- Shop Problems. *International Journal of Computer Science and Network Security*, 6(3A):98-104.

Jayakumar, S., Maganathan, R. & Shanthin, S. (2016). A Heuristic Approach or Solvintg Machine n-Jobs Flow Shop Scheduling Problem with Makespan Objective. *IOSR Journal of Mathematics*, 12(3): 23-26.

Johnson, S.M. (1954). Optimal two and Three Stage Production Schedules with Set-Up Time Included. *Naval Research Logistics Quarterly*, 1:61-68.

Palmer, D.S. (1965). Sequencing Jobs Through a Multi Stage Process in the Minimum Total Time: A Quick Method of Obtaining Near Optimum. *Operational Research Quarterly*, 16: 101-107.

Stevenson, W.J. (2007). *Production and Management*. McGraw-Hill Companies Inc. Boston.

Tsetimi, J. (2010). The Use of Heuristics for Single Machine Scheduling Problem. *Journal of Engineering for Development*, 9:1-14.

Yang, N.H. & Wang, J.B. (2011). Minimizing Total Weighted Completion Time Under Simple Linear Deterioration. *Applied Mathematics and Computation, 217(9): 4819-4826*.