MINIMIZE MAKESPAN IN HYBRID FLOW SHOP SCHEDULING PROBLEM USING-NEH METHOD

[1] Dr.T.Prabaharan, [2] M.L.Brabin Nivas
[1]Senior Professor, [2] Assistant Professor
[1] Departement of Mechanical Engineering,
[1] Mepco Schlenk Engineering College,Sivakasi, Tamil Nadu

Abstract : Flowshop scheduling using NP-hard (non-deterministic polynomial-time hard) is very hard. It is difficult to solve with NP method the number of unequal parallel machines choices per stage in multistage hybrid flowshop scheduling. This paper attempts to solve flowshop scheduling problem using NEH Method. The performance of the problem was benchmarked against available Hybrid problem to solve Hybrid flow shop .The outcome of this problem gives the best result than the other hybrid problem available present.

IndexTerms -. Minimum Makespan, Flowshop schedule

Introduction

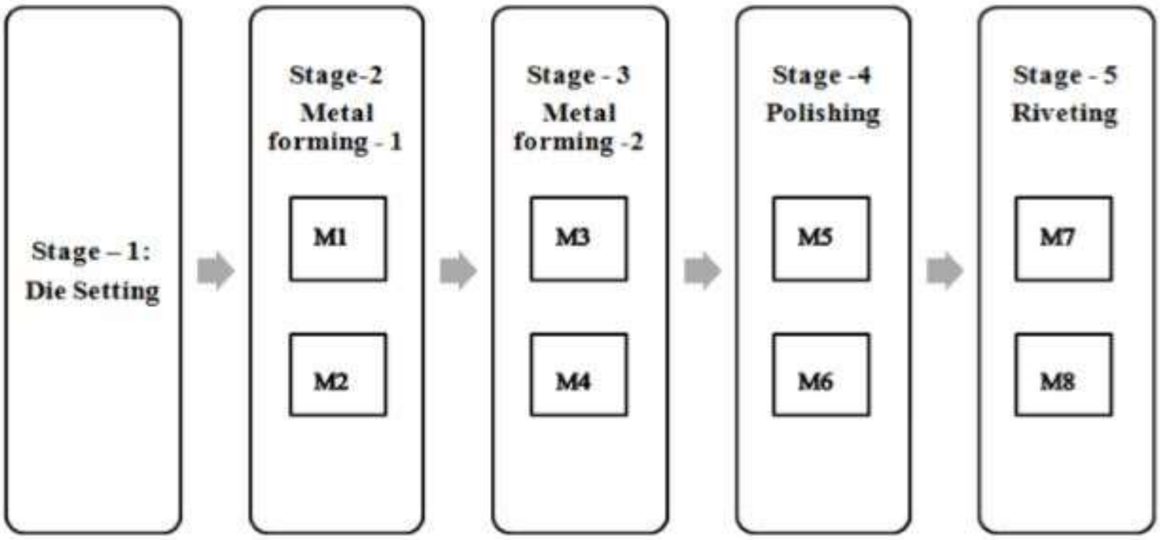
Scheduling is an important tool for manufacturing and engineering, where it can have a major impact on the productivity of a process. Flow shop scheduling problems are a class of scheduling problems with a workshop or group shop. If there is more than one machine and there are multiple jobs, then each job must be processed by corresponding machine or processor. That means with operation of each job must be processed on. Especially the maintaining of a continuous flow of processing tasks is desired with a minimum of *idle time* and a minimum of *waiting time*. Flow shop scheduling is a special case of job scheduling where there is strict order of all operations to be performed on all jobs. Solution methods of Flow shop scheduling are Branch and Bound, Dynamic programming, Heuristic algorithm and Meta-heuristics. This paper focuses on the hybrid flowshop scheduling problem. A Hybrid Flow Shop (HFS) consists of series of production stages, each of which has several machines operating in parallel, that can be identical, uniform or unrelated. Some stages may have only one machine, but at least one stage must have multiple machines. The flow of jobs through the shop is unidirectional. Each job is processed by one machine in each stage and it must go through one or more stage (Elmaghraby & Kamoub, 1997). As HFS is NP Hard many heuristic and metaheuristic methods are used to solve this problem such as (i) simple heuristic algorithms for the two stage flexible flow shop problems (ii) several heuristics for three-stage HFS scheduling problems to minimize makespan (iii) simple heuristic algorithms for flexible flow shop problems; and metaheuristic methods such as genetic algorithm, stimulated annealing algorithm, ant colony optimization algorithm, particle swarm optimization algorithm and artificial immune system algorithm, bat algorithm.

A HFS problem is solved by Ravianandan M and M. Om kumar using Improved Hybrid ACO Cuckoo Algorithm to minimize makespan. The problem is reviewed from International Journal of Computer Applications (0975 – 8887) Volume 115 – No. 18, April 2015 where Hybrid Flow Shop Scheduling using Improved Hybrid ACO Cuckoo Algorithm to Minimize Makespan is analysed. In this paper the above HFS problem is solved by using Flow shop scheduling using NEH Method by selecting best machine from each stages to minimise the makespan than the proposed Algorithm.

2. Problem definition

The problem is defined as follows. Let us consider a set of 10 jobs to be processed on 5 consecutive stages to minimize both makespan and mean flow time. Makespan is the completion time of the last job in the production system. Makespan is important for measuring the system utilization. Mean flow time is the average time spent by the jobs in the production system

ARRANGEMENT OF MACHINES



1 Hybrid Flow Shop – Metal Spinning Process

Job Processing time (in seconds)									
Sl No	Stage 1 - die Setting	Stage 2 - Metal Forming 1		Stage 3 - Metal forming 2		Stage 4 Polishing		Stage 5 - Riveting	
		M1	M2	M3	M4	M5	M6	M7	M8
1.	600	52	50	44	45	54	56	0	0
2.	580	63	60	46	48	53	53	0	0
3.	560	53	52	0	0	82	88	0	0
4.	600	54	52	0	0	64	64	15	12
5.	640	54	52	0	0	64	64	15	12
6.	600	30	31	0	0	9	8	0	0
7.	620	37	37	42	43	27	26	0	0
8.	600	36	36	45	43	43	45	0	0
9.	611	63	64	51	50	49	49	0	0
10.	588	54	54	52	57	50	50	0	0

Constraints shown

It is assumed that there is no no-wait constraint and that enough buffer space is made available just in case in this work. The assumptions are

1. Performance of only one operation at a time by each machine

2. Pre-emption is impossible

3. No re-sequence of job after the first stage

4. Process time includes set-up time as well

5. One job at one stage

6 The movement of job is stage by stage only

7. Every job can be processed by not more than a single machine at a time

A mathematical model of the proposed scheduling operation is given below and the notations are described along with.

J_i where $1 \leq i \leq n$ = Time at which the previous process was completed on a job.

M_k where $1 \leq k \leq m$ = Time at which the previous process was completed on a machine

$P_{i,j}$ = Process time for job i on machine j

Initially, $M_j = 0$ and $J_i = 0$

J_{i1} = Increment in machine and job completion times for a job i in the first stage

$JC_{i1} = P_{ij} \mid (M_j + P_{i-1j})$ is the minimum for any j,where j= 1, 2... mq1 is the selected machine in first stage

$$J_i = M_j + JC_{i1}$$

$$M_j = M_j + JC_{i1}$$

$JC_{i2,3\,.p}$ = Increment in machine and job completion times for a job in the second or third stage

$JC_{i2,3,p} = P_{i,j} \mid (\max(M_j, J_i) + P_{i,j})$ is the minimum for any j, where j= mq1+1,…,mq2 is the selected machine in second stage or any such stage.

$$J_i = \max(J_i, M_j) + JC_{i2,3,..p}$$

$$M_j = \max(J_i, M_j) + JC_{i2,3\ldots p}$$

MAKESPAN CALCULATION

Objective Function is to minimize make span i.e time between start of first job and completion of last job.

$$Cmax = Max(C_{i-1j}, C_{j-1i}) + P_{ij}$$

Where C_{i-1j}, C_{j-1i} = Completion time of previous operation.

P_{ij} = Processing time of next Operation.

Subject to machine availability in each stage to process the job

SELECTION OF MACHINE PATTERN

- Odd machine pattern (M_1,M_3,M_5,M_7)
- Even machine pattern (M_2,M_4,M_6,M_8)
- Better timing machine pattern.
- For these three patterns same sequence of job got by both LPT and SPT rules.
- **J_6–J_3–J_8–J_7–J_4–J_2–J_{10}–J_1–J_5–J_9 (SPT)**
- **J_6–J_7–J_8–J_4–J_5–J_3–J_1–J_{10}–J_2–J_9 (LPT)**

But the make span for better timing pattern is best compared with other two.

BETTER TIMING PATTERN

JOB	DIE SETTING TIME	STAGE 1	STAGE 2	STAGE 3	STAGE 4
J_1	600	50	44	54	0
J_2	580	60	46	53	0
J_3	560	52	0	82	0
J_4	600	52	0	64	15
J_5	640	52	0	64	15
J_6	600	30	0	8	0
J_7	620	37	42	26	0
J_8	600	36	43	43	0
J_9	611	63	50	49	0
J_{10}	588	54	52	50	0

APPLICATION OF NEH RULE

J_6-J_7

JOB	M_1	M_2	M_3	M_4
J_6	61	61	78	78
J_7	135	220	273	273

JOB	M1	M2	M3	M4

J_7-J_6

J7	74	159	212	212
J6	135	159	229	229

229 is the best make span time. So the sequence **$J_7 – J_6$** is selected

By applying the NEH Rule, the following sequence is obtained.

- J_8-J_7-J_6 is selected. (Best Time :318 seconds)
- J_4- J_8-J_7-J_6 is selected. (Best Time :424 seconds)
- J_5- J_4- J_8-J_7-J_6 is selected. (Best Time :530 seconds)
- J_3- J_5- J_4- J_8-J_7-J_6 is selected. (Best Time :689 seconds)
- J_3- J_5- J_1-J_4- J_8-J_7-J_6 is selected. (Best Time :799 seconds)
- J_3- J_5- J_1-J_4- J_8-J_{10}-J_7-J_6 is selected. (Best Time :899 seconds)
- J_3- J_5- J_1-J_4- J_8-J_{10}-J_2-J_7-J_6 is selected. (Best Time :1005 seconds)
- J_3- J_5- J_1-J_4- J_8-J_{10}-J_2-J_9-J_7-J_6 is selected. (Best Time :1101seconds)

From the above sequences, the best sequence is

J_3- J_5- J_1-J_4- J_8-J_{10}-J_2-J_9-J_7-J_6

RESULT OF CUCKOO ALGORITHM

NO OF JOBS	NO OF STAGES	MACHINES PER STAGE	ALGORITHM	SEQUENCE	MAKESPAN
10	5	2	FIFO	1 2 3 4 5 6 7 8 9 10	7141
10	5	2	ACO-CS	9 1 6 4 10 3 2 8 7 5	7039
10	5	1	NEH METHOD	3 5 1 4 8 10 2 9 7 6	6137

CONCLUSION

To conclude the whole analysis of the results, and with the aim of making a deeper analysis, the convergence behavior shown by the cuckoo algorithm is compared next with the ones shown by the NEH method. We have selected the NEH method for this comparison because they are the simpler techniques in terms of average results quality to proceed with initially. Analyzing the results, it can be concluded that the NEH performs better than the Cuckoo algorithm in instances with, approximately 900 seconds where the NEH has proved to be better in general terms in this problem. However, we think using other similar optimizations will reduce the makespan and can provide some better insight into these methods.

REFERENCES

[1] T.S. Arthanary and K.G. Ramaswamy, “An extension of two machines sequencing problem”, OPSEARCH, Journal of the Operational Research Society of India, Vol. 8, No. 4, pp. 10–22, 1971.

[2] Chung-Yee Lee and George L. Vairaktarakis, “Minimizing makespan in hybrid flowshops”, Operations Research Letters, Vol. 16, No. 3, pp. 149–158, 1994.

[3] J.N.D. Gupta, A.M.A. Hariri and C.N. Potts, “Scheduling a two-stage hybrid flow shop with parallel machines at the first stage”, Annals of Operations Research, Vol. 69, pp. 171–191, 1997.

[4] C. Sriskandarajah and S.P. Sethi, “Scheduling algorithms for flexible flowshops: worst and average case performance”, European Journal of Operational Research, Vol. 43, No. 2, pp. 143–160, 1989.

[5] H. Soewnadi and S.E. Elmaghraby, “Sequencing three- stage flexible flowshops with identical machines to minimize makespan”, IIE Transactions, Vol. 33, No. 11, pp. 985–994, 2001.

[6] S.A. Brah and L.L. Loo, “Heuristics for scheduling in a flow shop with multiple processors”, European Journal of Operational Research, Vol. 113, No. 1, pp. 113–122, 1999.

[7] F.Y. Ding and D. Kittichartphayak, “Heuristics for scheduling flexible flow lines”, Computers and Industrial Engineering, Vol. 26, No. 1, pp. 27–34, 1994.

[8] E. Figielska, “A new heuristic for scheduling the two-stage flowshop with additional resources”, Computers and Industrial Engineering, Vol. 54, No. 4, pp. 750–763, 2008.

[9] E. Rashidi, M. Jahandar M and M. Zandieh, “An improved hybrid multiobjective parallel genetic algorithm for hybrid flow shop scheduling with unrelated parallel machines”, The International Journal of Advanced Manufacturing Technology, Vol. 49, No. 9-12, pp. 1129–1139, 2010.

[10] C. Low, “Simulated annealing heuristic for flow shop scheduling problems with unrelated parallel machines”, Computers and Operations Research, Vol. 32, No. 8, pp. 2013–2025, 2005.

[11] H.M. Wang, F.D. Chou and F.C. Wu, “A simulated annealing for hybrid flow shop scheduling with multiprocessor tasks to minimize makespan”, The International Journal of Advanced Manufacturing Technology, Vol. 53, No. 5-8, pp. 761–776, 2011.

[12] K. Alaykyran, O. Engin and A. Doyen, “Using ant colony optimization to solve hybrid flow shop scheduling problems”, The International Journal of Advanced Manufacturing Technology, Vol. 35, No. 5-6, pp. 541–550, 2007.

[13] M.R. Singh and S.S. Mahapatra, “A swarm optimization approach for flexible flow shop scheduling with multiprocessor tasks”, The International Journal of Advanced Manufacturing Technology, Vol. 62, No. 1-4, pp. 267-277, 2012.

[14] Q. Niu, T. Zhou and S. Ma, “A quantum-inspired immune algorithm for hybrid flow shop with makespan criterion”, Journal of Universal Computer Science, Vol. 15, No. 4, pp.765–785, 2009.

[15] O. Engin and A. Doyen, “A new approach to solve hybrid flow shop scheduling problems by artificial immune system”, Future Generation Computer Systems, Vol. 20, No. 6, pp. 1083–1095, 2004.