A Constructive Heuristic for Makespan Minimization of Flowshop Scheduling

Umarali K
Sridharan R
National Institute of Technology
(umarali@nitc.ac.in)
(sreedhar@nitc.ac.in)

Radha Ramanan T
Indian Institute of Management Indore
(ramanan@iimidr.ac.in)

ISBN: 978-81-924713-8-9

This paper considers the permutation flow shop scheduling problem with the objective of minimizing the makespan. A constructive heuristic algorithm is proposed. The algorithm is a modification of the well-known NEH heuristic. The proposed heuristic algorithm, named as NEH-SD uses two parameters, the mean and standard deviation of processing times for sorting the jobs in sequence to have the initial sequence of jobs. The heuristic is tested against two well-known heuristics from the literature, namely, NEH and CDS. The computational experiments are carried out with 120 benchmark problem data sets. The computational experiments show that the proposed algorithm is a feasible alternative for practical application when solving n-job and m-machine flow-shop scheduling problems to give relatively good solutions in a short time interval.

1. Introduction

In the current competitive environment, most of manufacturing industries meet with the problem of effectively committing their resources among varieties of possible orders. The search for an optimal allocation of resources for performing a set of jobs within each work order is the main role of scheduling. According to Hejazi and Saghafian (2005), main problems in scheduling of jobs in manufacturing are, "priorities" and "capacity". Scheduling problem is an effort "to specify the order and timing of the processing of the jobs on machines, with single objective or multiple objectives." In this paper, we focus on flowshop environment where all jobs have to follow the same route in the same order and where machines are assumed to be set up in a series. The general flow-shop problem with a makespan (*Cmax*) objective can be denoted as an *n*/*m*/*F*/ *Cmax* that involves *n* jobs where each requiring operations on *m* machines, in the same job sequence [French, 1982]. The solution of such a problem is represented by the optimal job sequence that produces the smallest makespan, assuming no preemption of jobs. The general flow-shop problem is known to be NP-hard when number of machines is greater than two.

The rest of the paper is organized as follows. Section 2 contains a brief discussion on permutation flowshop scheduling problem. The literature survey is included in section 3. The proposed constructive heuristic method is explained in detail in section 4. The results are discussed in section 5. Conclusions and future scope are given in the section 6.

2. Flowshop Scheduling Problem

In discrete parts manufacturing industries, jobs with multiple operations use machines in the same order. In such a case, machines are installed in series. Raw materials initially enter the first machine and when a job has finished its processing on the first machine, it goes to the next machine. When the next machine is not immediately available, the job has to wait till the machine becomes available for processing. Such a manufacturing system is called a flowshop production system. Here, the machines are arranged in the order in which operations are to be performed on jobs. The technological order, in which the jobs are processed on different machines, is unidirectional. In a flowshop, a job i with a set of m operations i_1 , i_2 , i_3 , ..., i_m is to be completed in a predetermined sequence. In short, each operation except the first has exactly one direct predecessor and each operation except the last one has exactly one direct successor. Thus, each job requires a specific immutable sequence of operations to be carried out for it to be complete. Further, once started, an operation on a machine cannot be , interrupted. According to Baker (1974), this type of structure is referred as linear precedence structure. Here, we consider general flowshop scheduling with unlimited intermediate storage, where it is not allowed to sequence changes between machines. In this flow shop, referred to as permutation flow shop, the same sequence of jobs is maintained throughout. Here, our attention is limited to permutation schedules with constant setup times which are included in processing times. We assume the availability of all jobs at zero time.

3. Literature Review

The scheduling literature provides a rich knowledge of the general flow-shop scheduling problem to get permutation schedules with minimal makespan. This is a very popular topic in scheduling circles. Taylor [1911] and Gantt [1919] give the first scientific consideration to production scheduling. Pinedo [2008] is a good reference for all types of scheduling problems and scheduling systems including flow-shop environment. Production scheduling systems that emerged later were mostly connected to shop floor tracking systems and were dispatching rules to sequence the work . Similar scheduling systems are today implemented in ERP systems that were performed in the early 1990s. Modrak [2010] discusses manufacturing execution systems (MES) with integrated scheduling systems in the role of a link interface between a business level and shop floor.

Solution methodologies for the permutation flow-shop scheduling problem range from simple constructive to more complex as meta-heuristic approaches. Johnson [1954] first presented an algorithm that can find the optimum sequencing for an n-job and 2-machine problem. The concept of a slope index as a measure to sequence jobs was firstly introduced by Page 1961]. Later on, Palmer [1965] adopted this idea and utilized the slope index to solve job sequencing for the m-machine flow-shop problem. Gupta [1971] argued that the sequencing problem is a problem of sorting n items to minimize the makespan. He proposed alternative algorithm for calculating the slope index to schedule a sequence of jobs for more than two machines in a flow-shop scheduling problem. Campbell et al. [1970] proposed a simple heuristic extension of Johnson's algorithm to solve an m-machines flow shop problem. The extension is known in literature as the Campbell, Dudek, and Smith (CDS) heuristic. Nawaz et al. [1983] proposed the NEH algorithm, which is probably the most well- known constructive heuristic used in the general flow-shop scheduling problem. The basic idea is that a job with the largest processing time should have highest priority in the sequence. Results obtained by Kalczynski and Kamburowski [2007] have also given proof that many meta-heuristic algorithms are not better than the simple NEH heuristic.

The most emphasized names among the contributors of meta-heuristic approaches are as follows: Ogbu and Smith [1990] with their simulated annealing approach; Nowicki and Smutnicki [1996], who implemented tabu search to solve the flow-shop scheduling problem; And Reeves and Yamada [1998], who applied the genetic algorithm for PFSP. The new accession to the family of meta-heuristic scheduling algorithms is a water-flow like algorithm. The Hybrid algorithm, based on the genetic algorithm, was applied in order to find optimal makespan in an n-job and m-machine flow-shop production, In this paper, we focus on developing a constructive heuristics by modifying the NEH heuristics. The results of the proposed approach are compared with CDS and NEH heuristics.

4. The proposed NEH-SD Heuristic

In this section, we formally explain the steps of the constructive heuristic approach used to obtain a good initial solution. The general idea is that we use mean and standard deviation of the processing time to sort the jobs to get the initial sequence

A simulation model of the proposed heuristic was developed in Matlab and the experiments were conducted on a 2.30 GHz Intel Core processor with 2GB RAM. We ran our experiment with the objective of minimizing the makespan on Taillard's benchmark problem datasets, which has 120 instances with 10 each of one particular size. Taillard's datasets range from 20 to 500 jobs and 5 to 20 machines. We coded NEH and CDS in Matlab and ran on a processor with a 2.30 GHz and 2GB RAM to compare the results. The outputs of the proposed algorithm are compared with CDS and NEH heuristic for the above 120 problems.

5. Results and Discussion

In general, constructive heuristics of this type is used to get an initial solution in a short time period. In this approach, we get near optimum make-span values for all the problems. The makespan values obtained for different problem are tabulated below.

Table 1 20 *Job 5 Machine Problems*

Sl.No	CDS	NEH	NEH SD
1	1426	1286	1287
2	1368	1365	1383
3	1357	1248	1241
4	1409	1325	1308
5	1323	1305	1283
6	1279	1228	1227
7	1347	1278	1255
8	1358	1223	1253
9	1371	1291	1254
10	1209	1151	1133

Table 2 *20 Job 10 Machine Problems*

Sl.No	CDS	NEH	NEH SD
11	1743	1680	1660
12	1911	1729	1728
13	1652	1557	1609
14	1571	1441	1439
15	1568	1502	1532
16	1541	1453	1498
17	1635	1562	1576
18	1836	1609	1591
19	1735	1647	1656
20	1820	1695	1653

Table 3 *20 Job 20 Machine Problems*

SL.NO	CDS	NEH	NEH SD
21	2549	2410	2401
22	2336	2181	2197
23	2588	2411	2381
24	2451	2262	2249
25	2535	2397	2363
26	2407	2349	2320
27	2440	2362	2366
28	2330	2249	2275
29	2402	2320	2290
30	2430	2277	2295

Table 4 *50 Job 5 Machine Problems*

Sl.No	CDS	NEH	NEH SD
31	3020	2843	2882
32	3020	2843	2882
33	2871	2640	2633
34	2843	2782	2762
35	3055	2868	2890

Table 5 *50 Job 10 Machine Problems*

Sl.No	CDS	NEH	NEH SD
41	3423	3135	3128
42	3425	3032	3063
43	3241	2986	2973
44	3449	3198	3147
45	3415	3160	3192

Table 6 *50 Job 20 Machine Problems*

SL.NO	CDS	NEH	NEH SD
51	4413	4082	4063
52	4194	3921	3946
53	4287	3927	3888
54	4314	3969	3927
55	4206	3854	3945

36	3144	2850	2838
37	3006	2758	2779
38	2831	2721	2713
39	2780	2576	2564
40	2934	2793	2790

46	3368	3178	3165
47	3516	3277	3271
48	3522	3123	3172
49	3247	3002	2996
50	3505	3257	3214

56	4208	3914	3868
57	4231	3952	3934
58	4262	3938	3985
59	4282	3952	3941
60	4256	4079	3990

Table 7 *100 Job 5 Machine Problems*

Sl.No	CDS	NEH	NEH SD
61	5681	5519	5548
2	5466	5348	5314
63	5378	5219	5207
64	5273	5023	5023
65	5461	5270	5266
66	5259	5139	5139
67	5557	5265	5259
68	5387	5120	5110
69	5758	5489	5487
70	5723	5341	5346

Table 8 *100 Job 10 Machine Problems*

Sl.No	CDS	NEH	NEH SD
71	6209	5846	5918
72	5873	5453	5430
73	6024	5824	5749
74	6377	5929	6013
75	6018	5679	5599
76	5744	5375	5393
77	6201	5706	5704
78	6234	5760	5723
79	6349	6032	6024
80	6387	5918	5938

Table 9 *100 Job 20 Machine Problems*

SL.NO	CDS	NEH	NEH SD
81	6920	6541	6602
82	7087	6523	6499
83	7175	6639	6606
84	7040	6557	6579
85	7218	6695	6599
86	7307	6664	6678
87	7191	6632	6586
88	7358	6739	6761
89	7210	6677	6623
90	7230	6720	6677

Table10 *200 Job 10 Machine Problems*

Sl.No	CDS	NEH	NEH SD
91	11673	10942	10942
92	11438	10746	10716
93	11676	11025	11045
94	11376	11057	10953
95	11384	10645	10621
96	11210	10458	10429
97	11553	10989	10997
98	11470	10829	10825
99	11273	10574	10564
100	11544	10807	10839

Table 11 *200 Job 20 Machine Problems*

Sl.No	CDS	NEH	NEH SD
101	12605	11625	11573
102	12643	11675	11731
103	12770	11852	11835
104	12616	11803	11723
105	12483	11685	11683
106	12439	11629	11619
107	12584	11833	11830
108	12705	11913	11850
109	12754	11673	11680
110	12887	11869	11797

Table 12 *500 Job 20 Machine Problems*

SL.NO	CDS	NEH	NEH SD
111	28545	26697	26670
112	29494	27232	27271
113	28694	26919	26848
114	29063	27055	27036
115	28404	26846	26727
116	28833	26992	27055
117	28684	26823	26797
118	29195	27138	27222
119	28524	26631	26575
120	28887	26984	26968

In the computational experiment, we use 120 problem instances proposed by Taillard. The summary results for Taillard's 120 instances are shown in Tables 1 to 12. Each of the tables displays the results for CDS, NEH alone and NEH SD. The solutions of the proposed algorithm are compared with those of CDS and NEH. From the results, we can make the inference that the overall performance of the proposed heuristic is better than the two tested heuristics for Taillard's 120 problems

6. Conclusion

In the present study, a new constructive heuristic for permutation flowshop scheduling problem with makespan minimization is developed and analysed. The proposed NEH SD heuristic gives better solution than two other heuristics, namely CDS and NEH. The solution obtained using the proposed heuristic can be improved using meta-heuristics such as particle swarm optimization, ant colony optimization, etc.

7. References

1. Campbell, H. G. et al.: A Heuristic Algorithm for The n-job, m-machine Scheduling Problem, Management Science 16, 1970, pp. 630-637
2. Gantt, H. L.: Organizing for Work, New York: Harcourt, Brace, and Howe,1919
3. Gupta, J. N. D.: A Functional Heuristic Algorithm for the Flow Shop Scheduling Problem. Operational Research Quarterly 22, 1971, pp. 39-47
4. Hejazi, S. R., and Saghafian, S.: Flowshop Scheduling Problems with Makespan Criterion: A Review, International Journal of Production Research, 43(14), 2005, pp. 2895-2929

5. Johnson, S. M.: Optimal Two- and Three-Stage Production Schedules with SetuTimes Included, Naval Research Logistics Quarterly 1, 1954, pp. 61- 68
6. Kalczynski, P. J. et al.: On the NEH Heuristic for Minimising the Makespan in Permutation Flowshops, The International Journal of Management Science, 35 (1), 2007, pp. 53-60
7. Modrak, V. et al.: Flow Shop Scheduling Algorithm to Minimize Completion Time for n-jobs m-machines Problem, Tehnicki Vjesnik, 17(3), 2010, pp. 273-278.
8. Nawaz M, Enscore E, Ham I (1983) A heuristic algorithm for them- machine, n-job flowshop sequencing problem. Omega 11:91–95
9. French S, "Sequencing and scheduling: An introduction to the Mathematics of the job shop", New York, John wiley and sons, 1982.
10. Pinedo, M. "Scheduling: Theory, Algorithm, and Systems.",New York, Prentice Hall, 1995..
11. Taylor, F. W.: Principles of Scientific Management, New York and London: Harper & brothers, 1911Taillard, E.: Benchmarks for Basic Scheduling Problems, European Journal of Operational Research 64, 1993, pp. 278-285
12. Pinedo, M.: Scheduling: Theory, Algorithms and Systems, New Jersey: Springer, 2008
13. Reeves, C. R. et al.: Genetic Algorithms, Path Relinking, and Flow Shop Problem, Evolutionary Computation 6, 1998, pp. 45-60
14. Nowicki, E. et al.: A Fast Tabu Search Algorithm for the Permutation Flow-Shop Problem, Eur. J. Oper. Res. 91, 1996, pp. 160-175
15. [Ogbu, F. A. et al.: The Application of the Simulated Annealing Algorithm to the Solution of the n/m/c Subscript Max Flowshop Problem, Computers Ops Res. 17, 1990, pp. 243-253
16. Page, E. S.: An Approach to Scheduling of Jobs on the Machines, J. Royal Stat. Soc., 23, 1961, pp. 484-492
17. Kalczynski, P. J. et al.: On the NEH Heuristic for Minimising the Makespan in Permutation Flowshops, The International Journal of Management Science, 35 (1), 2007, pp. 53-60