ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645
Indian Journal of Science and Technology, *Vol 9(42), DOI: 10.17485/ijst/2016/v9i42/103302, November 2016*

Impact of Initial Partial Sequence in the Makespan, in Permutation Flow Shop Scheduling Heuristic Algorithms – An Analysis

A. Baskar[1]*, M. Anthony Xavior[2] and B. Dhanasakkaravarthi[1]

[1]Panimalar Institute of Technology, Chennai - 600123, Tamil Nadu, India;
a.baaskar@gmail.com, dhana_rada@yahoo.co.in
[2]SMBS, VIT University, Vellore – 632014, Tamil Nadu, India;
xavior_anto@hotmail.com

Abstract

Objectives: This paper analyzes the impact of a few new initial partial sequences on the makespan in a permutation flow shop scheduling problem. Taillard benchmark problems are used for the purpose of validation. **Methods/Statistical Analysis:** The popular NEH heuristic considers the first two jobs as its initial partial sequence after arranging them in descending order of their total processing times. The algorithms using different partial sequences are coded in MATLAB and a total of 120 number problem instances are used for the analysis which fall under twelve sets of 10 instances each. One-way ANOVA has been conducted for validating the results. **Findings**: It has been found that the initial partial sequences other than the first two jobs considered by the original NEH can also yield better makespans. Also, initial ordering of jobs according to the decreasing order of the average processing time and standard deviation of the processing times proposed by in. perform relatively better. In all the cases, job insertion technique is proved to be more powerful. The random algorithm that uses the job insertion technique do perform well with a deviation of 3.4342% which is better than many other known simple algorithms. The ANOVA confirms that the variants are statically not different from the NEH algorithm. But, it shows that a few variants perform better than the NEH for the Taillard benchmark problems. **Application/Improvements:** The results can be used as a seed solution and could be improved using metaheuristics. Further, the authors are working on other benchmarks and using tie breaking rules to know the impact..

Keywords: Initial Partial Sequence, Makespan, NEH Heuristic, Permutation Flow Shop, Scheduling

1. Introduction

The flow shop and job shop problems have been the interesting areas of research for over six decades. It started with in [1] algorithm meant for two machines and 'n' number of jobs. In algorithm yields the optimal makespan for any permutation flow shop sheduling problem with two machines. Many simple heuristic algorithms have been proposed over the years after in. In today's computer era, for any problem, finding exact solutions become easier with the application of powerful algorithms and high level languages. A few solution methods are available for solving these problems. However, the computation time and the corresponding cose increases exponentially with the problem size. Hence, the heuristics and meta-heuristics have gained popularity in this area. In [2] slope index algorithm is an early time approximate algorithm that could be used for any number of machines. In algorithm

**Author for correspondence*

was extended for any number of machines, by [3] and in the CDS algorithm proposed by them, the near optimal makespan is computed in (m-1) enumerations. Another approximate algorithm known as the Rapid Access (RA) algorithm was developed by [4]. RA algorithm combined the advantages of both the slope index and CDS heuristic algorithms. The constructive heuric proposed by [5] uses the powerful job insertion technique to construct the optimal sequence. In this algorithm, all the jobs are initially sequenced in descending order of their total processing times and the first two jobs are taken as the initial partial sequence. Even today, in industrial engineering research, the scheduling problems have a significant share. Eighteen constructive and improvement heuristics were analyzed in detail by [6] that include in heuristics. It was concluded by them that still NEH is one of the best simple, constructive algorithms for makespan minimization. Since then, many modifications and improvements have been applied on the NEH and analyzed (example: Chakraborty[7]. In [8] proposed the concept of dummy machine to improve the makespans obtained using Palmer, CDS, Gupta and RA heuristics. The improvement was observed to be up to 18% in the makespan. Tools like simulated annealing [9], fuzzy logic[10] are liberally used by many authors in the improvement process. A few meta-heuristics/ evolutionary algorithms have been proposed recently that increases the accuracy of the results either in the form of lowering the upper bounds or decreasing the processing time. The meta-heuristics take the initial seed solution from any simple heuristic and refine the solution further. A variety of evolutionary algorithms and their hybrids are available in the literature; many of them outperform the simple heuristics. A discrete firefly meta-heuristic was presented by [11] for optimising the makespan in a permutation flow shop problem. The results were compared with an existing ant colony optimization technique. The analysis indicates that the new method performs better for some well-known benchmark problems. In [12] proposed one such hybrid algorithm that combines the power of particle swarm and Tabu search based concepts. For larger problems with jobs not necessarily need to be processed in each machine, the heuristic of [13] can be used. In this paper, the second step of NEH is modified keeping other two steps the same. The initial partial sequences are changed and the impact analyzed using Taillard's benchmark problems [14] which are 120 in number. All the algorithms are coded in MATLAB 2008a and run in an i5 PC with 4 GB RAM. The ANOVA has been carried out using MS Excel.

2. Initial Partial Sequences Analyzed

The NEH2 and NEHR algorithms are already proposed by [15]. The middle two jobs constitute the initial partial

Table 1. Initial partial sequences analyzed

S.No	Algorithm	Initial Partial Sequence
1	NEH	First Two Jobs
2	NEHR	Randomly Two Jobs
3	NEH2	Middle Two Jobs
4	NEHS1	Job nos. [1, 3]
5	NEHS2	Job nos. [1, 4]
6	AB2S1	Job nos. [(n/2)-1, n]
7	AB2S2	Job nos. [(n/2), n]
8	NEH2S1	Job nos. [(n/2)-1, (n/2)+2]
9	NEH2S2	Job nos. [(n/2)-2, (n/2)+3]

'n' to be an even number

sequence in the NEH2 heuristic; whereas, AB2S1 and AB2S2 are two new initial partial sequences. Randomly two jobs are selected by the algorithm as the initial partial sequence in NEHR. To broaden the analysis, a few small neighborhood searches are carried out for the NEH and NEH2 algorithms for the makespan minimization which are listed in Table 1. The jobs are initially ordered according to their non increasing order of the total processing times. After the initial partial sequence is constructed, other jobs are inserted one by one at a suitable place that optimizes the partial makespan. These two steps are the elements of the NEH algorithm.

Following are the acronyms used in this paper for analyzing the algorithms.

n – Number of Jobs to be scheduled

m – Number of machines available for processing

3. Results and Discussion

The popular benchmark problems proposed by[14] are used for validating the results. They are 120 in numbers and grouped in 12 sets of 10 problem instances each. The number of machines are 5, 10 and 20. Whereas, the number of jobs are 20, 50, 100, 200 and 500. The deviations obtained from NEH, NEH2 and NEHR algorithms are reproduced in Table 2 along with other results. In this analysis, NEH heuristic algorithm has been considered as the reference heuristic. A lowest percentage deviation of 3.2404% is reported by the NEH2 algorithm among all the algorithms. This is closely followed by the NEHS2 algorithm with a deviation of 3.2415%. NEHS1 and NEHS2 are the simple neighbourhood search variants of NEH algorithm. They report 3.3212% and 3.2415% respectively which are better than the original NEH.

Table 2. (a) % Deviations in Makespans for different initial partial sequences (b) % Deviations in Makespans for different initial partial sequences

(a)				
Size(mxn)	**NEH**	**NEHS1**	**NEHS2**	**AB2S1**
5x20	3.300288	3.025795	3.099798	2.597573
10x20	4.601116	5.042629	4.184252	5.76967
20x20	3.730891	3.757845	3.578204	3.438556
5x50	0.727204	0.675023	0.675023	0.767995
10x50	5.072897	4.904886	4.778087	5.204228
20x50	6.648051	6.631219	6.655084	6.703285
5x100	0.527212	0.51154	0.504024	0.751175
10x100	2.21498	2.201115	2.188373	2.339929
20x100	5.344636	5.192349	5.473912	6.027814
10x200	1.230268	1.243634	1.304904	1.334068
20x200	4.435269	4.550695	4.385409	4.57197
20x500	2.066128	2.117366	2.070431	2.181249
Mean	**3.324912**	**3.321175**	**3.241458**	**3.473959**

(b)

Size(mxn)	AB2S2	NEH2	NEH2S1	NEH2S2	NEHR
5x20	3.27208	2.7900	3.162161	3.035379	3.8927
10x20	4.576939	3.6761	4.452189	4.436916	4.3975
20x20	3.00131	3.6663	3.456855	3.688779	3.7915
5x50	0.839139	0.8217	0.654299	0.621751	0.9419
10x50	5.329274	5.3638	5.062567	5.392344	5.3853
20x50	6.083032	6.5380	7.180808	6.343951	6.8520
5x100	0.612082	0.5054	0.511108	0.563655	0.5550
10x100	2.324834	2.1135	2.364034	2.239113	2.2384
20x100	5.600069	5.7233	5.439854	5.563105	5.3377
10x200	1.297395	1.4096	1.238414	1.290778	1.3456
20x200	4.540178	4.0681	4.592518	4.255121	4.3489
20x500	2.300165	2.2603	2.057632	2.185313	2.1243
Mean	**3.314708**	**3.2404**	**3.347703**	**3.30135**	**3.4342**

Table 3. No. of Instances better makespans are reported

S.No	Algorithm	No. of Instances
1	NEHS1	31
2	NEHS2	36
3	AB2S1	18
4	AB2S2	29
5	NEH2S1	32
6	NEH2S2	32

Figure 1. Percentage mean deviations for specific number of jobs.

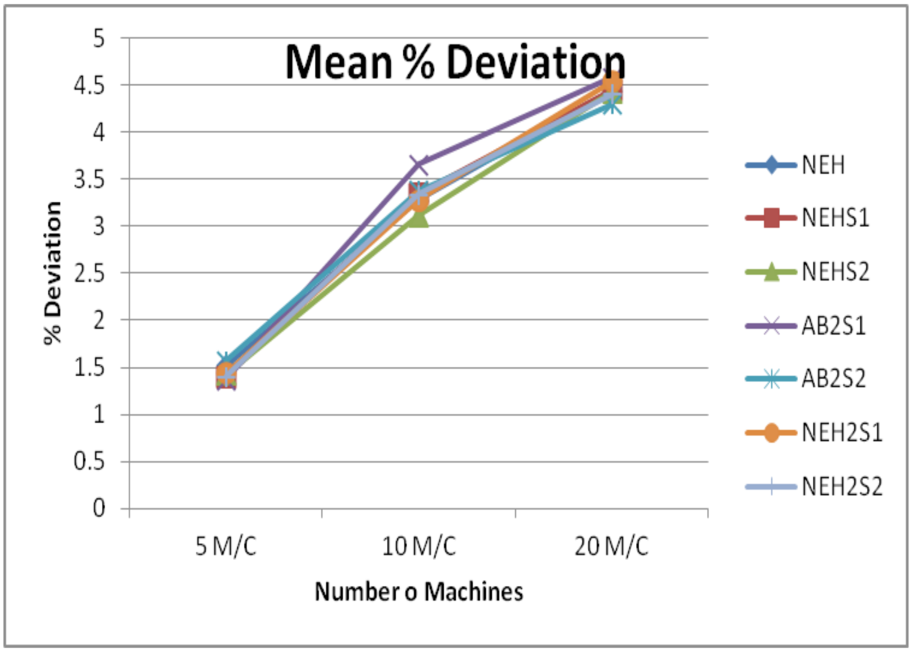

Figure 2. Percentage mean deviations for specific number of machines.

However, others do not perform well. The last column of Table 2(b) shows the deviations of NEHR algorithm from the known upper bounds; when two jobs are selected randomly as the initial partial sequence. The 3.4342% mean deviation reported by the random algorithm is not too far from that of the NEH, just 3.2873% higher. Table 3 presents the number of instances better makespans are reported by any individual algorithms. In case the same makespan is reported by more than one algorithm, both are considered for the count. It is found that, the NEHS2 algorithm outperforms other heuristics with a total count of 36 out of 120 problem instances. Figure 1 and Figure 2 present the deviations for specific number of machines and jobs. The deviations increase with the increase in the number of machines for all the heuristics. In contrast, the percentage deviation follows a varying pattern in case of the number of jobs. Maximum deviation is observed for the50 jobs and minimum deviation for the 500 jobs.

4. ANOVA

To analyze the interaction between the algorithms and the mean deviations of the reported makespans from the known upperbounds, one way ANOVA has been carried out at 95 % confidence level using MS Excel software.

Null hypothesis, H0: All means are equal

Alternative hypothesis, H1: At least one mean is different

It is observed that the P-Value is high (0.992288) and the F-Value is small (0.131739) which accepts the Null Hypothesis. To have the individual comparison with the reference NEH heuristic, one-sided paired t-test is also carried out using the MS Excel package. The results are tabulated in Table 4. Only the p-value of NEH vs AB2S1 is less than 0.05 which statistically confirms that the AB2S1 algorithm is inferior to the NEH algorithm. The p-values

Groups	Count	Sum	Average	Variance		
NEH	120	398.9894	3.324912	4.49987		
NEHS1	120	398.541	3.321175	4.464374		
NEHS2	120	388.975	3.241458	4.554434		
AB2S1	120	416.8751	3.473959	4.839283		
AB2S2	120	397.765	3.314708	4.223776		
NEH2S1	120	401.7244	3.347703	4.73668		
NEH2S2	120	396.162	3.30135	4.565417		
Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	3.600297	6	0.600049	0.131739	0.992288	2.109447
Within Groups	3794.176	833	4.554833			
Total	3797.776	839				

Table 4. One-sided paired-samples t-test, α = 0.05

H0	H1	p
NEH = NEHS1	NEH < NEHS1	0.2669
NEH = NEHS2	NEH < NEHS2	0.2993
NEH = AB2S1	NEH < AB2S1	0.0042
NEH = AB2S2	NEH < AB2S2	0.0617
NEH = NEH2S1	NEH < NEH2S1	0.1910
NEH = NEH2S2	NEH < NEH2S2	0.2906

of other pairs are greater than 0.05 which shows that the means are same for all and no statistical significance could be established. However, a few variants perform slightly better than the NEH algorithm.

5. Conclusion and Future Work

In this paper, six variants of three heuristic algorithms are analyzed. The well known NEH heuristic algorithm is considered as the reference algorithm and two other already proposed heuristics, the NEHR and the NEH2, are also included in the analysis. The powerful job insertion technique is used invariably by all the algorithms after arranging them in non-increasing order of their total processing times; both are the elements of original NEH. Only, the initial partial sequence is varied. It has been found that, out of the algorithms, the performances of NEH2 (3.240448%), NEHS2 (3.241458%), NEH2S2 (3.30135%), AB2S2 (3.314708%) and NEHS1 (3.321175%) are better than the popular NEH algorithm in terms of the mean gap from the known upper bounds for the Taillard's 120 problem instances. Even the random algorithm performs better with a deviation of 3.4342% which is better than many other known simple algorithms. It can be concluded that the initial partial sequences other than the first two jobs considered by the original NEH can also yield better makespans. Also, initial ordering jobs according to decreasing order of the average processing time and standard deviation of the processing times proposed by [16] improves the makespans further. However, in all the cases, job insertion technique is proved to be more powerful. Also, a few authors have improved the makespans using tie breaking strategies in NEH, one such being by [17]. The results obtained from efficient simple algorithms are generally used as the seed solution by many meta-heuristics. Then the results are improved further by the meta-heuristics. In the literature, it has been generally agreed by many researchers that the NEH algorithm is one of the best simple constructive heuristics and is used by many other meta-heuristics. The proposed algorithms also perform equally well and can be used at par with the NEH algorithm. Further work includes analysing the algorithms using other benchmarks and initial ordering of jobs by applying tie breaking rules.

6. References

1. Johnson SM. Optimal two-and three-stage production schedules with setup times included. Naval research logistics. 1954 Mar; 1(1):61–8.

2. Palmer DS. Sequencing jobs through a multi-stage process in the minimum total time—a quick method of obtaining a near optimum. Journal of the Operational Research Society. 1965 Mar; 16(1):101–17.
3. Campbell HG, Dudek RA, Smith ML. A heuristic algorithm for the n job, m machine sequencing problem. Management Science. 1970 Jun; 16(10):B 630–7.
4. Dannenbring DG. An evaluation of flow shop sequencing heuristics. Management Science. 1977 Jul; 23(11):1174–82.
5. Nawaz M, Enscore EE, Ham I. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. Omega. 1983 Dec; 11(1):91–5.
6. Ruiz R, Maroto C. A comprehensive review and evaluation of permutation flow shop heuristics. European Journal of Operational Research. 2005 Sep; 165(2):479–94.
7. Chakraborty UK, Laha D. An improved heuristic for permutation flow shop scheduling. International Journal of Information and Communication Technology. 2007 Apr; 1(1):89–97.
8. Baskar A, Xavior MA. Effects of dummy machines on make span in a few classical heuristics using Taillard bench mark problems. International Journal of Materials and Product Technology. 2012 Jan; 45(1–4):145–62.
9. Dalfard VM. A simulated annealing algorithm for JIT single machine scheduling with preemption and machine idle time. Indian Journal of Science and Technology. 2011 May; 4(5):1–7.
10. Nailwal KK, Gupta D, Sharma S. Two stage flow shop scheduling under fuzzy environment. Indian Journal of Science and Technology. 2015 Jul; 8(16):1–8.
11. Sayadi M, Ramezanian R, Ghaffari-Nasab N. A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems. International Journal of Industrial Engineering Computations. 2010 Jul; 1(1):1–10.
12. Rohini V, Natarajan AM. Comparison of genetic algorithm with particle swarm optimisation, ant colony optimisation and tabu search based on university course scheduling system. Indian Journal of Science and Technology. 2016 Jun; 9(21):1–5.
13. Bhongade A, Khodke P. Heuristics for production scheduling problem with machining and assembly operations. International Journal of Industrial Engineering Computations. 2012; 3(2):185–98.
14. Taillard E. Benchmarks for basic scheduling problems. European Journal of Operational Research. 1993 Jan; 64(2):278–85.
15. Baskar A, Xavior MA. Analysis of job insertion technique for different initial sequences in permutation flow shop scheduling problems. International Journal of Enterprise Network Management. 2015; 6(3):153–74.
16. Dong X, Huang H, Chen P. An improved NEH-based heuristic for the permutation flow shop problem. Computers and Operations Research. 2008 Dec; 35(12):3962–8.
17. Fernandez-Viagas V, Framinan JM. On insertion tie-breaking rules in heuristics for the permutation flowshop scheduling problem. Computers and Operations Research. 2014 May; 45:60–7.