A New Priority Rule for Initial Ordering of Jobs in Permutation Flowshop Scheduling Problems

B. Dhanasakkaravarthi[1]

Research Scholar, School of Mechanical Engineering
Sathyabama Institute of Science and Technology, India

A. Krishnamoorthy[2]

Professor, School of Mechanical Engineering
Sathyabama Institute of Science and Technology, India

Abstract—Scheduling in a permutation flowshop refers to processing of jobs in a set of available machines in the same order. Among the several possible performance characteristics of a flowshop, makespan remains one of the highest preferred metrics by researchers in the past six decades. The constructive heuristic proposed by Nawaz-Enscore-Ham (NEH) is one of the best for makespan minimization. The performance essentially depends on the initial ordering jobs according to a particular priority rule (PR). The popular priority rules are non-increasing order of the jobs' total processing time, the sum of average processing time and standard deviation and, the sum of average processing time, standard deviation and absolute skewness among others. The objective of this paper is to propose and analyse a new job priority rule for the permutation flowshop. The popular priority rules available in the literature are studied and, one of the best priority rules; the sum of average processing time and standard deviation is slightly modified, by replacing the standard deviation by mean absolute deviation (MAD). To assess the performance of the new rule, four benchmark datasets are used. The computational results and statistical analyses demonstrate the better performance of the new rule.

Keywords—Priority rule; flowshop scheduling; makespan; NEH algorithm

I. INTRODUCTION

Permutation flowshop scheduling problems (PFSSP) remain one of the most studied domains in operations research in the past six decades. PFSSP refers to scheduling '*n*' number of jobs for processing in '*m*' number of machines in the same order. According to Rinnooy Kan [1], PFSSP is proved to be NP-hard when the number of machines is greater than three. Since the PFSSP is np-hard, the computation time grows exponentially for larger problems and hence, exact solution becomes impossible or expensive. As a result, several dispatching rules, heuristics and metaheuristics have been proposed over the decades. Efficient heuristics report solutions with acceptable accuracy levels in a reasonable time. Priority rules or dispatching rules are some form of heuristics and have been studied in both academia and industry domains extensively for decades (Tay and Ho 2008)[2]. A few simple dispatching rules that are being extensively used in jobs' scheduling are: Shortest Processing Time (SPT) rule, Longest Processing Time (LPT) rule, Earliest Due Date (EDD) rule, First Come First Serve (FCFS) rule.

The optimization parameters for the PFSSP are generally; flow time, idle time or makespan (Liu et al., 2016) [3] to satisfy different production line requirements. They find

numerous real-time applications and could be combined with Internet of Things (IOT) for specific applications (Salis, 2021) [4].Among the many parameters that are being optimised, makespan minimization is widely considered by researchers over the years. Johnson's [5] algorithm proposed in 1954 yields an optimum solution for two machines and '*n*' jobs PFSSPs which was extended to three machines cases.

As the problem is NP-hard, the exact solution becomes impossible for larger problems and the computation time grows exponentially with the problem size. Earlier approximate heuristics could not yield the expected accuracy and the breakthrough came in 1983 when Nawaz-Enscore-Ham (NEH) [6] algorithm was proposed. NEH algorithm which has a complexity of $O(n^3.m)$ uses the largest processing time (LPT) dispatching rule and is considered as one of the best constructive heuristics for makespan minimization even today. Many improvements and extensions have been proposed by many authors over the years. NEH essentially consists of three phases:

- Pre-arranging the jobs according to the non-increasing order of their total processing times (Priority Rule).

- Selecting the first two jobs from the processed sequence as the initial partial sequence (Initial Sequence).

- Inserting other jobs one by one at a suitable place that minimises the partial makespan (Insertion Phase).

Ribas et al. (2010) [7] tested four priority rules combined with the powerful insertion technique of NEH including rules from NEHKK1 (Kalczynski and Kamburowski 2008) [8], N&M (Nagano and Moccellin 2002) [9], LPT, and a random job sequence. Baskar and Xavior (2015) [10] analysed the job insertion technique for different initial sequences.

The authors improved the solution quality of NEH by suitably modifying the first and third phases. Several priority rules for NEH are proposed and available in the literature and the ones proposed by Dong et al. [11] and Liu et al.[12] yield better results.

Framinan et al. [13] analysed 177 initial sequences for the NEH heuristic and concluded that SUM PIJ / DECR (i.e. original NEH) is ranked 1 among all for the makespan minimisation.

During the insertion phase, we come across several occasions when the partial makespan remains the same for

more than one partial sequence. NEH breaks such ties randomly. Several effective tie-breaking rules are proposed and analysed for the solution quality. The tie-breaking rules proposed by Fernandez and Framinan [14], Lie et al. [12], Benavides [15] and the ones recently by Baskar and Xavior [16] are a few to mention.

This paper considers the sum of the average processing times and means absolute deviation (MAD) for the initial ordering of jobs (Priority Rule) and analyses its impact using well-known benchmark datasets.

The structure of this paper is as follows: the new priority rule is presented in Section 2 followed by the benchmark and performance metrics used for the assessment in Section 3. The results and statistical analyses are detailed in Section 4 and Section 5 discusses about the conclusion, limitations and future work.

II. NEW PRIORITY RULE

The initial ordering of jobs or priority rule does affect the solution quality of the NEH heuristic. The priority rule considered by the original NEH is non-increasing order of the jobs' total processing times. Total processing time (TPT) for a particular job, j can be represented as,

$$\text{TPT}_j = \sum_{i=1}^{m} p_{ij} \tag{1}$$

m – Number of machines in the schedule. p_{ij} denotes the processing time of job 'j' in the machine 'i'.

Dong et al. [11] added the standard deviation of the processing times with the average total processing time, AVG_j for a job and reported improved results. Mathematically, standard deviation,

$$\text{SD}_j = \sqrt{\frac{1}{(m-1)} \sum_{i=1}^{m} (p_{ij} - AVG_j)^2} \tag{2}$$

Another priority rule proposed recently by Liu et al. [12] adds the absolute skewness with the priority rule of Dong et al. [11] and the rule improves the solution further. Skewness (SKE) for a job, j is defined as,

$$\text{SKE}_j = (\frac{1}{m} \sum_{i=1}^{m} (p_{ij} - AVG_j)^3) /$$
$$[\sqrt{\frac{1}{(m-1)} \sum_{i=1}^{m} (p_{ij} - AVG_j)^2}]^3 \tag{3}$$

In this paper, the standard deviation is replaced by another similar metric, the Mean Absolute Deviation (MAD) which is the average of the absolute deviations from the mean value.

For a job 'j',

$$\text{MAD}_j = \frac{1}{m} \sum_{i=1}^{m} |(p_{ij} - AVG_j)| \tag{4}$$

MAD was used as the variation measure in the tie-breaking rule of Liu et al. [12].Another metric, the median absolute deviation (MAD1) is also used instead of mean absolute deviation during the process; that is, descending order of average processing time + median absolute deviation as a priority rule. However, the results are not encouraging in this case.

For a job 'j',

$$\text{MAD1}_j = \text{Median}\,(|(p_{ij} - AVG_j)|) \tag{5}$$

The benchmarks and analyses results using different combinations are presented and discussed in the coming sections.

III. BENCHMARK AND PERFROMANCE MATRIC USED

For comparing the performance of similar heuristic algorithms in flowshop scheduling, several benchmark problem sets are available in the literature. One of the earlier benchmarks proposed for permutation flowshop is the one proposed by Carlier in 1978 [17] with varying jobs and machines combination. The instances have high processing times, up to 999-time units. DUM dataset proposed in 1998 by Demirkol et.al. [18] has a combination of jobs {20, 30, 40, 50} and machines {15, 20} resulting in 40 instances of 5 in each set. The processing time is randomly fixed between {0, 200} time units. The Taillard [19] instances proposed in 1993 are available under 12 groups of 10 instances each. The processing times in 5, 10 and 20 machines vary from 0 to 99 for 20, 50, 100, 200 and 500 numbers of jobs. The known upper bounds for all the 120 problems are available online [20] and are accessible for researchers.

Recently, Vallada et al. [21] proposed new hard benchmarks which are 480 in numbers. They are categorized into small and large problem sets. The small instances, termed as VFR (Small) are 240 in numbers with a combination of {10, 20, 30, 40, 50, 60} jobs and {5, 10, 15, 20} machines. Similarly, the larger instances have a combination of {100, 200, 300, 400, 500, 600, 700, 800} jobs and {20, 40, 60} machines, totaling 240 in numbers. The instances are grouped under 48 sets of 10 instances each. In line with Taillard's, these benchmarks do have processing times from 0 to 99. The known upper bounds are provided by the authors themselves.

The parameter used for the comparison of the performance of different heuristics is the Percentage Relative Deviation (PRD) which is defined as:

Percentage Relative Deviation, PRD =
$$\frac{(Makespan - KnownUpperBound)X100}{KnownUpperBound} \tag{6}$$

IV. COMPUTATIONAL RESULTS AND DISCUSSION

Different priority rules (initial ordering of jobs) considered are summarized below:

- PR-NEH: Descending order of their total processing times.

- PR-D: Descending order of the sum of their average processing time and standard deviation.

- PR-LJP: Descending order of the sum of their average processing time, standard deviation and absolute (skewness).

- PR-DB: Descending order of the sum of their average processing time and mean absolute deviation.

- "No PR" refers to no initial ordering of jobs; raw data are used as received.

Codes are generated for all priority rules in MATLAB R2012b and run in an i5 desktop PC with 4 GB RAM. Many heuristics initially order the jobs according to a priority rule and then refine the solution further using some strategy. The initial priority ordering may have significant impact on the final result. To start with, Taillard instances are initially ordered as per the priority rule (PR) considered and the makespans computed without any further processing. The average PRDs (APRDs) are presented in Table I. The bold digits indicate the best result for a particular problem set. There is not much difference in the results among the PRs considered. The best APRD of 21.40% is reported by PR-DB and the worst one by PR-D (21.64%); the difference being 1.12%. Even no "no priority rule" reports a better mean value of 21.61% and accounts for best results in 4/12 sets. Though PR-DB reports the lowest deviation, it reports best results in one case only, 10 machines and 100 jobs.

Subsequently, the impact of different initial sequences has been investigated. NEH considers the first two jobs of the processed sequence as the initial sequence. Baskar et al. [22] considered a few other sets of jobs as initial sequences and analysed (Table II) the impact using Taillard dataset. The performance was analysed by randomly selecting two jobs also from the processed sequence. The results show that the APRDs vary slightly with respect to the initial partial sequence. According to them, randomly selecting two jobs also results in a reasonably good APRD with 3.43% which may be due to the job insertion strategy that was originally used by the NEH algorithm. It is to be noted that many heuristics could not even better this mean value of 3.43% for the Taillard benchmark.

Now, to assess the performance; the PR is applied to the classic NEH algorithm and the results are tabulated in Table III to Table VI. The benchmark used are; Carlier's [17] proposed in 1978, Taillard's [19], in 1993, Demirkol's [18] in 1998 and the latest VFR benchmark proposed in 2015 by Vallada et al.[21]. The better APRDs are shown in bold and italics in all the tables.

For the Carlier 8 instances (Table III); Dong et al. priority rule, PR-D reports an APRD of 1.44%, an increase of 3.60% in the APRD over PR-NEH which is taken as the reference for the analysis in this paper. That is, for this dataset, the solution quality deteriorates when PR-D is used for initial ordering the jobs. The APRD of the new priority rule, PR-DB is 1.21%, a significant 12.95% improvement over the rule used by NEH.

For the comparatively larger dataset of Demirkol's which are 40 in numbers also, the proposed priority rule, PR-DB reports better APRD than the priority rules, PR-NEH and PR-D with an improvement of 4.70% over the reference PR-NEH. The improvement of the rule, PR-D over PR-NEH being 1.61%. Here, the lower bounds (LB) are considered instead of upper bounds (UB) for better comparison as the makespans better the UBs provided by Demirkol et al. in many cases. While using Taillard's dataset; three more priority rules are considered as detailed below:

- PR-LJP1: Descending order of the sum of their average processing time, mean absolute deviation and absolute (skewness).

- PR-LJP2: Descending order of the sum of their average processing time, mean absolute deviation and skewness.

- PR-DB1: Descending order of the sum of their average processing time and median absolute deviation.

TABLE I.   APRDs of Priority Rules – Taillard Benchmark

Size (mxj)	No PR	PR-NEH	PR-D	PR-LJP	PR-DB
5x20	*24.98*	26.27	27.41	27.66	26.24
10x20	28.77	28.47	28.15	*27.96*	28.16
20x20	21.43	21.50	20.81	*20.56*	21.31
5x50	*15.32*	16.46	17.21	17.63	16.91
10x50	*25.05*	28.86	27.10	26.66	26.34
20x50	29.73	29.15	28.13	*27.76*	27.89
5x100	13.63	12.06	12.22	*11.98*	12.72
10x100	20.92	19.08	20.21	20.24	*18.85*
20x100	25.51	*23.50*	24.47	24.77	24.61
10x200	15.67	15.70	*14.78*	15.31	15.11
20x200	22.28	*21.39*	22.32	21.74	22.30
20x500	*15.99*	16.31	16.80	16.51	16.33
Mean	21.61	21.56	21.64	21.56	21.40

TABLE II.   APRDs of Initial Partial Sequences Applied to NEH - Taillard Benchmark

Size (mxj)	Jobs 1 and 2 (NEH)	Middle 2 Jobs	Jobs 1 and 3	Jobs 1 and 4	Randomly 2 Jobs
5x20	3.30	2.79	3.03	3.10	3.89
10x20	4.60	3.68	5.04	4.18	4.40
20x20	3.73	3.67	3.76	3.58	3.79
5x50	0.73	0.82	0.68	0.68	0.94
10x50	5.07	5.36	4.90	4.78	5.39
20x50	6.65	6.54	6.63	6.66	6.85
5x100	0.53	0.51	0.51	0.50	0.56
10x100	2.21	2.11	2.20	2.19	2.24
20x100	5.34	5.72	5.19	5.47	5.34
10x200	1.26	1.41	1.24	1.30	1.35
20x200	4.41	4.07	4.55	4.39	4.35
20x500	2.07	2.26	2.12	2.07	2.12
Mean	3.32	3.24	3.32	3.24	3.43

TABLE III.   APRDs of Priority Rules Applied to NEH- Carlier Benchmark

Size (mxj)	BM	UB	PR-NEH	PRD	PR-D	PRD	PR-DB	PRD
5x11	Carl1	7038	7038	*0*	7038	*0*	7038	*0*
4x13	Carl2	7166	7376	*2.93*	7376	*2.93*	7376	*2.93*
5x12	Carl3	7312	7399	*1.19*	7399	*1.19*	7399	*1.19*
4x14	Carl4	8003	8003	*0*	8129	1.57	8021	0.22
6x10	Carl5	7720	7835	*1.49*	7843	1.59	7843	1.59
9x8	Carl6	8505	8773	3.15	8773	3.15	8570	*0.76*
7x7	Carl7	6590	6590	*0*	6590	*0*	6590	*0*
8x8	Carl8	8366	8564	2.37	8457	*1.09*	8617	3.00
Mean				1.39		1.44		1.21
%Improvement				Ref.		-3.60		12.95

TABLE IV.   APRDs (from LB) of Priority Rules Applied to NEH- Demirkol Benchmark

Instance	LB	PR-NEH	PRD	PR-D	PRD	PR-DB	PRD
flcmax_20_15_3	3354	4071	21.38	4018	*19.80*	4065	21.20
flcmax_20_15_6	3168	3898	23.04	3878	22.41	3870	*22.16*
flcmax_20_15_4	2997	3672	22.52	3617	*20.69*	3625	20.95
flcmax_20_15_10	3420	4248	24.21	4223	23.48	4217	*23.30*
flcmax_20_15_5	3494	4007	*14.68*	4028	15.28	4038	15.57
flcmax_20_20_1	3776	4779	26.56	4641	*22.91*	4674	23.78
flcmax_20_20_3	3758	4567	21.53	4535	*20.68*	4598	22.35
flcmax_20_20_9	3902	4699	20.43	4666	19.58	4611	*18.17*
flcmax_20_20_2	3881	4606	*18.68*	4619	19.02	4626	19.20
flcmax_20_20_10	3823	4487	17.37	4515	18.10	4462	*16.71*
flcmax_30_15_3	4020	4770	18.66	4729	17.64	4692	*16.72*
flcmax_30_15_4	4080	4912	20.39	4924	20.69	4890	*19.85*
flcmax_30_15_9	4022	4857	20.76	4737	*17.78*	4770	18.60
flcmax_30_15_8	4490	5070	12.92	5056	12.61	4982	*10.96*

flcmax_30_15_6	4184	5041	20.48	5013	19.81	4888	*16.83*
flcmax_30_20_3	4806	5664	17.85	5648	17.52	5561	*15.71*
flcmax_30_20_1	4772	5891	*23.45*	5995	25.63	5940	24.48
flcmax_30_20_6	5004	5919	*18.29*	5989	19.68	5970	19.30
flcmax_30_20_10	4899	5523	12.74	5532	12.92	5464	*11.53*
flcmax_30_20_2	4757	5629	18.33	5470	*14.99*	5591	17.53
flcmax_40_15_5	5560	6286	13.06	6380	14.75	6193	*11.38*
flcmax_40_15_9	5119	5931	15.86	5907	*15.39*	5947	16.18
flcmax_40_15_2	5290	6113	15.56	6105	15.41	6102	*15.35*
flcmax_40_15_10	5596	6206	10.90	6271	12.06	6115	*9.27*
flcmax_40_15_8	5576	6394	14.67	6329	*13.50*	6347	13.83
flcmax_40_20_3	5693	6816	*19.73*	6865	20.59	6866	20.60
flcmax_40_20_9	5998	6929	*15.52*	7065	17.79	6995	16.62
flcmax_40_20_6	5990	7154	19.43	7160	19.53	7097	*18.48*
flcmax_40_20_7	6170	7026	*13.87*	7107	15.19	7080	14.75
flcmax_40_20_5	6011	6910	14.96	6846	13.89	6842	*13.82*
flcmax_50_15_6	6290	7264	15.48	7206	14.56	7111	*13.05*
flcmax_50_15_5	6355	6928	*9.02*	7026	10.56	6972	9.71
flcmax_50_15_1	6198	6909	11.47	6860	*10.68*	6916	11.58
flcmax_50_15_8	6312	7180	13.75	7158	13.40	7132	*12.99*
flcmax_50_15_2	6531	7330	12.23	7267	*11.27*	7278	11.44
flcmax_50_20_2	6740	8138	20.74	8063	19.63	8021	*19.01*
flcmax_50_20_1	6736	7602	12.86	7550	*12.08*	7673	13.91
flcmax_50_20_7	6756	7965	*17.90*	8081	19.61	7993	18.31
flcmax_50_20_8	6897	7924	14.89	7890	14.40	7617	*10.44*
flcmax_50_20_4	6830	8256	20.88	8218	20.32	8098	*18.57*
Mean			17.43		17.15		16.61
%Improvement			Ref.		1.61		4.70

TABLE V.   APRDs of Priority Rules Applied to NEH- Taillard Benchmark

Size (mxj)	PR-NEH	PR-D	PR-LJP	PR-LJP1	PR-LJP2	PR-DB	PR-DB1
5x20	3.30	2.70	2.71	2.95	3.04	2.74	*1.93*
10x20	4.60	4.08	3.68	4.02	*3.37*	3.87	4.89
20x20	3.73	3.82	*2.91*	3.20	3.13	2.95	3.73
5x50	0.73	0.89	0.88	*0.70*	0.80	0.92	0.93
10x50	5.07	4.90	4.84	*4.71*	4.82	5.44	5.56
20x50	6.65	6.12	6.42	6.50	6.67	6.45	*6.08*
5x100	0.53	*0.41*	0.54	0.57	0.50	0.50	0.60
10x100	2.21	*2.16*	2.24	2.33	2.36	2.40	2.39
20x100	5.34	5.65	*4.99*	5.28	5.07	5.16	5.71
10x200	1.26	1.27	1.24	*1.23*	1.31	1.29	*1.23*
20x200	4.41	4.57	*4.14*	4.24	4.26	4.27	4.39
20x500	2.07	2.12	2.12	2.14	2.10	*2.06*	2.09
Mean	3.32	3.22	3.06	3.16	3.12	3.17	3.29
%Improvement	Ref.	3.01	7.83	4.82	6.02	4.52	0.91

TABLE VI.   AVERAGE PERCENT RELATIVE DEVIATION OF DIFFERENT PRIORITY RULES FOR VFR BENCHMARK

Size (mxj)	PR-NEH	PR-D	PR-DB	Size (mxj)	PR-NEH	PR-D	PR-DB
5x10	2.18	*1.51*	1.67	20x100	5.71	5.61	*5.27*
10x10	1.63	*1.46*	2.66	40x100	5.67	*5.31*	5.43
15x10	*1.53*	2.17	2.47	60x100	4.95	*4.51*	4.85
20x10	1.99	*1.52*	1.59	20x200	4.23	*4.04*	4.19
5x20	*1.51*	2.76	2.19	40x200	4.71	4.66	*4.51*
10x20	*4.82*	4.93	5.34	60x200	4.55	4.35	*4.28*
15x20	4.33	*3.93*	4.50	20x300	3.00	3.03	*2.99*
20x20	4.12	*3.50*	3.89	40x300	4.08	3.90	*3.88*
5x30	*1.43*	1.64	1.45	60x300	3.93	*3.91*	3.93
10x30	*5.26*	5.46	5.29	20x400	2.58	2.46	*2.24*
15x30	5.83	5.44	*5.31*	40x400	3.66	3.51	*3.43*
20x30	*5.41*	5.49	5.47	60x400	3.56	3.47	*3.42*
5x40	1.09	0.79	*0.70*	21x500	2.27	2.23	*2.00*
10x40	4.97	*4.52*	4.75	40x500	3.20	3.11	*3.06*
15x40	6.05	5.87	*5.56*	60x500	*3.12*	3.20	3.13
20x40	*5.14*	5.29	5.58	20x600	*1.57*	1.64	1.62
5x50	*0.55*	0.82	0.84	40x600	3.13	2.98	*2.86*
10x50	4.58	*4.45*	4.59	60x600	2.93	2.94	*2.92*
15x50	*6.52*	6.90	6.76	20x700	1.40	*1.23*	1.32
20x50	*5.96*	6.00	6.71	40x700	2.77	*2.60*	2.70
5x60	0.89	0.48	*0.34*	60x700	2.75	*2.71*	2.78
10x60	3.96	*3.94*	4.25	20x800	1.23	1.15	*1.14*
15x60	*5.79*	5.91	5.91	40x800	*2.43*	2.52	2.50
20x60	6.45	6.42	*6.14*	60x800	2.71	*2.67*	2.68
Mean	3.83	3.80	3.92	Mean	3.34	3.24	3.21
%Improvement	Ref.	0.78	-2.35	%Improvement	Ref.	2.99	3.89

The PR-LJP proposed by Liu et al. [12] is proved to yield better results for the Taillard dataset. PR-LJP adds a third metric, absolute (skewness) to the priority rule, PR-D. However, when this absolute (skewness) is added to the sum of average processing time and mean absolute deviation (PR-LJP1); the APRD comes down to 3.16% from 3.06%. The APRD slightly improves to 3.12% when the absolute (skewness) is replaced by skewness (PR-LJP2). For this dataset, PR-D reports an APRD of 3.22% and PR-DB, 3.17%. The improvement of PR-DB is 4.52% over PR-NEH which is better than PR-D (3.01% improvement). When the median absolute deviation (MAD1) replaces the mean absolute deviation (PR-DB1), we get an APRD of 3.29% (Table V) for the Taillard instances.

The results of VFR (Small) and VFR (Large) datasets are given in Table VI. The performance is different here. For the VFR (Small) dataset, the performance of PR-DB worsens by 2.35% whereas; PR-D reports 0.78% improvement over the reference priority rule, PR-NEH. For the smaller problems, the priority rule, PR-D reports the best results. However, when the number of problem sets are considered, the reference rule, PR-NEH accounts for better results in 11/24 sets followed by PR-

D, 8/24 sets and the new rule, PR-DB does well in only 5/24 cases.

However, for the VFR (Large) dataset, the performance of the newly proposed rule, PR-DB is better than PR-D and PR-NEH. The improvement over PR-NEH is 3.89% for this larger dataset. The new rule also reports best results in 13/24 problem sets. PR-D comes next with better results in 8/24 cases. The priority rule, PR-NEH which performs extremely well in smaller problems of VFR, could not match that performance and accounts for only 3/24 problem sets.

The summary of the results is presented in Table VII.

TABLE VII.   SUMMARY OF RESULTS (APRDs)

Benchmark	PR-NEH	PR-D	PR-DB
Carlier	1.39	1.44	**1.21**
Demrikol (over LB)	17.43	17.15	**16.61**
Taillard	3.32	3.22	**3.17**
VFR (Small)	3.83	**3.80**	3.92
VFR (Large)	3.34	3.24	**3.21**

For all the datasets except VFR (Small), the newly proposed priority rule, PR-DB outperforms PR-D and PR-NEH.

Paired t-tests are carried out at 95% confidence level using MINITAB17 and the results are presented in Table VIII.

TABLE VIII. PAIRED T-TEST ON DIFFERENT BENCHMARKS

Pairs	T-Value	P-Value
Carlier		
PR-NEH vs PR-D	-0.18	0.862
PR-NEH vs PR-DB	0.55	0.597
Demirkol		
PR-NEH vs PR-D	1.30	0.202
PR-NEH vs PR-DB	3.75	*0.001*
Taillard		
PR-NEH vs PR-D	1.16	0.270
PR-NEH vs PR-DB	1.46	0.172
VFR (Small)		
PR-NEH vs PR-D	0.38	0.710
PR-NEH vs PR-DB	-0.85	0.403
VFR (Large)		
PR-NEH vs PR-D	3.86	*0.001*
PR-NEH vs PR-DB	4.54	*0.000*

Since only PR-D and PR-DB are similar priority rules, they are compared with PR-NEH and listed in Table VII and Table VIII. The analyses could prove the statistical significance in three cases with probability values of 0.001, 0.001 and 0.000 for the pairs; PR-NEH vs PR-DB (Demirkol), PR-NEH vs PR-D (VFR-Large) and PR-NEH vs PR-DB (VFR-Large) respectively.

V. CONCLUSION AND FUTURE WORK

This paper proposes a new priority rule, PR-DB for the permutation flowshop scheduling problems. The popular NEH algorithm considers the jobs according to the non-increasing order of their total processing times. Dong et al. added another metric, the standard deviation of the processing times of a particular job for initial ordering. In this work, the standard deviation used in the priority rule of Dong et al., PR-D is replaced by Mean Absolute Deviation. The PR is applied to the classic NEH heuristic for makespan minimisation. Computational results show that the performance is better than PR-D for the Carlier, Demirkol, Taillard and VFR (Large) datasets. The results do not improve in the case of VFR (Small) instances. The paired t-tests are carried out to assess the statistical significance. The main advantage of the new rule is that it is simple yet powerful. It can be easily applied in scheduling the jobs in any engineering industry. Similarly, it can be combined with any popular tie-breaking rule to improve the solution quality further. Also, the results obtained could be used as the seed solution for any metaheuristic for better solution and schedules. Future work includes the assessment of PR-DB for other potential benchmarks also.

REFERENCES

[1] Rinnooy kan ah. Machine scheduling problems: classification, complexity, and computations. PhD thesis, University of Amsterdam. 1976.

[2] Tay JC, Ho NB. Evolving dispatching rules using genetic programming for solving multi-objective flexible job-shop problems. Computers & Industrial Engineering. 2008 Apr 1;54(3):453-73.

[3] Liu W, Jin Y, Price M. A new Nawaz–Enscore–Ham-based heuristic for permutation flow-shop problems with bicriteria of makespan and machine idle time. Engineering Optimization. 2016 Oct 2;48(10):1808-22.

[4] Salis A. Towards the Internet of Behaviors in Smart Cities through a Fog-To-Cloud Approach. HighTech and Innovation Journal. 2021 Dec 1;2(4):273-84.

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

[6] Nawaz M, Enscore Jr EE, Ham I. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. Omega. 1983 Jan 1;11(1):91-5.

[7] Ribas I, Companys R, Tort-Martorell X. Comparing three-step heuristics for the permutation flowshop problem. Computers & Operations Research. 2010 Dec 1;37(12):2062-70.

[8] Kalczynski PJ, Kamburowski J. An improved NEH heuristic to minimize makespan in permutation flow shops. Computers & Operations Research. 2008 Sep 1;35(9):3001-8.

[9] Nagano MS, Moccellin JV. A high quality solution constructive heuristic for flow shop sequencing. Journal of the Operational Research Society. 2002 Dec;53(12):1374-9.

[10] 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.

[11] Dong X, Huang H, Chen P. An improved NEH-based heuristic for the permutation flowshop problem. Computers & Operations Research. 2008 Dec 1;35(12):3962-8.

[12] Liu W, Jin Y, Price M. A new improved NEH heuristic for permutation flowshop scheduling problems. International Journal of Production Economics. 2017 Nov 1;193:21-30.

[13] Framinan JM, Leisten R, Rajendran C. Different initial sequences for the heuristic of Nawaz, Enscore and Ham to minimize makespan, idletime or flowtime in the static permutation flowshop sequencing problem. International Journal of Production Research. 2003 Jan 1;41(1):121-48.

[14] Fernandez-Viagas V, Framinan JM. On insertion tie-breaking rules in heuristics for the permutation flowshop scheduling problem. Computers & Operations Research. 2014 May 1;45:60-7.

[15] Benavides AJ. A New Tiebreaker in the NEH heuristic for the Permutation Flow Shop Scheduling Problem. EasyChair; 2018 Sep 14.

[16] Baskar A, Xavior MA. New idle time-based tie-breaking rules in heuristics for the permutation flowshop scheduling problems. Computers & Operations Research. 2021 Sep 1;133:105348.

[17] Carlier J. Ordonnancements a contraintes disjonctives. RAIRO-Operations Research. 1978;12(4):333-50.

[18] Demirkol E, Mehta S, Uzsoy R. Benchmarks for shop scheduling problems. European Journal of Operational Research. 1998 Aug 16;109(1):137-41.

[19] Taillard E. Benchmarks for basic scheduling problems. european journal of operational research. 1993 Jan 22;64(2):278-85.

[20] Taillard E. Summary of best known lower and upper bounds of Taillard's instances. Available in http://ina2. eivd. ch/collaborateurs/etd/problemes.dir/ordonnancement.dir/ordonnancement. html. 2005.

[21] Vallada E, Ruiz R, Framinan JM. New hard benchmark for flowshop scheduling problems minimising makespan. European Journal of Operational Research. 2015 Feb 1;240(3):666-77.

[22] Baskar A, Xavior MA, Dhanasakkaravarthi B. Impact of Initial Partial Sequence in the Makespan, in Permutation Flow Shop Scheduling Heuristic Algorithms–An Analysis. Indian Journal of Science and Technology. 2016 Nov 17;9(42).