IMPROVED HEURISTIC FOR MINIMIZING TOTAL ELAPSED TIME IN PERMUTATION FLOW SHOP

[1]Kewal Krishan Nailwal, [2]Deepak Gupta, [3]Kawal Jeet
[1]Assitant Professor, Department of Mathematics, Apeejay College of Fine Arts, Jalandhar, Punjab
[2]Professor and Head, Department of Mathematics and Humanities, M.M University, Mullana, Ambala, Haryana
[3]Associate Professor, Department of Computer Science, D.A.V College, Jalandhar, Punjab

Abstract: This paper presents a heuristic based on the well known CL heuristic with tie breaking rule. The objective under study is the minimization of total elapsed time in permutation flow shop. The paper proposes nine CL heuristic variants, out of which eight heuristics improves upon the original CL heuristic. The performances of all the nine variants of CL heuristic tie breaking rule are tested against original version as well as against the acclaimed NEH heuristic on Taillard problem instances found in the literature of flow shop scheduling with the best known solutions. The improved CL heuristics with tie breaking rule uses a new priority rule instead of the original CL heuristic.

IndexTerms - Flow shop scheduling, total elapsed time, heuristic, permutation.

I. INTRODUCTION

The scheduling system constituting flow shop for the optimization of one or more objectives deals with the execution of distinct machines in series. Numerous real world applications exist in the field of flow shop scheduling. Reid and Stark [30] report the study of a small alloy shop with four machines. The manager and the shop foreman anticipate the prerequisite for job processing which as input is fed to a computer program for job scheduling. Kim et al. [20] studied the production of electronic printed circuit boards consisting of three stages namely surface mounting, automatic insertion of components and manual insertion. The three stages described may involve more than one workstation in series making it a multi-stage flow shop. Bartholdi and Eisenstein [2] described a textile shop study in which the operator works on an item and passes it further to the next crew on the line over multiple stages. Besides these, flow shop has strong background in the setups of cosmetics; pharmaceuticals; food industry and household utensils manufacturing company e.g. see Ying et al. [41], Ying and Liao [42] and Wolde et al. [40]. In the flow shop scheduling problem, *n*-jobs have to be processed through *m*-machines with same technological route for each job. The basic assumptions given by Baker [1] that are commonly made for this problem are: All jobs and machines are available at time zero, For jobs pre-emption is not allowed, during processing all the machines are available and no breakdown takes place, each job is processed through each of the machine once and only once, each machine can perform only one task at a time, machines may remain idle and the setup times of jobs on machines are ignored as considered to be negligibly small. Recently,Takano and Nagano [36] discussed the problem by relaxing few of the assumptions of unlimited buffering and setup time included in processing time. The problem for minimizing total elapsed time in a flow shop is a classical and is a typical combinatorial optimization problem. Total elapsed time is defined as the total time elapsed before the completion of given set of jobs on given number of machines. Mathematically, let some job $i\,(1 \le i \le n)$ is to be scheduled on machine $l(1 \le l \le m)$ in the same technological order with criteria to be optimized as minimization of total elapsed time $C_{\max}$. Let $t_{i,l}$ and $C_{i,l}$ be the time of processing and completion time of the job *i* on the machine *l*, then the calculation for $C_{\max}$ follows the formula as given by:

$$C_{i,l} = \max\{C_{i,(l-1)}, C_{(i-1),l}\} + t_{i,l} \quad \text{where } C_{i,0} = 0 \text{ and } C_{0,l} = 0 \;\forall\; i,l.$$

Total elapsed time, $C_{\max} = C_{n,m}$.

With the notation of Graham et al. [15], the problem of minimizing total elapsed time in flow shop can be put as $F_m / prmu / C_{\max}$.

Ignall & Schrage [16] and Lomnicki [22] during the same time phase independently solved the problem $F_m / prmu / C_{\max}$ by using branch-and-bound methods which is exact techniques for solving the problem. Being a strongly NP-hard for more than two machines does not yield the solution in polynomial time even for small instances see Garey et al. [9]. The alternative to exact technique is heuristics which finds near optimal or optimal solutions in better time frame. Firstly, Johnson [17] proposed the solution for $F_2 / prmu / C_{\max}$ giving optimal solution. This becomes a benchmark in the scheduling literature and many heuristics namely CDS proposed by Campbell et al. [4], HFC heuristic proposed by Koulamas [21], NEHKK1 proposed by Kalczynski and Kamburowski [19] and Gupta and Chauhan [10] used the idea of Johnson [17] for solving this flow shop problem in scheduling. Page [25] suggested that the sorting might be applied to the job sequencing problem as sequencing problem is unlike the sorting problem in data processing. Gupta [12] shows the dominance over the CDS heuristic by developing an algorithm based on sorting.

Palmer [26] used the idea of priority rule for solving *n*-job, *m*-machine flow shop problem with minimizing criteria as total elapsed time. The modification of [26] in the scheduling history is the heuristic proposed by Gupta [11]. Dannenbring [6] combined the benefits of both [26] and [4] and gave three heuristics namely RA, RACS and RAES. After these heuristics Nawaz et al. [24] used the idea of priority rule for jobs processing namely NEH heuristic for minimizing total elapsed time. NEH heuristic uses a powerful insertion technique. The literature after NEH was proposed mainly be categorized into two types; First category of literature defends the NEH heuristic as superior heuristics and other tried to improve upon NEH heuristics combining the benefits of several other heuristics from the literature. Under first category Park et al. [28], Turner and Booth [37], Watson et al. [39] and Kalczynski and Kamburowski [18] defend the superiority of NEH over other heuristics prevalent in literature. In Second category, Sarin and Lefoka [34] finds better results in their heuristic than NEH for the number of machines less than 100. On the contrary, NEH performed better for large problems. The heuristic NEHNM & NEHNM1 proposed by Nagano and Moccellin [23], Chakraborty and Laha [5], NEH-D proposed by Dong et al. [7] and CL_{WTS}& CL_{WOTS} proposed by Ying and Lin [43] improved the solution of NEH. Gupta et al. [13] proposed an algorithm for minimizing total elapsed time with one or more alternative sequences of jobs processing. Rad et al. [29] proposed high performing heuristic FRB5 based on local search algorithm which improves the solution by re-inserting the jobs. Rossi et al. [32] reported G1 to G15 which are based on the different combinations of initial order and construction stage taken from heuristics such as NEH, NEH-D, NEHKK1 etc. Again, by re-inserting the jobs, solution is improved upon in proposed heuristics. The performance measure total elapsed time values being evaluated on benchmark problems suggested by Vallada et al. [38] along with the Taillard instances [35]. NEHAB1, NEHAB2, NEHAB3, NEHAB4, NEHAB5, NEHAB6, NEHAB7 are the seven versions proposed by Baskar [3] during the same time out which NEHAB1 gave the improved results than NEH and others on the Taillard's instances.

The extensive review and grouping of heuristics as per development stages can be seen in Framinan et al. [8], Ruiz and Maroto [33], Reza Hejazi and Saghafian [31], Gupta and Stafford [14] and Pan and Ruiz [27]. Framinan et al. [8] describes that a general framework in the development of heuristics consists of three stages namely index development, solution construction and solution improvement. For the first stage i.e. index development stage, we used the priority rule as described by NEH-D algorithm [7]. We generated nine CL_{WTS} variants for same priority rule with different parameter values of $\alpha = 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90$. We named them as $CL_{WTS}1$, $CL_{WTS}2$, $CL_{WTS}3$, $CL_{WTS}4$, $CL_{WTS}4$, $CL_{WTS}5$, $CL_{WTS}6$, $CL_{WTS}6$, $CL_{WTS}7$, $CL_{WTS}8$ and $CL_{WTS}9$ and the results supports the superiority of $CL_{WTS}2$ over CL_{WTS} and NEH algorithm. A different tie breaking rule is proposed in the second stage of development i.e. construction stage and is described later in the section 2. The various versions of heuristic proposed are compared with CL_{WTS} and NEH heuristic over the Taillard 120 benchmark instances. The rest of paper is organized as follows: Section 2 describes the steps of the heuristic algorithm proposed; Computational results are given in Section 3 and Section 4 draws the conclusion from the results obtained which are followed by references.

II. Proposed Heuristic

The proposed constructive heuristic described below finds the solution of the problem $F_m / prmu / C_{\max}$ efficiently. Before explaining the heuristic proposed, the tie breaking rule used in it is explained. The various steps involved are described as follows:

2.1 The tie breaking rule

Out of the sequences having same total elapsed time generated during the run of algorithm one sequence is to be selected from them. For this, a tie breaking rule for these sequences is adopted:

For the sequences having tie for total elapsed time, MIT (Minimum idle time on machines) is determined using the relation MIT = $\sum_{i=2}^{n}\sum_{l=2}^{m}\max(C_{i(l-1)} - C_{(i-1)l}, 0)$. The tie between the sequences is removed by selecting the sequence having minimum MIT. The tie for MIT is removed by selecting a sequence with smaller index.

2.2 Steps of proposed heuristic

Step 1: Compute T_i of each job i (i =1,2,3,…,n) on the given m-machines by the expression:

$$T_i = \alpha AV_i + (1-\alpha) SD_i$$

$$\text{Where, } AV_i = \frac{1}{m}\sum_{l=1}^{m} t_{i,l}$$

$$SD_i = \sqrt{\frac{1}{m}\sum_{l=1}^{m}(t_{i,l} - AV_i)^2}$$

Step 2: Store the jobs in decreasing values of T_i obtained in step 1.

Step 3: Repeat the steps of CL_{WTS} heuristic proposed by Ying and Lin [43] to get the final sequence with minimum value of total elapsed time with the new tie breaking rule explained in section 2.1. The sequence with minimum total elapsed time is stored as the final sequence.

Different versions of above algorithms are generated with $\alpha = 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90$ and are termed as $CL_{WTS}1$, $CL_{WTS}2$, $CL_{WTS}3$, $CL_{WTS}4$, $CL_{WTS}5$, $CL_{WTS}6$, $CL_{WTS}7$, $CL_{WTS}8$ and $CL_{WTS}9$. As the changes for improvement is done in the step 1 of CL_{WTS} heuristic and the tie breaking rule, the total number of sequences generated (partial and complete) remains the same in the proposed heuristic or we can say that the computational complexity of proposed algorithm i.e. $\Theta(n^3 m)$ is same as the computational complexity of CL_{WTS}.

III. Computational Results

The various version $CL_{WTS}i$ (i = 1,2,3,4,5,6,7,8,9) of CLWTS are evaluated on 120-Taillard's benchmark problems and is found that CL_{WTS} 2 performs well than others. Twelve total elapsed time values with their final sequences are given in table 1. CL_{WTS} 2 is compared with the original CLWTS heuristic and the famous NEH over the benchmark instances given above in this section. The proposed algorithms are coded in MATLAB-2008a and are made to run on i-5 processor. The values of original CL_{WTS} heuristic are taken from Ying and Lin [43]. The comparison of the proposed heuristics is shown in the table 2, table 3, table 4 and table 5 along with other heuristics considered. These tables describe the problem description, the value of the total elapsed time obtained from the heuristics considered. The Average Relative Percentage Deviation (ARPD) denotes the average of the Relative Percentage Deviation for the particular set of the problems. The ARPD of the heuristics considered is compared against the proposed heuristic algorithm on Taillards problem instances and the results are given in table 6.

From the results of table 6, The overall average relative percentage deviation of the , NEH, CL_{WTS} and various version $CL_{WTS}i$ (i = 1,2,3,4,5,6,7,8,9) of CL_{WTS} on all 120 Taillard problem instances stands at 3.32, 3.101, 3.175, 2.912, 2.99, 3.055, 2.946, 3.08, 3.032, 3.003 and 2.997 respectively. The ARPD of NEH, CL_{WTS} and CL_{WTS} 2 (in table 6) are plotted against the 120- Taillard problem instances in the fig 1. From the table 6 and the fig1, it is clear that the CL_{WTS} 2 is superior to NEH, CL_{WTS} and other various versions of proposed heuristics considered for comparisons.

Table 1. Final Sequences and total elapsed time obtain by $CL_{WTS}2$

Problem	Final Sequence	Total elapsed time
Ta 001	17,3,15,14,2,8,5,4,9,18,1,19,13,16,6,7,11,12,10,20	1297
Ta 011	18,5,2,9,12,17,7,13,10,20,11,3,6,15,8,14,19,4,1,16	1664
Ta 021	16,14,8,7,15,17,10,11,6,12,18,9,13,5,1,20,2,4,3,19	2374
Ta 031	41,12,31,30,6,26,45,28,46,27,24,18,1,39,10,11,23,5,21,22,32,34,17,4,14,47,40,25,42,9,13,43,8,2,15,29,50,16,33,7,3,35,19,44,20,48,49,38,37,36	2745
Ta 041	18,49,8,44,3,25,20,28,30,40,7,38,26,46,36,37,27,13,6,34,10,33,2,31,29,42,15,4,24,21,16,5,43,19,9,14,35,41,17,11,12,1,22,32,50,23,48,47,45,39	3113
Ta 051	35,43,29,37,31,11,12,24,15,22,38,6,46,4,47,27,36,1,16,19,14,20,41,34,7,39,8,40,44,49,45,48,42,17,23,13,2,5,10,21,33,18,50,26,32,30,25,9,28,3	4039
Ta 061	26,10,59,27,72,25,56,76,20,6,11,12,4,60,62,23,40,19,24,90,91,64,42,51,65,93,21,5,38,46,82,96,85,95,15,37,2,78,53,32,50,71,49,9,81,70,77,54,73,57,94,74,63,88,75,67,80,29,98,66,97,58,34,41,16,35,22,31,28,44,8,18,39,43,100,86,79,83,17,89,7,36,92,33,69,61,48,47,99,45,13,3,1,52,68,14,55,87,30,84	5503
Ta 071	70,58,61,56,96,80,62,10,37,17,86,41,78,32,44,31,21,1,53,60,35,74,89,55,20,6,67,64,29,91,69,43,87,79,77,81,9,11,39,50,95,93,82,19,36,2,92,5,47,49,65,24,98,72,71,14,73,23,76,68,3,48,46,97,22,7,8,16,42,33,4,59,57,28,83,27,85,25,51,30,66,34,94,100,45,88,15,13,84,18,90,38,99,63,52,75,54,40,26,12	5872
Ta 081	22,78,54,5,51,40,83,55,21,11,82,61,88,25,1,100,34,99,47,59,79,2,30,74,37,44,35,53,31,16,66,81,57,49,77,80,36,71,46,33,86,95,7,45,17,12,6,3,9,24,48,56,62,98,28,60,10,20,70,87,52,18,23,32,19,26,73,91,68,13,65,72,15,85,50,58,39,94,93,92,67,76,96,4,64,27,14,89,29,90,75,43,8,97,69,84,63,42,38,41	6598
Ta 091	23,73,29,66,24,151,197,87,94,135,77,71,1,18,128,172,108,2,147,39,164,16,70,121,173,10,113,200,13,47,176,35,196,127,115,3,132,174,123,85,134,191,69,33,195,152,107,19,148,12,27,99,183,198,17,170,14,106,36,75,111,163,28,52,34,22,43,42,186,30,156,21,6,181,9,109,83,142,72,149,190,89,116,8,100,48,193,166,159,68,57,185,138,20,192,11,129,102,189,84,177,55,165,199,78,146,131,175,79,117,182,144,103,38,81,125,105,92,179,26,82,145,130,112,160,143,150,97,56,7,110,139,136,53,25,153,74,90,32,37,96,180,141,168,59,86,64,140,120,126,119,187,49,88,46,50,155,60,184,95,118,98,161,41,45,58,76,101,114,44,80,40,104,171,4,62,5,157,91,51,31,122,63,137,162,67,154,93,54,167,124,194,158,169,178,188,65,15,61,133	10906
Ta 101	83,76,150,97,131,66,151,126,115,152,89,144,91,163,138,3,67,92,79,65,63,147,132,182,57,107,111,99,13,59,137,113,74,198,7,39,90,174,98,22,72,85,88,135,140,184,190,8,128,127,21,95,105,94,117,125,37,6,25,33,20,136,96,158,145,48,116,139,120,24,181,176,36,178,17,146,123,106,173,47,73,153,70,161,168,93,69,122,61,4,40,75,10,133,101,102,110,195,179,54,43,56,5,11,32,172,167,169,175,160,118,143,2,87,197,194,18,35,68,177,155,141,50,86,186,64,142,53,15,187,12,34,188,130,42,103,104,148,71,9,58,1,80,119,162,124,78,189,29,52,199,46,154,109,51,100,156,108,121,112,77,159,82,164,31,30,129,28,16,44,165,185,170,149,49,26,157,81,134,180,191,84,14,55,27,166,200,41,196,171,19,62,183,60,23,192,38,193,45,114	11657
Ta 111	452,295,183,128,49,88,58,288,256,16,347,392,398,282,304,216,234,22,485,145,113,287,268,116,362,315,370,85,45,114,312,135,327,278,43,293,306,369,303,200,267,151,124,83,157,335,467,9,245,324,208,491,155,168,325,473,78,407,73,338,368,279,5,198,342,143,475,439,318,377,290,449,201,31,224,323,470,379,412,242,6,138,97,53,101,378,192,104,167,344,106,162,136,10,99,497,91,471,289,329,66,225,79,423,219,107,149,388,29,414,319,86,354,274,108,61,307,161,2,488,264,33,372,322,487,498,217,281,493,461,231,291,387,453,120,331,380,226,482,62,298,51,382,37,418,67,321,438,25,472,127,189,23,442,356,229,451,277,34,258,431,456,24,68,411,220,495,193,339,153,455,248,413,355,499,476,243,207,440,450,286,105,102,75,36,443,90,186,490,366,408,100,257,328,1,233,361,38,125,444,119,296,202,44,218,359,235,176,500,56,390,209,59,285,111,346,345,409,197,484,468,441,92,394,39,384,134,422,159,76,163,87,164,158,284,496,391,402,480,297,188,154,46,436,54,142,363,460,178,187,406,477,211,169,240,98,84,166,152,459,237,457,437,19,74,337,404,255,272,117,479,210,156,232,172,252,276,419,494,222,146,270,213,236,310,144,489,332,17,330,426,165,212,81,486,194,203,148,254,326,121,401,415,173,11,432,397,3,13,367,371,399,174,302,182,14,271,492,299,190,405,311,57,35,341,63,386,26,280,421,429,195,206,374,93,215,427,253,181,320,463,132,446,351,316,396,199,433,416,462,69,348,259,395,205,15,333,364,223,357,301,352,275,478,177,263,221,376,147,424,383,65,294,358,184,464,360,179,41,227,8,246,30,60,89,129,309,80,454,465,481,292,94,139,96,239,170,447,334,410,340,266,420,448,373,131,141,82,308,230,425,269,417,123,7,238,381,42,365,466,247,27,185,52,12,214,109,434,110,265,64,317,28,261,50,175,249,336,400,305,273,95,435,353,474,349,251,262,343,445,375,122,160,350,458,483,140,260,72,48,250,133,18,228,171,393,428,103,70,283,196,180,385,40,4,112,314,47,115,469,150,403,137,313,126,389,118,241,32,71,130,204,55,21,191,244,77,20,430,300	26583

Table 2. Total elapsed time Values for variants of CL_{WTS} heuristic with Taillard 20-job Instances

Problem Description		Total elapsed time										
Problem Instance	Upper Bound	NEH	CL_{WTS}	CL_{WTS} 1	CL_{WTS} 2	CL_{WTS} 3	CL_{WTS} 4	CL_{WTS} 5	CL_{WTS} 6	CL_{WTS} 7	CL_{WTS} 8	CL_{WTS} 9
20x5												
1	1278	1286	1305	1292	1297	1297	1297	1297	1284	1288	1290	1294
2	1359	1365	1371	1377	1367	1365	1383	1383	1373	1377	1383	1383
3	1081	1159	1135	1147	1136	1107	1118	1122	1134	1132	1140	1118
4	1293	1325	1323	1354	1309	1354	1353	1323	1323	1315	1327	1314
5	1235	1305	1305	1250	1283	1283	1283	1283	1305	1305	1305	1305
6	1195	1228	1210	1234	1215	1212	1217	1210	1220	1195	1210	1210
7	1239	1278	1270	1251	1254	1251	1254	1253	1254	1266	1252	1253
8	1206	1223	1224	1232	1214	1239	1250	1236	1223	1225	1221	1244
9	1230	1291	1292	1268	1264	1269	1261	1267	1292	1277	1275	1292
10	1108	1151	1127	1151	1143	1143	1154	1137	1144	1131	1131	1131
20x10												
1	1582	1680	1646	1650	1664	1639	1632	1658	1677	1677	1686	1686
2	1659	1729	1711	1721	1735	1703	1701	1701	1718	1718	1718	1725
3	1496	1557	1559	1555	1548	1527	1557	1541	1552	1558	1551	1560
4	1377	1439	1455	1407	1431	1410	1414	1427	1418	1421	1419	1433
5	1419	1502	1502	1523	1475	1489	1487	1473	1532	1474	1474	1490
6	1397	1453	1433	1475	1471	1455	1465	1452	1476	1480	1458	1453
7	1484	1562	1526	1529	1518	1522	1546	1511	1533	1518	1518	1523
8	1538	1609	1610	1609	1578	1637	1637	1612	1612	1593	1593	1610
9	1593	1647	1647	1667	1634	1653	1620	1636	1638	1637	1628	1659
10	1591	1653	1649	1696	1698	1685	1675	1672	1682	1669	1654	1665
20x20												
1	2297	2410	2397	2386	2374	2390	2404	2380	2441	2425	2402	2410
2	2099	2150	2150	2194	2153	2146	2152	2137	2168	2137	2153	2160
3	2326	2411	2411	2413	2429	2399	2390	2405	2381	2408	2418	2400
4	2223	2262	2290	2277	2274	2281	2296	2319	2262	2282	2268	2262
5	2291	2397	2384	2355	2400	2365	2382	2363	2381	2368	2385	2404
6	2226	2349	2349	2308	2287	2346	2308	2378	2306	2306	2306	2315
7	2273	2362	2360	2382	2318	2329	2356	2353	2354	2393	2385	2360
8	2200	2249	2249	2257	2269	2266	2273	2259	2264	2249	2249	2249
9	2237	2320	2323	2363	2310	2321	2381	2292	2335	2310	2306	2290
10	2178	2277	2270	2268	2309	2271	2294	2244	2309	2222	2222	2223

Table 3. Total elapsed time Values for variants of CL_{WTS} heuristic with Taillard 50-job Instances

Problem Description		Total elapsed time										
Problem Instance	Upper Bound	NEH	CL_{WTS}	CL_{WTS} 1	CL_{WTS} 2	CL_{WTS} 3	CL_{WTS} 4	CL_{WTS} 5	CL_{WTS} 6	CL_{WTS} 7	CL_{WTS} 8	CL_{WTS} 9
50x5												
1	2724	2733	2729	2744	2745	2729	2725	2724	2729	2742	2724	2724
2	2834	2843	2848	2882	2882	2882	2882	2882	2882	2882	2882	2882
3	2621	2640	2633	2635	2626	2639	2633	2637	2643	2641	2641	2638
4	2751	2782	2762	2778	2762	2762	2762	2766	2762	2762	2802	2773
5	2863	2868	2886	2882	2873	2873	2873	2877	2877	2877	2877	2877
6	2829	2850	2839	2834	2835	2835	2835	2838	2852	2852	2835	2839
7	2725	2758	2732	2725	2739	2736	2736	2736	2736	2736	2734	2736
8	2683	2721	2688	2734	2711	2699	2692	2683	2702	2709	2716	2718
9	2552	2576	2565	2565	2554	2577	2574	2575	2565	2555	2565	2568
10	2782	2790	2804	2796	2797	2796	2796	2802	2789	2787	2794	2800
50x10												
1	2991	3135	3155	3111	3113	3112	3147	3126	3159	3136	3160	3135
2	2867	3032	3076	3008	3011	3025	3022	3061	3039	3034	3036	3010
3	2839	2986	3013	2994	2965	2986	3011	3014	2983	3004	3016	2992
4	3063	3198	3156	3157	3088	3121	3137	3162	3175	3145	3120	3165
5	2976	3160	3185	3102	3076	3150	3158	3135	3172	3168	3134	3200
6	3006	3178	3127	3137	3163	3193	3133	3165	3158	3161	3118	3120
7	3093	3277	3259	3289	3238	3289	3217	3271	3294	3264	3234	3273
8	3037	3123	3147	3125	3093	3098	3102	3173	3135	3142	3162	3144
9	2897	3002	3047	3013	3027	3005	3035	2983	3018	3022	3005	3059

10	3065	3257	3204	3198	3197	3213	3198	3187	3198	3204	3207	3205
50x20												
1	3850	4082	4069	4123	4039	4061	4067	4042	4047	4083	4030	4059
2	3704	3921	3958	3948	3916	3917	3960	3938	3924	3974	3919	3907
3	3640	3927	3882	3898	3878	3891	3896	3853	3831	3893	3948	3854
4	3723	3969	3998	3916	3939	3932	3945	3945	3950	3995	3954	3953
5	3611	3835	3834	3880	3787	3880	3807	3819	3857	3877	3839	3852
6	3681	3914	3859	3934	3934	3887	3877	3872	3906	3886	3975	3869
7	3704	3952	3931	3907	3887	3981	3911	3945	4005	3934	3921	3918
8	3691	3938	3925	3894	3886	3888	3907	3909	3876	3905	3899	3918
9	3743	3952	3949	3966	3963	3972	3946	3980	3976	3960	3945	3998
10	3756	4079	4012	3994	3971	4037	3984	3999	3994	3987	3992	3939

Table 4. Total elapsed time Values for variants of CL_{WTS} heuristic with Taillard 100-job Instances

Problem Description		Total elapsed time										
Problem Instance	Upper Bound	NEH	CL_{WTS}	CL_{WTS} 1	CL_{WTS} 2	CL_{WTS} 3	CL_{WTS} 4	CL_{WTS} 5	CL_{WTS} 6	CL_{WTS} 7	CL_{WTS} 8	CL_{WTS} 9
100x5												
1	5493	5519	5501	5529	5503	5493	5493	5493	5493	5493	5514	5495
2	5268	5348	5289	5284	5284	5284	5285	5284	5284	5284	5298	5287
3	5175	5219	5216	5206	5206	5197	5208	5195	5195	5207	5221	5219
4	5014	5023	5023	5059	5023	5036	5029	5027	5021	5021	5021	5023
5	5250	5266	5256	5255	5255	5255	5255	5256	5256	5256	5255	5256
6	5135	5139	5139	5139	5139	5139	5139	5139	5139	5139	5139	5139
7	5246	5259	5284	5281	5278	5265	5279	5281	5288	5257	5255	5282
8	5094	5120	5123	5126	5113	5124	5104	5107	5123	5112	5106	5107
9	5448	5489	5482	5467	5460	5456	5485	5482	5487	5487	5467	5458
10	5322	5341	5344	5354	5339	5334	5338	5328	5342	5350	5354	5337
100x10												
1	5770	5846	5831	5901	5872	5856	5850	5827	5870	5861	5846	5807
2	5349	5428	5431	5418	5442	5421	5461	5428	5408	5410	5419	5416
3	5676	5824	5777	5740	5771	5785	5725	5756	5753	5731	5792	5741
4	5781	5929	6006	6023	5966	5944	5960	5942	5958	5977	5960	5906
5	5467	5661	5633	5598	5632	5634	5629	5634	5562	5655	5608	5634
6	5303	5375	5363	5346	5371	5404	5362	5379	5400	5356	5362	5383
7	5595	5704	5699	5702	5706	5703	5709	5687	5701	5754	5707	5690
8	5617	5760	5727	5722	5695	5716	5714	5743	5717	5713	5704	5714
9	5871	5932	5990	5938	5946	5938	5903	5925	5930	5934	5914	5955
10	5845	5891	5903	5939	5928	5928	5928	5928	5928	5928	5928	5928
100x20												
1	6202	6541	6588	6575	6598	6519	6561	6552	6558	6565	6691	6553
2	6183	6523	6482	6533	6566	6482	6591	6554	6511	6553	6568	6450
3	6271	6639	6584	6556	6566	6485	6543	6580	6588	6594	6637	6544
4	6269	6557	6567	6571	6604	6548	6560	6543	6485	6522	6621	6549
5	6314	6695	6642	6645	6621	6638	6603	6609	6658	6629	6648	6602
6	6364	6664	6694	6684	6670	6777	6701	6670	6656	6686	6692	6689
7	6268	6632	6617	6713	6690	6636	6649	6580	6585	6666	6566	6554
8	6404	6739	6818	6749	6771	6727	6788	6839	6727	6788	6734	6820
9	6275	6677	6650	6673	6635	6629	6628	6605	6603	6607	6588	6588
10	6434	6677	6699	6787	6772	6710	6771	6733	6686	6679	6722	6740

Table 5. Total elapsed time Values for variants of CL_{WTS} heuristic with Taillard 200-job and 500-job Instances

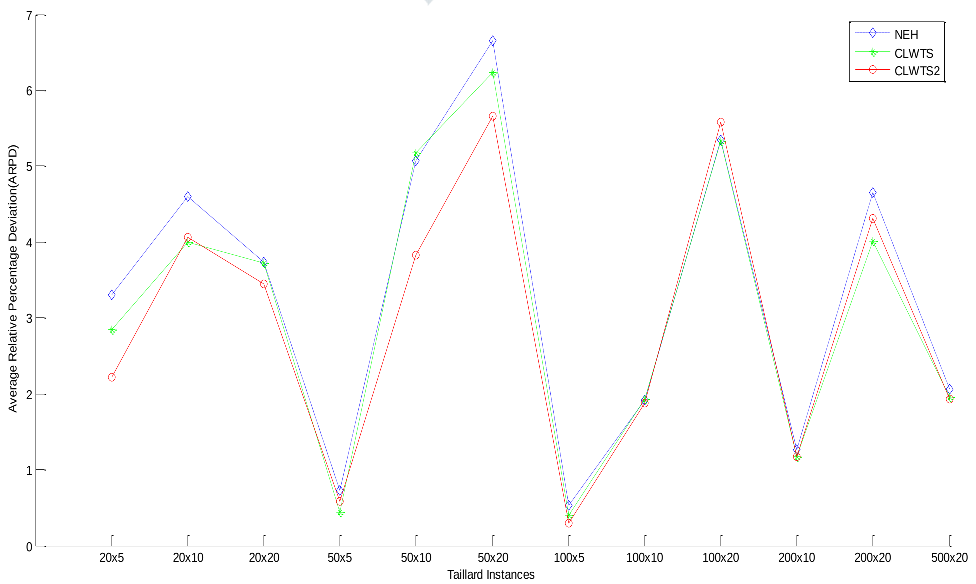
Problem Description		Total elapsed time										
Problem Instance	Upper Bound	NEH	CL_{WTS}	CL_{WTS} 1	CL_{WTS} 2	CL_{WTS} 3	CL_{WTS} 4	CL_{WTS} 5	CL_{WTS} 6	CL_{WTS} 7	CL_{WTS} 8	CL_{WTS} 9
200x10												
1	10862	10942	10942	10921	10906	10931	10986	10888	10885	10942	10941	10936
2	10480	10716	10741	10669	10661	10685	10731	10727	10615	10670	10678	10623
3	10922	11025	11027	11030	11045	11060	11060	11041	11034	11094	11030	11038
4	10889	11057	11050	11050	11050	11051	11050	11051	11051	11051	11050	11051
5	10524	10645	10597	10656	10646	10608	10624	10664	10566	10583	10629	10612
6	10329	10458	10467	10443	10421	10418	10482	10427	10426	10446	10427	10406
7	10854	10989	10962	10963	10956	10949	10955	10937	10918	10916	10923	10941

8	10730	10829	10817	10878	10869	10840	10878	10818	10799	10828	10838	10845
9	10438	10574	10558	10594	10558	10561	10524	10570	10585	10572	10532	10522
10	10675	10807	10790	10818	10840	10809	10782	10799	10758	10759	10819	10758
200x20												
1	11195	11594	11530	11625	11657	11571	11533	11611	11552	11577	11518	11529
2	11203	11675	11648	11786	11692	11718	11633	11712	11693	11715	11748	11740
3	11281	11852	11788	11743	11763	11835	11792	11787	11792	11680	11808	11757
4	11275	11803	11695	11741	11706	11694	11695	11761	11682	11713	11751	11743
5	11259	11685	11668	11629	11636	11620	11624	11667	11656	11621	11639	11657
6	11176	11629	11676	11653	11701	11655	11651	11604	11650	11657	11679	11705
7	11360	12103	11809	12042	12043	12076	12075	12070	12075	12049	11992	12022
8	11334	11913	11753	11832	11891	11774	11765	11788	11830	11792	11768	11866
9	11192	11673	11678	11703	11619	11654	11687	11580	11622	11696	11637	11695
10	11288	11869	11838	11771	11704	11791	11691	11756	11750	11794	11772	11835
500x20												
1	26059	26670	26659	26672	26583	26673	26690	26593	26672	26642	26685	26757
2	26520	27232	27145	27188	27146	27204	27148	27065	27124	27164	27196	27161
3	26371	26848	26835	26927	26940	26922	26902	26953	26909	26837	26801	26866
4	26456	27055	26935	26976	26988	26977	26983	26971	27040	26960	26903	26918
5	26334	26727	26890	26768	26756	26808	26843	26793	26725	26752	26875	26754
6	26477	26992	26990	27026	27007	26933	27061	26976	26997	27041	26942	27033
7	26389	26797	26726	26827	26776	26799	26752	26682	26880	26779	26719	26817
8	26560	27138	27165	26971	27061	27221	27237	27104	27019	27054	27049	26976
9	26005	26631	26555	26670	26477	26528	26539	26529	26506	26548	26510	26601
10	26457	26984	26877	26969	26966	26988	26970	26932	26905	26976	26969	26932

Table 6. Average Relative Percentage Deviation on Taillard Instances

Problem Instances	Average Relative Percentage Deviation(APRD)										
	NEH	CLWTS	CLWTS 1	CLWTS 2	CLWTS 3	CLWTS 4	CLWTS 5	CLWTS 6	CLWTS 7	CLWTS 8	CLWTS 9
20x5	3.301	2.84	2.823	2.217	2.481	2.899	2.416	2.78	2.426	2.615	2.68
20x10	4.601	4.003	4.375	4.06	3.848	3.971	3.614	4.656	4.025	3.712	4.401
20x20	3.731	3.716	3.82	3.454	3.415	3.961	3.48	3.807	3.331	3.308	3.218
50x5	0.727	0.444	0.77	0.579	0.601	0.525	0.568	0.629	0.649	0.751	0.697
50x10	5.073	5.169	4.363	3.827	4.56	4.469	4.849	5.023	4.861	4.571	4.934
50x20	6.648	6.238	6.355	5.655	6.32	5.923	5.928	6.103	6.449	6.262	5.836
100x5	0.527	0.404	0.487	0.296	0.266	0.324	0.28	0.348	0.306	0.351	0.303
100x10	1.919	1.926	1.861	1.875	1.879	1.725	1.737	1.692	1.856	1.716	1.604
100x20	5.34	5.331	5.562	5.575	5.028	5.418	5.21	4.883	5.252	5.538	4.927
200x10	1.258	1.173	1.239	1.172	1.134	1.285	1.147	0.877	1.086	1.092	0.963
200x20	4.645	4.015	4.407	4.306	4.284	4.069	4.238	4.208	4.202	4.218	4.428
500x20	2.066	1.954	2.037	1.924	2.058	2.085	1.885	1.954	1.944	1.905	1.969
Total	3.32	3.101	3.175	**2.912**	2.99	3.055	2.946	3.08	3.032	3.003	2.997

Fig 1. Plot of ARPDs of Heuristics versus Taillard Problem Instances



IV. Conclusion

The paper solves a permutation flow shop scheduling problem for minimizing total elapsed time by presenting a constructive heuristic algorithm. Out of the nine variants proposed, eight heuristics namely $CL_{WTS}2$, $CL_{WTS}3$, $CL_{WTS}4$, $CL_{WTS}5$, $CL_{WTS}6$, $CL_{WTS}7$, $CL_{WTS}8$ and $CL_{WTS}9$ improves upon the solution of CL_{WTS} and NEH heuristic. The best heuristic $CL_{WTS}2$ outperforms all the variants along with NEH and CL_{WTS} heuristic on the 120-set of instances given by Taillard. The less average relative percentage deviation (ARPD) from the best known solutions found in the flow shop scheduling literature of the heuristic proposed shows the superiority over the heuristic compared. Only one objective namely total elapsed time is discussed in this work but the modification of the proposed heuristic can be tested for bi-objective flow shop scheduling problems.

V. Acknowledgment

This research is fully supported by University Grants Commission (UGC) of India under the minor research project F.No. 8-4(73)/2015(MRP/NRCB).

References

[1] Baker, K. R. (1974). Introduction to sequencing and scheduling. John Wiley & Sons.

[2] Bartholdi III, J. J., & Eisenstein, D. D. (1996). A production line that balances itself. Operations Research, 44(1), 21-34.

[3] Baskar, A. (2016). Revisiting the NEH algorithm-the power of job insertion technique for optimizing the makespan in permutation flow shop scheduling. International Journal of Industrial Engineering Computations, 7(2), 353-366.

[4] Campbell, H.G., Dudek, R.A. & Smith, M.L. (1970). A heuristic algorithm for n-jobs, m-machines sequencing problem. Management Science, 16(10), 630-637.

[5] Chakraborty, U. K., & Laha, D. (2007). An improved heuristic for permutation flowshop scheduling. International Journal of Information and communication technology, 1(1), 89-97.

[6] Dannenbring, D. G. (1977). An evaluation of flow shop sequencing heuristics. Management science, 23(11), 1174-1182.

[7] Dong, X., Huang, H., & Chen, P. (2008). An improved NEH-based heuristic for the permutation flowshop problem. Computers & Operations Research, 35(12), 3962-3968.

[8] Framinan, J. M., Gupta, J. N., & Leisten, R. (2004). A review and classification of heuristics for permutation flow-shop scheduling with makespan objective. Journal of the Operational Research Society, 55(12), 1243-1255.

[9] Garey, M. R., Johnson, D. S., & Sethi, R. (1976). The complexity of flowshop and jobshop scheduling. Mathematics of operations research, 1(2), 117-129.

[10] Gupta, A., & Chauhan, S. (2015). A heuristic algorithm for scheduling in a flow shop environment to minimize makespan. International Journal of Industrial Engineering Computations, 6(2), 173-184.

[11] Gupta, J. N. D. (1971). A functional heuristic algorithm for the flowshop scheduling problem. Journal of the Operational Research Society, 22(1), 39-47.

[12] Gupta, J. N. D. (1976). A heuristic algorithm for the flowshop scheduling problem. Revue française d'automatique, informatique, recherche opérationnelle. Recherche opérationnelle, 10(V2), 63-73.

[13] Gupta, D., Nailwal, K. K., & Sharma, S. (2014). A heuristic for permutation flowshop scheduling to minimize makespan. In Proceedings of the Third International Conference on Soft Computing for Problem Solving (pp. 423-432). Springer, New Delhi.

[14] Gupta, J. N., & Stafford, E. F. (2006). Flowshop scheduling research after five decades. European Journal of Operational Research, 169(3), 699-711.

[15] Graham, R. L., Lawler, E. L., Lenstra, J. K., & Kan, A. R. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. Annals of discrete mathematics, 5, 287-326.

[16] Ignall, E., & Schrage, L. (1965). Application of the branch and bound technique to some flow-shop scheduling problems. Operations research, 13(3), 400-412.

[17] Johnson, S. M. (1954). Optimal two-and three-stage production schedules with setup times included. Naval Research Logistics (NRL), 1(1), 61-68.

[18] Kalczynski, P. J., & Kamburowski, J. (2007). On the NEH heuristic for minimizing the makespan in permutation flow shops. Omega, 35(1), 53-60.

[19] Kalczynski, P. J., & Kamburowski, J. (2008). An improved NEH heuristic to minimize makespan in permutation flow shops. Computers & Operations Research, 35(9), 3001-3008.

[20] Kim, Y. D., Lim, H. G., & Park, M. W. (1996). Search heuristics for a flowshop scheduling problem in a printed circuit board assembly process. European Journal of Operational Research, 91(1), 124-143.

[21] Koulamas, C. (1998). A new constructive heuristic for the flowshop scheduling problem. European Journal of Operational Research, 105(1), 66-71.

[22] Lomnicki Z.A. (1965). A Branch and Bound algorithm for the exact solution to some flow shop scheduling problems. Operational Research Quarterly, 16, 89-100.

[23] Nagano, M. S., & Moccellin, J. V. (2002). A high quality solution constructive heuristic for flow shop sequencing. Journal of the Operational Research Society, 53(12), 1374-1379.

[24] Nawaz, M., Enscore, E. E., & Ham, I. (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. Omega, 11(1), 91-95.

[25] Page, E. S. (1961). An approach to the scheduling of jobs on machines. Journal of the Royal Statistical Society. Series B (Methodological), 484-492.

[26] Palmer, D. S. (1965). 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, 16(1), 101-107.

[27] Pan, Q. K., & Ruiz, R. (2013). A comprehensive review and evaluation of permutation flowshop heuristics to minimize flowtime. Computers & Operations Research, 40(1), 117-128.

[28] Park, Y. B., Pegden, C. D., & Enscore, E. E. (1984). A survey and evaluation of static flowshop scheduling heuristics. The International Journal of Production Research, 22(1), 127-141.

[29] Rad, S. F., Ruiz, R., & Boroojerdian, N. (2009). New high performing heuristics for minimizing makespan in permutation flowshops. Omega, 37(2), 331-345.
[30] Reid, R. A., & Stark, W. A. (1982). Job Sequencing Program Minimizes Processing Time. Industrial Engineering, 14(1), 26-33.
[31] Reza Hejazi, S., & Saghafian, S. (2005). Flowshop-scheduling problems with makespan criterion: a review. International Journal of Production Research, 43(14), 2895-2929.
[32] Rossi, F. L., Nagano, M. S., & Neto, R. F. T. (2016). Evaluation of high performance constructive heuristics for the flow shop with makespan minimization. The International Journal of Advanced Manufacturing Technology, 87(1-4), 125-136.
[33] Ruiz, R., & Maroto, C. (2005). A comprehensive review and evaluation of permutation flowshop heuristics. European Journal of Operational Research, 165(2), 479-494.
[34] Sarin, S., & Lefoka, M. (1993). Scheduling heuristic for the n-jobm-machine flow shop. Omega, 21(2), 229-234.
[35] Taillard, E. (1993). Benchmarks for basic scheduling problems. european journal of operational research, 64(2), 278-285.
[36] Takano, M. I., & Nagano, M. S. (2017). A branch-and-bound method to minimize the makespan in a permutation flow shop with blocking and setup times. Cogent Engineering, 4(1), 1389638.
[37] Turner, S., & Booth, D. (1987). Comparison of heuristics for flow shop sequencing. Omega, 15(1), 75-78.
[38] Vallada, E., Ruiz, R., & Framinan, J. M. (2015). New hard benchmark for flowshop scheduling problems minimising makespan. European Journal of Operational Research, 240(3), 666-677.
[39] Watson, J. P., Barbulescu, L., Whitley, L. D., & Howe, A. E. (2002). Contrasting structured and random permutation flow-shop scheduling problems: search-space topology and algorithm performance. INFORMS Journal on Computing, 14(2), 98-123.
[40] Wolde, M. G., Berhan, E., & Jilcha, K. (2018). Production improvement with flow shop scheduling heuristics in Household utensils manufacturing company. Cogent Engineering, 5(1), 1430007.
[41] Ying, K. C., Gupta, J. N., Lin, S. W., & Lee, Z. J. (2010). Permutation and non-permutation schedules for the flowline manufacturing cell with sequence dependent family setups. International Journal of Production Research, 48(8), 2169-2184.
[42] Ying, K. C., & Liao, C. J. (2004). An ant colony system for permutation flow-shop sequencing. Computers & Operations Research, 31(5), 791-801.
[43] Ying, K. C., & Lin, S. W. (2013). A high-performing constructive heuristic for minimizing makespan in permutation flowshops. Journal of Industrial and Production Engineering, 30(6), 355-362.