COMBINDED HEURISTIC MEHODS FOR TOTAL FLOW TIME MINIMIZATION IN PERMUTATION FLOW SHOPS SCHEDULING

Wikanda Phaphan[1], Jeerayut Wetweerapong[1,2] and Tawun Remsungnen[1]

[1]Applied Mathematics Research Group (AMRG), Department of Mathematics
Faculty of Science Khon Kaen University, Khon Kaen, Thailand 40002.
[2]Center of Excellence in Mathematics (CEM), Faculty of Science
at Mahidol University - Phayathai Campus 272 Rama 6 Road, Ratchathewi, Bangkok 10400.

ABSTRACT

In this paper, we deal with the classical permutation flow shop problem, where the n jobs are being processed by the same order on m machines. We propose an NEHLF method which is the composition of two modified NEH methods of Framinan (NEHF) and Laha (NEHL). The numerical results show that the NEHLF method outperforms both the NHEF and NEHL methods, while their complexities are of the same order.

KEYWORDS: Flow Shops Scheduling; Total Flow Time; Heuristics.

1. INTRODUCTION

The permutation flow shop scheduling problem (PFSP) is a production problem for finding the best sequence of n jobs that to be processed by m machines in order to minimize the given objective function. This case can be found in manufacturing facilities where the jobs (parts) are moved from machine to machine by material handling devices with no passing is allowed. The problem has been proved to be strongly NP-complete [1] and its total number of possible schedules (sequence) is $n!$ for n jobs. The total flow time and the makespan are of important performance measures, which lead to rapid turn-around of jobs and minimization of in-process inventory [2].

Several heuristic algorithms have been proposed to obtained best solution for PFSP in a short time. The constructive method of Nawaz et al. (denoted by NEH) aimed to minimize the makespan had shown to be most effective one [3-4]. The class, called improvement heuristics based on the NEH method has been studied by many groups Li et al. [5-9]. The methods of Framinan [8] (denoted by NEHF) and Laha [9] (denoted by NEHL) improve NEH method for minimize the total flow time by additional the pairwise interchange and effective neighborhood sequences. The NEHF was found to be better than the method of Woo [7], which has same complexity while the NEHL method significantly enhances the performance of NEHF method. Note that, the total number of generated sequences of NEHL method is approximately two times larger than those of NEHF methods.

However, the set of generated sequences of NEHF method is not a subset of those generated in NEHL method, then it is possible to improve both methods by union their generated sequences sets, which is the main aim of this study.

2. THE METHOD OF FRAMINAN AND LEISTEN

The NEHF method improves upon the idea of the NEH heuristic which is designed for minimizing makespan time. The NEH heuristic obtains a solution by first computing the total processing time *Tj* and sorting the jobs in descending order of *Tj*. Set k=2, select the *k*th job from the sorted list and insert it into *k* possible positions. Then select the best sequence and iterate by incrementing *k* until every jobs are inserted. The detail of method is described as follows:

Step 1: Calculate the total processing time *Tj* as the following,

$$T_j = \sum_{i=1}^{m} t_{ij}$$

for each the *j*th job, $j = 1,2,3,\dots,n$.

Step 2: Sort the jobs in ascending order of the total processing times on all machines.

Step 3: Set k = 2. Compute the total flow time of the first two jobs from the sorted list. Swap the first two jobs and compute its total flow time. Choose the order of the first two job which makes the better total flow time.
Step 4: Increment k, k=k+1. Select the kth job from the sorted list and insert it into k possible positions of the best partial sequence. Among the k partial sequences, select the best one with the least total flow time.
Next, swap jobs in positions i and j of the above partial sequence for all $i, j\ 1 \leq i < k,$ $i < j \leq k$. Select the best partial sequence among $k(k-1)/2$ partial sequences which has minimal total flow time.
Step 5: If $k = n$, then STOP; else, go to Step 4.

By simple inspection, Step 4 determines the complexity of the algorithm by first procedure building k partial sequences and swapping procedure making another $k(k-1)/2$ partial sequences. In total, $k(k+1)/2$ partial sequences should be performed. The complexity of total flow time calculation for each partial schedule of k jobs on m machines is $O(km)$, then the overall complexity of the NEHF method is $O(n^4m)$.

3. THE METHOD OF LAHA AND SARIN

This NEHL heuristic improves upon the idea of the NEHF method by modification pertaining to step 4 of the above procedure. Now, step 4 is executed as follows.

Step 4: Increment k, $k = k + 1$. Select the kth job from the sorted list and insert it into k possible positions of the best partial sequence. Select the best one with the minimal total flow time from these k jobs partial sequences. Next, place each job (except for the kth job of the sorted list) of the best partial sequence into its having $k-1$ positions and select the best partial sequence which makes the better total flow time.

Step 4, now, builds $k + (k-1)^2$ and requires computing total flow time. The complexity of the heuristic is $O(n^4m)$ and is the same complexity as the NEHF method.

4. THE COMBINATION OF NEHF AND NEHL METHODS

In step 4, the NEHF heuristic builds $k + k(k+1)/2$ partial sequence, while the NEHL heuristic builds $k + (k-1)^2$ partial sequence. It is easy to see that the NEHL heuristic creates more partial sequence greater than the NEHF heuristic and both methods create some different partial sequences. The proposed heuristic improves upon the idea of combination of these two methods (NEHFL) by modification pertaining to step 4. Now, step 4 is executed as follows.

Step 4: Increment k, $k = k + 1$. Select the kth job from the sorted list and insert it into k possible positions of the best partial sequence. Select the best one with the minimal total flow time from these k jobs partial sequences. Next, place each job (except for the kth job of the sorted list) of the best partial sequence into its having k-1 positions and select the best partial sequence which makes the better total flow time. Swap jobs in positions i and j of the above partial sequence for all $i, j\ 1 \leq i < k,\ i < j \leq k$. Select the best partial sequence among $k(k-1)/2$ partial sequences which has minimal total flow time.

Step 4, creates $k + (k-1)^2 + k(k+1)/2$ partial sequence and requires computing total flow time. The complexity of the proposed heuristic is $O(n^4m)$. This is the same complexity as the two methods above.

In order to illustrate the proposed heuristic, let us consider the following 5-jobs, 5-machines problem instance:

	J1	***J2***	***J3***	***J4***	***J5***
M1	55	22	61	63	17
M2	51	24	4	71	33
M3	62	34	94	73	87
M4	77	57	22	55	82
M5	26	14	81	57	61

According to Step 1, the initial ordering of the jobs is *{*2–3–1–5–4*}*. Therefore, two partial sequence are *{2-3}* and *{3-2}*. The flow time of the first is 435, while the flow time of the second is 538. Therefore, *{*2–3*}* is the best partial sequence. For $K = 3$, the job to be tried is 1. The partial sequences are *{*1–2–3*}*, *{*2–1–3*}*, and *{*2–3–1*},* yielding partial flow times of 992, 831, and 789, respectively. Accordingly, *{*2–3–1*}* is selected. A general pairwise inserting is operated on *{*2–3–1*}*.The resulting partial sequences are *{3- 2- 1}, {3-1-2}, {3-2-1}* and *{2-1-3}*. The flow times of the partial sequences are 896, 971, 896 and 831, thus the best partial sequences is {2-3-1}. Since no one of these improves the result obtained by the seed sequence, *{*2–3–1*}* is retained as best partial sequence. The resulting of swapping are *{3-2-1}, {1-3-2}* and *{2-1-3}*, obtaining flow times of 896, 1015, and 831, respectively, thus the best is *{2–3–1}* . for $K = 4$, the following partial sequences are tried: *{5–2–3–1}*, *{2–5–3–1}*, *{2–3–5–1}*, and *{2–3–1–5}*. The flow times of the partial sequences are 1383, 1278, 1299, and 1270. Therefore, the selected partial sequence is *{2–3–1–5}*. The general inserting applied to the selected partial sequence gives the following sequences: *{3-1-2-5}, { 3-1-5-2}, {3-2-1-5}, {2-1-3-5}, {2-1-5-3}, {1-2-3-5}, {2-1-3-5}* and *{2-3-5-1}*. The flow times of these sequences are 1477, 1526, 1381, 1345, 1365, 1518, 1345 and 1299, respectively. Thus the best partial sequence is *{2–3–1–5}*. By swapping, it gives the partial sequence as follows: *{3–2–1–5}*, *{1–3–2–5}*, *{5–3–1–2}*, *{2–1–3–5}*, *{2–5–1–3}*, and *{2–3–5–1}*. The flow times of these sequences are 1381, 1541, 1478, 1345, 1246, and 1299, respectively. The partial sequence *{2–5–1–3}* is selected. In the same manner, for $K = 5$ the partial sequence *{2–5–4–1–3}* with a flow time of 1744 is reached.

5. COMPUTATIONAL EXPERIMENTS

Three methods - NEHF, NEHL and the proposed NEHFL – were run on 36 different problem sizes with $n = 6, 7, 8, 10, 20, 30, 40, 50, 60$ and $m = 5, 10, 15, 20$ and with generated random numbers, uniformly distributed $(low = 1, high = 100)$, as processing times. Thirty replications are conducted for each combination of machines and jobs. Hence, a total of 1080 problems were considered in this experiment.

In order to compare the performance of these heuristics, we consider two measures, namely, average relative percentage deviation $(ARPD)$ and percentage of optimal solutions. For a set of problems, we define $ARPD$ for a heuristic as follows:

$$ARPD = \frac{100}{30}\sum_{i=1}^{30}\left(\frac{Heuristic_i - Best_i}{Best_i}\right)$$

where Heuristic$_i$ is the objective function value obtained for the *i*th instance by a heuristic and Best$_i$ is the best solution taken as the best of the three generated by the heuristics.

It can be also seen from Table 1 that the proposed heuristic gives better results than those obtained by the NEHF and NEHL methods for all problem instances. The $ARPD$ values for the proposed heuristic range from 0.0000% to 0.2748%, while the number of times an optimal solutions is obtained ranges from 40% to 100%, whereas NEHF method has $ARPD$ value in the range from 0.0310-1.7433% and 0–77% for the number of times an optimal solutions is obtained and the $ARPD$ for the NEHL method ranges from 0.0094-0.7074% and the number of times the best solution is obtained varies from 17% to 93%. Clearly, the proposed NEHFLoutperforms the NEHF and the NEHL methods**.**

Next, we show statistical significance of the better results obtained by our method over those obtained by the NEHF and NEHL methods. The number of problem instances for each problem is taken as 30. Thus, each problem set comprises 30 pairs of total flow time values. For each problem set, the mean and the standard deviation of the 30 differences in total flow times are computed. We test the null hypothesis, $H_0: \mu = 0$. The *t*-statistic is computed as follows:

$$t = \sqrt{N}\frac{\bar{X} - \mu_0}{S}$$

Where N is the sample size, $\bar{X}$ is the sample mean, *S is* the sample standard, deviation $\mu_0 = 0$ and degrees of freedom = *N* – 1, determine $\text{critical region} = \alpha = 0.1$

If $p - value \leq \alpha$, that the proposed method better than both method is statistically significant.

The results are presented in Table 2 and Table 3. Note that the proposed method is statistically better than the method of NEHF method in 33 out of the 36 cases and better than the NEHL method in 24 out of the 36 cases.

6. CONCLUSION

In this paper, we have presented a composite NEHFL heuristic to improve the existing good heuristic NEHF and NEHL for minimizing the total flow time in permutation flow shops. The proposed method has the same complexity as NEHF and NEHL methods. Our modification significantly enhances the performance of those methods.

Table 1. Comparison of heuristics for 36 self uniform random generated problems.

n	m	No. of problems	NEHF		NEHL		NEHLF	
			ARPD	Percent best	ARPD	Percent best	ARPD	Percent best
6	5	30	0.3218	77	0.2831	87	0.0182	97
	10	30	0.1843	77	0.1196	87	0.0665	93
	15	30	0.0310	77	0.0094	87	0.0259	93
	20	30	0.2157	70	0.0598	93	0.0055	97
7	5	30	0.4125	53	0.2105	83	0.0000	100
	10	30	0.6115	50	0.0985	87	0.0358	90
	15	30	0.3526	63	0.1360	83	0.0531	83
	20	30	0.2603	63	0.0634	87	0.0095	93
8	5	30	0.4983	53	0.2559	87	0.0094	93
	10	30	0.3568	50	0.0765	87	0.0000	100
	15	30	0.5317	50	0.1409	83	0.0113	97
	20	30	0.4982	57	0.0178	93	0.0000	100
10	5	30	0.7503	37	0.2599	67	0.0659	80
	10	30	0.7234	20	0.1861	73	0.0130	97
	15	30	0.4902	37	0.2289	73	0.0478	90
	20	30	0.2048	67	0.1612	73	0.1431	73
20	5	30	1.3757	7	0.4867	30	0.1131	77
	10	30	1.3592	7	0.5057	37	0.1492	73
	15	30	1.3168	7	0.3778	43	0.0759	73
	20	30	0.9777	13	0.2415	43	0.1314	73
30	5	30	1.0734	17	0.6519	17	0.1447	67
	10	30	1.4060	7	0.6409	33	0.1567	67
	15	30	1.5877	0	0.4847	30	0.1665	73
	20	30	1.3825	17	0.3589	40	0.2215	50
40	5	30	1.5215	7	0.4765	17	0.0492	77
	10	30	1.5007	13	0.6779	20	0.2392	67
	15	30	1.3828	3	0.7074	20	0.1630	77
	20	30	1.4017	7	0.2893	53	0.2748	40
50	5	30	1.0089	10	0.6918	20	0.1801	70
	10	30	1.7433	0	0.6070	33	0.2015	67
	15	30	1.5707	3	0.3935	37	0.2435	60
	20	30	1.7284	0	0.3182	43	0.2510	57
60	5	30	0.8403	10	0.5218	23	0.2077	67
	10	30	1.4453	10	0.5018	27	0.2250	63
	15	30	1.3220	7	0.5108	47	0.2452	47
	20	30	1.4221	0	0.5178	20	0.1033	80

Table 2. Result of statistical tests (NEHL versus NEHFL).

n	m	No. of problems	Framinan and Leisten versus the proposed method			
			Total flow time difference		t	p-value
			Mean	Std. dev		
6	5	30	7.00	15.83	2.422	0.011
	10	30	4.90	28.25	0.950	0.175
	15	30	0.33	3.11	0.587	0.281
	20	30	15.33	39.85	2.108	0.022
7	5	30	11.53	22.06	2.864	0.004
	10	30	26.87	42.87	3.432	0.001
	15	30	20.17	47.19	2.341	0.013
	20	30	21.80	40.90	2.919	0.003
8	5	30	17.03	25.87	3.607	0.001
	10	30	20.83	34.09	3.347	0.001
	15	30	41.87	66.73	3.436	0.001
	20	30	51.03	80.00	3.494	0.001
10	5	30	31.13	45.90	3.715	0.000
	10	30	55.13	63.85	4.730	0.000
	15	30	47.10	78.88	3.271	0.001
	20	30	8.17	69.76	0.641	0.263
20	5	30	170.57	157.01	5.950	0.000
	10	30	252.37	218.54	6.325	0.000
	15	30	341.97	277.65	6.746	0.000
	20	30	282.97	297.75	5.205	0.000
30	5	30	260.67	249.29	5.727	0.000
	10	30	485.77	383.54	6.937	0.000
	15	30	694.27	569.72	6.675	0.000
	20	30	677.67	584.58	6.349	0.000
40	5	30	687.57	518.42	7.264	0.000
	10	30	781.60	724.91	5.906	0.000
	15	30	934.93	588.30	8.705	0.000
	20	30	1014.97	964.09	5.766	0.000
50	5	30	570.27	539.60	5.789	0.000
	10	30	1379.57	951.10	7.945	0.000
	15	30	1441.77	977.36	8.080	0.000
	20	30	1866.40	981.45	10.416	0.000
60	5	30	599.87	776.80	4.230	0.000
	10	30	1493.77	1309.21	6.249	0.000
	15	30	1571.07	1306.64	6.586	0.000
	20	30	2216.30	1392.95	8.715	0.000

Table 3. Result of statistical tests (NEHL versus NEHFL).

n	m	No. of problems	Dipak Laha and Subhash C. Sarin versus the proposed method			
			Total flow time difference		t	p-value
			Mean	Std. dev		
6	5	30	6.17	20.16	1.675	0.052
	10	30	2.07	13.11	0.864	0.197
	15	30	-0.73	6.72	-0.598	0.277
	20	30	4.10	21.34	1.052	0.151
7	5	30	5.67	19.41	1.599	0.060
	10	30	2.90	15.67	1.014	0.160
	15	30	5.90	28.32	1.141	0.132
	20	30	4.60	17.20	1.465	0.077
8	5	30	8.33	37.08	1.231	0.114
	10	30	4.47	16.22	1.509	0.071
	15	30	10.43	34.28	1.667	0.053
	20	30	1.80	6.97	1.414	0.084
10	5	30	8.23	23.38	1.929	0.032
	10	30	13.73	33.08	2.274	0.015
	15	30	20.27	71.45	1.554	0.066
	20	30	2.57	40.45	0.348	0.365
20	5	30	49.93	111.21	2.459	0.010
	10	30	74.20	121.17	3.354	0.001
	15	30	83.07	170.82	2.663	0.006
	20	30	34.37	160.43	1.173	0.125
30	5	30	141.73	189.21	4.103	0.000
	10	30	182.93	330.06	3.036	0.003
	15	30	155.23	396.80	2.143	0.020
	20	30	80.90	433.83	1.021	0.158
40	5	30	202.33	240.10	4.616	0.000
	10	30	268.13	518.26	2.834	0.004
	15	30	417.80	740.48	3.090	0.002
	20	30	14.03	689.03	0.112	0.456
50	5	30	350.50	580.29	3.308	0.001
	10	30	367.90	912.57	2.208	0.018
	15	30	170.50	868.43	1.075	0.146
	20	30	81.17	918.84	0.484	0.316
60	5	30	305.40	836.49	2.000	0.027
	10	30	352.17	1021.81	1.888	0.035
	15	30	389.43	1262.02	1.690	0.051
	20	30	696.57	1124.28	3.394	0.001

REFERENCES

[1] Garey MR, Johnson DS, Sethi R. *The complexity of flowshop and jobshop scheduling. Mathematics of Operations Research 1976*;1:117–29.

[2] Baker KR. *Introduction to sequencing and scheduling. New York: Wiley*; 1974.

[3] Taillard E. *Some efficient heuristic methods for the flow shop sequencing problem. European Journal of Op erational Research 1990*;47:65-74.

[4] Ruiz R, Maroto C. *A comprehensive review and evaluation of permutation flowshop heuristics. European Journal of Operational Research 2005*;165:479-94.

[5] Li X, Wang Q, Wu C. *Efficient composite heuristics for total flowtime minimization in permutation flow shops. Omega 2009*;*37*, in press.

[6] Allahverdi A, Aldowaisan T. *New heuristics to minimize total completion time in m-machine flow shop. International Journal of Production Economics 2002*;7:71–83.

[7] Woo HS, Yim DS. *A heuristic algorithm for mean flowtime objective in flowshop scheduling. Computers and Operations Research 1998;25(3)*:175-82.

[8] Framinan JM, Leisten R. *An efficient constructive heuristic for flowtime minimization in permutation flow shops. Omega 2003;31*:311–7.

[9] Dipak Laha, Subhash C. Sarin. *Aheuristic to minimize total flowtime in permutation flowshop. Omega 2009;37*: 734–739.