ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

OPTIMIZING THE TOTAL COMPLETION TIME IN PROCESS PLANNING USING THE RANDOM SIMULATION ALGORITHM

Baskar A. and Anthony Xavior M.
School of Mechanical and Building Sciences, VIT University, Vellore, India
E-Mail: a.baaskar@gmail.com

ABSTRACT

In a shop floor, minimization of makespan (Total Completion Time) has been an interesting area for many researchers for over six decades. The problem for the process planning engineer is to find a processing order of the 'n' jobs, the same for each machine, such that the make span is minimized, that is, the 'n' jobs are finished as soon as possible. In this paper, one attempt has been made to develop and use one 'Random Simulation Algorithm', with the objective of improving the makespan. Benchmark problems proposed by Taillard are used here for the validation purpose. These values have been compared with the makespans obtained from the original NEH algorithm and the 'NEF family' of algorithms proposed by the authors. For the 120 number of problem instances analyzed, the new algorithm reports better makespans, than the original NEH algorithm, in 114 cases. The ANOVA indicates that, the Random Simulation Algorithm performs slightly better.

Keywords: NEH heuristic, process planning, permutation flow shop scheduling, makespan, random simulation algorithm.

INTRODUCTION

Reduction in the total completion time of processing any job helps in the speedy realization of the product which is the main objective of any business. If the order is not to be changed, this can be referred as a permutation flow shop scheduling problem, in short, PFSP. For a two machines and 'n' jobs problem, Johnson's algorithm produces the optimum makespan. The problem is NP complete [1] for more than two machines, n jobs. In view of the complexity of these problems, most of the researches concentrate on the heuristic procedures to get near optimum solutions

After the advent of the computers and the increased computing capabilities over the years; many algorithms have been developed, coded in high level languages and run for computing the makespans, and the corresponding sequences. It has been generally accepted that the NEH heuristic algorithm [2], proposed by Nawaz *et al*. performs reasonably better among the simple heuristics for the general permutation flow shop scheduling problems.

The authors have already proposed a 'family of NEH heuristics' with different starting sequences for the minimization of the makespan. The performance has been analyzed using the well known benchmark problems proposed by Taillard [3] and Ruben Ruiz [4], and observed to be satisfactory.

In the proposed 'Random Simulation Algorithm', the initial partial sequence is constructed by randomly selecting two jobs. These two jobs will be initially scheduled for the optimum makespan, and then the remaining jobs will be inserted one by one in the partial sequence. At each step, the new job will be inserted at a place, which minimizes the partial makespan among the possible ones. Codes were generated in MATLAB and run in an i5 PC with 4 GB RAM. For each problem instance, ten trials have been made and the least makespan obtained has been selected. These values have been compared with the makespans obtained from the original NEH algorithm and the 'NEF family' of algorithms.

The proposed new algorithm produces better makespans in most of the cases. For the 120 number of problem instances analyzed, the new algorithm reports better makespans, than the original NEH algorithm, in 114 cases. Also, the average number of sequences per problem instance, having makespans better than the original NEH, is 4.2333 (a maximum of 10 is possible, as 10 trials have been made per simulation). The ANOVA indicates that, all the three algorithms considered have almost the same values of the standard deviations; the Random Simulation Algorithm performs slightly better. In the analysis, the F value is small (0.57) and the P value is > 0.05 (0.566) and hence, the Null Hypothesis (all means are the same) is accepted.

BRIEF LITERATURE REVIEW

Most of the researches in this field of flow shop scheduling, are directed towards minimizing the makespan. The difficulties in obtaining the exact solutions, especially for larger problems, influenced the researchers to go for heuristics and meta heuristics. Starting from Johnson for two machines and 'n' jobs [5], many heuristic and meta heuristic algorithms have been proposed over the years. Out of the several heuristic approaches developed during earlier periods, those proposed by Palmer [6], Campbell *et al*. [7], Gupta [8], Dannenbring [9] are note worthy. These early constructive heuristic algorithms have been inspirations for the researchers to develop many other well-known scheduling techniques. Most of the studies, like by Terner and Booth [10] and Rad *et al*. [11] conclude that, the NEH heuristic algorithm developed by Nawaz, Enscore and Ham is the most efficient so far, among the existing simple heuristics. Framinan *et al*. [12] considered twenty two different approaches for the indicator value and eight different sorting criteria, totaling 176 approaches for every objective function. Additionally, for every objective function, the RANDOM choice of a sequence was considered. It was concluded that, for the makespan criterion, the job insertion technique and ordering the jobs by non increasing sums of processing times on the machines, used by the NEH algorithm, outperform others. Many meta heuristics use NEH algorithm for obtaining the seed solutions. A simple probabilistic methodology for solving the Permutation Flowshop Sequencing Problem has been presented by Juan *et al*. [13]. Monte Carlo Simulation was used with 15 iterations, for each problem instance.

RANDOM SIMULATION ALGORITHM

The proposed algorithm is an extension of the well known NEH heuristic algorithm. Hence, it will be appropriate to review the NEH procedure first before describing the algorithm. NEH Algorithm for the make span minimization can be stated as follows:

Step-1: Ordering the jobs by non increasing sums of processing times on the machines

Step-2: Taking the first two jobs and scheduling them in order to minimize the partial make span as if there were only these two jobs

Step-3: For k= 3 to n, Step 4 to be repeated

Step-4: Inserting the k th job at the place, which minimizes the partial make span among the possible ones.
Total No. of sequences to be enumerated = n(n+1)/2 – 1.

The authors [14] have also proposed a family of NEH heuristics, NEH1, NEH2 and NEH3. The brief description is as follows:

All the jobs are ordered in the decreasing order of total processing times, the initial sequences vary.

In the original NEH algorithm, first two jobs are considered as the initial sequence; remaining jobs are inserted one by one, starting form the third job.

In NEH1, the first and the last jobs are taken as the initial sequence; remaining jobs are inserted one by one, starting form the second job.

Similarly, in the third case of NEH2 algorithm, the middle two jobs are taken as the initial sequence; remaining jobs are inserted one by one starting form the first job.

Finally, for the NEH2 algorithm, the last two jobs are taken as the initial sequence; remaining jobs are inserted one by one starting form the first job.
Now, the new algorithm can be stated as follows:

Step-1: for a given permutation flow shop scheduling problem, the minimum makespan and the corresponding sequence are computed, using the NEH Algorithm. This is the ‘NEH Makespan’ and the sequence is the ‘Reference Sequence’.

Step-2: for the same problem, the makespans are computed separately using the NEH family of heuristics. The minimum makespan and the corresponding sequence among these three are selected. This is the ‘NEH family best Makespan’.

Step-3: now, two jobs are selected randomly as the initial partial sequence. The optimal partial sequence for these two jobs is computed using the Johnson’s Algorithm

Step-4: from the reference sequence, other jobs are inserted one by one from the beginning, at the place, which minimizes the partial makespan among the possible ones. The makespan and the sequence are computed.

Step-5: steps 3 and 4 are repeated for the required number of trials; the makespan and the corresponding sequence are computed, in each trial. Separate counter is provided for counting the number of trials yielding makespans better than that of the NEH makespan. The authors have taken 10 trials for each problem instance.

Step-6: the minimum value among the makespans obtained from the trials is found. This is the ‘Random Best Makespan’.

Step-7: The lower bounds are computed. The makespans are statistically analyzed for any set of Benchmark problems.

COMPUTATIONAL RESULTS AND DISCUSSION

The problem instances proposed by Taillard are used for the validation purpose. Taillard benchmark is composed of 12 groups of 10 instances each, totaling 120 instances. Each group is characterized by a combination of jobs and machines (n×m). The groups are {20, 50, 100} × {5, 10, 20}, 200× {10, 20}, and 500× 20. The processing times are randomly generated between 1 and 99 time units. The seeds proposed by Taillard are used for this. For each problem instance, using the Random Simulation, ten trials are made for obtaining a minimum makespan. In addition, the NEH makespan and the NEH family best makespan are also computed. The results are tabulated in Tables 1 to 3. Table-1 lists the output for the first forty Taillard problems. The problems size varies from 5 machines, 20 jobs to 5 machines, 50 jobs. Subsequently, Tables 2 and 3 report the results for the remaining 80 problems. Lower bounds are directly taken from the Taillard work. It can be seen that the number of makespans better than the NEH makespan is not consistent within the same group. Also, due to the random nature of the algorithm, the values and count vary each time we repeat the simulation and change the number of trials.

Out of the 120 problems, the Random Simulation could report better makespans than the NEH in 99 problem instances and equals the NEH makespans in 15 problems. Only in case of 6 problems, the makespans are worse than the NEH. The average number of sequences per problem instance, having the makespans better than the original NEH, is 4.2333. (10 trials were conducted per simulation. Therefore, a maximum of 10 is possible).

In addition, ten simulations for the problem instances, 20 m/c 50 jobs and 20 m/c 100 jobs, each simulation with 10 trials each, have been carried out. The results are tabulated in Tables 4 and 5. The average number of sequences per problem instance, having makespans better than the original NEH, are 5.4 (6.4) and 4.19 (4.5) respectively. In brackets, the values obtained in a single simulation are indicated. It may be noted that, as

ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

the number of simulations and trials increases, the possibility of getting better makespans also increases. In the case of 20 m/c 50 jobs problem set, out of the 10 problems, the makespans improved in 8 and in the other case of 20 m/c 100 jobs, the improvement reported in all the 10 problems.

The authors repeated the simulation and changed the number of trials per simulation for a few more times and the average number of sequences per problem instance, having makespans better than the original NEH do not change much. The rate of improvements also reduces significantly with the number of simulations and trials and stops at a particular moment. That marks the saturation point of the power of NEH's insertion technique and selection of the initial partial sequence.

ANOVA

One way ANOVA has been carried out, considering the relative deviations of makespans from the Lower Bound for the three; NEH, NEH family best, and the Random Simulation Algorithms. MINITAB16 software is used for this analysis. Algorithm 0 represents the original NEH, 1 represents the NEH family best and 2 represent the Random best algorithms. The analysis was carried out at 95% confidence level. Figure-1 indicates the box plot for the relative deviations of the three algorithms considered. It clearly indicates that the simulation based algorithm has less deviation band than the others.

Relative Deviation, RD =

$$\frac{(Makespan - LowerBound)X100}{LowerBound}$$

Test for equal variances: RD versus algorithm

```
95% Bonferroni confidence intervals for
standard deviations

Algorithm N   Lower    StDev      Upper
  0       120 6.23691 7.21068  8.51988
  1       120 5.96253 6.89347  8.14507
  2       120 5.85282 6.76662  7.99519

Bartlett's Test (Normal Distribution)
Test statistic = 0.51, p-value = 0.775
Levene's Test (Any Continuous Distribution)
Test statistic = 0.25, p-value = 0.781
```

One-way ANOVA: RD versus algorithm

```
Source     DF      SS    MS    F     P
Algorithm  2     55.2 27.6 0.57 0.566
Error      357  17290.8  48.4
Total      359  17346.0

S = 6.959 R-Sq= 0.32% R-Sq(adj)= 0.00%

Individual 95% CIs for Mean Based on
Pooled StDev
---+---------+---------+---------+------
         (-----------*-----------)
   (------------*-----------)
(------------*-----------)
---+---------+---------+---------+------
  6.0       7.0       8.0       9.0

Pooled StDev = 6.959
Level  N    Mean   StDev
0      120  7.906  7.211
1      120  7.292  6.893
2      120  6.961  6.767
```

Figures 2 to 4 show the Probability Plot, Interval Plot and the Test for Equal Variances, respectively for the Relative Deviations. From the probability plot, it can be seen that about 50% of the solution makespans have the relative deviations less than 5% from the lower bound. The deviations span also gets lowered for the Random Algorithm, whereas, the span is at its highest for the NEH algorithm. The mean and standard deviations for the random best algorithm clearly outperforms the other algorithms, but at the cost of computation time.

It may be recalled that, the null hypothesis in ANOVA is that, the means of all the groups are the same. The alternative is that, at least one is different. So, for our analysis with the three algorithms:

H_0: all means are same
H_A: at least one mean is different

It can be seen that the F value is small (0.57) and the P value is > 0.05 (0.566) and hence, the Null Hypothesis is accepted.

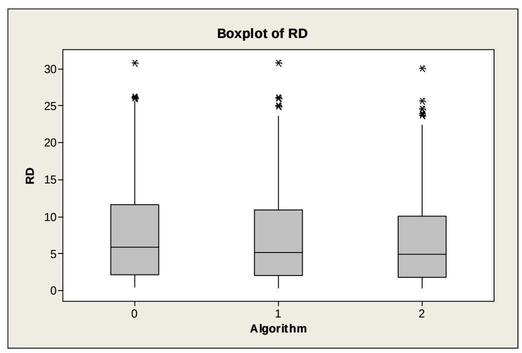

Figure-1. Box plot for the relative deviations.

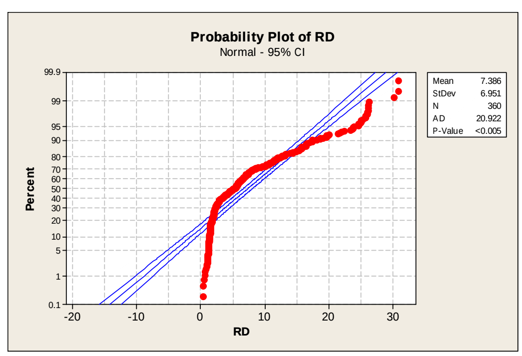

Figure-2. Probability plot for the relative deviations.

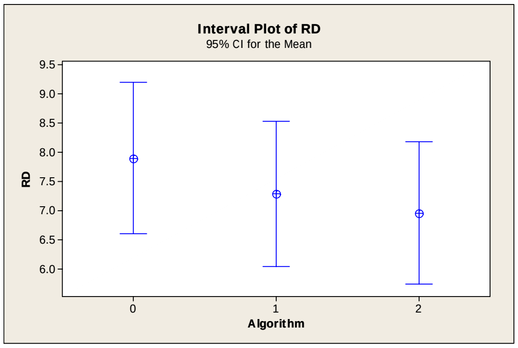

Figure-3. Interval plot for the algorithms.

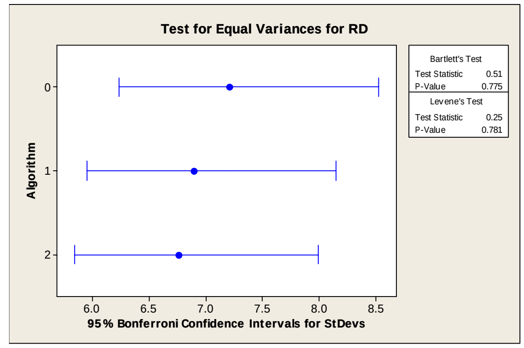

Figure-4. Test for equal variances.

Table-1. Results of simulation for the first forty problems.

Problem instance	Lower bound	NEH Makespan	NEH family best Makespan	Random best Makespan	Count- better than NEH (10max)
5 machines	1232	1286	1286	1286	0
20 jobs	1290	1365	1365	1365	0
	1073	1159	1132	1100	10
	1268	1325	1309	1304	4
	1198	1305	1244	1244	2
	1180	1228	1228	1210	5
	1226	1278	1269	1251	9
	1170	1223	1223	1221	1
	1206	1291	1270	1260	7
	1082	1151	1122	1127	6
10 machines	1448	1680	1631	1626	8
20 jobs	1479	1729	1707	1719	2
	1407	1557	1557	1546	3
	1308	1439	1404	1399	4
	1325	1502	1490	1448	5
	1290	1453	1444	1434	3
	1388	1562	1544	1518	7
	1363	1609	1589	1587	5
	1472	1647	1638	1638	3
	1356	1653	1646	1628	5
20 machines	1911	2410	2357	2379	8
20 jobs	1711	2150	2114	2150	0
	1844	2411	2411	2398	7
	1810	2262	2262	2240	2
	1899	2397	2371	2366	3
	1875	2349	2281	2294	8
	1875	2362	2362	2323	3
	1880	2249	2249	2249	0
	1840	2320	2320	2276	5
	1900	2277	2253	2231	3
5 machines	2712	2733	2731	2731	2
50 jobs	2808	2843	2843	2843	0
	2596	2640	2625	2625	8
	2740	2782	2782	2781	1
	2837	2868	2864	2868	0
	2793	2850	2835	2835	7
	2689	2758	2741	2735	7
	2667	2721	2719	2695	9
	2527	2576	2563	2565	6
	2776	2790	2786	2786	1

VOL. 9, NO. 4, APRIL 2014 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

Table-2. Results of simulation for the second forty problems.

Problem instance	**Lower bound**	**NEH Makespan**	**NEH family best Makespan**	**Random best Makespan**	**Count- better than NEH** (10max)
10 machines	2907	3135	3133	3112	2
50 jobs	2821	3032	3032	2986	5
	2801	2986	2986	2986	0
	2968	3198	3198	3140	9
	2908	3160	3126	3098	5
	2941	3178	3139	3148	4
	3062	3277	3224	3231	3
	2959	3123	3123	3118	1
	2795	3002	3002	3002	0
	3046	3257	3236	3198	8
20 machines	3480	4082	4053	4025	8
50 jobs	3424	3921	3914	3871	7
	3351	3927	3872	3849	4
	3336	3969	3916	3886	8
	3313	3835	3835	3805	2
	3460	3914	3856	3860	9
	3427	3952	3905	3881	6
	3383	3938	3938	3915	5
	3457	3952	3916	3914	5
	3438	4079	3969	3965	10
5 machines	5437	5519	5519	5514	3
100 jobs	5208	5348	5284	5284	9
	5130	5219	5206	5204	7
	4963	5023	5023	5023	0
	5195	5266	5261	5261	4
	5063	5139	5139	5139	0
	5198	5259	5257	5255	1
	5038	5120	5105	5105	6
	5385	5489	5489	5487	1
	5272	5341	5341	5341	0
10 machines	5759	5846	5846	5812	1
100 jobs	5345	5453	5443	5427	5
	5623	5824	5764	5756	10
	5732	5929	5929	5912	2
	5431	5679	5613	5593	8
	5246	5375	5360	5354	4
	5523	5704	5681	5685	1
	5556	5760	5760	5754	2
	5779	6032	5988	6012	6
	5830	5918	5903	5902	8

VOL. 9, NO. 4, APRIL 2014 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

Table-3. Results of simulation for the last forty problems.

Problem instance	**Lower bound**	**NEH Makespan**	**NEH family best Makespan**	**Random best Makespan**	**Count- better than NEH** (10max)
20 machines	5851	6541	6541	6541	0
100 jobs	6099	6523	6523	6473	6
	6099	6639	6594	6550	8
	6072	6557	6518	6542	3
	6009	6695	6623	6623	10
	6144	6664	6664	6656	1
	5991	6632	6609	6569	6
	6084	6739	6739	6752	0
	5979	6677	6630	6587	10
	6298	6677	6677	6665	1
10 machines	10816	10942	10941	10942	0
200 jobs	10422	10716	10660	10644	7
	10886	11025	11025	11025	0
	10794	11057	11057	11057	0
	10437	10645	10645	10618	1
	10255	10458	10458	10437	3
	10761	10989	10989	10938	7
	10663	10829	10829	10785	3
	10348	10574	10558	10535	5
	10616	10807	10758	10772	5
20 machines	10979	11625	11594	11587	6
200 jobs	10947	11675	11675	11690	0
	11150	11852	11761	11785	7
	11127	11803	11766	11730	7
	11132	11685	11670	11666	4
	11085	11629	11582	11636	0
	11194	11833	11754	11772	5
	11126	11913	11825	11754	9
	10965	11673	11641	11628	4
	11122	11869	11743	11748	8
20 machines	25922	26670	26670	26671	0
500 jobs	26353	27232	27162	27119	7
	26320	26848	26848	26923	0
	26424	27055	27010	26861	9
	26181	26727	26727	26766	0
	26401	26992	26986	26969	1
	26300	26797	26748	26759	3
	26429	27138	27138	27015	5
	25891	26631	26580	26495	5
	26315	26984	26952	26918	4

Table-4. Ten simulations for the problem instances, 20 m/c 50 jobs.

Problem no.	Lower bound	NEH Makespan	Random best Makespan (one simulation)	Random best Makespan (10 simulations)	Avg. count- better than NEH, 10max.*
1	3480	4082	4025	4017	6.7(8)
2	3424	3921	3871	3869	3(7)
3	3351	3927	3849	3835	7.7(4)
4	3336	3969	3886	3895	6.5(8)
5	3313	3835	3805	3782	1.5(2)
6	3460	3914	3860	3824	6(9)
7	3427	3952	3881	3893	5.2(6)
8	3383	3938	3915	3888	4.1(5)
9	3457	3952	3914	3891	3.7(5)
10	3438	4079	3965	3942	9.6(10)

*In brackets, the values obtained in a single simulation
Avg. 5.4 (6.4)*

Table-5. Ten simulations for the problem instances, 20 m/c 100 jobs.

Problem no.	Lower bound	NEH Makespan	Random best Makespan (one simulation)	Random best Makespan (10 simulations)	Avg. count- better than NEH, 10max*
1	5851	6541	6541	6467	1.3(0)
2	6099	6523	6473	6421	6.1(6)
3	6099	6639	6550	6537	7.4(8)
4	6072	6557	6542	6504	1.3(3)
5	6009	6695	6623	6570	9(10)
6	6144	6664	6656	6631	0.6(1)
7	5991	6632	6569	6543	6.4(6)
8	6084	6739	6752	6704	0.3(0)
9	5979	6677	6587	6563	8.7(10)
10	6298	6677	6665	6641	0.8(1)

*In brackets, the values obtained in a single simulation
Avg. 4.19 (4.5)*

CONCLUSIONS

The proposed new Random Simulation Algorithm produces better makespans in most of the cases. For the 120 number of Taillard problem instances analyzed, the new algorithm reports better makespans in 99 cases, same makespan in 15 cases when compared with the original NEH algorithm. Also, the average number of sequences per problem instance, having makespans better than the original NEH, is 4.2333 (a maximum of 10 is possible, as 10 trials have been made per simulation).

The one way ANOVA indicates that, all the three algorithms considered have almost the same values of the standard deviations; the Random Simulation Algorithm performs slightly better. In the analysis, the F value is small and the P value is > 0.05 and hence, the Null Hypothesis (all means are the same) is accepted. It is proposed that the output of the simulation algorithm can be used as a candidate solution to refine the solution further using metaheuristics.

REFERENCES

[1] A.H.G. Rinnooy Kan. 1976. Machine Scheduling Problems: Classification, Complexity, and Computations: The Hague: Martinus Nijhoff.

[2] M.Nawaz, E.E. Enscore Jr. and I. Ham. 1983. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. OMEGA, The International Journal of Management Science. 11 (1): 91–95.

[3] E. Taillard. 1993. Benchmarks for basic scheduling problems. European Journal of Operational Research. 64: 278–285.

[4] Problem instances available in: http://soa.iti.es

[5] S.M. Johnson. 1954. Optimal two- and three-stage production schedules with setup times included. Naval Research Logistics Quarterly. 1: 61-68.

[6] D.S. Palmer. 1965. Sequencing jobs through a multistage process in the minimum total time: a quick method of obtaining a near optimum. Operations Research Quart. 16: 101–107.

[7] H.G. Campbell, R.A. Dudek and M.L. Smith. 1970. A heuristic algorithm of the n-job, m-machine sequencing problem. Management Science. 16: B630–B637.

[8] J.N.D.Gupta. 1971. A functional heuristic algorithm for the flow-shop scheduling problem. Operations Research. 22: 39-47.

[9] D.G. Dannenbring. 1977. An evaluation of flow shop sequencing heuristics. Management Science. 23: 1174-1182.

[10] S. Turner and D. Booth. 1987. Comparison of heuristics for flow shop sequencing. OMEGA, The International Journal of Management Science. 15(1): 75-78.

[11] S.F. Rad, R. Ruiz and N. Boroojerdian. 2009. New high performing heuristics for minimizing makespan in permutation flow shops. OMEGA, the International Journal of Management Science. 37: 331-345.

[12] J.M. Framinan, R.Leisten and C. Rajendran. 2003. 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. 41(1): 121-148.

[13] A.A. Juan, A. Guix, F. Adelantado, P. Fonseca i Casas and R.Ruiz. 2010. Simulation for Alternative Solutions to the Flowshop Sequencing Problem. Integrationsaspekte der Simulation, Technik, Organisation und Personal, Gert Zülch and Patricia Stock (Hrsg.), Karlsruhe, KIT Scientific Publishing. pp. 349-356.

[14] A. Baskar and M. Anthony Xavior. 2013. Optimization of Makespan in Flow Shop Scheduling Problems using Combinational NEH family of Heuristics - An Analysis. International Journal of Applied Engineering Research. 8(10): 1205-1216.