IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

An Improved NEH Algorithm Applied to Permutation Flow Shop Scheduling

Ekta Singhal[*1], Dr Naveen Hemrajani[2]
Suresh Gyan Vihar University, Jaipur, India
ektasinghal36@gmail.com

Abstract

Flowshop Scheduling is used to determine the optimal sequence of n jobs to be processed on m machines in the same order.The permutation flowshop represents a particular case of the flowshop scheduling problem having as goal the deployment of an optimal schedule for N jobs on M machines. Solving the flowshop problem consists in scheduling n jobs (i= 1…..n) on m machines (j=1….m). A job consists in m operations and the jth operation of each job must be processed on machine j. So, one job can start on machine j if it is completed on machine j-1 and if machine j is free. Each operation has a known processing time pij. For the permutation flowshop the operating sequences of the jobs are the same on every machine. If one job is at the ith position on machine 1, then this job will be at the ith position on all the machines. Such problems are NP-Complete and hence optimal solutions are not guaranteed but heuristics have been shown to produce good working solutions.

NEH (Nawaz, Enscore, Ham) Algorithm is an efficient algorithm that works by minimizing the makespan for Permutation flowshop Scheduling Problems PFSP. The proposed algorithm is obtained by modifying the NEH algorithm and produces improved quality solutions (i.e. makespan) with algorithmic complexity same as the original algorithm.

Keywords: Flow shop Scheduling, makespan, heuristics,PFSP, NP-Complete.

Introduction

A mobile ad-hoc network (MANET) is a network A two-stage parallel flow shops scheduling problem having the following characteristics:

1. All jobs are available for machine processing simultaneously at time zero.
2. Two parallel flow shops are available, each of which consists of two stages
3. Two proportional machines are available at each stage.
4. Each job has to be processed in exactly one machine in each stage starting with stage 1 and ending with stage 2.
5. Switching jobs between flow shops is not allowed.
6. The jobs can be processed in either the slower flow shop (flow shop 1) or in the faster flow shop (flow shop 2).
7. The processing time in flow shop 1 is a multiple of the processing time in flow shop

Problems such as the one described above can be formulated as a two-stage parallel flow shops[6] with proportional processing times. Figure 1.1 illustrates a schematic diagram of this scheduling situation with two stages.

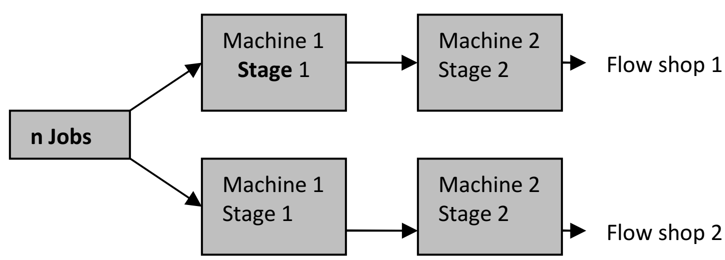


Figure 1.1- Two-stage parallel flow shops with proportional processing times

At each stage j, for a job, there are preferred machines and non-preferred machines. A preferred machine processes a job faster than a non-preferred machine. The objective is to minimize the makespan[10].

Problem Definition

The permutation flow shop problem requires to find the order in which n jobs are to be processed on m consecutive machines. The jobs are processed in the order machine 1, machine 2, . . . machine m. Machines can only process one job at a time and jobs can be processed by only one machine at a time without preemption.

No job can jump over any other job, meaning that the order in which jobs are processed in machine 1 is maintained throughout the system[15]. Moreover, no machine is allowed to remain idle when a job is ready for processing. All jobs and machines are available at time 0.

Objective

For each job two parameters are computed:
tp (i, j)→ processing time of job j on machine i
tc (i, j)→ completion time of job j on machine i
The completion time of all jobs is can be computed as:
tc (M1, J1) = tp (M1, J1)
tc (Mi, J1) = tc (Mi-1, J1) + tp (Mi, J1)
tc (M1, Jj) = tc (M1, Jj-1) + tp (M1, Jj)
tc (Mi, Jj) = max {tc (Mi-1, Jj), tc (Mi, Jj-1)} + tp (Mi, Jj)
The objective is to find an n-job sequence so as to minimize the makespan i.e. tc (Mm, Jn).

Genetic Algorithm For NEH

(Nawaz Enscore Ham) APPLIED TO FLOW SHOP SCHEDULING
It is a constructive heuristic.
Step 1: Sort the *n* jobs in non-increasing order of their total processing times

Step 2: Take the first two jobs and schedule them in order to minimise the partial makespan as if there were only these two jobs
Step 3: For *k*= 3 to *n* do Step 4
Step 4: Insert the *k*th job at the place, which minimises the partial makespan among the *k* possible ones.

Proposed Methodology for Improved NEH Algorithm Applied to Flow Shop Scheduling Improved Heuristic

Step 1: Sort the *n* jobs in non-increasing order of their total processing times
***Step 2*:** Take the first four jobs from the sorted list and form 4! = 24 partial sequences (each of length 4). The best *k* (*k* is a parameter of the algorithm) out of these 24 partial sequences are selected for further processing. The relative positions of jobs in any partial sequence is not altered in any later (larger) sequence
***Step 3*:** Set *z* = 5
***Step 4*:** The *z*th job on the sorted list is inserted at each of the *z* positions in each of the *k* (*z* − 1)-job partial sequences, resulting in (*z* × *k*) *z*-job partial sequences
***Step 5*:** The best *k* out of the *z* × *k* sequences are selected for further processing
***Step 6*:** Increment *z* by 1
***Step 7*:** If *z* > *n*, accept the best of the *k* *n*-job sequences as the final solution and stop.
Otherwise go to step 4.

Comparision (Example)

Ex:

Machines → Jobs ↓	M 1	M2	M3	M4	M5
J1	5	9	8	10	1
J2	9	3	10	1	8
J3	9	4	5	8	6
J4	4	8	8	7	2
J5	3	5	6	3	7

Table 5.1-Comparison of NEH algorithm and Improved Heuristic

Total processing times of jobs
Job 1= 5+9+8+10+1= 33
Job 2= 9+3+10+1+8= 31
Job 3= 9+4+5+8+6= 32
Job 4= 4+8+8+7+2= 29
Job 5= 3+5+6+3+7= 24
Sorting in non-increasing order of total processing times
J1, J3, J2, J4, J5
NEH Algorithm

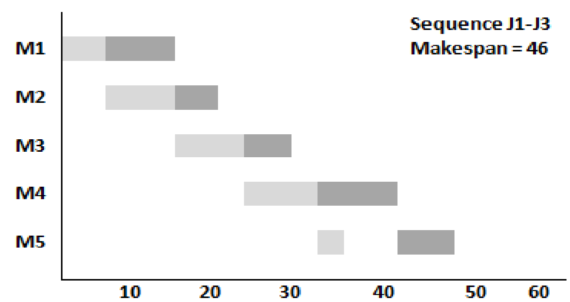


Fig: 5.1

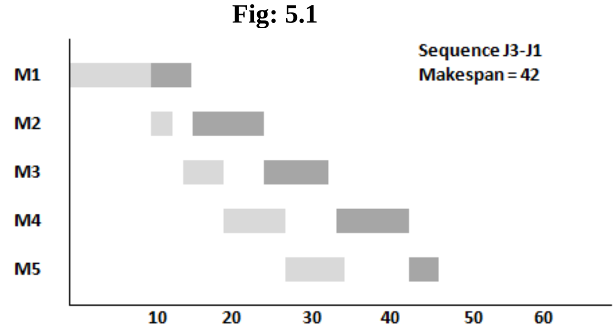


Fig: 5.2

Sequence: J1-J3 Makespan: 46
Sequence: J3-J1 Makespan: 42
Select sequence J3-J1

 ISSN: 2277-9655

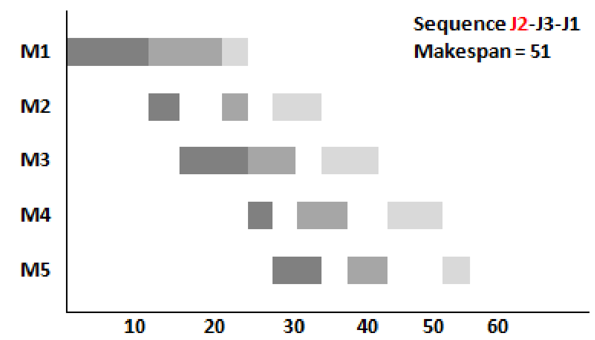


Fig: 5.3

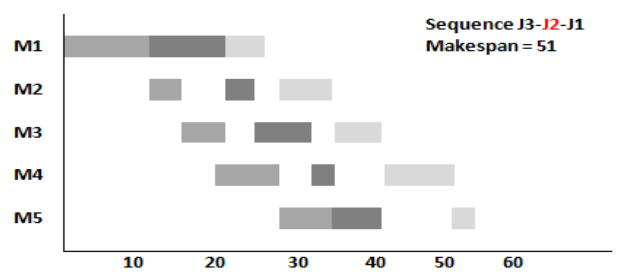


Fig: 5.4

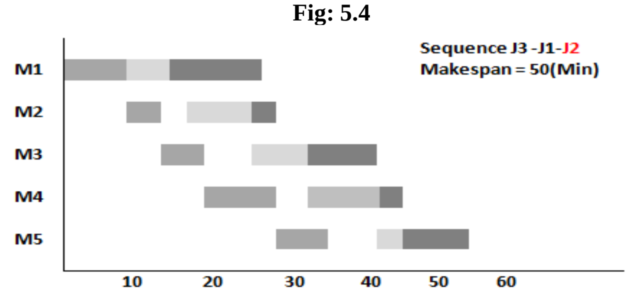


Fig: 5.5

Sequence: **J2**-J3-J1
Makespan: 51
Sequence: J3-**J2**-J1
Makespan: 51
Sequence: J3-J1-**J2**
Makespan: 50

Select sequence J3-J1-J2

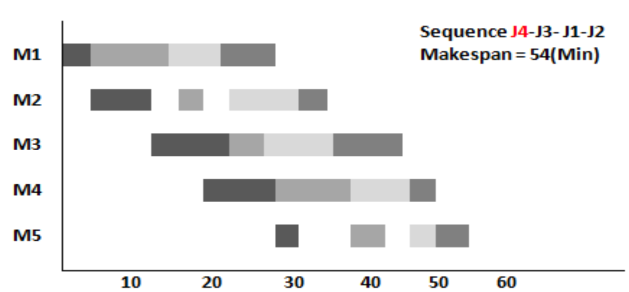


Fig: 5.6

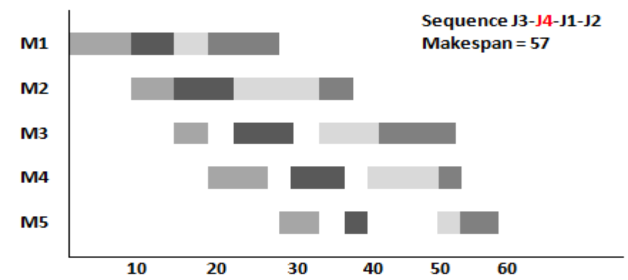


Fig: 5.7

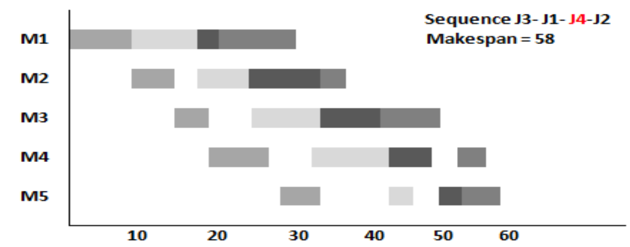


Fig: 5.8

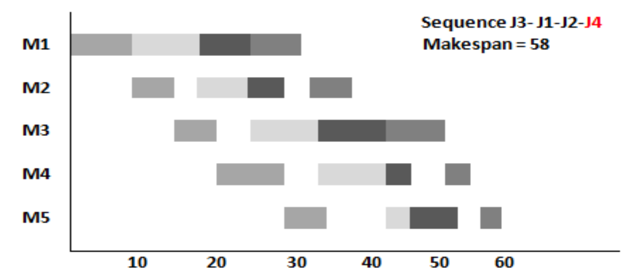


Fig: 5.9

Sequence: **J4**-J3-J1-J2
Makespan: 54
Sequence: J3-**J4**-J1-J2
Makespan: 57
Sequence: J3-J1-**J4**-J2
Makespan: 58
Sequence: J3-J1-J2-**J4**
Makespan: 58

Select sequence J4-J3-J1-J2

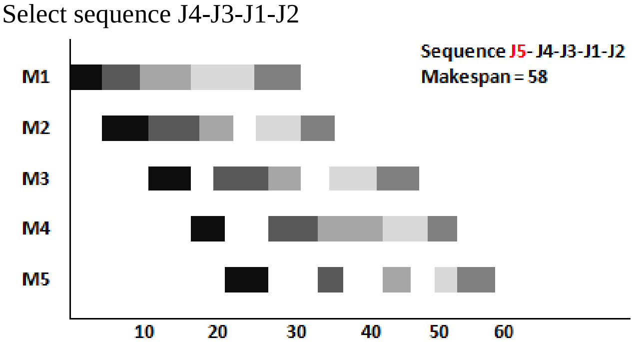


Fig: 5.10

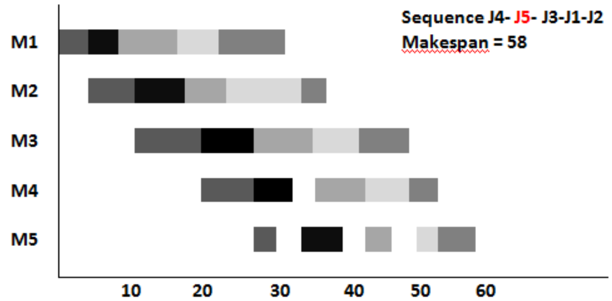


Fig: 5.11

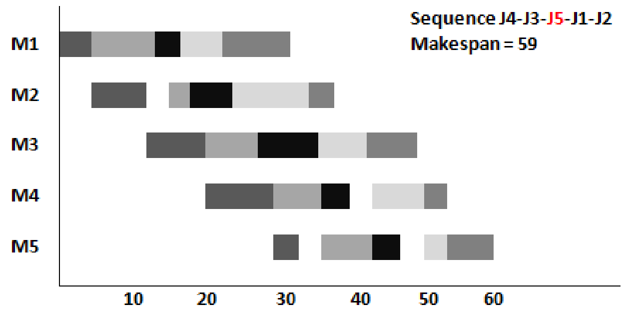


Fig: 5.12

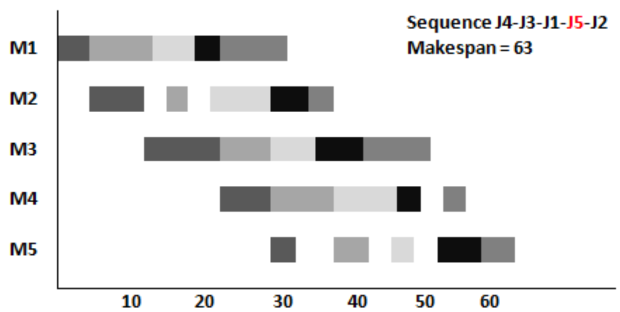


Fig: 5.13

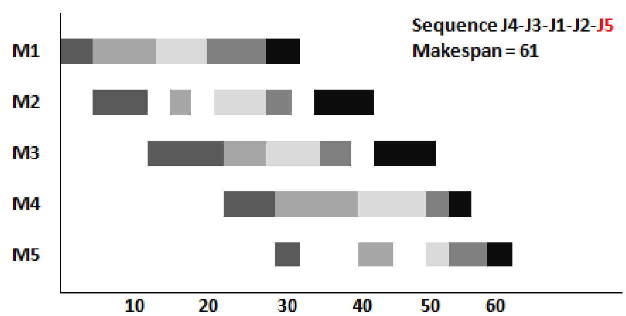
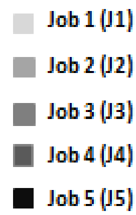


Fig: 5.14



Sequence: **J5**-J4-J3-J1-J2
Makespan: 58
Sequence: J4-**J5**-J3-J1-J2
Makespan: 58
Sequence: J4-J3-**J5**-J1-J2
Makespan: 59
Sequence: J4-J3-J1-**J5**-J2
Makespan: 63
Sequence: J4-J3-J1-J2-**J5**
Makespan: 61

Thus, the best sequence is **J5-J4-J3-J2-J1** and **J4-J5-J3-J1-J2** with makespan of 58.

Improved Heuristic

Taking first four jobs from the sorted order to form 24 partial sequences.

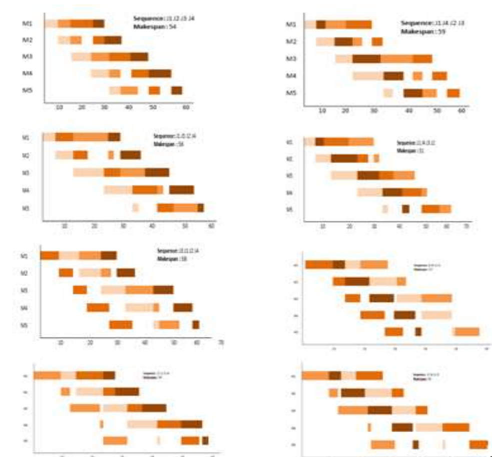


Fig: 6.1

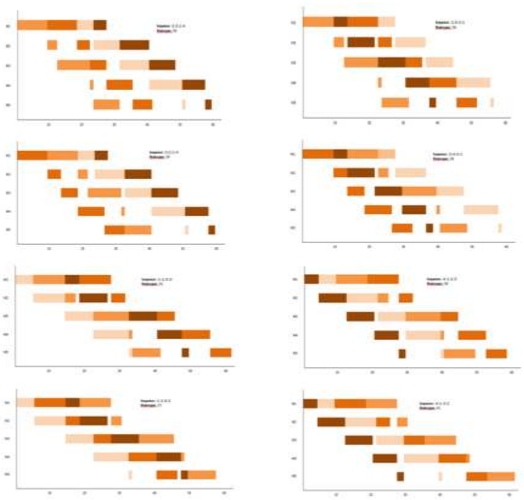


Fig: 6.2

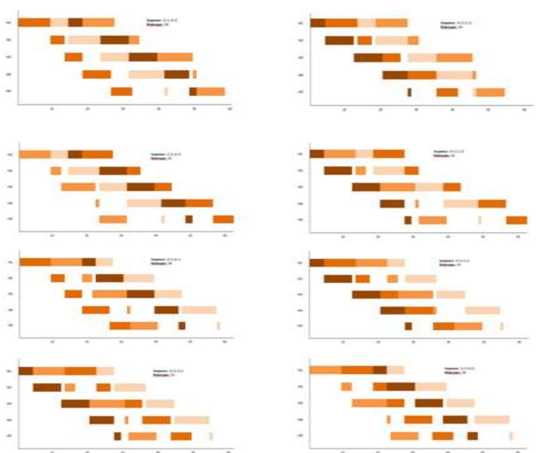


Fig: 6.3

Sequence: J1-J2-J3-J4
Makespan: 54
Sequence: J1-J4-J2-J3
Makespan: 59
Sequence: J1-J3-J2-J4
Makespan: 56
Sequence: J1-J4-J3-J2
Makespan: 61
Sequence: J3-J1-J2-J4
Makespan: 58
Sequence: J3-J4-J1-J2
Makespan: 57
Sequence: J2-J1-J3-J4
Makespan: 58
Sequence: J2-J4-J1-J3
Makespan: 62
Sequence: J2-J3-J1-J4
Makespan: 59
Sequence: J2-J4-J3-J1
Makespan: 56

Similar method will be applied for rest of the sequences.
The parameter of the algorithm k is taken as 8.

Selecting the best 8 sequences for further processing.

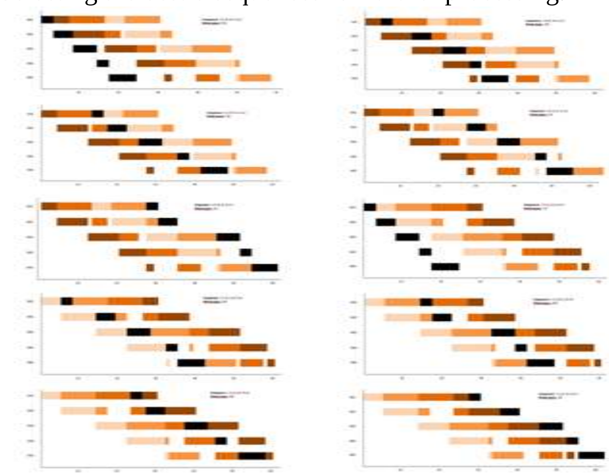


Fig: 6.4

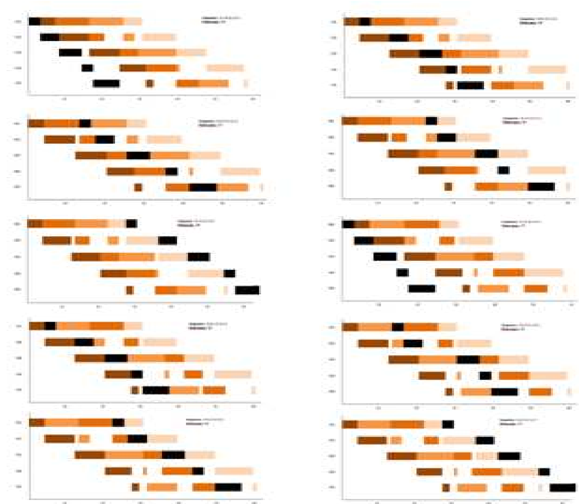


Fig: 6.5

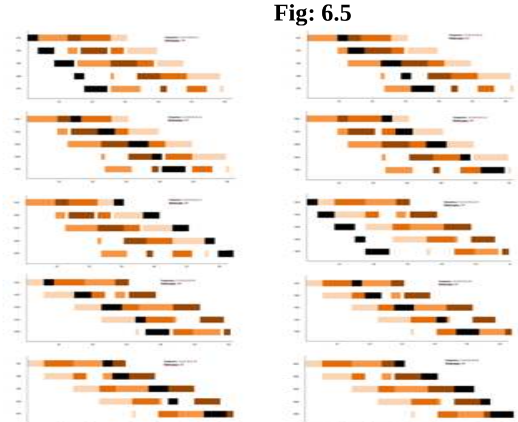


Fig: 6.6

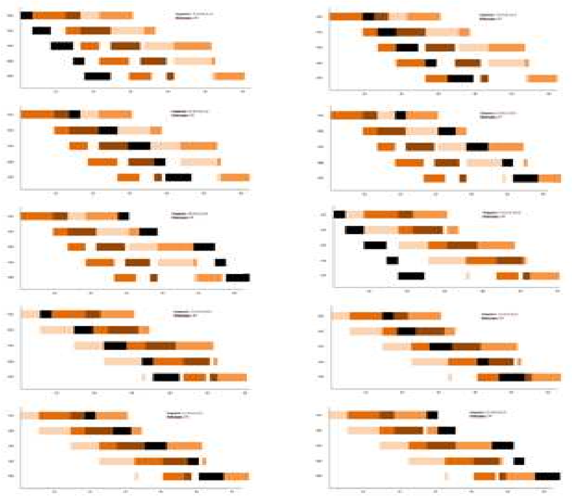


Fig: 6.7

J4-J2-J3-J1

Sequence: J5-J4-J2-J3-J1
Makespan: 57
Sequence: J4-**J5**-J2-J3-J1
Makespan: 60
Sequence: J4-J2-**J5**-J3-J1
Makespan: 60
Sequence: J4-J2-J3-**J5**-J1
Makespan: 60
Sequence: J4-J2-J3-J1-**J5**
Makespan: 63

Similar method will be applied for Sequences (**J4-J3-J1-J2), (J4-J3-J2-J1), (J2-J4-J3-J1),(J1-J2-J3-J4), (J1-J3-J2-J4), (J3-J4-J1-J2), (J1-J3-J4-J2)**

Thus, the best sequences are
J5-J1-J2-J3-J4
J5-J4-J2-J3-J1
With makespan of **57**.

Complexity

Complexity of NEH Algorithm

The total number of enumerations in Neh is given by

$$n(n+1)/2$$

which clearly states that the complexity of this algorithm is $\Theta(n^2)$.

Complexity of improved heuristic

The total number of enumerations in case of the improved heuristic is given by[18]

$$4! + \sum, (z=5 \text{ to } n)\ k * z$$
$$= 4! + k * \sum, (z=5 \text{ to } n)\ z$$

Where, k denotes the algorithm parameter,
And n is the number of jobs.

Hence, the algorithmic complexity of this approach is $\Theta(n^2)$.

Conclusions

The improved heuristic proposed for PFSP yields better result than original NEH algorithm while maintaining the same algorithmic complexity.
The main focus of this work has been cyclic permutation flow-shop scheduling problem. In today's world of multitude customer choices, manufacturers often have to produce multiple types of components in large quantities.

As shown using an example, the improved heuristic generates lower makespan as compared to the NEH algorithm and also we can generate more options of job sequences that can be implemented for greater production with lower makespan.

The objective was to design a mathematical formulation for the problem and develop heuristic algorithms to obtain the optimal or near-optimal solution which minimizes the total completion time (i.e. makespan) of the system with same complexity.

Future Scope

NEH is considered to be the best known heuristic for PFSPs. But the proposed heuristic has been proved to outperform NEH.

Hence, this heuristic has a great scope in industry where n jobs are required to be scheduled on m machines for greater production, efficient planning of resources and maintaining proper control over the industry.

References

[1] Dipak Laha , Uday Kumar Chakraborty, (2007), Int. J. Information and Communication Technology, Vol. 1, No. 1
[2] Nawaz, M., Enscore Jr, E. and Ham, I., (91-95), The International Journal of Management Science. v11. 91-95.
[3] Pawel J. Kalczynski, Jerzy Kamburowski, (2008), Journal Computers and Operations Research, Volume 35, Issue9
[4] Bestwick, P.F. and Hastings, N.A.J. (1976) 'A new bound for machine scheduling', Operational Research Quarterly, Vol. 27, pp.479–490.
[5] Brown, A.P.G. and Lomnicki, Z.A. (1966) 'Some applications of the branch and bound algorithm to the machine scheduling problem', Operational Research Quarterly, Vol. 17, pp.173–182.
[6] Campbell, H.G., Dudek, R.A. and Smith, M.L. (1970) 'A heuristic algorithm for the

n-job, m-machine sequencing problem', Management Science, Vol. 16, pp.630–637.

[7] Cormen, T.H., Leiserson, C.E. and Rivest, R.L. (1990) 'Introduction to Algorithms', Cambridge,MA: MIT Press.

[8] Dannenbring, D.G. (1977) 'An evaluation of flowshop sequencing heuristics', Management Science,Vol. 23, No. 11, pp.1174–1182.

[9] Gupta, J.N.D. (1971) 'A functional heuristic algorithm for the flow-shop scheduling problem',Operational Research Quaterly, Vol. 22, No. 1.

[10] Ignall, E. and Schrage, L. (1965) 'Application of the branch-and-bound technique to some flowshop scheduling problems', Operations Research, Vol. 13, pp.400–412.

[11] Johnson, S.M. (1954) 'Two and three stage production schedules with set-up times included', Naval Research Logistics Quarterly, Vol. 1, pp.61–68.

[12] Koulamas, C. (1998) 'A new constructive heuristic for the flowshop scheduling problem', European Journal of Operations Research, Vol. 105, pp.66–71.

[13] Kreyszig, E. (1972) 'Advanced Engineering Mathematics', New York: John Wiley.

[14] Lomnicki, Z.A. (1965) 'A branch-and-bound algorithm for the exact solution of the three-machine scheduling problem', Operational Research Quarterly, Vol. 16, pp.89–100.

[15] Nawaz, M., Enscore Jr., E.E. and Ham, I. (1983) 'A heuristic algorithm for the m-machine n-job flowshop sequencing problem', OMEGA International Journal of Management Science, Vol. 11, pp.91–95.

[16] Turner, S. and Booth, D. (1987) 'Comparison of heuristics for flowshop sequencing', OMEGA,Vol. 15,pp.75–78