

Heuristics for production scheduling problem with machining and assembly operations**A S Bhongade* and P.M. Khodke***Department of Mechanical Engineering, Government College of Engineering, Amravati, M.S. 444604, India***ARTICLE INFO***Article history:*

Received 30 June 2011
 Received in revised form
 September, 01, 2011
 Accepted 01 September 2011
 Available online
 11 September 2011

Keywords:

Production scheduling
Assembly flow shop
Makespan
Heuristics

ABSTRACT

This work deals with production scheduling problem in an assembly flow shop, having parts machining followed by their subsequent assembly operations. Limited heuristics available on the problem, are based on unrealistic assumption that every part is processed on all machines. In this paper, two heuristics NEH_BB and Disjunctive are proposed to solve assembly flow shop scheduling problem where every part may not be processed on each machine. Exhaustive computational experiments are conducted with 60 trials each. The methods are found to be applicable to large size problems. The objective functions used for comparison are makespan and computational time. Disjunctive method takes very less computational time as compared to NEH_BB and hence claimed to be the better among available approaches for finding solution in assembly flow shop problems.

1. Introduction

A production scheduling problem deals with processing j jobs or parts on w machines and determining the sequence and timing of each operation on each machine such that some given performance criterion is maximized or minimized. Most of the production scheduling problem in the literature considers only machining operations in job shop or flow shop manufacturing system. However, manufacturing systems, where several products are produced and each product is made from several parts and/or subassemblies that require a series of machining operations in first stage and assembly operations at later stage are common in many industries. Here, the tasks required to complete a product (final assembly) consist of both machining and assembly operations. A final assembly or sub assembly may have to wait not only for a machine but also for its mating part. This joint consideration increases the complexity of assembly shop scheduling problems when compared to scheduling in the general shop. Because of the computational complexity and the fact that the generation of optimal schedules more likely requires excessive computational time, independent of the methodology, the pursuit of 'pure' optimal scheduling methods is impractical. This means that the primary challenge faced by researchers is to develop efficient solution methodologies that generate near-optimal solutions. This work is an attempt to solve scheduling problem of such systems.

* Corresponding author. Tel.: +91-22-26707026
 E-mail: ajay_bhongade@rediffmail.com (A. S. Bhongade)

Majority of production shop scheduling problems falls into two classes i.e. flow shop and job shop scheduling. In flow shops, operations on the jobs are processed in fixed order on a set of machines arranged, sequentially. On the other hand, the job shop problem involves a set of jobs with no clear flow pattern and the job visits a set of machines in any order (Pinedo, 2002). The production scheduling problem for these two classes with machining and assembly operations are accordingly named as assembly job shop problem (AJSP) and assembly flow shop problem (AFSP).

In AJSP, the operations of a part are carried out serially on one set of machines by following the precedence relationship, while those of another part belonging to the same assembly may be carried out in parallel on another set of machines. The performance of the system is affected mainly by product structure complexity (single-level, multi-level assembly), dispatching rule and scheduling method. Fry et al. (1989) and Philipoon et al. (1991) investigated the performance of some dispatching rules to minimise mean flow time or mean tardiness. Doctor et al. (1993) developed a heuristic algorithm to solve the model. The capability of the algorithm was limited to assembly jobs with no more than three levels of assemblies. McCoy and Egbelu (1998) and Park and Kim (1999) modeled the problem using mathematical programming approach. They developed the heuristic solution procedures considering two-level assembly product structure. Moon-Won and Kim (2000) developed mixed integer linear programming model and a branch and bound algorithm. Results of computational experiments using two-level product structure on randomly generated test problems showed that the B&B algorithm found optimal solutions of problems of moderate sizes in a reasonable amount of computation time. Extensive simulation study for assembly job shops with different levels of shop utilization, product structure up to three-level and job due-date settings, has been carried out (Thaigrajan, 2005). It evaluates all the dispatching rules under consideration with respect to a number of measures of performance. Lu et al. (2011) considered single level product structure and experimented on the application of order review/release (ORR) combined with dispatching rules in optimizing due date and flow time related performance measures. It shows incorporation of ORR control into the existing control of dispatching rules is highly important for better performance of assembly job shop control. Pereira and Santoro (2011) proposed a new heuristic method named integrative scheduling simulation, which was proposed based on the use of sequencing rules. It investigates the effect of product structure, scheduling method and sequencing rules on the system performance.

Most of the available literature on AFSP considers the situation, where each part of a product is first processed concurrently on each of m independently operating machines; these machines constitute the first stage of the problem. In the second stage, an assembly machine assembles the m prepared parts to form the product. Lee et al. (1993) considered n products, each product being assembled only from two types of parts. An approximate solution is obtained by using three modified Johnson heuristics and branch and bound method. The same approach is adopted by Potts et al. (1995) for assembly of personal computers. Hariri and Potts (1997) developed the problem with computational experience. Koulamas and Kyparisis (2001) and Hatami et al. (2010) introduced an intermediate transportation stage in between the machining and assembly stage. Hatami et al. (2010) also considered sequence dependent set up time and transfer time. Xi et al. (2003) proposed a series of heuristic algorithms to find the optimal solution for problem similar to Lee et al. (1993). Tozkapan et al. (2003) developed a branch and bound procedure with an objective of minimizing the total weighted flowtime. The problems with respect to maximum lateness, mean completion time and weighted sum of makespan are addressed by Allahverdi and Fawaz (2006; 2007; 2008; 2009). Torabzadeh and Zandehi (2010) found cloud theory based simulated annealing algorithm performing better than simulated annealing algorithm of Allahverdi and Fawaz (2008). However, little work on AFSP deals with situation closer to industrial application, where all parts are processed serially through the same set of machines in the first stage and machines in the next stage assemble all these parts into a final product. Park and Kim (1999) introduced mathematical model and suggested heuristic solution procedure to solve two-level assembly problem with one machine each in three stages namely machining, subassembly and

assembly. Yokoyama (2001) developed an integrated scheduling model and obtained a solution using a branch and bound method. Effectiveness of method is demonstrated by performing computational experiments up to ten products, total 150 parts, three machines in the first stage and one machine each in later subassembly and assembly stage. This model is further extended for computations on maximum 25 products and total 625 parts considering two machines in the first stage and one in the second stage (Yokoyama & Santos, 2005). Similar work, considering setup time with four machines in the first stage and one machine in the second stage is also reported (Yokoyama, 2008).

This indicates limited research is available on this most realistic area of AFSP, where each part is processed on multiple machines in the first stage. Moreover, heuristics available assume part to be processed on every machine and deals with maximum four machines in the machining stage. Also, the methods used to find solution are applicable to moderate size problems. It is, therefore, necessary to search for other heuristic capable of finding better solution for problems of any size in AFSP. Therefore, in this work, scheduling model for this area of AFSP is developed, with some parts having no operations on one or more machine. Two different heuristics are proposed for large size problems. The objective is to find sequence of parts on each machine so as to minimize the completion time (makespan) of all products.

The AFSP is formulated in next section. Section 3 presents algorithms for three heuristics to find solution. Preliminary experiments are demonstrated in section 4 and results are discussed. The algorithms are validated by performing computational experiments in section 5, followed by conclusions.

2. Problem formulation

In AFSP, as shown in Fig.1, several products $1, 2$ are manufactured on multiple machines w . Few parts as J_{12}, J_{13} are assembled in the sub-assembly, and remaining parts and the sub-assembly are assembled to form final product.

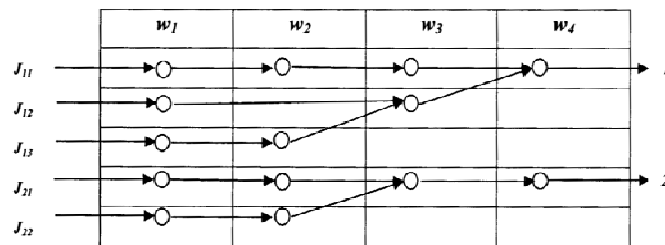


Fig. 1. Assembly flow shop

Common assumptions made and notations used are as follows,

1. Each part can be processed at the most on one machine at a time.
2. Each machine can process only one part at a time.
3. No pre-emption is allowed, i.e. the processing of a part on a machine cannot be interrupted.
4. All parts are independent and are available for processing at zero time.
5. The set up times of parts and assemblies are negligible and therefore are ignored.
6. The machines are continuously available.
7. In-process inventory is allowed. If next machine in a sequence needed by a part is not available, it waits and joins the queue at that machine.
8. Each machine processes parts in the same order, i.e., if part J_{hi} is processed on a machine before part J_{hi+1} , then it is true for all machines.
9. Processing of any part for a sub-assembly must be done after processing of all parts for the previous sub-assembly is completed. Similarly, processing of any part for a product is done after processing of all parts of the previous product is completed.
10. All parts may or may not require all machines.

11. Transit time is zero. It means a part processing starts on a machine w_{j+1} as soon as it leaves machine w_j .

Notations:

h = product number

H = total number of products

J_{hi} = name of i^{th} part for product h

J_h = number of parts for the product h

J = total number of parts for all products

w_j = machine number

W = total number of machines for machining and assembly

$P(j_{hi}, w_j)$ = processing time of part J_{hi} on machine w_j

O_k = operation number

$makespan_h$ = completion time for product h

$makespan$ = completion time for all products

$STWS(J_{hi}, w_j)$ = Start time of operation on machine w_j for part J_{hi}

$FTWS(J_{hi}, w_j)$ = Finish time of operation on machine w_j for part J_{hi}

Consider assembly flow shop where H is the number of total products and J is the number of total parts. Each product h is made up of several parts. These parts are processed on machines in the technological sequence. Further, they form sub-assembly and assembly on the last machines as shown in Fig. 1.

In this work, first the order of the parts for each product giving minimum completion time is decided. Then the order of products (with a predefined order of parts) which gives minimum completion time is found out. To decide the order of the parts for each product giving minimum completion time, several combinations for sequence of parts are obtained. For every combination, completion time for each product is calculated using Eqs. (1-7).

The start time of the first operation of the first part in sequence is assumed to be zero as stated in Eq. (1). The finish time of the operation of the part is calculated using Eq. (2).

For J_{hi} with $i=1$, and for w_j with $j = 1$ or 2 or as applicable

$$STWS(J_{hi}, w_j) = 0, \quad (1)$$

$$FTWS(J_{hi}, w_j) = STWS(J_{hi}, w_j) + P(J_{hi}, w_j), \quad (2)$$

Start time and finish time of subsequent operations of same part J_{hi} with $i=1$ on machines w_j , $j=2,3,\dots,W$ are calculated by Eq. (3) and Eq. (2), respectively.

$$STWS(J_{hi}, w_j) = \max\{FTWS(J_{hi}, (w_j - 1)), FTWS(J_{hi}, (w_j - 2))\dots\}, \quad (3)$$

Eq. (4) and Eq. (2) are used for start time and finish time operations on the first machine w_j , $j=1$ of subsequent parts $J_{hi}, i=2,3,\dots, J_k$.

$$STWS(J_{hi}, w_j) = \max\{FTWS((J_{hi} - 1), w_j), FTWS((J_{hi} - 2), w_j)\dots\}, \quad (4)$$

Eq. (5) or Eq. (6) are used for start time and Eq. (2) is used for finish time operation on machines w_j , $j = 2,3,\dots,W$ of subsequent parts $J_{hi}, i=2,3,\dots, J_k$.

$$STWS(J_{hi}, w_j) = \max\{ \max [FTWS((J_{hi} - 1), w_j), FTWS((J_{hi} - 2), w_j)\dots], \max [FTWS(J_{hi}, (w_j - 1)), FTWS(J_{hi}, (w_j - 2))\dots] \}, \quad (5)$$

for machining operations

$$STWS(J_{hi}, w_j) = \max\{ \max [FTWS(J_{hi}, (w_j - 1)), FTWS(J_{hi}, (w_j - 2))\dots], \max [FTWS((J_{hi} - 1), (w_j - 1)), FTWS((J_{hi} - 1), (w_j - 2))\dots], \max [FTWS((J_{hi} - 1), w_j), FTWS((J_{hi} - 2), w_j)\dots] \}, \quad (6)$$

for assembly operations

Completion time of product is the finish time of last operation of last part of a product h and is calculated using Eq. (7).

$$makespan_h = FTWS(J_h, W) \quad (7)$$

The same sets of equations are used to determine start time and finish time for all products. The start time of first operation is assumed zero for the product which is first in order and is calculated using Eq. (4) for the rest of operations. Completion time for all products is calculated using Eq. (8).

$$makespan = FTWS(J, W) \quad (8)$$

Directed graph, as shown in Fig. 2, is used for operation based representation of problem, where set of vertices $\{O_1, O_2, O_3, \dots\}$ stands for operations. Direction of arrow shows precedence relationship among the operations. Accordingly, processing operation i.e. machining or assembly is identified and assigned a number. Two dummy operations are added at the beginning and at the end. Required data related to processing time, part number, work station number etc. are assigned to each operation.

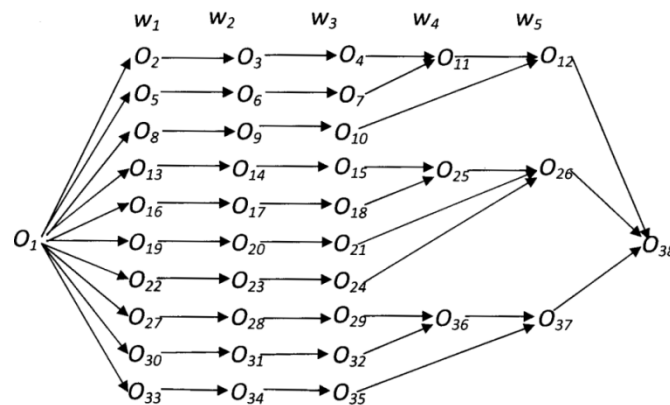


Fig. 2. Directed graph

3. Proposed heuristics

Heuristics used in this work are named as Total Enumeration, NEH_BB and Disjunctive. The sequence of parts to be processed is decided on the basis of makespan of possible alternatives in the first two heuristics. In Disjunctive, it is based on the processing time of parts on the first machine and availability of machine and part for subsequent operations. Furthermore, Total enumeration heuristic calculates makespan for all possible alternatives. NEH_BB determines lower and upper bound values of makespan at every step and the solution giving least makespan is further branched off. The detailed procedures for these heuristics are described in this section.

3.1 Total enumeration heuristic

The solution is obtained using the procedure given below,

1. Determine all possible sequences of parts for product h , which is equal to $J_h!$;
2. Calculate the $makespan_h$ for each sequence and determine the sequence which gives minimum $makespan_h$ for product h ;
3. Repeat step 1 and 2 for remaining products to decide sequence of their parts which is considered fixed for further procedure;
4. Determine all possible sequences ($H!$) of products;
5. Calculate the $makespan$ for every sequence obtained in step 4 and determine the one which gives minimum $makespan$.

3.2 NEH_BB heuristic

Here, branch and bound heuristic is combined with well known NEH heuristics (Nawaz et al., 1983) for flow shop. The procedure is divided in two phases. In the first phase, the sequence of parts for minimum *makespan_h* is decided for every individual product and in the second phase, the sequence of product is decided for minimum *makespan*.

Phase I – Decide sequence of parts

1. Calculate total processing time of each part of product *h* for all operations till assembly;
2. Form an initial set of parts by arranging them in descending order of processing time calculated in step 1 (Nawaz & Ham, 1983) where the set refers to order of parts processed;
3. Determine subsequent sets by interchanging (branching) the part at first position of initial set with parts at rest positions; Thus number of sets formed including the initial one is J_h .
4. Calculate the *makespan_h* for all sets obtained in step 3; The set with minimum value of *makespan_h* is identified and the part at first position is decided.
5. The set obtained in step 4 becomes initial set to decide the part at second position.
6. Further branching is done by interchanging the part at the second position of the set obtained in step 4 with parts at subsequent remaining positions. (Thus J_h-1 sets are formed)
7. Repeat step 4 and decide part at second position and continue the same procedure till part at J_h-1 position is decided which gives sequence of parts for minimum *makespan_h*;
8. Repeat steps 2- 7 for remaining products to decide their sequence of parts with minimum *makespan_h*;
9. This sequence of parts for each product is considered fixed for further procedure in phase – II.

Phase II – Decide sequence of products

10. Order of products in the initial set is kept same as given in the problem.
11. Steps 3 to 7 are followed by considering the products instead of parts and by considering the same sequence of parts obtained in step 9 (in phase – I), decide the sequence of products for minimum *makespan*.

The flow chart for the heuristic is shown in Fig 3.

3.3 Disjunctive heuristic

Here, the number of stages depends on the number of machines rather than the number of parts. Hence, graph theory, as shown in Fig. 2, becomes operative. In this heuristic, the order of parts to be processed (referred as operations in the Fig. 2) on each machine is decided. As the operations which are not connected with each other are compared, the heuristic is named as disjunctive. In first phase, the sequence of operations on different machines is decided for every individual product. In second phase, the sequence of product is decided for minimum *makespan*.

Phase I – Decide sequence of operations on machines for each product

1. Set of operations on every machines are identified for product *h* and bottleneck machine $BN(h)$ is identified.
2. Every set obtained in step 1 for machine before $BN(h)$ is arranged in ascending order of processing time and others are arranged in descending order of processing time.
3. Thus sequence of the parts for product *h* is decided on first machine.

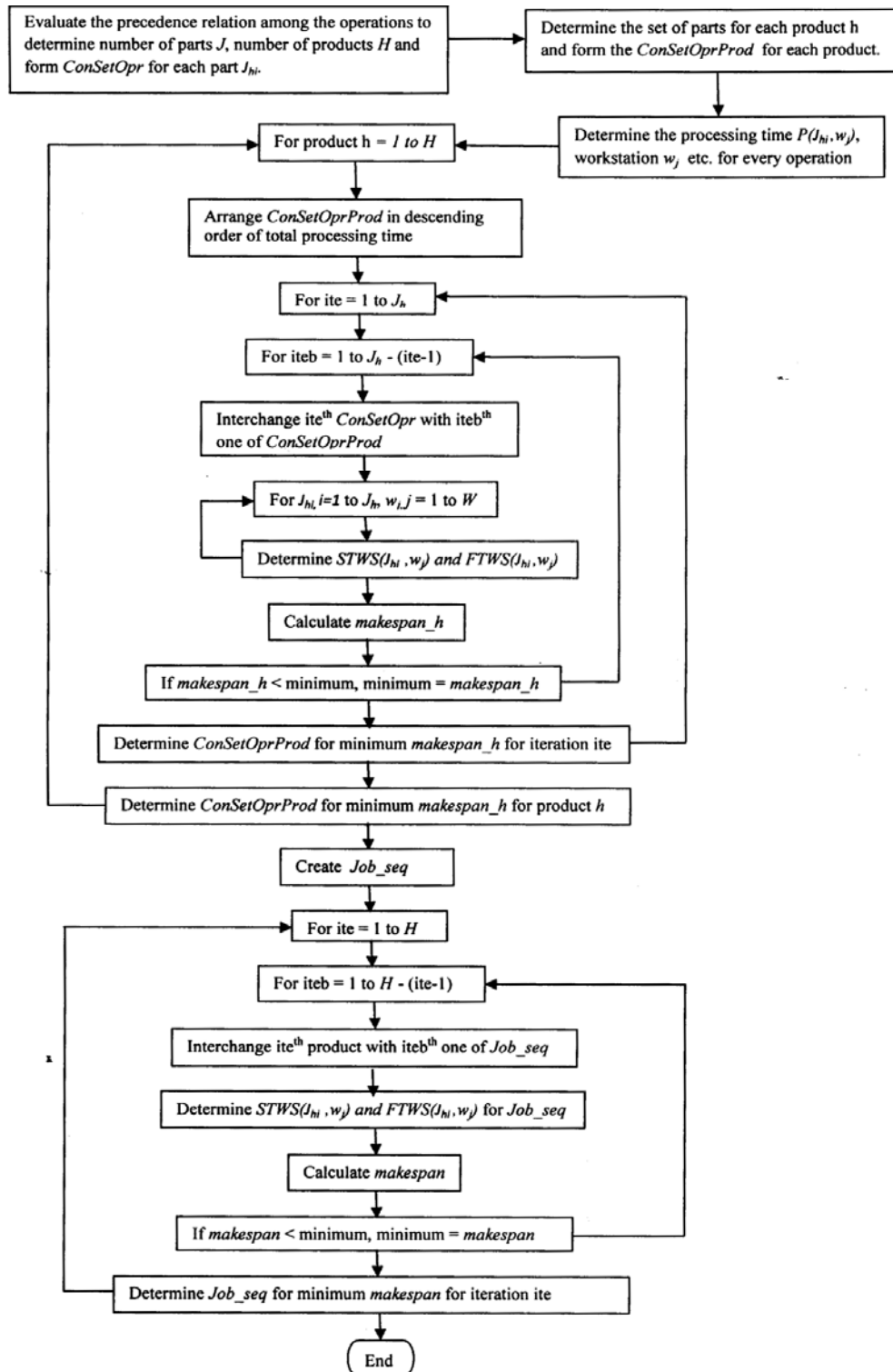


Fig. 3. Flow chart for NEH_BB method

4. The order of parts on subsequent machine is decided on the basis of availability of the part and also availability of machine. However, preference is given to the order obtained in step 2 for that machine. It gives sequence of parts S_M_h to be processed on every machine.
5. Repeat the steps 1- 4 for remaining products to decide their S_M_h ;
6. S_M_h for every individual product is considered fixed for further procedure in phase – II.

Phase II – Decide sequence of products

7. Sequence of products is decided by using branch and bound method as used in phase II of NEH_BB heuristic for minimum *makespan*.

The flow chart for the heuristic is shown in Fig 4.

4. Preliminary experimentation

In order to make the procedure of different heuristic clear, preliminary experiments on proposed approaches are conducted on a simple numerical example.

Table 1
Processing times of parts

Part No.	Processing Time on Machine				
	Machining			Assembly	
	$w1$	$w2$	$w3$	$w4$	$w5$
1	12	14	9		
2	5	9	14	11	16
3	5	12	12	-	
4	9	7	11		
5	9	7	12	15	
6	10	11	5	-	21
7	10	13	4	-	
8	14	5	9		
9	12	13	6	8	15
10	14	10	12	-	

4.1 Numerical example

The numerical example by Yokoyama (2001) with the number of products $H=3$ and the number of machines $W=5$ is taken. Processing times for machining and assembly operations on respective machine are shown in Table 1. A precedence relationship among operations of numerical example is shown in Fig. 2.

The data is processed by the computational program coded in MATLAB. Computations are executed on a system with 2.19G Intel(R) core (TM) 2 Duo CPU and 2GB RAM under the Windows XP-2002 operating environment.

Table 2
Preliminary experimental results of three heuristic procedures

Problem	01	02	03	04
Total parts (J)	10	10	15	35
Number of products (H)	3	3	1	5
Number of machines (W)	5	5	5	5
Total number of operations (O_k)	36	30	39	94
Makespan				
Total Enumeration	137	130	N.A.	460.45
NEH_BB	137	130	156	462.13
Disjunctive	142	125	188	475.33
Order of products				
Total Enumeration	1, 2, 3	1, 2, 3	N.A.	2,1,3,5,4
NEH_BB	1, 2, 3	1, 2, 3	1	5,4,1,2,3
Disjunctive	1, 3, 2	1, 2, 3	1	4,5,2,1,3
Time (CPU)				
Total Enumeration	1.093	0.9063	N.A.	4700
NEH_BB	0.8906	0.7344	43.23	81.8
Disjunctive	0.4844	0.5469	0.312	8.34

Results obtained from computational program for the numerical example are shown in Table 2 under problem '01' indicating minimum makespan, order of products and time taken for three heuristics. The results are also demonstrated on Gantt charts drawn for each of the heuristics as shown in Fig 5.

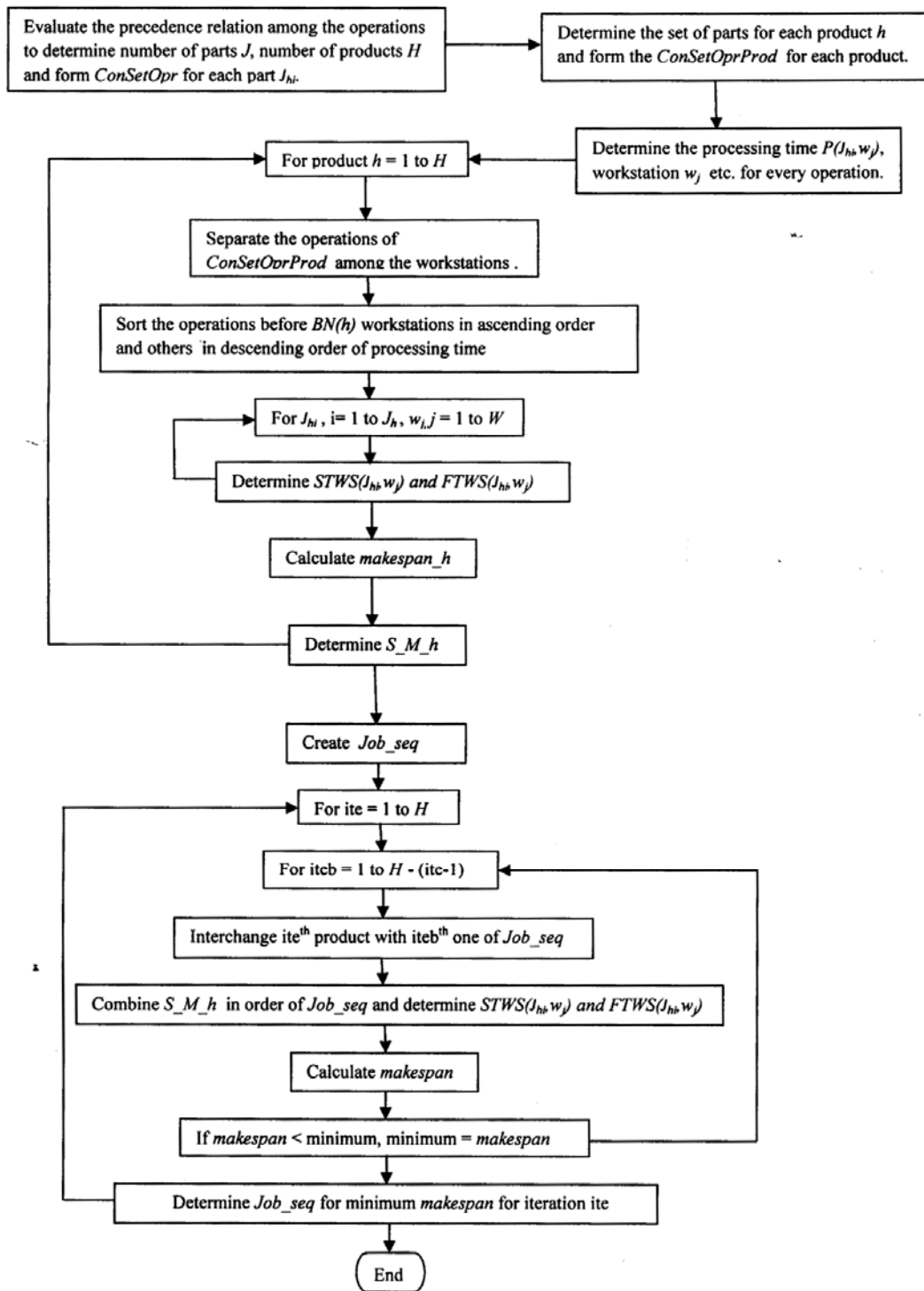


Fig. 4 Flow chart for disjunctive method

Gantt chart indicates schedule of parts, schedule of products, start time and finish time of each operation on different machines, and completion time of each product (indicated at top). The number in shaded region corresponds to operation number given in Fig. 2.

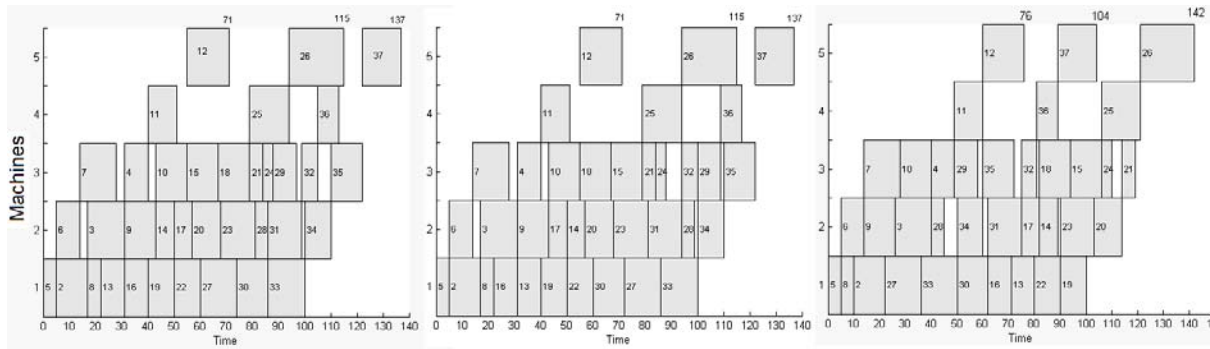


Fig. 5. Gantt chart for operations - Total Enumeration Method, NEH_BB and Disjunctive heuristics

4.2 Preliminary experiments

Three more problems are developed from numerical example (problem '01'), given in section 4.1, by changing number of operations, number of parts and number of products. Experiments corresponding to this problems, as defined below, are performed on the computational program.

Problem '02': All the parts may not require every machine and hence few of machining operations are removed from problem '01'.

Problem '03': Considering only one product, number of the parts is increased to 15.

Problem '04': Number of products as well as number of parts per product are increased to 5 and 7 respectively.

The processing times are generated randomly (0, 20) for problem '03' and '04'.

4.3 Observations

A scrutiny of experimental results, given in Table 2, lead to following observations.

1. It is the first attempt to solve AFSP where the parts may not have operation on all machines. Hence, no data is available in literature for comparison.
2. Total enumeration heuristic is not applicable for problem '03' owing to more number of permutations due to increase in parts per product. The computation time of Total enumeration heuristic grows exponentially for more than seven number of product or number of part per product.
3. Makespan obtained by Total enumeration heuristic is better (near optimal), except for the problem '02' where Disjunctive heuristic gives the best solution. This is because, the problem '02' is created by removing few of machining operations from problem '01' and different way of Disjunctive method to allocate operations. In Disjunctive, the allocation of operation on the next machine is decided by the preference of operations on that machine and completion of preceding operation. In other two methods, the allocation of operation on the next machine is decided by completion of preceding operation.
4. Makespan of NEH_BB and Total enumeration approaches are closer to each other while disjunctive approach gives more makespan.
5. Computation time depends on number of iterations. It is found to be least with Disjunctive heuristic.

5. Experimentation results and discussion

In order to show validity of proposed methods, several computational experiments are made with a larger scale. In view of limitations of Total Enumeration heuristic in handling large size problems, further experimentation is conducted using NEH_BB and Disjunctive heuristics. Fractional factorial experimental design using L4 orthogonal array (Phadke, 1989) was employed. The three factors at two different levels to vary problem size are number of products, parts per product and number of machines, as shown in the Table 3.

Table 3

Factor setting

Factors	Level 1	Level 2
<i>H</i>	10	20
<i>J_h</i>	10	20
<i>W</i>	5	10

Four basic problems (Problem A, B, C, and D) designed with distinct combination of factor levels is shown in Table 4. The number of parts ranges from 100 to 400. Further, five instances of each basic problem are created by varying processing time. These dataset of instances can be made available. Processing times for machining and assembly operations are generated with random number (0, 20) and (10, 30), respectively. Each instance is then replicated thrice. Hence, total 60 experimental trails are conducted on NEH_BB and Disjunctive heuristics each. The results of both the heuristics for problems A, B, C, and D are shown in Table 4 along with problem size, total number of parts and number of operations. The makespan values for three replicates of same instance are averaged and percentage difference of mean makespan value (MMV%) of each instance is obtained using Eq. (9), relative to the best solution for that instance.

$$MMV\% = \frac{(\text{Average makespan for instance} - \text{Best makespan})}{\text{Best makespan}} \times 100 \tag{9}$$

The values of MMV% obtained for different instances of the same problem size are then averaged for each heuristic and indicated by average of MMV%. Average MMV% indicates that NEH_BB approach yields better solution in majority of problems except problem D. In problem D where problem size is large with highest number of parts, NEH_BB approach considers comparatively less percentage of total permutations and hence Disjunctive approach gives better solution. Table 4 also indicates mean relative percentage increase in computation time.

The result shows that computation time in NEH_BB is very high as compared to disjunctive approach. To determine the effect of each factor of problem size on the computation time, averages of all the values of CPU time for each level of factors are calculated. The plots of relative change in this average CPU time for each factor levels are shown in Fig. 6. The number of products seems to have significant effect, while number of machines has least effect on the computation time. The computing time in NEH_BB is affected more with *J_h*, as number of iterations needed is proportional to *J_h* to decide sequence of parts.

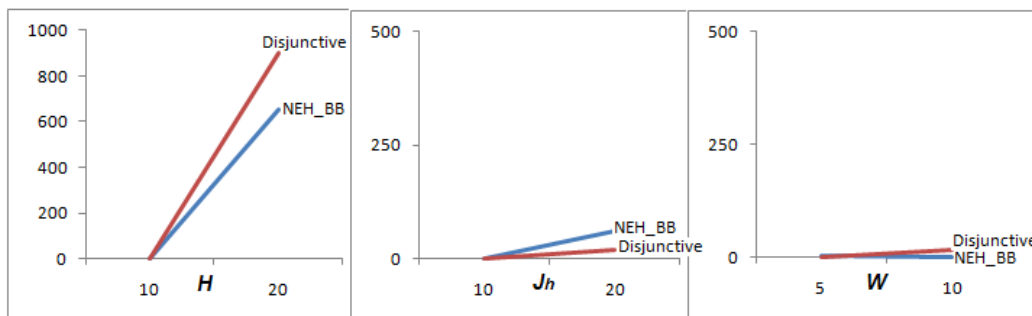


Fig. 6 Relative change in average CPU time for number of products, parts per product and number of machines

Table 4

Relative performance of two heuristics for large size problems with respect to the best heuristic

Problem size	Total parts	Number of operations	Instances	Percentage difference of mean makespan value (MMV%)		
				NEH_BB	Disjunctive	
$H/J_i/W$				NEH_BB	Disjunctive	
Problem A						
10/10/5	100	290	1	0	0	
			2	0	2.9148	
			3	0	1.4749	
			4	0.5543	0	
			5	0	1.1435	
Average of MMV%				0.1108	1.1066	
Mean relative percentage increase in time				2256.44	0	
Problem B						
10/20/10	200	1440	1	0	1.7217	
			2	0	1.6569	
			3	0	1.7209	
			4	0	1.6457	
			5	0	3.1685	
Average of MMV%				0	1.9827	
Mean relative percentage increase in time				4537.86	0	
Problem C						
20/10/10	200	1520	1	0	0.0929	
			2	0	0.1863	
			3	0	0.6155	
			4	0	0.2225	
			5	0	1.2167	
Average of MMV%				0	0.4668	
Mean relative percentage increase in time				2819.75	0	
Problem D						
20/20/5	400	1120	1	0	0.0803	
			2	0	0.3324	
			3	0.5660	0	
			4	0.1365	0	
			5	0.1422	0	
Average of MMV%				0.1689	0.0825	
Mean relative percentage increase in time				3764.43	0	

Although both heuristics are found suitable for finding solution for large size problems, yet, Disjunctive approach is proposed to be preferred due to less CPU time.

6. Conclusions and future scope

The production scheduling model for assembly flow shop considering both machining and assembly operations, with some parts having no operations on one or more machine, has been studied with an objective to obtain near optimal solution for scheduling on the basis of completion time (makespan) of all products. Two approaches NEH_BB and Disjunctive are proposed and conclusions are as under.

1. NEH_BB is an extension of well established NEH approach by introducing branch and bound heuristic to obtain sequence of parts. Disjunctive performs sorting of operations to obtain sequence of parts. Both heuristics performs branch and bound iterations to obtain sequence of products.
2. Both NEH_BB and Disjunctive proposed heuristics found to be capable of providing solution for large scale problems, former giving better solution in majority of situations.
3. NEH_BB heuristic is found to give better solution except the situation where problem size is large. For large problem size, percentage of total permutations considered for analysis by NEH_BB heuristic is comparatively less and therefore Disjunctive approach gives comparatively better solution.
4. The computation time of the NEH_BB heuristic is disproportionately high and practically not recommended for obtaining initial solution.

The work can be extended to improve the solution using metaheuristics like Genetic algorithm, simulated annealing etc. In view of less computation time and also better solution for large problem size, Disjunctive heuristic may provide good initial solution.

References

- Allahverdi A., & Al-Anzi, F. S. (2006). A PSO and a Tabu search heuristics for the assembly scheduling problem of the two-stage distributed database application. *Computers and Operation Research*, 33(4), 1056-1080.
- Allahverdi, A. & Al-Anzi, F. S. (2007). The two-stage assembly flowshop scheduling problem with bicriteria of makespan and mean completion time. *International Journal of Advance Manufacturing Technology*, 37 (1), 166–177.
- Allahverdi A., & Al-Anzi, F.S. (2008). The two-stage assembly flowshop scheduling problem with bicriteria of makespan and mean completion time. *International Journal of Advance Manufacturing Technology*, 37(1-2), 66-177.
- Al-Anzi, F. S. & Allahverdi, A. (2009). Heuristics for a two-stage assembly flow shop with bicriteria of maximum lateness and makespan. *Computers and Operation Research*, 36, 2682-2689.
- Doctor, S. R., Cavalier, T. M. & Egbelu, P. J. (1993). Scheduling for machining and assembly in a job-shop environment. *International Journal of Production Research*, 31 (6), 1275-1297.
- Fry, T.D., Olife, M.D., Minor, E.D., & Keong Leong, G. (1989). The effect of product structure and sequencing rule on assembly shop performance, *International Journal of Production Research*, 27 (4), 671-686.
- Hariri, A. M. A., & Potts, C. N. (1997). A branch and bound algorithm for the two-stage assembly scheduling problem. *European Journal of Operational Research*, 103(3), 547-556.
- Hatami, S., Ebrahimnejad, S., Tavakkoli-Moghaddam, R. & Maboudian, Y. (2010). Two meta-heuristics for three-stage assembly flowshop scheduling with sequence dependent setup times. *International Journal of Advance Manufacturing Technology*, 50(9-12), 1153-1164.
- Koulamas, C., & Kyparisis, G.J. (2001). The three-stage assembly flowshop scheduling problem. *Computers and Operation Research*, 28(7), 689-704 .
- Lee, C. Y., Cheng, T .C. E., & Lin, B. M. T. (1993). Minimizing the makespan in the 3-machine assembly-type flowshop scheduling problem. *Management Science JSTOR*, 39(5), 616–625 .
- Lu, H. L., Huang, G. Q. & Yang, H. D. (2011). Integrating order review/release and dispatching rules for assembly job shop scheduling using a simulation approach. *International Journal of Production Research*, 49(3), 647–669.
- McCoy, D. H. C., & Egbelu, P. J. (1998). Minimizing production flow time in a process and assembly job shop. *International Journal of Production Research*, 36 (8), 2315-2332.

- Moon-Won, P., & Kim, Y. D. (2000). A branch and bound algorithm for a production scheduling problem in an assembly system under due date constraints. *European Journal of Operational Research*, 123(3), 504-518 .
- Nawaz, M., Enscore, Jr. E. E. & Ham, I. (1983). Heuristic algorithm for the m-machine, n-job flowshop sequencing problem. *Omega, International Journal of Management Science*, 11(1), 91–95.
- Park, M. W., & Kim, Y. D. (1999). A heuristic algorithm for a production planning problem in an assembly system. *Journal of Operational Research Society*, 50, 138-147.
- Pinedo, M. (2002). *Scheduling: Theory, Algorithms and Systems*, Second ed. Prentice-Hall, Englewood Cliffs, NJ.
- Pereira, M. T., & Santoro, M. C. (2011). An integrative heuristic method for detailed operations scheduling in assembly job shop systems. *International Journal of Production Research*, 1–17, iFirst.
- Phadke, M.S. (1989). *Quality engineering using robust design*. Prentice hall, Englewood Cliff, NJ.
- Potts, C.N., Sevast'janov, S.V., Strusevich, V.A., Van, Wassenhove L.N. & Zwaneveld, C.M., (1995). The two stage assembly scheduling problem: complexity and approximation. *Operations Research*, 43,346–55.
- Philipoon, P.R., Russel, R.S. L., & Fry, T.D., (1991). A preliminary investigation of multi-attribute based sequencing rules for assembly shops, *International Journal of Production Research*, 29 (4), 739-753.
- Thiagarajan, S., & Rajendran, C. (2005). Scheduling in dynamic assembly job-shops to minimize the sum of weighted earliness, weighted tardiness and weighted flowtime of jobs. *Computers and Industrial Engineering*, 49 (4),463–503.
- Torabzadeh, E., & Zandieh, M. (2010). Cloud theory-based simulated annealing approach for scheduling in the two-stage assembly flowshop. *Advance Engineering Software*, 41(10-11),1238-1243.
- Tozkapan A., Kirca, O., & Chung, C. S. (2003). A branch and bound algorithm to minimize the total weighted flowtime for the two-stage assembly scheduling problem. *Computers and Operations Research*, 30(2), 309-320.
- Yokoyama, M. (2001). Hybrid flow shop scheduling with assembly operations. *International Journal of Production Economics*, 73(1), 103-116.
- Yokoyama, M. & Santos, D. L. (2005). Three-stage flow-shop scheduling with assembly operations to minimize the weighted sum of product completion times. *European Journal of Operational Research*, 161(3), 754-770 .
- Yokoyama, M. (2008). Flow-shop scheduling with setup and assembly operations. *European Journal of Operational Research*, 187(3), 1184-1195.
- Xi , S., Morizawa, K., & Nagasawa, H. (2003). Powerful heuristics to minimize makespan in fixed, 3-machine, assembly-type flowshop scheduling *European Journal of Operational Research*, 146(3), 498-516 .