Middle-East Journal of Scientific Research 24 (11): 3469-3474, 2016
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2016.3469.3474

Minimization of Makespan Using Inverse Branch and Bound Heuristic

[1]T. Illakiya, [2]R. Franklin Issac and [3]R. Pearlin Helina

[1]Assistant Professor, Department of Computer Science and Engineering,
Rajalakshmi Institute of Technology, Chennai, India
[2]Research Scholar, Department of Mechanical Engineering, Anna University, Chennai, India
[3]Department of Computer Science and Engineering, Rajalakshmi Institute of Technology, Chennai, india

Abstract: The flow shop scheduling problem is categorized as NP hard in nature; it is difficult to solve using mathematical or computational aspect alone. This article aims to solve the Permutation Flow Shop (PFS) problem through an effective computational approach for minimizing the makespan. The mathematical model helps in finding the trigger jobs of each stage towards lower bound. In this search, a requirement of reduction in computational time is done through a inverse of branch and bound technique. The effect of mathematical approach through computational support and strong neighborhood search technique the benchmark problems are solved. From the results, it is noticed that the minimizing of makespan is successful achieved compared to existing approaches.

Key words: Branch and bound · Heuristic · Flowshop · Makespan

INTRODUCTION

Scheduling is the process of arranging resources in production environment to perform a collection of task. It is the allocation of resources over time to perform a collection of tasks. Sequencing and scheduling are forms of decision making which play a crucial role in manufacturing as well as in service industries. In the current competitive environment, effective sequencing and scheduling has become necessity for survival in market place. Companies have to meet shipping dates committed to the customer, as failure to do so may result in a significant loss of good will. They also have to schedule activities in such a way as to use the resources available in an efficient manner. Shop floor is a production or manufacturing environment. The area in which manufacturing facility where assembly or production is carried out, either by an automated system or by workers or a combination of both. The shop floor may be include equipment, inventory and storage areas. The time it takes to complete a prescribed procedure. Process time is the period during which one or more inputs are transformed into a finished product a manufacturing procedure.

Types of shop floor are flexible flow shop, open shop, closed shop, hybrid flow shop and job shop. A flexible flow shop (FFS) consists of a flow line with several parallel machines on some or all production stages. Multiple products are produced in such a flow line. While all the products follow the same linear path through the system, all of them may not visit all the stages of production. On each of the stages, one of the parallel machines has to be selected for the production of a given product. The production of a product consists of multiple operations, one for each production stage. When an operation is started on a machine, it must be finished without interruption. single machine flow shop is having one machine and n jobs to be processed. Job shop is machine order can change and job order cannot be changed. It consists of n jobs and m machines each machine can handled at most one operation at a time. Each operation needs to be processed during an uninterrupted period of a given length on a given machine. Open shop is having there are m machines and there is no restriction in the routing of each job through the machine. In other words, there is no specified flow pattern for any job. Closed shop is a job shop however, all production orders are generated as a result of inventory replenishment decisions. In other words, the production is not affected by the customer order.

The constrain of flow shop is job order may change and machine order could not be change. The flow of work is unidirectional. There are *n* machines and *m* jobs.

Corresponding Author: T. Illakiya, Department of Computer Science and Engineering,
Rajalakshmi Institute of Technology, India.

Each job contains exactly *n* operations. The *i*-th operation of the job must be executed on the *i*-th machine. No machine can perform more than one operation simultaneously. For each operation of each job, execution time is specified. Operations within one job must be performed in the specified order. The first operation gets executed on the first machine, then the second operation on the second machine and so until the *n*-th operation. Jobs can be executed in any order, however. Problem definition implies that this job order is exactly the same for each machine. The problem is to determine the optimal such arrangement.

There are m machines in series and jobs can be processed in one of the following ways:

- Permutational: jobs are processed by a series of machine in exactly same order
- 2.Non-permutational: job are processed by a series of m machine not in the same order.

r_i is the ready-time, release time, or arrival time. This is the time at which the job is released to the shop by some external job generation process. It is significant as the earliest time that processing of the first operation of the job could begin. d_i is the due-date. This is the time at which some external agency would like to have the job leave the shop. It is the time by which the processing of the last operation should be completed. Waiting time is the amount of time a process has been waiting in the ready queue. $C_{i,}$ is the completion time of the job i. The time at which processing of the last operation of the job is completed. $F_{i,}$ is the flow-time of job i. The total time that the job spends in the shop. Flow time is also called manufacturing interval and the shop-time. Mean flow time is an average time spent by a job in the shop and comprises of processing time, waiting time and transfer time. Lateness is defined as difference between completion time and due date of the job. Lateness considers the algebraic difference of each job, regardless of the sign of the difference. Tardiness considers only positive differences. Job which are completed after their due date. Earliness considers only negative differences. Job completed ahead of their due dates. Make span is the time difference between the start and finish of a sequence of jobs or tasks.

Branch and Bound algorithmic technique to find the optimal solution by keeping the best solution found so far. The algorithm depends on the efficient estimation of the lower and upper bounds of a region/branch of the search space. The branch and bound method uses a sequence tree. Each node in the tree contains a partial sequence of jobs. For n job problem there are n-1 number of levels for a tree. At zero, root node will be placed with all n empty sequence positions. At level 1, there will be n number of nodes. Each node will contain partial sequence of jobs. The first position in the sequence will be occupied by a job in numerical order. Similarly, each node at $(n-1)^{th}$ level will be branched to (n-2) number of nodes. The process will continue till each node has exactly one leaf. Generation of all sequences is combinational in nature and, will result in enormous number of sequence even for small number of jobs. For example for a 10 job problem there will be 10! Sequences. To reduce the combinational effort, lower bound are calculated at every level for each node. The formula used to compute lower bound is pertinent to objective function of scheduling problem. Branching is carried out only from those nodes with minimum lower bound. By doing so, only small proportion of the nodes is explored resulting in fewer amounts of computations. The branch and bound (B&B) method is applied in almost every scheduling problem.

Existing System: Generally the scheduling problems are classified as fixed batch size problem (static problem) and stochastic process problems (dynamic problem). In fixed batch size problem, the number machine may be single or many. If it is single then it is called as single machine problem. Most of the single machine problem is considered while considering a critical machine. In some rare cases the multi machine problems are considered with critical machine or special machine. This problem is routed to parallel, serious or hybrid shop. Flow shop manufacturing is a common production system found in many manufacturing facilities, assembly lines and industrial processes. A Permutation Flow Shop (PFS) is a shop design of machines arranged in series in which the jobs need to be processed in a same order without eliminating any machine. In industries, scheduling plays a vital role in finding better solution to achieve some criteria. It is known to be a tedious task. Therefore, many researchers focused their efforts on finding an optimal solution with acceptable aspects by using heuristics.

In 1954, a simple algorithm was framed by Johnson [1], for flowshop scheduling problems in the order of ‘n’ jobs in ‘2’ machines. The NP-completeness of the flow shop scheduling problems had been discussed widely by Quan and Ling [2]. In 1965’s Palmer [3] was the first to propose heuristic with a slope index procedure, which was an effective and simple methodology in tracing a better makespan. The significant work in the development of an efficient heuristic was done by Campell, Dudek and Smith [4]. Their algorithm consists essentially in splitting the ‘m’ machine problem into a series of an equivalent two-machine flow shop problem and solving it by Johnson’s rule. Dannenbring [5] had developed a procedure called 'rapid access', which attempts to combine the advantages of Palmer's slope index and CDS procedures.

Stinson and Simith [6] had proposed a radically different approach called travelling salesman problem with two steps. The solution was found to be better than Palmer [7] and CDS methods, but with increased computational effort.

Since the problem is known to be NP-hard, the meta-heuristics are required efficiently to solve industry size problems. Thus, the meta-heuristic with search technique were developed to reach the near optimal solutions for the PFS problem [8]. For applying a search technique in a PFS, an initial solution is generated and then it applies a move mechanism to search the neighbourhood of the current solution to choose the better one [9]. An application to the PFS problem is proposed in various combinatorial optimization problems [10]. Pugazhenthi and Anthony propose a heuristic to minimizing the idle time, of critical machine when the material flows in manufacturing. In this paper they said the reduction of idle time, of critical machine indirectly minimizing the makespan [11, 12]. Schuster and Framinan [13] used the neighbourhood search technique which is specially designed for flow shop problems. This technique is better compared to other instances. A step of search starts with the current feasible solution x ª X to which is applied a function m ª M(x) that transforms x into x', a new feasible solution (x' = m(x)). This transformation is called a *move* and {x': x' = m(x); x, x' ∈ X; m ∈ M(x)} is called the *neighbourhood* of x.

In this article an attempt to minimise the makespan of a PFS problem through the combined effect of mathematical and computational aspects through Inverse Branch and Bound (IBB) technique.

Proposed System

Assumptions:

- The first machine is assumed to be ready whichever and whatever job is to be processed on it first.
- Machines may be idle
- Each job is processed through each of the m machines once and only once. Furthermore a job does not become available to the next machine until and unless processing on the current machine is completed i.e. splitting of job or job cancellation is not allowed.
- In-process inventory is allowed. If the next machine on the sequence needed by a job is not available, the job can wait and joins the queue at that machine.

Notations:

P_{ij}	=	Processing time of j^{th} job in i^{th} machine
T_i	=	Summation of processing time of N jobs in i^{th} machine
a_{ij}	=	summation of processing time of j^{th} job in 1^{st} to $(k-1)^{th}$ machine
b_{ij}	=	summation of processing time of j^{th} job in $(k+1)^{th}$ to M^{th} machine
A_i	=	minimum of a_{ij} for i^{th} machine
B_i	=	minimum of b_{ij} for i^{th} machine
S_i	=	summation of T_i, A_i and B_i for i^{th} machine
LB	=	minimum of S_i
Z	=	pivot machine
ZA, ZB	=	pivot jobs
k	=	representation of pivoting machine
i	=	representation of machine from 1to M.
j	=	representation of job from 1 to N.

Algorithm:

Step 1: Define the number of machines 'M' and jobs 'N'.

Step 2: Assign the processing time of 'N' jobs in 'M' machines. And frame the PFS problem NxM matrix.

Step 3: Calculate a_{ij} and b_{ij} values using the equations (1) and (2).

$$a_{ij} = \sum_{i=1}^{k-1} P_{ij} \tag{1}$$

$$b_{ij} = \sum_{i=k+1}^{m} P_{ij} \tag{2}$$

Step 4: Calculate T_i, A_i and B_i values using the equations (3), (4) and (5)

$$T_i = \sum_{j=1}^{n} P_{ij} \tag{3}$$

$$A_i = \min(a_{ij}) \tag{4}$$

$$B_i = \min(b_{ij}) \tag{5}$$

Step 5: Calculate the S_i values for 'M' machines using the equation (6).

$$S_i = T_i + A_i + B_i \quad (6)$$

Step 6: Calculate the LB value for the NxM PFS problem using the equation (7).

$$LB = \max(S_i) \quad (7)$$

Step 7: Identify the Z machine by the below stated condition in equation (8).

$$\text{if}(LB == T_k + A_k + B_k) \qquad \text{Then, } Z = k; \quad (8)$$

Step 8: Identify the pivot jobs ZA and ZB using the condition stated in equation (9) and (10)

$$\text{if}(A_k == a_{kj}) \qquad \text{Then, } ZA = j; \quad (9)$$

$$\text{if}(B_k == b_{kj}) \qquad \text{Then, } ZB = j; \quad (10)$$

Step 9: Place the ZA and ZB pivoted jobs in the sequence under the condition, if the pivoted job is ZA, ($Z \neq 1$) && ($ZA \neq 1$) then place the ZA at beginning of the sequence. If the pivoted job is ZB, ($Z \neq M$) && ($ZB \neq N$) then place the ZB at end of the sequence.

Step 10: After the step 9 is successful eliminate the ZA and ZB jobs from the NxM PFS problem.

Step 11: Repeat the step 3 to step 10, till N = 1.

Step 12: Arrange the jobs in a sequence according to the pivoting conditions.

Step 13: Find the makespan from the identified sequence.

RESULTS AND DISCUSSIONS

The benchmark problems proposed by Taillard [14] are tested against the newly proposed heuristic (IBB heuristic) for the various sizes of the problems with 20, 50 & 100 jobs through 5, 10 & 20 machines. The results obtained from the MATLAB environment for the CDS heuristic, Palmer heuristic and IBB heuristic are compared and tabulated in Table 1 to 9.

From the Table 1 to 9, it can be seen that by finding the cumulative % of success of IBB in reaching the LB is better compared to others and it is shown in Table 10 and Fig. 1.

From the Table 10, the overall % nearer to LB is calculated and it is shown in Fig. 2. It is observed that the IBB heuristic is better about 4.2% and 0.5% when compared to CDS and Palmer heuristics.

Table 1: 20 jobs through 5 machines

Taillard Seeds	Lower Bound	CDS	Palmer	IBB
873654221	1232	1409	1384	1336
379008056	1290	1424	1439	1360
1866992158	1073	1255	1162	1185
216771124	1268	1485	1490	1338
495070989	1198	1367	1360	1273
402959317	1180	1387	1344	1280
1369363414	1226	1403	1400	1303
2021925980	1170	1395	1313	1313
573109518	1206	1360	1426	1239
88325120	1082	1196	1229	1170

Table 2: 20 jobs through 10 machines

Taillard Seeds	Lower Bound	CDS	Palmer	IBB
587595453	1448	1829	1790	1752
1401007982	1479	2021	1948	1906
873136276	1407	1773	1729	1884
268827376	1308	1678	1585	1585
1634173168	1325	1781	1648	1597
691823909	1290	1813	1527	1518
73807235	1388	1826	1735	1628
1273398721	1363	2031	1763	1735
2065119309	1472	1831	1836	1831
1672900551	1356	2010	1898	1855

Table 3: 20 jobs through 20 machines

Taillard Seeds	Lower Bound	CDS	Palmer	IBB
479340445	1911	2833	2818	2571
268827376	1711	2564	2331	2236
1958948863	1844	2977	2678	2510
918272953	1810	2603	2629	2438
555010963	1899	2733	2704	2452
2010851491	1875	2707	2592	2370
1519833303	1875	2670	2456	2398
1748670931	1880	2523	2435	2383
1923497586	1840	2583	2754	2392
1829909967	1900	2707	2633	2372

Table 4: 50 jobs through 5 machines

Taillard Seeds	Lower Bound	CDS	Palmer	IBB
1328042058	2712	2920	2774	2735
200382020	2808	3032	3041	2987
496319842	2596	3034	2777	2789
1203030903	2740	3156	2860	2898
1730708564	2837	3188	2963	3013
450926852	2793	3154	3090	2852
1303135678	2689	2969	2845	2878
1273398721	2667	3236	2826	2745
587288402	2527	3255	2733	2634
248421594	2776	3167	2915	2820

Table 5: 50 jobs through 10 machines

Taillard Seeds	Lower Bound	CDS	Palmer	IBB
1958948863	2907	3660	3478	3122
575633267	2821	3645	3313	3256
655816003	2801	3659	3321	3251
1977864101	2968	3707	3511	3220
93805469	2908	3664	3427	3118
1803345551	2941	3584	3323	3356
49612559	3062	3806	3457	3222
1899802599	2959	3758	3356	3102
2013025619	2795	3548	3414	3101
578962478	3046	3964	3104	3440

Table 6: 50 jobs through 20 machines

Taillard Seeds	Lower Bound	CDS	Palmer	IBB
1539989115	3480	4759	4272	4268
691823909	3424	4398	4303	4087
655816003	3351	4471	4210	4160
1315102446	3336	4776	4233	4062
1949668355	3313	4642	4376	4095
1923497586	3460	4505	4312	4013
1805594913	3427	4758	4306	4134
1861070898	3383	4554	4310	4033
715643788	3457	4470	4547	4157
464843328	3438	4549	4197	4115

Table 7: 100 jobs through 5 machines

Taillard Seeds	Lower Bound	CDS	Palmer	IBB
896678084	5437	5592	5749	5495
1179439976	5208	5657	5316	5389
1122278347	5130	5619	5325	5340
416756875	4963	5286	5049	5225
267829958	5195	5623	5317	5311
1835213917	5063	5259	5274	5233
1328833962	5198	5557	5376	5342
1418570761	5038	5509	5263	5303
161033112	5385	5821	5606	5686
304212574	5272	5740	5427	5342

Table 8: 100 jobs through 10 machines

Taillard Seeds	Lower Bound	CDS	Palmer	IBB
1539989115	5759	6858	6161	5937
655816003	5345	6284	5889	5523
960914243	5623	6609	6127	6134
1915696806	5732	6783	6313	6089
2013025619	5431	6436	6070	6019
1168140026	5246	6138	5870	5633
1923497586	5523	6456	6442	5738
167698528	5556	6602	6168	6279
1528387973	5779	6356	6081	6420
993794175	5830	6852	6259	6338

Table 9: 100 jobs through 20 machines

Taillard Seeds	Lower Bound	CDS	Palmer	IBB
450926852	5851	7586	7075	6769
1462772409	6099	7709	7058	6922
1021685265	6099	7481	7221	7030
83696007	6072	7895	7039	6907
508154254	6009	7657	7259	6730
1861070898	6144	7590	7109	7159
26482542	5991	8167	7279	7075
444956424	6084	7892	7567	7225
2115448041	5979	7604	7271	7095
118254244	6298	7965	7305	6893

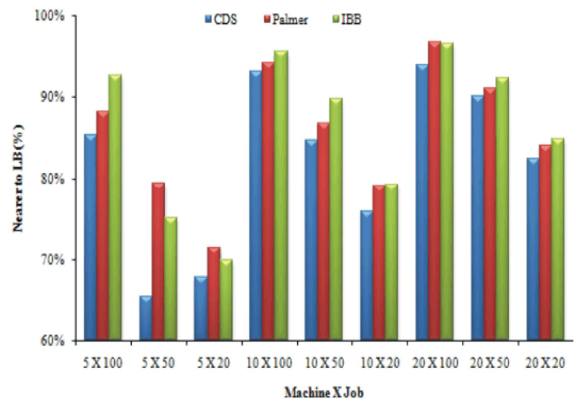

Fig. 1: Comparison of heuristics based on the % nearer to LB

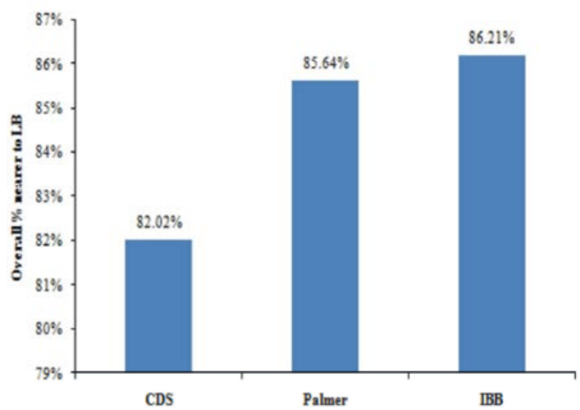

Fig. 2: Comparison of heuristics based on the % nearer to LB

M X N	CDS	Palmer	IBB
5 X 100	85.25%	88.13%	92.61%
5 X 50	65.46%	79.45%	75.11%
5 X 20	67.83%	71.40%	69.90%
10 X 100	93.03%	94.21%	95.55%
10 X 50	84.59%	86.77%	89.74%
10 X 20	75.88%	79.14%	79.27%
20 X 100	93.87%	96.65%	96.55%
20 X 50	89.98%	90.96%	92.33%
20 X 20	82.33%	84.02%	84.83%

Comparison of heuristics based on the overall % nearer to LB.

CONCLUSION

The attempt has been very successful made to find a best neighbour in a sequence to minimize the makespan of a flowshop problem. The newly proposed IBB heuristic performed well in achieving the above objective. This work was evaluated through a set of benchmark problems in MATLAB environment, which concludes IBB heuristic is better about 4.2% and 0.5% when compared to CDS and Palmer heuristics. In further, the performance of IBB heuristic is to be increased by applying a advanced optimization technique like genetic algorithm.

Future Work: In further, the performance of IBB heuristic is to be increased by applying a advanced optimization technique like genetic algorithm. A case study is going to carried out in a manufacturing industry and test the effectiveness of the heuristic.

REFERENCES

1. Johnson, S.M., 1954. “Optimal Two and Three stage Production schedule with Setup Times Included, ” Naval Research Logistics Quarterly, 1(1).
2. Quan-Ke, Pan and Ling Wang, 2012. Effective heuristics for the blocking flowshop scheduling problem with makespan Minimization. OMEGA, 40(2): 218-229.
3. Palmer, D.S., 1965. Sequencing Jobs through a Multi-Stage Process in the Minimum Total Time – A Quick Method of Obtaining a near Optimum. Operations Research, 16: 101-107.
4. Campbell, H.G., R.A. Dudek and M.L. Smith, 1970. ”A heuristic Algorithm for the n- job m- machine Sequencing Problem”, management Science, 16(10).
5. Dannenbring, D.G., 1977. An Evolution of Flow-Shop Sequencing Heuristics. Management Science, 23: 1174-1182.
6. Stinson, Simith, D.T. and G.L. Hogg, 1982. A state of art survey of dispatching rules for manufacturing job shop operations. International Journal of Production Research, 20: 27-45.
7. Palmer, K., 1984. Sequencing rules and due date assignments in a job shop. Management Science, 30(9): 1093-1104.
8. Taillard, E., 1990. “Some efficient heuristic methods for the flow shop sequencing problem” EJOR, 47, pp: 65-74.
9. Glover, F., 1987. Tabu search. Preliminary draft, US West Chair in System Science, Center For Applied Artificial Intelligence, University of Colorado.
10. Franklin Issac, R., R. Pearlin Helina, T. Illakiya, S. Chitra, M. Yogalakshmi and M. Ishwariya, 2016. “Computational Inverse Branch and Bound Heuristic for Flowshop Problem”, International Journal of Innovative Research in Technology, Science & Engineering (IJIRTSE), 2(6): 65-72.
11. Pugazhenthi, R. and M. Anthony Xavior, 2014. ‘Minimizing Material Processing Time and Idle Time of a Critical Machine in a Flow Shop’, Advanced Materials Research, 984: 106-110.
12. Pugazhenthi, R. and M. Anthony Xavior, 2014. "A Characteristic Study of Exponential Distribution Technique in a Flowshop using Taillard Benchmark Problems." Pakistan Academy of Sciences”, pp: 187-192.
13. Schuster, C.J. and J.M. Framinan, 2003. “Approximate procedure for no-wait job shop scheduling”, Oper. Res. Lett., 31: 308-318.
14. Taillard, 1993. Bench Marks for Basic Scheduling Problems, European Journal of Operational Research, 64(2): 278-285.