International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 – 5161

Available at http://inpressco.com/category/ijcet

Research Article

New Heuristic for Job Shop Scheduling Problem

M. Vishnu Vardhan Reddy[†*], P. Ram Reddy[‡] and B. Dattatraya Sarma[†]

[†]Department of Mechanical Engineering, MLRITM, Dundigal, Hyderabad – 43, TS, India
[‡]Department of Mechanical Engineering, Mallareddy Engineering College, Hyderabad – 43, India
[†]Principal, Sri Venkateswara College of Engineering, Nellore, A.P., India

Accepted 13 Dec 2015, Available online14 Dec 2015, **Vol.5, No.6 (Dec 2015)**

Abstract

As the job shop scheduling problems are difficult to solve in combinatorial optimization, research has been devoted to developing efficient heuristics and applying meta-heuristics. Due to their complexity, most forms of the JSSP are formally classified as NP-hard problems. The major implication of this classification is that there is no known solution technique that is guaranteed to obtain an optimal schedule, other than searching through all possibilities. However, an exhaustive search is almost always impractical given the vast number of permutations for even relatively small JSSPs. Consequently, researchers have focused their efforts on developing heuristic techniques that aim to construct the best possible schedule in a limited amount of time. One such heuristic method, Lowest Operation Time, is proposed in the present paper to solve Job Shop Scheduling Problems. Here the attention is focused on the long wait FT10 bench mark problem suggesting a heuristic procedure to solve it in minimum number of iterations to arrive at the desired makespan also a new scheduling measure namely Average Flow Time is reported that helps in picking up the best heuristic.

Keywords: *Job-shop scheduling; Heuristic, priority, dispatching rules, makespan.*

1. Introduction

Sequencing and scheduling is a form of decision making that plays a crucial role in manufacturing and service industries. In the current competitive environment effective sequencing and scheduling has become a necessity for survival in market place. Companies have to meet shipping dates that have been committed to customers, as failure to do so may result in a significant loss of goodwill. They also have to schedule activities in such a way as to use the resources available in an efficient manner.

Scheduling deals with the allocation of scare resources to tasks over time. It is a decision making process with the goal of optimizing one or more objective functions. The resources and task in an organization can take many forms. The resources may be machines in a workshop, runways at an airport, crew at a construction site and so on. The tasks may be operations in a production process, take-offs and landings at an airport, stages in a construction project. Scheduling is the allocation of start and finish time to each particular task. Therefore scheduling can bring productivity in shop floor by providing a calendar for processing a set of tasks. It is nothing but scheduling various tasks on set of resources such that certain performance measures are optimized. There are three types of scheduling single machine scheduling, Flow shop scheduling, Job Shop Scheduling. Scheduling of tasks is very difficult in case of Job Shops and as such research attention has been focused on better solution methods to solve Job Shop Scheduling Problems.

1.1 Job Shop Scheduling.

All the figures In job shop problem, It is assumed that each job has m different operations. If some of the jobs are having less than m operations, required number of dummy operations with zero process times are assumed. By this assumption, the condition of equal number of operations for all the jobs is ensured. In job shop scheduling problem, the process sequences of the jobs are not the same. Hence, the flow of each job in job shop scheduling need not be unidirectional.

Each operation j in the operation sequence of the job i in the job shop problem will be described with triplet (i, j ,k) where k is the required machine for processing the j^{th} operation of the i^{th} job. The time complexity function of the job shop problem is combinatorial in nature. Hence, heuristic approaches are popular in this area.

1.2 Job Shop Scheduling Problems (JSSP)

A Job shop scheduling problem consists of a finite set J of n jobs $\{J_i\}_{i=1 \text{ to } n}$ to be processed on a finite set M of m machines $\{M_k\}_{k=1 \text{ to } m}$. Each job J_i must be processed on

*Corresponding author: **M. Vishnu Vardhan Reddy**

every machine and consists of a chain of operations $O_{i1}, O_{i2}, .., O_{im}$ which have to be scheduled in a predetermined given order(precedence constraint). Each machine can process only one job and each job can be processed by only one machine at a time (capacity constraints).The duration in which all operations for all jobs are completed is referred to as 'make span' and the amount of time each job spends in the system is known as 'flow time'. Many jobs in industry and elsewhere require completing a collection of tasks while satisfying temporal and resource constraints. Temporal constraints say that some tasks have to be finished before others can be started: resource constraints say that two tasks requiring the same resource cannot be done simultaneously. The objective is to create a schedule specifying when each task is being processed and what resources it will use which satisfy all the constraints while taking as little overall time as possible. This is the job shop schedule problem.

2. Literature Review

Job shop scheduling problems are N-P hard. Brucker and Gareystated this and thus getting solution of these types of problems are very difficult. Many heuristic approaches have been developed in the last decades by researchers to optimize programming problems of job shop schedules. Some of them are: despatch rules, genetic algorithm, artificial immune system, fuzzy logic, simulated annealing etc.

Brandimarte (1993) was the first to apply this heuristic method to solve job shop scheduling. BruckerCarlier and Pisonand devised branch and bound methods for the solution to small problems. To solve larger size problems Blazewicz developed effective local search method. The results in his method were found for at least one preferred program and which reduced the searching efforts.

In-Chan Choi aimed to develop local search algorithm to solve job shop scheduling problem. The objective function is to minimize makespan. Sequence dependent setup condition is added to the problem. Erscher et al. a branch and bound method with three parts. Step 1 is to calculate the lower limit, the step 2 is branching and step 3 is the removal of the node . Hurik, Jarisch and Thole (1999) and Dazere-Peres use different methods of tabu search for scheduling jobshop problems. Mastrolilli and Gambardella (2000) worked on FJSS's neighbourhood functions that can be used in Meta heuristic optimization techniques. This method performed better than any other method in terms of computational results and quality of the solution. D. A. Kooncenused data mining to find the programming model for the job shop scheduling problems. Propose of this work is to apply the methodology of data mining to explore the pattern. The problem objective is to minimize makespan. Genetic algorithm is used to generate a good solution. Data mining is used to find the relationship between sequences and predict the next job in sequence. The result of data mining can be used to summarize new rule which gives the result like result of the genetic algorithm. A. Tamilara used a new method for solving job-shop scheduling problem using hybrid Genetic Algorithm (GA) with Simulated Annealing (SA). This method introduces a reasonable combination of local search and global search for solving JSSP.

Chandrasekharan introduced three new dispatching rules for dynamic flow problem shop and job shop problem. The performance of these rules, compared with 13 sequencing rules. The case study is the simulation study for the problem of flow shop scheduling. The problems are modified again by random route jobs. The problems are changed shop scheduling problem programming problem flow workshop. The study can conclude that the performance of dispatching rules is being influenced by the routing of job and shop floor settings. Hiroshi used shift bottleneck procedure to solve the job shop scheduling problem. The problem objective is to minimize the total cost of holding. The specific restriction adds to the problem. The additional constraint is any limitation to the later work. The experiment shows that the bottleneck procedure can reduce the time change calculation. Anthony presented Memmetic algorithm for job shop for delay. The mean minimum and maximum delay between the start of operations. In this paper we present a framework for solving job shop scheduling problem based on a disjunctive graph to modify the problem and solve it by the Memmetic algorithm. rule which gives the result like result of the genetic algorithm.

Jansen scheduling problem solved job shop under the assumption that the jobs have controllable processing time. That means you can reduce the processing time of work by paying some cost. Jansen presented two models. The first is the continuous model and the other is model reduction. The test could prove manifest that both of them can solve the approximation scheme is fixed polynomial time when the number of machines and the number of operations. Job shop scheduling problem with minimizing Makespan is investigated.

Objective of Present Work

To find a feasible schedule of jobs on the machines and to calculate the Makespan and the Average Flow Time (AFT) for FT10 problem using LOT Heuristic developed.

3. Lowest Operation Time (LOT) Heuristic

To solve NP hard JSSP problems, no specific technique or tool but exhaustive search methods are available. Hence many heuristics were proposed to get near optimum solutions to JSSP. One such heuristic is proposed here named 'Lowest Operation Time'. One of examples of job shop problems that is shown in table .1 are to be solved by using LOT Algorithm.

Table 1 FT 10 Problem

Job (I)	Machine Sequence (II)									
A	1	2	3	4	5	6	7	8	9	10
B	1	3	5	10	4	2	7	6	8	9
C	2	1	4	3	9	6	8	7	10	5
D	2	3	1	5	7	9	8	4	10	6
E	3	1	2	6	4	5	9	8	10	7
F	3	2	6	4	9	10	1	7	5	8
G	2	1	4	3	7	6	10	9	8	5
H	3	1	2	6	5	7	9	10	8	4
I	1	2	4	6	3	10	7	8	5	9
Job (I)	**Processing Time (III)**									
A	29	78	9	36	49	11	62	56	44	21
B	43	90	75	11	69	28	46	46	72	30
C	91	85	39	74	90	10	12	89	45	33
D	81	95	71	99	9	52	85	98	22	43
E	14	6	22	61	26	69	21	49	72	53
F	84	2	52	95	48	72	47	65	6	25
G	46	37	61	13	32	21	32	89	30	55
H	31	86	46	74	32	88	19	48	36	79
I	76	69	76	51	85	11	40	89	26	74

3.1 Methodology of LOT Heuristic

At time t_0 it is assumed that all the machines are available and all jobs are candidates for scheduling. The procedure aims at finding the job with least early start time. If there is more than one candidate job with minimum early start time, i.e, in case of a tie the Lowest Operation Time rule is used to resolve the tie and the job with lowest operation time is chosen for scheduling. Suppose if job i is scheduled first on its stage1machine m_{i1}, since preemptions are not permitted, m_{i1} is occupied until the time $t = t_0 + p_{i1}$ (p_{i1} being the processing time of job i on m_{i1}). At the completion of first stage processing of this job, machine m_{i1} becomes available to process other jobs. Job i moves forward to machine m_{i2} where it immediately begins its stage 2 processing or enters a queue for m_{i2}. Once an operation is scheduled, the start time of the scheduled job for the next operation and ready time of the machine on which the operation is scheduled are recomputed each iteration.

3.2 Algorithm for Lowest Operation Time (LOT) Heuristic

1) Prepare a list of schedulable operations.
2) Identify the operation with lowest processing time.
3) Schedule the operation on the required machine.
4) Remove the scheduled operation from the list and add successor to the scheduled operation.
5) Repeat the process until all the operations are scheduled.
6) Determine the sequence of operations on all the machines and the schedule.
7) Obtain the Makespan and the Average Flow Time for the schedule.

One of examples of job shop problems that is shown in table .1 are to be solved by using LOT Algorithm.

3.3 FT 10 Problem

From the above example it was proved that the LOT heuristic can be applied for any job shop scheduling problem. So, we have chosen to solve 10 × 10 job shop problem shown in table 1 by using LOT method.

The description of the FT 10 problem shown in the above table is as follows:

1) There are 10 jobs namely A, B, C, …….., J and 10 machines 1, 2, ….,10.
2) The technological order of processing the jobs is given in column II i.e., Machine sequence. For example the routing of the job B is 1, 3, 5, 10, 4, 2, 7, 6, 8 and 9 and it can be read from the second row under the column Machine sequence as against the name of the job i.e., B.
3) The III column i.e., Processing time denote the operation time required on the machine at a particular stage. For example processing times of job B during various stages are 43, 90, 75, 11, 69, 28, 46, 46, 72, 30 i.e., job B requires 43 time units on machine 1 and 90 time units on machine 3 and so on.

The objective is to find a feasible schedule of jobs on the machines and to calculate the Makespan and the Average Flow Time (AFT).

A program is developed using C language to solve the above FT10 problem using LOT Algorithm.

4. Results and Discussion

The program is so designed to get the desired output, iteration-wise, giving the schedule to arrive at the optimum result.

Input Data for FT 10 Problem

Enter the number of Jobs and Machines
10 10
Enter the number of operations for job A
10
Enter Machine sequence for job A 1 2 3 4 5
6 7 8 9 10
Enter process times for job A 29 78 09 36 49
11 62 56 44 21

Enter the number of operations for job B
10
Enter Machine sequence for job B 1 3 5 10
4 2 7 6 8 9
Enter process times for job B 43 90 75 11
69 28 46 46 72 30
Enter the number of operations for job C
10

Enter Machine sequence for job C
2 1 4 3 9 6 8 7 10 5
Enter process times for job C
91 85 39 74 90 10 12 89 45 33

Enter the number of operations for job D
10
Enter Machine sequence for job D
2 3 1 5 7 9 8 4 10 6
Enter process times for job D
81 95 71 99 09 52 85 98 22 43

Enter the number of operations for job E
10
Enter Machine sequence for job E
3 1 2 6 4 5 9 8 10 7
Enter process times for job E
14 06 22 61 26 69 21 49 75 53

Enter the number of operations for job F
10
Enter Machine sequence for job F
3 2 6 4 9 10 1 7 5 8
Enter process times for job F
84 02 52 95 48 72 47 65 06 25

Enter the number of operations for job G
10
Enter Machine sequence for job G
2 1 4 3 7 6 10 9 8 5
Enter process times for job G
46 37 61 13 32 21 32 89 30 55

Enter the number of operations for job H
10
Enter Machine sequence for job H
3 1 2 6 5 7 9 10 8 4
Enter process times for job H
31 86 46 74 32 88 19 48 36 79

Enter the number of operations for job I
10
Enter Machine sequence for job I
1 2 4 6 3 10 7 8 5 9
Enter process times for job I
76 69 76 51 85 11 40 89 26 74
Enter the number of operations for job J
10
Enter Machine sequence for job J
2 1 3 7 9 10 6 4 5 8
Enter process times for job J
85 13 61 07 64 76 47 52 90 45

Table 2 Output

E 1 3	E 2 1	E 3 2	A 1 1	H 1 3	B 1 1	G 1 2	G 2 1	E 4 6	E 5 4
G 3 4	G 4 3	G 5 7	G 6 6	G7 10	E 6 5	E 7 9	E 8 8	E 9 10	E 10 7
I 1 1	I 2 2	I 3 4	I 4 6	A 2 2	A 3 3	A 4 4	A 5 5	A 6 6	A 7 7
A 8 8	A 9 9	A1010	D 1 2	F 1 3	F 2 2	F 3 6	I 5 3	I 6 10	I 7 7
J 1 2	J 2 1	J 3 3	J 4 7	J 5 9	J 6 10	J 7 6	J 8 4	H 2 1	H 3 2
H 4 6	H 5 5	H 6 7	H 7 9	H8 10	H 9 8	H10 4	G 8 9	G 9 8	G10 5
I 8 8	I 9 5	I 10 9	B 2 3	B 3 5	B4 10	B 5 4	B 6 2	B 7 7	B 8 6
B 9 8	B10 9	J 9 5	J10 8	C 1 2	C 2 1	C 3 4	C 4 3	C 5 9	C 6 6
C 7 8	C 8 7	C 9 10	C10 5	D 2 3	D 3 1	F 4 4	F 5 9	F 6 10	F 7 1
F 8 7	F 9 5	F 10 8	D 4 5	D 5 7	D 6 9	D 7 8	D 8 4	D9 10	D10 6

Table 3 Make span, Iterations, Average Flow Time of various methods

	Make span	**Iterations**	**Average Flow Time**
General SB routine	1094	400	1057.1
Shifting bottle neck/wt	1174	600	753.6
Shifting bottle neck/T_{max}	1340	1100	849.4
Local search method	940	6100	823.4
Lowest operation time	1291	100	864.9
Dash method	930	1100	869.4

The table 2 output of the program shows scheduling of jobs in sequence. It includes names of jobs, operation number and machine number. For example, E 1 3 can be defined as the first operation of job E is run on machine3. The output of the program also includes the schedule of operations on each machine with job name followed by start finish times of the operation. In addition to the schedule of operations on each machine the output also indicates the Makespan, Average Flow Time for the schedule.

Conclusion

The proposed LOT algorithm is capable of solving job shop problems. This is significant because of the way this optimization progressed, through the division of search space by the machines in the problem. We are not aware of any other researcher taking this approach in the job shop realm. At first thought, it doesn't seem like much more than a big combinatorial problem, but after probing deeper, it is easy to get lost in it, especially the different classifications of schedules and the different ways to search them. What's interesting about this problem is its history and stubbornness to remain difficult to solve even with modern computing. Table 3 Make span, Iterations, Average Flow Time of various methods

By observing table 3 we had found that,

- In general SB routine, Makespan is decreased at the cost of more iterations but the Average Flow Time is increased.
- In shifting Bottle Neck/wT, Makespan is decreased at the cost of more iterations.
- In Shifting Bottle Neck/T_{max}, Makespan is increased with increase in iterations and Average Flow Time is decreased.
- Local Search Method also follow the trend of Shifting Bottle Neck/wT but the iterations are around 6100.

Dash method also takes more iteration to get optimum Makespan and Average Flow Time. By the above results , we conclude that the algorithm i.e., Lowest Operation Time got slight increase of optimum Make span and Average Flow Time in very less number of iterations. By this we can say that Lowest Operation Time Heuristic takes very less time and very less iteration to attain optimum Make span and Average Flow Time and it can be applied wherever we want less number of iterations.

References

Brucker P., 1995. Scheduling algorithms 2nd edn. Springer. Berlin Heidelberg New York.

M. R. Garey, D. S. Johnson, Ravi Sethi. *The complexity of flow shop and Job shop scheduling.* M*athematics* of *operations research*. Vol. I. No. 2. May. 1976, pp. 117-119.

Brandimarte P.,(1993). Routing and scheduling in a flexible job shop by tabu search, *Annals of operation research*, 41(3), pp 157-183

Peter Brucker, Bernd Jurisch, Bernd Sievers (1994) A branch and bound algorithm for Jobshop problem. *Discrete Applied Mathematics*, 49(1-3):107-127

Carlier J., Pison E., (1989).An algorithm for solving Job shop problem. *Management Science*, 35(2): 164-176

Adams J, E Balas , D Zawack .,(1988), The Shifting BottleNeck Procedure for Job Shop Scheduling, Management science, 34(3), pp. 391-401.

Blazewickz J., (1996). The Job shop scheduling Problem: Conventional and new solutions techniques. *European Journal of Operational Research,* 93,pp, 1-33.

Choi I. and Choi, D., (2002). A local search algorithm for job shop scheduling problems with alternative operations and sequence dependent setups. *Computer and industrial engineering*, 42(1), 43-58

Erscher J.F., Roubellat J.P., Verntes , (1976). Finding some essential characteristics of the feasible solutions for a scheduling problem. *Operational research*, 24(4), pp. 774-783

Heurik E., Jarisch B. and Thole M., (1994). Tabu search for the job shop scheduling problem with multipurpose machines. *Operation research Spectrum*, 15, pp. 205-215

Mastrolli M. and Gambardella C. M.,(2000). Effective neighborhood functions for the flexible jobshop problem. *Journal of scheduling*, 3(1), pp. 3-20.

Koonce D. A. and Tsai S. C., (2000). Using data mining to find patterns in genetic algorithm to a job shop schedule, *Computer and industrial engineering*, 38(3), pp. 361-374.

Holthaus O. and Rajendran C., (1990). A comparative of study of dispatching rules in dynamic flow shops and job shops. *European journal of operational research* , 116(1), pp. 156-170.

Ohta H. and Nakatani T., (2006). A heuristic job shop scheduling algorithm to minimize total holding cost of completed and in progress products subject to no tardy job. *International journal production economics*, 11, pp. 19-29.

Caumond A., Lacomme P., and Tchernev N.,(2008) A memmetic algorithm for job shop with time lag. *Computer and operational research*.35(7), PP. 2331–2356

Jansen K., Mastrolilli M. and Oba R.S., (2005). Approximation scheme for job shop scheduling problem with controllable processing time. *European journal of operational research*, 1672, pp. 97-319.