ISSN 1996-3386

Asian Journal of Industrial Engineering

http://knowledgiascientific.com

Knowledgia SCIENTIFIC
A Place to Publish Outstanding Research

Asian Journal of Industrial Engineering 2 (1): 9-16, 2010
ISSN 1996-3386

Deep Memory Greedy Search and Allocation of Job Shop Scheduling For Optimizing Shop Floor Performance

[1]T. Hemamalini, [2]A.N. Senthilvel and [3]S. Somasundaram
[1]Department of Mathematics, Tamilnadu College of Engineering, Karumathampatty, 641014, Tamilnadu, India
[2]Department CSE, Coimbatore Institute of Technology, India
[3]Department Mathematics, Coimbatore Institute of Technology, Coimbatore-641014, India

Abstract: The objective of this study was to improve shop floor performance through proper allocation of jobs in machines taking due time into consideration. In general, a constructive optimization method tries to give the best possible solution. On the other hand, an iterative process improvises an assumed initial solution, which may take a long time, since we do not know which would give the optimal solution in a stipulated time frame. Hence, we used the constructive method of optimization, barring the time taken to get the solution. A novel Deep Memory Greedy Search (DMGS) method is proposed to solve the above problem. The simulation run is conducted for different combination of jobs and machines with job processing time and results are illustrated. The performance charts are also plotted for visual analysis of the problem solving instances.

Key words: Parallel machine scheduling, job shop, simulation, deep memory greedy search

INTRODUCTION

In the Job Shop Scheduling Problem (JSP), a finite set of jobs is processed on a finite set of machines. Each job is characterized by a fixed order of operations, each of which is to be processed on a specific machine for a specified duration. Each machine can process at most one job at a time and no preemption is allowed. A schedule is an assignment of operations to time slots on the machines. Panneerselvam (2002) studied optimized scheduling leads to increased efficiency and capacity utilization, reducing the time required to complete jobs and consequently increase the profitability of an organization.

The JSP is NP-hard discussed by Lenstra and Kan (1979) has continuously challenged to computational researchers. As the due date approaches processing speed of the jobs can not be altered as discussed like in the Ferris and Valch (1992). This study is concerned with planning the sequence of operation for the components manufactured in CNC machining centres.

A factory processes M jobs in N machines. Each machine has M! alternatives for job sequencing. Hence $(M!)^N$ alternatives in total.

Corresponding Author: T. Hemamalini, Department of Mathematics, Tamilnadu College of Engineering, Karumathampatty, 641014, Tamilnadu, India

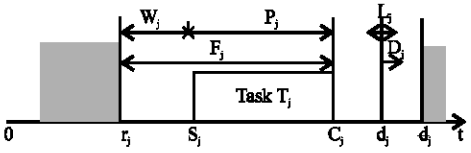


Fig. 1: Graph plot of job task parameters

Koulamas (2006) considered the two machine scheduling problem with setup times that are proportionate to the length of the already scheduled jobs, which they call past-sequence dependent setup times. They showed that the single machine problem with the objective functions of makespan, total completion time, total absolute differences in completion times and a linear combination of the last two objective functions can be solved in O (n long n) time by a sorting procedure.

Scheduling Criteria

A basic role of operations scheduling is to generate an optimal schedule. Such a scheduling decision should be made according to a certain measure of performance, or scheduling criterion is elaborated by Sule (2000) and Hitomi (2002):

- Maximum flow time or Makespan (C_{max}); Mean flow time ($\bar{F}$); Maximum lateness or tardiness (L_{max}) or (D_{max}); Mean lateness or tardiness ($\bar{L}$) or ($\bar{D}$); Number of tardy jobs (m); Average number of work-in-process jobs (or inventory); Facility utilization in the workshop. The last measure is maximized; others are minimized

In Fig. 1 where, P_j: Processing Time, S_j: Start tome of j^{th} job, r_j: Release time, d_j: Dead line, $\dot{d}_j$: Due date C_j: Completion time = Sj+Pj, W_j: Waiting time = S_j+r_j, F_j: Flow time = C_j-r_j, L_j: Lateness = C_j+d_j ; D_j: Tardiness = D_j = max{C_j-dj,0}.

Parallel Machine Scheduling

Parallel Machine Scheduling (PMS) is used to schedule jobs processed on a series of same function machines, with optimized objective. Suppose that N jobs J_i (i = 1, 2,…, N) have to be processed on M machines M_i (I = 1, 2,…, M). The M same-function machines are called identical, uniform and unrelated parallel machines if the processing requirement P_{ij} satisfies P_{ij} = P_j for any M_i, P_{ij} = P_j/S_i for a given speed S_i of M_i and P_{ij} = P_j/S_{ij} for a given job-dependent speed S_{ij} of M_i, respectively. Balas *et al.* (2005) formulated the job shop sequence dependent setup time problem as an asymmetric traveling salesman problem with a special type of precedence constraints, which can be solved by a dynamic programming algorithm whose complexity is linear in the number of operations. The parallel machine scheduling is an allocation of one time interval on one of the machines.

SCHEDULING

Job Scheduling with Deadlines and Profits

Assume that there are n jobs, each of which takes unit time and a processor on which we would like to schedule them in as profitable a manner as possible. Each job has a profit associated with it, as well as a deadline; if the job is not scheduled by its deadline, then we don't get its profit. In many cases of parallel machine scheduling problems, customers want

their jobs to be completed on or before their due dates. If some jobs are finished earlier than their due dates, the producer will incur an inventory cost that is called earliness penalty. If some jobs are finished later than their due dates, the producer will incur a tardiness penalty. The research carried out by Cheng (1987), Ferris and Vlach (1992) and Pinedo and Chao (1999) gave great contribution to the understanding of the due date problems. The problem of Single Machine Setup Time sequence independent with the objective of minimizing the tardiness considered by Gupta and Schaller (2006), proposed a branch-and-bound algorithm for the problem with and without the group technology assumption. He also proposed a heuristic to solve larger-sized problems. His computational experiments revealed that total tardiness can be significantly reduced by removing the group technology assumption. He solved problems with up to 10 families and 20 jobs in each family. Schaller *et al.* (2000) studied the same problem with the objective of minimizing the earliness and tardiness by following the same procedure.

C_{max} Minimization about Common Due Date

The problem of determining the optimal due-dates and optimal sequence of n independent jobs that are to be processed by a single machine. Each job is assigned a common due date and penalties will be incurred when a job is finished either early or late. The objective is to determine the optimal common due-date value k* and the optimal job sequence σ^* to minimize the maximum deviation of job completion time about the common due-date.

Prime Objective Mathematical Model

$$f(s) = \sum_{j=2}^{m} P_1 (C_{ij}-D_{ij}) + \sum_{j=2}^{n} P_2 (D_{ij}-C_{ij}) \qquad i=1 \qquad i=1$$

Non Pre Emptive /Parallel Machines />2/C_{max}

Where:

f (s) = Net penalty function

*P_1 = Penalty cost per unit time of late jobs

*P_2 = Penalty cost per unit time of early completion of jobs

C_{ij} = Completion time of ith job in jth machine

D_{ij} = Due date of ith job in jth machine. The objective of the above function (f (s)) is to find a schedule to minimize the penalty

Proposed Methodology

Deep Memory Greedy Search Method

Erwin (2004) discussed a optimization as a broadcast domination analysis assigns a value f (v) to a vector v, such that its distance from every vertex of the graph is with in f (v), where f (v) is optimized. Analogy to the above problem we assign a position p (x) for a process x, such that its penalty, on delaying it, gets reduced. In this method we attempt to construct an optimum solution in stages. At each stage we make a decision that appears to be the best (under some criterion) at that time. A decision made in one stage is not changed in a later stage, so each decision should assume feasibility. Best is taken right now, without regard for future consequences. A deep memory greedy way to an optimal task assignment is to assign the tasks in stages.

Deep Memory Greedy Search Algorithm

Step 1 : Start
Step 2 : Dynamically allocate the array for machines and jobs. Initialize the minimum number with a very large number
Step 3 : Insert the processing time and target (due) times into respective vector array
Step 4 : Find the difference array based on smallest negative element to be the first element in the array. Quick sort algorithm is used
Step 5 : While (number of jobs =0) repeat the following steps 6-9
Step 6 : while (number of not allocated machine › allocated machines) repeat steps 7-8
Step 7 : if (minimum number › (so far allocated processing time + new job time to be allocated. Store the result as minimum number and note the particular machines)
Step 8 : Then the new minimum number will be equal to so far allocated processing time + new job time to be Allocated. Store the minimum number for the particular machine
Step 9 : Allocate the job for the particular machine and reduce the number of jobs to be allocated by one
Step 10 : End of allocation

Assume if there are two machines and ten jobs. First two assignments are simply the first two jobs from the sorted array. Remaining assignments are calculated based on the availability of the particular machine.

SOFTWARE AND SIMULATION

A software program is developed to simulate the scheduling for n number of jobs and m number of machines using deep memory greedy search technique. The code is created using JAVA Language. In this software the values C_{max}, % machine utilization, total penalty, tardiness and machine wise job allocation can be determined for the given problem. Comparison can be done for the best C_{max}, tardiness values, machine utilization and idle time of the individual machine. The graphical representation of the scheduled jobs in the machines is also done in the software.

Objective of the Simulation

- Minimize the total flow time or makespan (C_{max})
- Minimize tardiness
- Increase % machine utilization
- Reduction of total penalty cost
- Minimize the number of tardy jobs

INFERENCE

A test instance of 4 cases had been made and the results are tabulated. Table 1 describes about the input parameters with different criteria. Table 2 shows the detail about the actual data considered for the algorithm. Simulation output illustrates how the jobs

Table 1: Analysis criteria and input

Input given	Due date analysis
No. of Jobs-25,	Criteria I: Due date as per company requirements (DC_oR)
No. of Machines-5,	Criteria II: Due date as per customer requirements (Dc_uR)
Processing Time and Due Time	Criteria III: Common Due Date (CDD)

Table 2: Processing time and requirement

Job ID	Code	Material	Total processing time (min)	DC_oR	DC_uR	CDD
WSR	D1401	CastIron	22.2	30	35	28.6
BRG COV	D4041	Stainless steel	23.8	14	19	28.6
CAP	BS101	Brass	31.0	28	33	28.6
NIPPLE	BS012	Brass	43.2	30	35	28.6
SQ.BLOCK	CI101	Cast Iron	28.2	16	21	28.6
DUMMY	CI201	Cast Iron	21.6	16	21	28.6
PLUG CASE	AL401	Aluminium	26.1	14	19	28.6
REV BOSS	ALS81	Aluminium	20.2	14	19	28.6
STEM CAP	AL514	Aluminium	28.9	42	47	28.6
BOARD RING	CI118	CastIron	37.8	38	43	28.6
STEM RING	AL611	Aluminium	42.9	30	35	28.6
GUARD BASE	G1001	JL Alu	37.1	15	20	28.6
GUARD ACT	G1002	LM07	27.8	15	10	28.6
LEVER BUSH	S1124	C45	28.4	60	55	28.6
BUSH RING	O112	GM	25.4	60	55	28.6
ACTUATOR CLIP	K141	T- Steel	33.7	35	30	28.6
STEM LOCK	K147	C45	51.0	35	30	28.6
LOCK RING	O1121	Cast Iron	21.9	45	40	28.6
INNER LEVER	J1124	C45	34.0	55	50	28.6
STUD BASE	J1127	C45	34.2	32	27	28.6
PULLY CRAD	S111	C45	30.0	30	25	28.6
SOFT PIN	S112	Brass	44.0	16	11	28.6
JELLY ROD	J121	Cast Iron	25.8	12	7	28.6
DOUBLE CR	D1001	C45	17.0	10	5	28.6
SL CAP	S1125	Cast Iron	16.0	22	17	28.6

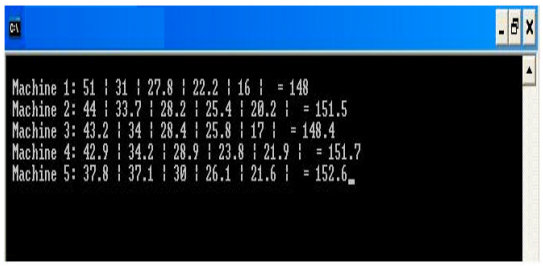


Fig. 2: Machinewise allocation for 5M25J Problem. Simulation output for calculation of C_{max}

are allocated to each machine and the utilization time of each machine. Figure 2 shows the output screen shot of the simulation for machinewise allocation of 5 machines and 25 jobs problem. Figure 3-5 are the comparison of gantt chart for due date as per company requirements (DC_oR), due date as per customer requirements (DC_uR) and Common Due Date (CDD), respectively. Analysis provides better results compared to the results with Cheng and Chen (2006). Table 3 shows the results obtained using deep memory greedy search algorithm for 4 different cases with 3 different due date criteria is compared for C_{max}, Total penalty and % machine utilization. Table 4 shows the comparison of C_{max}-lower bound values for the 4 cases considered with 3 due date criteria's. For the common due date case the difference is minimum compared to other two cases. Since, the

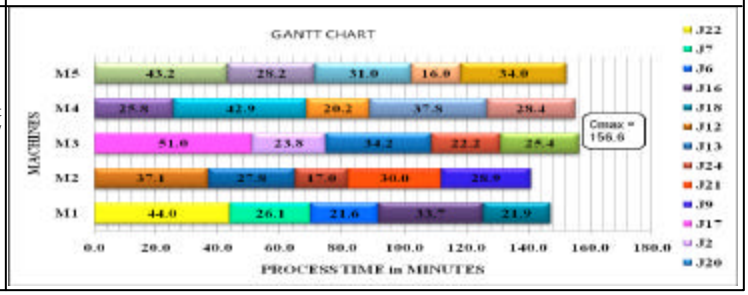
Due date criteria	DC_0R
Gantt chart (PM/$5/m/n)	

Fig. 3: Scheduling as per the company requirement

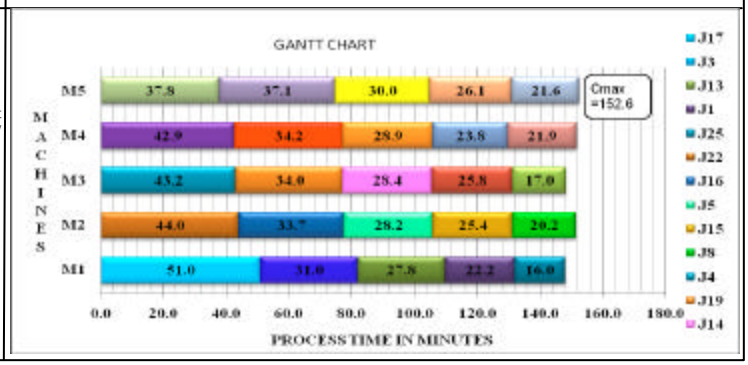
Due date criteria	CDD
Gantt chart (PM/$5/m/n)	

Fig. 4: Scheduling as per the customer requirement

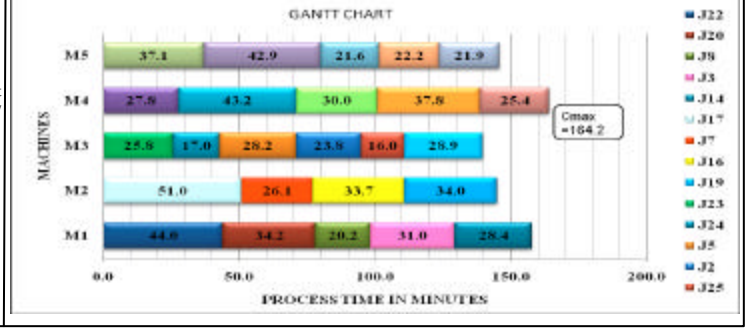
	DC_0R
Gantt chart (PM/$5/m/n)	

Fig. 5: Scheduling as per the common due date requirement

Table 3: Comparison for criteria I, II and III with objective function

Objectives/due date criteria	DC_oR	DC_uR	CDD
C_{max}	156.6	164.2	152.6
Tardiness			
Min	13.2	17.2	9.2
Max	102.3	109.2	124.0
% machine utilization			
Min	89.9	85.1	97.0
Max	100.0	100.0	100.0
Net penalty	1665.0	1647.0	1823.0
Lower bound LB	150.4	150.4	150.4
C_{max}-LB	6.2	13.8	2.2

Table 4: C_{max} value comparison for criteria I, II and III with different cases

Cases	Case I N = 25 M = 5	Case II N = 25 M = 4	Case III N = 25 M = 7	Case IV N = 25 M = 9
Due time (company)	156.6	200.5	116.2	97.2
Due time (Customer)	164.2	195.6	119.6	91.5
Common due time	152.6	197.9	116.5	87.1

Table 5: % machine utilization (minimum of all machines)

Cases	Case I N = 25 M = 5	Case II N = 25 M = 4	Case III N = 25 M = 7	CASE IV N = 25 M = 9
Due time (company)	89.9	88.0	81.4	75.2
Due time (customer)	85.1	91.2	79.8	77.4
Common due time	97.0	93.0	82.9	81.2

Table 6: Tardiness comparison for criteria I, II and III with different cases

Cases	Case I N = 25 M = 5	Case II N = 25 M = 4	CAse III N = 25 M = 7	Case IV N = 25 M = 9
Due time (company)	102.3	140.5	66.3	45.1
Due time (customer)	109.2	140.6	64.6	49.0
Common due time	124.0	169.3	87.9	58.5

Table 7: Comparison of C_{max} with benchmark values

	Makespan	
Test instances	Benchmark	Obtained
5M25J	228.08	230.60
5M30J	302.30	303.20
9M25J	126.71	128.50
9M30J	167.94	179.70
12M25J	95.03	99.20
12M30J	125.96	128.10
16M25J	71.28	99.20
16M30J	94.47	101.20
20M25J	57.02	99.20
20M30J	75.58	99.20
25M50J	98.29	107.40
30M50J	81.91	99.20
30M100J	173.53	186.70

difference is minimum we can get the optimum schedule of allocation of jobs for the given machines. Table 5 provides the percentage of machine utilization (minimum of all machines) by keeping the number jobs as 25 and varying the number of machines as 4, 5, 7 and 9. Table 6 shows the comparison of Tardiness for DC_oR, DC_uR and CDD. When number of jobs are less DC_oR outperforms DC_uR. Table 7 compares the C_{max} values obtained through Deep Memory Greedy search algorithm with Benchmark values. Hence the algorithm proves to be a better for finding optimal solution for job shop scheduling problems.

The approach was promising one because of its generality in nature and its effectiveness in finding very near optimal solution to the difficult job shop scheduling problems.

CONCLUSIONS

Thus, it is concluded from the results obtained that Deep Memory Greedy search method of allocation proves to be a better heuristic search algorithm to solve complex job shop scheduling problem. It had been proved through different case analysis that Deep Memory Greedy search method of allocation is one of the near best algorithms for solving combinatorial optimization problems.

The results show that Deep Memory Greedy Search Algorithm can be used to resolve industrial scheduling problems, considering variation in input data from time to time. With the idea that Deep Memory Greedy search method of allocation algorithm has a uniform probability distribution over local optima and global optima and its better usage for multi-objective problems, it can be used to resolve Multi-objective shop floor problems.

REFERENCES

Balas, E., N. Simonetti and A. Vazacopoulos, 2005. Job shop scheduling with setup times, deadlines and precedence constraints. Proceedings of the 2nd Multidisciplinary International Conference on Scheduling: Theory and Applications, July 18-21, New York, USA., pp: 520-532.

Cheng, T.C.E., 1987. Minimizing the maximum deviation of job completion time about a common due-date. Comp. Mathematics Appl., 14: 279-284.

Cheng, T.C.E. and Z.L. Chen, 2006. Parallel machine scheduling problems with earliness and tardiness penalties. J. Operat. Res. Soc., 45: 685-695.

Erwin, D.J., 2004. Dominating broadcasts in graphs. Bull. Inst. Comb. Appl., 42: 89-105.

Ferris, M.C. and M. Vlach, 1992. Scheduling with earliness and tardiness penalties. Naval Res. Logistics, 39: 229-245.

Gupta, J.N.D. and J.E. Schaller, 2006. Minimizing flow time in a flow-line manufacturing cell with family setup times. J. Operat. Res. Soc., 57: 163-176.

Hitomi, K., 2002. Manufacturing System Engineering. 2nd Edn., Viva Books press, Delhi, India, ISBN: 9788185617886, pp: 1-1.

Koulamas, C.P., 2006. Scheduling two parallel semiautomatic machines to minimize machine interference. Comput. Operat. Res., 23: 945-956.

Lenstra, J.K. and A.H.G.R. Kan., 1979. Computational complexity of discrete optimization problems. Ann. Discrete Mathematics, 4: 121-140.

Panneerselvam, R., 2002. Production Operations Management. Prentice Hall Private Ltd., New Delhi, ISBN: 8120327675, pp: 270–330.

Pinedo, M. and X. Chao, 1999. Operation scheduling: With applications in manufacturing and services. Prentice Hall, New Jersy, ISBN: 0-07-289779-1, pp: 154-178.

Schaller, J.E., J.N.D. Gupta and A.J. Vakharia, 2000. Scheduling a flowline manufacturing cell with sequence dependent family setup times. Eur. J. Operat. Res., 125: 324-339.

Sule, D.R, 2000. Industrial Scheduling. PWS Publishing Co., London, ISBN: 1420044206, pp: 45-57.