Available online at www.ijmcmr.com
Trigin Publisher

Int. J. of Mechanical Computational and Manufacturing Research, Vol.1. No.1, (2012), 12-18, ISSN: 2301-4148

Tool and Job Scheduling in a Tool Constrained Enviroment Using Heuristic Approach

Ananda Kumar Eriki[1] and V. Diwakar Reddy[2]

[1]*School of Engineering, Science and Technology, Nilai University College, No. 1, Banda Baru Nilai, 71800, Putra Nilai, Negri Sembilan, Malaysia,* E-mail: *erikiananda@yahoo.co.in*

[2]*Faculty of Mechanical Engineering, Sri Venkateswara University College of Engineering, Tirupati, 517502, Chittoor (Dist), Andhra Pradesh, India. E-mail: vdrsvuce@gmail.com*

Abstract: This paper deals with finding the optimal schedule of a Job Shop Scheduling problem. Scheduling is one of the most important issues in the planning and operation of manufacturing system. The aim of this paper is to find the optimal scheduling of the Job Shop Problem consisting of ‘N’ jobs with ‘M’ machines and ‘T’ tools. Each job needs different types of machines and tools for the execution of sequence of operations. Tools are stored in a common tool magazine (CTM) in an automated Job Shop system. In this study addresses with a scheduling algorithm like Branch and Bound Algorithm (B&B) and Heuristics approaches like Simulated Annealing Algorithm (SAA). The above algorithms adopt the procedure of Extended Giffler and Thompson Algorithm (EGTA) for active feasible schedules and adopt different techniques for optimal schedule. To reduce the total job completion time minimum makespan (MS) time is used here as the objective function. Branch and Bound method is used here to find the optimal schedule of jobs that can meet minimum makespan time for tool constrained automated job shop system. Simulated annealing algorithm is used to solve the large size problems also and the performance of this algorithm is compared with branch and bound algorithm for known results, thereby the optimum schedule can be found.

Keywords: Tool and Job Scheduling, Common tool magazine (CTM), Automated Job Shop (AJS), Branch and Bound (BB), Heuristic Approach, Scheduling Environment.

1 INTRODUCTION

Production is a transformation process that converts raw material into finished products that have value in the market place.The products are made by a combination of manual labor, machinery; tools and energy. The transformation process made by a combination of a sequence of steps is referred to as production operation, such as continuous, mass, batch and job shop production. Rigid lines are now a day progressively replaced by flexible cells of FMS. Many industrial problems, in particular the scheduling are very complex in nature and quite hard to solve by convention optimization techniques, various algorithm have been employed to solve scheduling problems are heuristic evolutionary and artificial intelligent technique. To solve scheduling problem in FMS/Automated Job Shop scheduling we used Giffler and Thompson based “Branch and Bound Technique and “Simulated Annealing Algorithm” is the numerical arrival of procedure to arrive at an active feasible schedule. Scheduling is said that an inch of gold cannot buy an inch of time.

This saying highlights the importance of time to induvidual as well as organizations. A competitive organization does things ahead of other. That is also the quality of an organization which is a leader in its business. The essence of scheduling is that it helps an organization to use the available time and resources effectively and efficiently. Scheduling is very important to reduce idle time, improve job completion time and many such desirable criteria. The general job shop consits of ‘m’ number of machines each different from all others and ‘n’ number of jobs each different from all other, determining an efficient schedule for the general job shop problems has been the subject of research for more than 50 years. For ‘N’ jobs and ‘M’ machines in the general case there will be $(N)^M$ feasible sequences. Scheduling problems are known to be complex even for simple formulation. Hong-Bae [1] has developed a tool provisioning problem and suggested direction to schedule parts and tools simulateneously to maximize the overall system performance under a given budget constraint.

M Selim Akburk [2] has proposed optimization procedure (Geometric programe) for solving joint lot sizing and tool management problems simulatneously to minimize the total production cost. Scheduling problem can be described by four types:

1. The job and operations to be performed
2. The number and type of machine that comprise the shop
3. Conditions that govern the maner in which the

Corresponding author: erikiananda@yahoo.co.in (Ananda Kumar Eriki)

ICID: 1049218 (Index Copernicus)

assignment can be made
4. The criteria by which a schedule will be evaluated.

Problems differ in number of jobs that are to be processed, the manner in which the jobs arrive at the shop and in the order in which the different machines are required to process the operation of individual jobs. The nature of jobs arrivals are classified in to two categories:

1. Static Arrival
2. Dynamic Arrival

In a static problem a certain number of jobs arrive simultaneously in a shop that can immediately undergo processing. No further jobs are expected, this is called static arrival. In dynamic problem certain number of jobs can be expected even during the processing time of other jobs. In many research works, assumption was made to have individual tool magazines for each machines leads to more expenses. In such situations many tools will be idle most of the time. To cut down the tool expenses, and to have optimum tool utilization, a common tool magazine with several machines can be employed.

The concept of common tool magazine that shares with and serves for several machining center reduces the cost of duplicating tools in each and every machining center is of particular interest to computerized automated manufacturing system. Further, the loss of productivity due to some machines being idle (waiting for tool availability) may be reduced by means of proper scheduling under shared tool magazine, this type of manufacturing system is necessary for manufacturing environment in which tools are particularly expensive like Tool Room Scheduling etc.

Common Tool Magazine

The concern of this paper is to generate joint job/tool schedule in automated job shop manufacturing system/FMS consisting of several machines and a common tool magazine. Group Technology (GT) algorithm is extended to provide one active feasible schedule by resolving arbitrarily machine conflicts and tool conflicts choosing any one of the conflicts during stages of conflicts. It is not practicable to enumerate all the above feasible schedules and mathematical model computations will be time consuming. The apparent difficulties encountered with mathematical model computations have forced practitioners to seek alternative computational method. The solution techniques for such computationally complex scheduling problems concentrate on branching and search heuristics.

Sequencing and Scheduling

Job is started first on some machine or work center, is only the ordering of jobs, scheduling is ordering of operations on all the machines of the job shop simultaneously as well as synchronizing them over the schedule period without overlapping of successor and predecessor operations. Scheduling is the allocation of resources overtime in order to perform a set of operations and meet certain objectives while respecting a set of constraints [7].

Scheduling problems are identified to be having four levels of decisions”

1. Task determination: How much to do?
2. Allocating facility: Assignment
3. Sequencing: In which order to do?
4. Time tabling: When to do?

2 OBJECTIVE

The objective of this paper is to utilize the available resources effectively with makespane (MS) minimization as performance criteria. Production environment that improves the productivity will increase the performance and efficiency of the whole production system. The performance can be measured on many aspects such as hardware utilization, tardiness, in-process inventory, makespane time, lateness.

Assumptions:

1. Each machine/tool can process only one job at a time
2. Each machine is continuously availabe to perform operation
3. Each job contains one or more operations to be performed in a particular order. Change of operation sequence is not permissible
4. Each operation can be performed only by the machine identified for it and onot by any other machine
5. There is only one machine of each type
6. Only one type of tool is available for each variety
7. The processing times of successive operation of a particular job may not be overlapped. A job can be in process on at most one operation at time
8. Each operation once started, must be completed (Interruption is not allowed)
9. The processing time of a job at each facility includes loading, unloading, changover and setup time of a job and tool
10. The job visiting the same machine for more than one time is not allowed

Problem Specification

Generation of opitimal joint operation-tool schedules of ‘N’ different jobs that require processing on ‘M’ different machines with ‘T’ tools, which are shared for many operations, and stored in a Comon Tool Magazine with minimum makespan criteria.

Operating environment

In an automated job shop manufacutring system there are 'N' numbers or different type of jobs to be processed in 'M' number of machines with different tools which are stored in a tool magazine [4]. Each mahcine needs a separate tool magaine which consits of 'T' number of tools, in such situation:

1. each machines leads to more expenses because of duplication tools
2. many tools will be idle most of the time

To cut down the tool expenses and to have optimum tool utilization, CTM with several machines can be employed, in such situation:

1. tools are stored in a CTM, and are moved throughout the machines by means of tool handling system.
2. Conflict may arise whenever the same tool or machine is required by two or more jobs at the same time

In these circumstances, one of the jobs will be served first and others have to wait for the release of tool or machine.

3 HEURISTICS APPROACH

3.1. Branch and Bound Technique (BBT)

Branch and Bound methodology for pruning the conflict resolution tree of active feasible schedules for the optimal schedule, instead of resolving all the conflict one by one and searching for the optimal with the Giffler and Thomson algorithm [3] proposed a lower bound (LB) technique to find the optimal schedule for the makespan criterion. The earliest completion time of all the jobs at every scheduling point is the LB for resolving the conflict among jobs. The conflict is resolved by selecting one of the contending jobs that has the minimum LB value to obtain the schedule. Branch and Bound methodology for pruning the conflict resolution tree of active feasible schedules for the optimal schedule, instead of resolving all the conflicts one by one and searching for the optimal with the GT algorithm. A lower bound technique is used to find the optimal schedule for the makespan criterion.

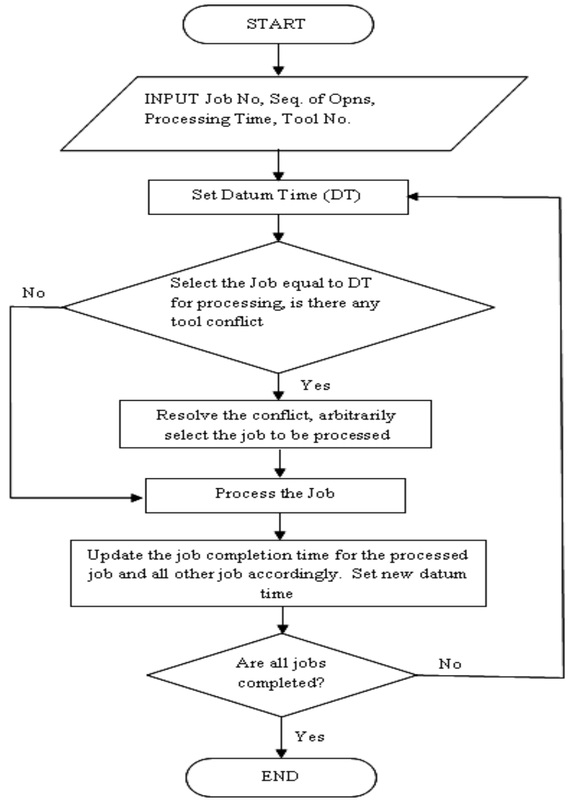


Figure 1: Flow Chart

BBT algorithm

The lower bound on any solution that may be generated via this particular branch of the tree and designated LB (M-N) may be determined as follows:

LB (M-N) = max Cj (M-N)

(j = 1, 2, 3… n) where n is the total number of jobs.

a) BBT conflict_fn ( )

Step 1. Find lower bound value LB (Ci) of all contending jobs.

LB (Ci) = Maximum of {EJ (i, Ci)} ∀ i and Ci

where EJ (i, Ci) is the earliest completion time of job i when conflict job Ci is loaded.

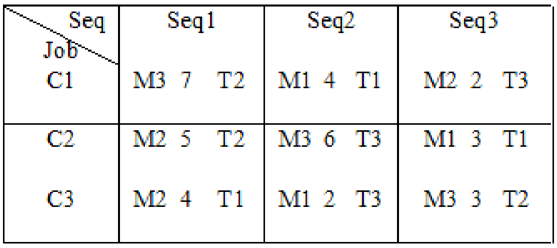
Step 2. Find minimum value of lower bound LBMIN = minimum of LB (Ci) ∀ Ci

Step 3. Resolve conflict and select the job "Ci" which has LB (Ci) = LBMIN

When applying the Branch and Bound method, there is a possibility of obtaining the same minimum lower bound (LBMIN) value for more than one job. This situation leads to a TIE between them. If the ties are resolved arbitrarily, then the resultant one may not be the optimal.

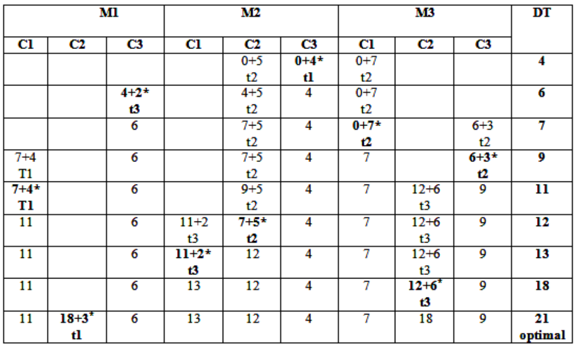
But in our case we have resolved all the ties one by one and selecting the one which gives the minimum lower bound value. So that the optimal one may not be missed at all and we will get the optimal solution by using branch and bound method. Branch and bound method, solving larger size problems may be difficult one. But with the help of present-day computers we can solve.

Table 1 –3jobs with 3 machines & 3tools



Seq / Job	Seq1	Seq2	Seq3
C1	M3 7 T2	M1 4 T1	M2 2 T3
C2	M2 5 T2	M3 6 T3	M1 3 T1
C3	M2 4 T1	M1 2 T3	M3 3 T2

Table 2 – The optimal schedule with minimum makespan time



M1			M2			M3			DT
C1	**C2**	**C3**	**C1**	**C2**	**C3**	**C1**	**C2**	**C3**	
				0+5 t2	**0+4* t1**	0+7 t2			**4**
		4+2* t3		4+5 t2	4	0+7 t2			**6**
		6		7+5 t2	4	**0+7* t2**		6+3 t2	**7**
7+4 T1		6		7+5 t2	4	7		**6+3* t2**	**9**
7+4* T1		6		9+5 t2	4	7	12+6 t3	9	**11**
11		6	11+2 t3	**7+5* t2**	4	7	12+6 t3	9	**12**
11		6	**11+2* t3**	12	4	7	12+6 t3	9	**13**
11		6	13	12	4	7	**12+6* t3**	9	**18**
11	**18+3* t1**	6	13	12	4	7	18	9	**21 optimal**

3.2. Simulated Annealing Algorithm (SAA)

It is suitable to deal with classical Job Shop Scheduling Problems (JSSPs), most application try to minimize the makespan. They adopt the disjunctive graph representation and the moves are based on reversing critical arcs or changing some other precedence releation on the critical path, SAA requre high computational time to achieve good solutions [5].

Tree diagram of B&B method

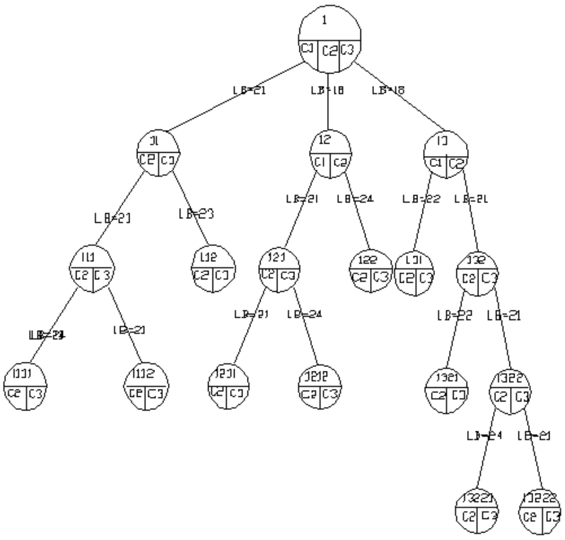


Figure 2: Tree diagram for B&B method

Simulated Annealing Algorithm is employed in such a way that it finds best priority sequence through random generation of initial priority sequence set at high temperature and Randomly Inserted Perturbation scheme (RIPS) for further improvements[6]. The parameters and steps of the Simulated Annealing Algorithm are as follows:

Step1: Initialization: Set AT=450; fr_cnt=0; accept=0; total=0;

Step2: Generation of initial solution arbitrarily generate two initial priority jobs sequences S and B.

Find the makespan time s corresponding to S and B using new algorithm procedure and assign to both M_s and M_B respectively.

Step3: Checking termination of SAA.When (fr_cnt=5) or AT=20 then go to step16. Else proceed to step4

Step4: Generation of neighbours. Generate number of nearer sequences to S using randomly insertion perturbation scheme (RIPS).

Step5: Find makesapan time of all sequences generated in step 4

using Priority Dispatching Rules. Sort he minimum makespan time and store it in M_S.

Step6: Compute ΔS (MS, MS’). If (ΔS<=0) then proceed to Step7. Else go to step10.

Step7: Assign S=S’, $M_{S=}M_S$’ and accept=accept+1

Step8: Compute ΔB (MB, MB’). If (ΔB<=0) then proceed to step9. Else go to step12

Step9: Assign B=S’, $M_{B=}M_B$’ and fr_cnt=0 and go to step12.

Step10: Compute P and sample U.

If U>P then go to step12.else proceed to step11.

Step11: Assign S=S’, $M_{S=}M_S$’ and accept=accept+1.

Step12: Set Total=Total+1.

Step13: If (Total>2*n) or (accept>n/2), then proceed to step 14.Else go back to step 4.
Step14: Compute per= (accept*100/total)
If per<15 then set fr_cnt=fr_cnt+1.Else go to step15.
Step15: Set AT=AT *0.9, accept=0, total=0 and go back to step3.
Step16: The algorithm frozen. B contains the best sequence. M_B has the minimum makespan time.

Simulated Annealing is a point-by-point method. The algorithm begins with an initial point and a high temperature AT. A second point is created at random in the vicinity of the initial point and the difference in the function values (ΔE) at these two points is calculated. If the second point has a smaller function value, the point is accepted, otherwise the point is accepted with a probability exp (-ΔE/AT).This completes one iteration of the simulated annealing procedure. In the next generation, another point is created at random in the neighborhood of the current point and the Metropolis is used to accept or reject the point. In order to simulate the thermal equilibrium at every temperature, a number of points (n) are usually tested at a particular temperature, before reducing the temperature.

The algorithm is terminated when a sufficiently small temperature is obtained or a small enough change in function values is found. Four Basic ingredients are needed to apply simulated annealing in practice a concise problem representation, a neighborhood function, a transition mechanism and a cooling schedule. As for the choice of the cooling schedule, there exist some general guidelines. However, no general rules are known that guide the choice of the other ingredients. The way they are handled is still a matter o experience, taste, and a skill left to the annealing practitioner, and it is expected that this will not change in the near future.

3.2.1 Initialization:

The annealing algorithm is set to 475.The freezer count is set to 0. The accept total are also set to 0.Arbitarily two priority sequence are developed and assigned to S and B.

3.2.2 Generation of neighbors:

The proposed perturbation scheme, RIPS can be best explained by a simple numerical illustration. Consider a seed sequence {2-3-1-5-4} with five jobs. The job in the first (i.e. an extreme position) can be inserted at any position to its right. Hence the job in the first position is inserted between 2 and n (here n=5), and a random number generated 2 and n is used to select the job position. Suppose the selected position is 3.Job [1] is inserted in position 3, yielding a new sequence {3-1-2-5-4}. Considering the job in the second (i.e. non extreme) position of sequence S and choose randomly two positions for its insertion. Note that this job can be inserted at any position between (2+1)th and *n*th positions(i.e. a position to its right) and between 1st and position 4 is selected to the right of job 3.The new sequence thus generated are {3-2-1-5-4} and {2-1-5-3-4}. Similarly ,for the jobs in positions 3 and 4,we select two positions randomly ,one to the right and one to the left, and obtain the resulting sequences {2-1-3-5-4}, {2-3-5-1-4},{5-2-31-4} and {2-3-1-4-5}. The same position can be continued for all jobs found in position 2 through *n*-1 in the original sequence. As for the job in the *n*th position(another extreme position),only one position is randomly selected towards the left of the job, i.e. between positions 1 and n-1.The randomly selected position be 2 and the resulting sequence be {2-4-3-1-5}. Generate 2*(n-1) number of sequences with n jobs using RIPS perturbation scheme.

3.2.3 Evaluation and assignment

The makespan time for each sequence is calculated using priority dispatching rules for the randomly generated sequence. This value is compared with previous minimum. If the new minimum is better than previous, new minimum is stored and accept is incremented. Total is incremented.

3.2.4 Check for iteration

If (total <2 *n) or (accept>n/2) condition are not satisfied, it will proceed to previous step. If these conditions are satisfied fr_cnt is incremented and annealing temperature AT is set to AT*0.9.Then it will proceed to next step.

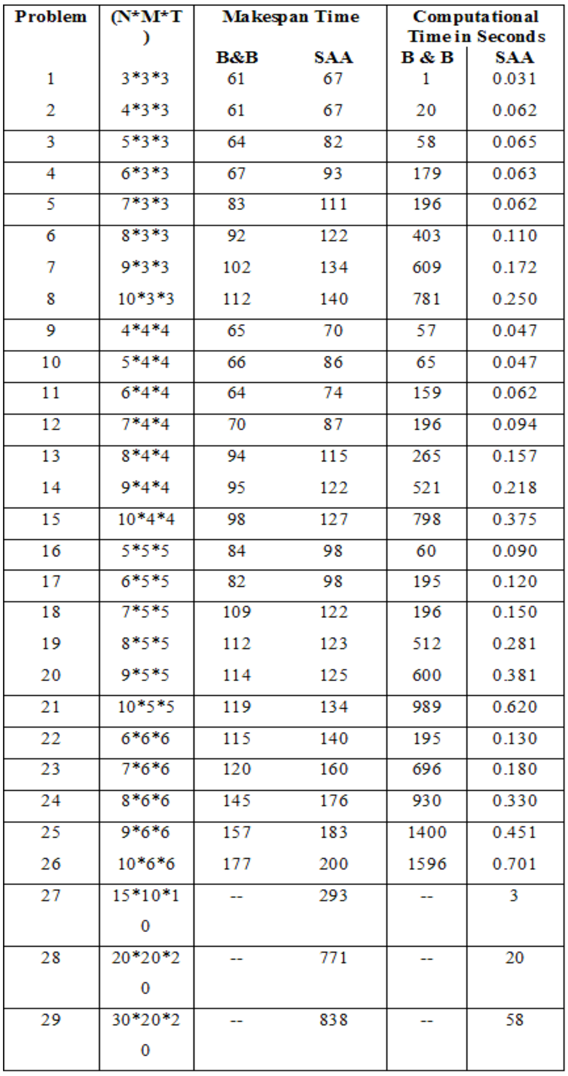
3.2.5 Checking for Termination

If annealing temperature AT is less than 20 or fr_cnt value equal to 5 then algorithm terminates. Else the second step will be repeated.

4 RESULTS AND DISCUSSION

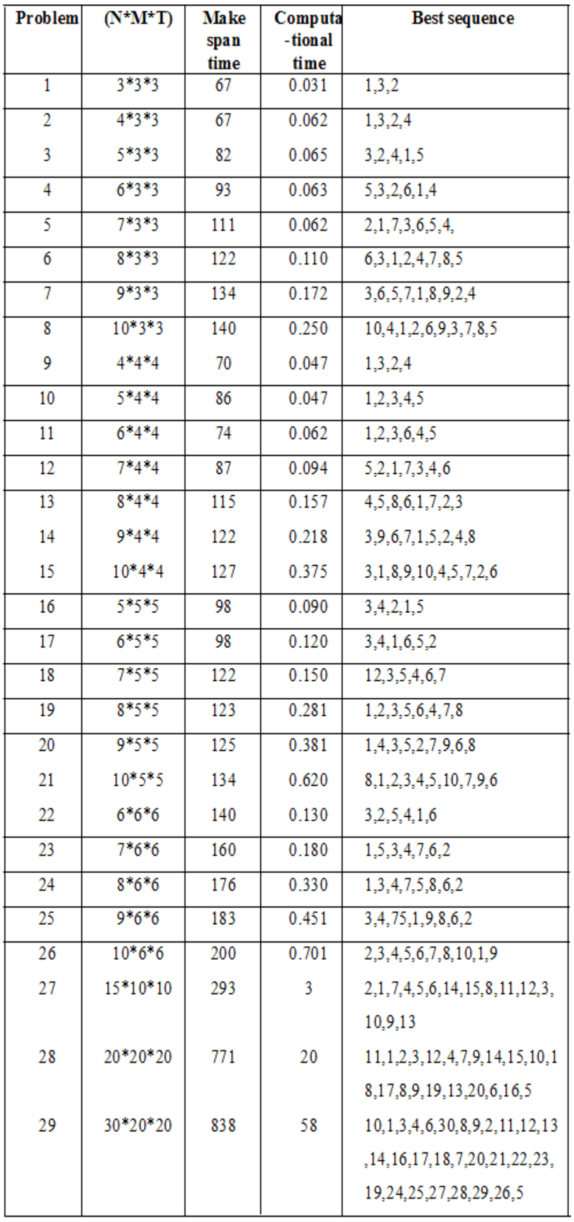
The proposed methodology experienced with twenty nine sample problems of various size is shown in table 3 are tabulated and compared for two different methodologies (B&B and SAA), for the performance comparison in terms of makespan time and computational time.

Table 3 – Makespan Time Comparison of B&B and SAA



Problem	(N*M*T)	Makespan Time		Computational Time in Seconds	
		B&B	SAA	B & B	SAA
1	3*3*3	61	67	1	0.031
2	4*3*3	61	67	20	0.062
3	5*3*3	64	82	58	0.065
4	6*3*3	67	93	179	0.063
5	7*3*3	83	111	196	0.062
6	8*3*3	92	122	403	0.110
7	9*3*3	102	134	609	0.172
8	10*3*3	112	140	781	0.250
9	4*4*4	65	70	57	0.047
10	5*4*4	66	86	65	0.047
11	6*4*4	64	74	159	0.062
12	7*4*4	70	87	196	0.094
13	8*4*4	94	115	265	0.157
14	9*4*4	95	122	521	0.218
15	10*4*4	98	127	798	0.375
16	5*5*5	84	98	60	0.090
17	6*5*5	82	98	195	0.120
18	7*5*5	109	122	196	0.150
19	8*5*5	112	123	512	0.281
20	9*5*5	114	125	600	0.381
21	10*5*5	119	134	989	0.620
22	6*6*6	115	140	195	0.130
23	7*6*6	120	160	696	0.180
24	8*6*6	145	176	930	0.330
25	9*6*6	157	183	1400	0.451
26	10*6*6	177	200	1596	0.701
27	15*10*10	--	293	--	3
28	20*20*20	--	771	--	20
29	30*20*20	--	838	--	58

Table 4 – SAA Results:



Problem	(N*M*T)	Make span time	Computa-tional time	Best sequence
1	3*3*3	67	0.031	1,3,2
2	4*3*3	67	0.062	1,3,2,4
3	5*3*3	82	0.065	3,2,4,1,5
4	6*3*3	93	0.063	5,3,2,6,1,4
5	7*3*3	111	0.062	2,1,7,3,6,5,4,
6	8*3*3	122	0.110	6,3,1,2,4,7,8,5
7	9*3*3	134	0.172	3,6,5,7,1,8,9,2,4
8	10*3*3	140	0.250	10,4,1,2,6,9,3,7,8,5
9	4*4*4	70	0.047	1,3,2,4
10	5*4*4	86	0.047	1,2,3,4,5
11	6*4*4	74	0.062	1,2,3,6,4,5
12	7*4*4	87	0.094	5,2,1,7,3,4,6
13	8*4*4	115	0.157	4,5,8,6,1,7,2,3
14	9*4*4	122	0.218	3,9,6,7,1,5,2,4,8
15	10*4*4	127	0.375	3,1,8,9,10,4,5,7,2,6
16	5*5*5	98	0.090	3,4,2,1,5
17	6*5*5	98	0.120	3,4,1,6,5,2
18	7*5*5	122	0.150	12,3,5,4,6,7
19	8*5*5	123	0.281	1,2,3,5,6,4,7,8
20	9*5*5	125	0.381	1,4,3,5,2,7,9,6,8
21	10*5*5	134	0.620	8,1,2,3,4,5,10,7,9,6
22	6*6*6	140	0.130	3,2,5,4,1,6
23	7*6*6	160	0.180	1,5,3,4,7,6,2
24	8*6*6	176	0.330	1,3,4,7,5,8,6,2
25	9*6*6	183	0.451	3,4,75,1,9,8,6,2
26	10*6*6	200	0.701	2,3,4,5,6,7,8,10,1,9
27	15*10*10	293	3	2,1,7,4,5,6,14,15,8,11,12,3,10,9,13
28	20*20*20	771	20	11,1,2,3,12,4,7,9,14,15,10,18,17,8,9,19,13,20,6,16,5
29	30*20*20	838	58	10,1,3,4,6,30,8,9,2,11,12,13,14,16,17,18,7,20,21,22,23,19,24,25,27,28,29,26,5

1. The results show that Branch and Bound Algorithm and Simulated Annealing Algorithm are capable of providing optimal /near by optimal solution. However finding solutions to larger size problems using B&B algorithm is not feasible one.
2. The computational time increases exponentially for Branch and bound Algorithm compared with Simulated Annealing Algorithm.
3. Solution quality is not compared since there is no Bench Marking Problems.
4. However the nature of solutions obtained in each randomly inserted Perturbation in simulated annealing process confirms optimal solution.

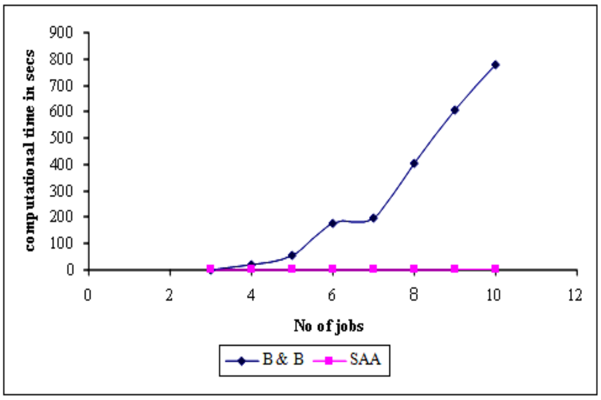


Figure 3: Graph (B&B vs SAA)

CONCLUSION

In this paper, extended Giffler and Thompson Algorithm based Branch and Bound Technique and Simulated Annealing Algorithm based heuristic, for automated job Shop/FMS scheduling problems with makespan as the criterion, has been addressed. The Branch and Bound Algorithm(B&B) and Simulated Annealing Algorithm(SAA) are suitable for Automated Job Shop/Flexible Manufacturing system(FMS) scheduling problems because:

1. The solution is closer to other methodologies and capable of providing solutions to optimal.
2. The heuristic search process can be regarded as better than simulation in the sense that it guarantees near optimal solutions in actual cases.
3. Even the computational time for Branch and Bound Algorithm is higher; planning can be done in small size problems.

REFERENCES

[1]. Hong-Bae Jun, Yeong-Dae kim, and Hyo-Won suh. 1999. Heuristics for a Tool Provisioning Problem in a Flexible Manufacturing System with an Automatic Tool Transporter, IEEE Transactions on Robotics and Automotion, 15 (3) June, pp. 488-496.

[2]. Selim Akburk.M and Siraceddin Oxen. 1999. Joint lot sizing and tool management in a CNC environment. Computers in industry, 40, pp. 61-75.

[3]. Giffler and Thomson. G.L, 1960. Algorithms for solving production scheduling problems. International Journal of Operations Research, 8, pp. 487-500.

[4]. G. Moslehi, M.Mirzaee, M. Vasei, M. Modarres, A. Azaron. 2009. Two-machine flow shop scheduling to minimize the sum of maximum earliness and tardiness. Int. J. Production Economic 122, pp. 763-773.

[5]. T.K. Varadharajan, Chandrasekharan R. 2005. A multi objective simulated-annealing algorithm for scheduling in flowshops to minimize the makespan and total flowtime of jobs. International Journal of Operational Research 167, pp. 772-795.

[6]. Dong-Won Kim, Kyong-Hee Kim, Jang .W, F.F.Chen. 2002. Unrelated parallel machine scheduling with setup times using simulated annealing. Robotics and Computer Integrated Manufacturing 18, pp. 223-231.

[7]. Maria J. Alves, Climaco .J. 2000. An intelligent reference point approach for multiobjective mixed-integer programming using branch-and-bound. European Journal of Operational Research 124, pp. 478-494.