SEQUENCING AND SCHEDULING OF JOB AND TOOL IN FLEXIBLE MANUFACTURING SYSTEM

[1*]R.Vijayan, [2]T.Arun Christopher, [3]M.A.Athiya

[1,2,3] Department of Mechanical Engineering, Veltech Engineering College, Chennai, Tamil Nadu, India
[1*]srajendranvijayan@gmail.com

Abstract

This thesis deals with the 'N' Jobs and 'T' Tools, Sequencing and Scheduling Problem in a Flexible Manufacturing System consisting of 'M' Machining Centers. Each job requires a set of tools for the execution of a sequence of operations. The tools are stored in a Common Tool Magazine that shares with and serves for several machining centers reduces the cost of duplicating tools in each and every machining center. The concern of the thesis is the design and development of algorithm for the generation of active schedules and optimal sequence of jobs that can meet minimum make span schedule for the tool constraint job shop type production system. The thesis addresses three different heuristics namely Priority Dispatching Rule Algorithm, Genetic Algorithm and Simulated Annealing Algorithm. The above algorithms adopt the procedure of Extended Giffler and Thompson Algorithm for active feasible schedules and adopt different techniques for optimal schedule. The performance is compared with regard to makespan criteria and computational time. It is observed that the computational time increases with job size. The proposed methodology is relatively new and can be used for any objective suitable by modifying the fitness parameter corresponding to the desired performance criterion. A package has been developed in C and Visual Basic to generate optimal sequence of jobs and active feasible schedule.

Key words: Sequencing, Scheduling, Flexible Manufacturing Cell, Extended Giffler and Thompson Algorithm, Genetic Algorithm.

I. INTRODUCTION

Scheduling literature addresses wide range of problems described with machine environment, job description, and objective function [1], [2]. Operation scheduling is the allocation of jobs to be processed on the corresponding machines in a given time span, for a workshop consisting of production resources including machines, operative workers, and tools (cutting tools, inspection gauges, work holding fixtures, etc.) FMS/FMC/Automated Job shop manufacturing system will be best suited to produce variety of parts with flexibility of even small lots with nearly the efficiency of large volume Mass production [3]. In many research works, assumption was made to have individual tool magazine for each machining center [3], [4], [5]. This approach leads to more expenses, and 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 (CTM) with several machining centers can be employed [6], [7].

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 FMS/FMC systems. This paper deals with scheduling in the following environment consisting of 'm' no. of machining centers. Machines in the system are assumed to have no on-board tool magazines. Tools are stored in a common tool magazine (CTM), and are delivered by means of tool handling systems consisting of shuttles and robots, which interface with common tool magazine. The jobs correspond to the manufacture of a physical part, for which a part program (i.e. a sequence of operations) is given. Each operation requires a given processing time and a specific tool. Most often, operations are long enough to allow the tool handling system to deliver the next tool without forcing the machine to wait. We will assume that the time taken to transfer a tool from one center to the other is negligible; accounting for the finite speed of shuttles and the robot would only slightly complicate the tool-scheduling problem, without affecting the overall complexity [6].

Conflict may be arised whenever same tool is required by two or more jobs/machines at the same time. In those circumstances, one of the jobs will be served first and others have to wait for the release of tool. 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 and may well traded-off by the cost saving we obtain by sharing tools on different machines. This type of manufacturing system is necessary for manufacturing environment in which tools are particularly expensive. Scheduling in such environment requires taking three types of decisions:

1. The jobs waiting for the operation must be sequenced
2. Each job must be assigned to any of the available machining centers.
3. Each operation of a job must be scheduled, taking tool availability into account.

Obviously, such decisions are not independent from each other, and the overall system performance depends on their interaction. The objective function is minimize the makespan time and maximize the hardware utilization for the tool constraint FMC system. The problem considered here is essentially an identical parallel machine problem with additional resources; since these problems are NP-hard even without additional resources, their solution calls for suitable heuristic approaches). [6]. In this context, the concern of this paper is to generate joint job sequence and tool schedule in FMC/FMS consisting of several machines and a common tool magazine (CTM).

The rest of the paper is organized as follows. In Section 2, the manufacturing environment and the problem are defined. The proposed schedule generation algorithm is addressed in Section 3. In section 4, the proposed methodology is addressed. In section 5, the illustration with an example, comparative analysis and finally, conclusions are given.

II. PROBLEM DESCRIPTION

A. Existing Operating Environment

In Automated Job Shop Manufacturing namely, FMS, consisting of several machining centers. Each Job requires a set of tools for the execution of a sequence of operations. Each machining center has a separate tool magazine which consists of 'N' number of tools.

M/c Centre (1) **M/c Centre (2)** **M/c Centre (M)**

Tool Box (1) **Tool Box (2)** **Tool Box (T)**

In such situation

- Each machines leads to more expenses because of duplicating tools.
- Many tools will be idle most of the time.

B. PROPOSED OPERATING ENVIRONMENT

To cut down the tool expenses, and to have optimum tool utilization, COMMON TOOL MAGAZINE (CTM) with several machines can be employed.

M/c Centre (1) **M/c Centre (2)** **M/c Centre (M)**

Common Tool Magazine (CTM)

In such situation

- Tools are stored in a CTM, and are moved throughout the machining center by means of a tool handling system.
- Conflict may arise whenever the same tool is required by two or more jobs at the same time.

In these circumstances, one of the jobs will be served first and others has to wait for the release of tool

C. Assumptions

- Each Job is composed of a set of operations and the operation order on machines is pre-specified.
- Each operation is characterized by the required machine, tool and its processing time.
- Each machine/tool can process only one job at a time.
- Operation cannot be interrupted i.e., each operation once started must be completed.
- A job does not visit the same machine twice.
- Availability of tools in each variety is considered as one, which can be used more than once of any job.
- The operation time of a job includes the loading, unloading, tool changeover and setup times (both tool and job) along with processing time.
- Machine is capable of doing many operations
- Machine failure is not considered

D. Problem Definition

"Generation of minimum makespanship schedule of "N" different Jobs which require processing on "M" different Machines with "T" Tools, that are shared for many operations, in a job shop type production system".

III. PROPOSED SCHEDULE GENERATION HEURISTIC

The problem to determine optimum scheduling with minimizing total tardiness in a above said manufacturing environment with common tool magazine comes under combinatorial category. For this type of problems, if we want to obtain the better solution, we will have to use either complete enumeration technique or suitable heuristics.

A. EXTENDED GIFFLER AND THOMPSON ALGORITHM:

Step 1: Construct a table subdivided into blocks one for each machine. Subdivide each machine block into columns, one column for each job that requires processing on the machine.

Step 2: Enter the first line of table, in the appropriate machine blocks, the processing time of the first operation of each job. The entries indicate the earliest possible completion time of immediate waiting operations of all jobs.

Step 3: Set the datum time equal to smallest of the entries and enter it on the right hand side of the table on the same line.

Step 4: Select the jobs whose operation time in the table matches with datum time.

Step 5: Check for machine conflict (more than one job waiting in the machine)?

If there is not conflict go to step 7.

Step 6: List all the jobs contending for conflict machine. Resolve conflict arbitrarily and select that job as the candidate for assignment in the conflict machine. Check for any tool conflict (same tool requirement for waiting jobs in other machines) for the selected job. If so, the tool conflict is also resolved arbitrarily and the operation along with the machine and tool are earmarked. Go to step 9.

Step 7: Check for tool conflict (same tool requirement for waiting jobs in other machines)?

If there is no tool conflict, earmark the operation. Go to step 9.

Step 8: List all the jobs contend for the tool. Resolve conflict arbitrarily and select the job as the candidate for assignment with that tool under consideration. Check for machine conflict for the selected job. If so, machine conflict is also resolved arbitrarily and the candidate along with machine and tools are earmarked.

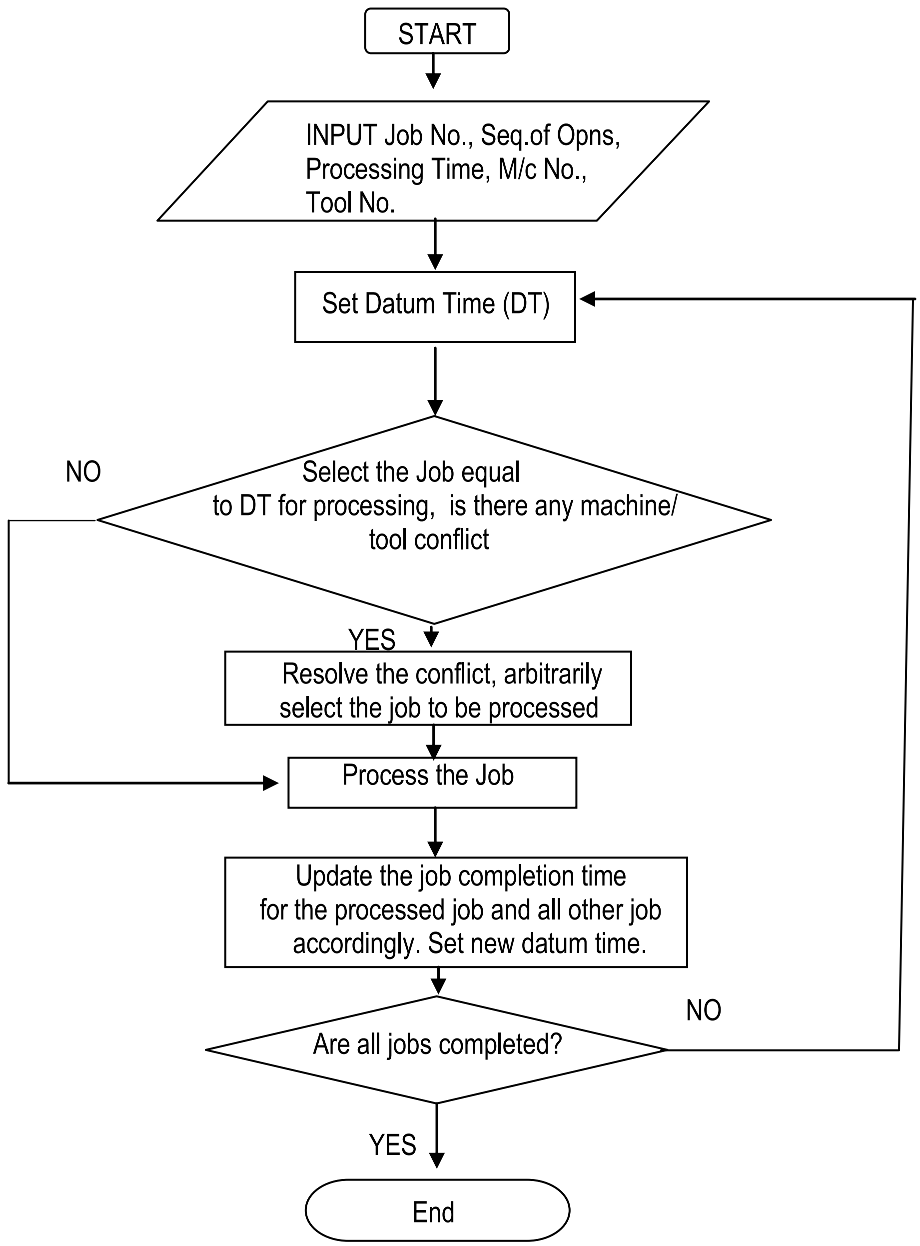
Step 9: The operation of the earmarked along with its processing machine and tool is assigned.

Step 10: Update earliest possible completion time of immediate waiting operations including the next operation of the job just assigned in the next line of the table.

Step 11: Check for the assignment of all operations If 'yes' go to step 12. Otherwise, go to step 3.

Step 12: End

B. FLOWCHART FOR EXTENDED GIFFLER AND THOMPSON ALGORITHM:



IV. GENETIC ALGORITHM

Methodology

The proposed GA evaluates the active feasible schedules of each population that are generated by resolving the conflicts one by one the sequences represented as chromosomes and evolves the best loading sequence. The coding adopted in this GA is as follows: a gene denotes the job number; position of the gene represents the priority rank of the job; one chromosome describes a job loading sequence; number of chromosomes forms a population. The optimal or near optimal solution is obtained through number of generations with the genetic procedure of random generation of members for initial population and generation of members for new population based on selection of fittest members of previous population, crossover and mutation [10]. The fitness parameter 'total tardiness time' is found out by generating an active feasible schedule using the concept and adaptation of branch and bound technique and resolving the tie and conflict stages one by one.

A. Genetic Algorithm

Steps in Genetic Algorithm

Step 1: Generation of Initial Population:

Generate initial population randomly consisting of number of chromosomes (c) such that pop_size = 2 * n

Step 2: Evaluation:

In this step, the population is evaluated for fitness and probability of selection of each chromosome is to be found out. The process of evaluating the fitness of chromosomes (c) is given below.

Step 2.1: Evaluation of Objective Function

Objective function f(c) is evaluated using Branch and Bound algorithm such that

f(c) = makespan time of the corresponding sequence of the chromosomes (c).

Step 2.2: Fitness Value

Convert the objective function to a fitness value fit(c) suitable for the minimization objective function. The best conversion function that has been found to be generally useful is the exponential i.e., $fit(c) = e^{-k*fit(c)}$

Where constant k = 0.05

Step 2.3: Probability

Convert the new fitness parameter to expected probability of selection (p(c)) of chromosomes(c) are found using formula $p(c) = fit(c) / \Sigma fit(c)$. The cumulative probabilities of survival (cp(c)) of all chromosomes are found using formula $cp(c) = \Sigma p(c)$

Step 3: Selection of New Population:

A random selection procedure generates the next population of the same size. A random number r between 0 and 1 is obtained and a chromosome (c) is selected which satisfies the following condition $cp(c-1) \leq r \leq cp(c)$.

This selection process is repeated a number of times equal to the population size. The method used here is more reliable in that it generates that the most fit individuals will be selected, and that the actual number of times each is selected will be its expected frequency ±1. This procedure enables the fittest chromosome to have multiple copies and the worst to die off.

Step 4: Cross Over:

Random numbers between 0 and 1 are generated for all chromosomes and those chromosomes that obtain a random number less than the p_cross value are the chromosomes selected for crossover. The value for p_cross has been assumed to be 0.4, so that at least 40% of the chromosomes selected for the new population will undergo the crossover operation and produce offspring. The genetic literature addresses many crossover operators [3]. Notable among them are: partially mapped crossover (PMX), ordinal mapped crossover (OMX) and edge crossover (EX). PMX (Partially Matched Crossover) is used in this method [10].

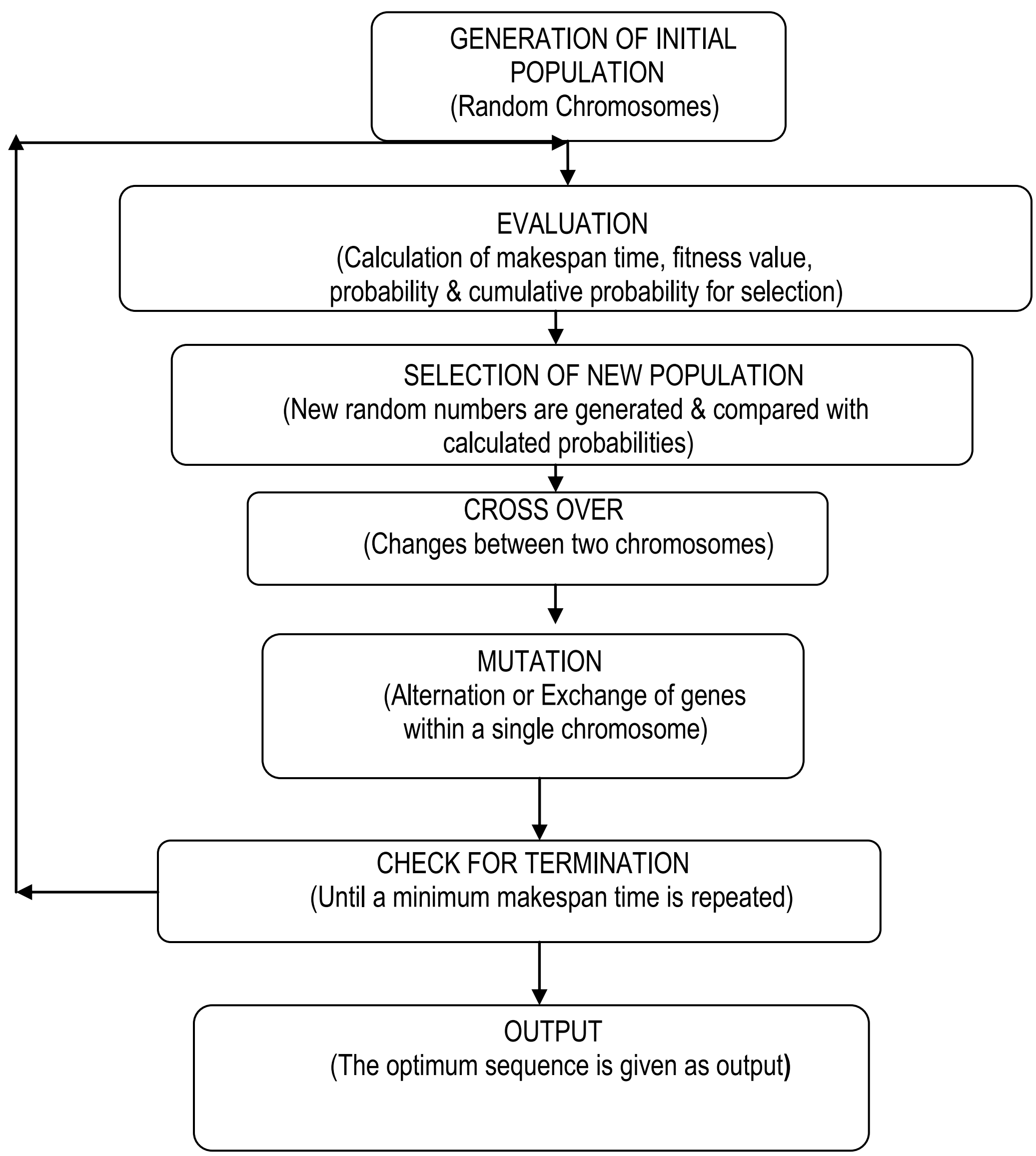
Step 5: Mutation:

Random numbers between 0 and 1 are generated for all chromosomes and those chromosomes that obtain a random number less than the (p_mute) value are the chromosomes selected for mutation. The value of the probability of mutation (p_mute) has been assumed to be 0.05. Actually, the mutation here exchanges the gene within the chromosome.

Step 6: Check for Termination:

The repetition of the whole process (iteration) of evaluation, selection, reproduction and mutation depends on the size of the problem. No generalization is possible with respect to the behavior of model constraint. The number of iterations is considered as the termination criterion and is equal to n^2. The best chromosome in each iteration is stored, and the best among those stored is the optimal one.

B. Flow Chart for GA Procedure



C. Simulated Annealing Algorithm:

In this, Simulated Annealing is employed in such a way that it finds best priority sequence through random generation of initial priority sequence set at high temperature and pair-wise exchange perturbation scheme for further improvements. The parameters and steps of the Simulated Annealing Algorithm are as follows:

Step 1: Initialization: Set at = 475; fr_cnt = 0; accept = 0; total = 0;
Step 2:Generation of initial solution:
Arbitrarily generate two initial priority job sequences S and B. To find the makespan time corresponding to S and B using EGT procedure and assign to both M_S and M_B
Step 3: Checking termination of SA:
If (fr_cnt = 5) or at = 20 then go to step 16. Else proceed to step 4.
Step 4: Generation of neighbours:
Generate number of nearer sequences to S using pair wise perturbation scheme.
Step 5: Find makespan times of all sequences generated in step 4 using EGT procedure. Sort the minimum makespan time and store it in M_S.
Step 6: Compute $delta_S$ (M_S, M_S').
If ($delta_S$<=0) then proceed to step 7. Else go to step 10.
Step 7: Assign S = S', M_S= M_S' and accept = accept + 1.
Step 8: Compute $delta_B$ (M_B, M_B').
If ($delta_B$<=0) then proceed to step 9. Else go to step 12.
Step 9: Assign B = S', M_B = M_S' and fr_cnt = 0, and go to step 12.
Step 10: Compute P and sample U.
If U>P then go to step 12. Else proceed to step 11.
Step 11: Assign S = S', M_S= M_S' and accept = accept + 1.
Step 12: Set total = total + 1.
Step 13: If (total>2 * n) or (accept > n/2), then proceed to step 14. Else go back to step 4.
Step 14: Computer per = (accept*100/total).
If per<15, then set fr_cnt = fr_cnt + 1.
Step 15: Set at=at*0.9, accept = 0, total = 0 and go back to step 3.
Step 16: The algorithm frozen. B contains the best sequence. M_B has the minimum makespan time.

V. ILLUSTRATIVE EXAMPLE

The input data of jobs that requires processing on 2 identical machines for 15 problems were scheduled. In the proposed branch and bound based GA methodology, if all ties and all conflict stages are resolved one by one we can get the optimal or closer to optimal solution. The results of a sample of fifteen problems are given the table 2 using BB and GA. It shows that we are able to get optimal or near optimal solutions for all the problems concerned and is given.

PROBLEM 1:

O J	1	2	3
1	3_2	8_1	7_3
2	6_3	4_2	8_1
3	9_1	5_3	6_2

PROBLEM 2

O J	1	2	3
1	19_2	11_3	17_1
2	13_1	6_3	24_2
3	16_3	14_1	12_2
4	8_2	22_1	18_3

PROBLEM 3:

O J	1	2	3	4
1	15_2	12_3	8_4	9_1
2	7_3	9_4	12_1	6_2
3	11_2	6_1	15_4	7_3
4	9_4	10_3	9_1	6_2
5	11_1	7_2	6_3	14_4

PROBLEM 4:

O J	1	2	3	4	5
1	11_5	7_2	10_4	9_1	12_3
2	8_1	10_2	10_3	11_5	8_4
3	10_1	14_4	13_3	12_2	6_5
4	10_4	8_5	6_3	10_2	11_1
5	6_2	9_3	6_1	15_4	12_5

PROBLEM 5

O J	1	2	3	4
1	11_2	7_1	6_3	3_4
2	7_4	7_3	5_2	11_1
3	12_1	13_4	9_3	15_2
4	7_4	5_1	12_2	20_3
5	8_4	12_2	12_1	14_3
6	10_4	14_2	7_3	8_1

PROBLEM 6:

O J	1	2	3	4	5
1	15_4	12_3	8_2	9_5	10_1
2	7_2	13_4	6_3	9_5	6_1
3	8_3	15_4	13_5	10_1	11_2
4	11_2	6_1	15_4	14_3	10_5
5	14_1	10_4	12_5	16_2	18_3
6	9_5	12_3	15_1	13_4	10_2
7	6_1	15_5	14_4	13_2	15_3

PROBLEM 7:

O J	1	2	3	4	5	6	7
1	6_2	7_1	15_7	12_3	8_5	11_4	14_6
2	7_1	9_5	11_3	10_2	14_4	13_7	14_6
3	9_4	9_6	15_1	11_5	13_2	15_7	11_3
4	8_2	15_1	13_5	15_6	13_7	9_4	6_3
5	10_3	12_2	6_7	9_1	13_5	9_6	15_4
6	7_2	6_1	10_6	10_5	10_7	6_3	7_4
7	7_5	13_4	6_2	10_1	7_6	9_3	12_7
8	11_4	7_7	13_5	15_6	13_3	9_4	6_2

PROBLEM 8:

O J	1	2	3	4	5
1	8_3	7_2	10_1	15_5	13_4
2	7_4	9_3	10_2	6_1	4_5
3	10_4	12_2	6_1	15_3	12_5
4	11_1	10_2	9_3	9_5	12_4
5	15_3	11_2	12_1	6_4	14_5
6	10_3	10_4	9_2	6_1	10_5
7	6_1	12_3	15_5	11_2	10_4
8	9_2	9_5	12_4	11_3	13_1
9	15_5	11_2	15_3	12_1	9_4

PROBLEM 9:

O J	1	2	3	4	5	6
1	10_1	25_4	16_5	8_3	1_6	20_2
2	6_4	9_2	17_5	3_6	5_3	17_1
3	6_2	10_3	15_1	8_4	6_6	5_5
4	19_1	21_3	24_5	19_2	22_4	12_6
5	10_3	21_5	17_4	21_6	8_1	12_2
6	10_3	20_5	5_6	8_4	24_1	19_2
7	6_2	17_1	6_6	15_4	16_5	9_3
8	6_1	5_5	22_3	8_3	17_4	10_6
9	19_4	12_6	15_1	21_5	10_2	13_3
10	7_2	13_1	6_6	10_3	9_5	12_4

PROBLEM 10:

O J	1	2	3	4	5
1	11_3	7_5	13_1	15_4	9_2
2	7_5	6_1	9_2	7_3	7_4
3	10_4	10_5	9_3	13_2	7_1
4	9_3	6_1	9_4	12_5	8_2

5	9_4	13_2	6_5	15_1	15_3
6	10_5	11_3	6_1	6_4	8_2
7	15_3	10_1	12_5	11_4	10_2
8	10_5	9_2	10_1	14_3	13_4
9	7_3	8_2	13_1	10_4	14_5
10	13_3	11_2	10_4	9_1	9_5
11	11_5	15_4	6_2	13_1	9_3

PROBLEM 11:

O J	1	2	3	4
1	13_4	12_2	15_3	11_1
2	15_3	8_2	12_1	7_4
3	6_2	15_4	11_3	6_1
4	7_1	9_3	7_2	10_4
5	6_1	13_4	14_2	6_3
6	3_2	6_3	9_1	7_4
7	12_4	12_3	15_2	5_1
8	14_3	10_4	10_2	11_1
9	11_2	11_1	8_4	15_3
10	7_1	6_2	13_4	14_3
11	14_4	10_1	15_3	10_2
12	15_3	7_1	11_2	12_4

PROBLEM 12:

O J	1	2	3	4	5
1	24_5	16_3	20_4	10_2	10_1
2	25_3	30_5	40_2	15_4	9_1
3	20_4	25_2	15_1	10_3	5_5
4	25_1	35_3	45_2	15_5	20_4
5	30_2	20_5	40_3	10_4	10_1
6	20_3	20_1	30_5	15_4	10_2
7	15_4	15_5	20_1	10_3	8_2
8	40_2	10_4	15_1	25_5	30_3
9	12_1	23_5	14_4	19_3	21_2
10	35_4	45_2	27_1	18_3	9_5
11	24_3	32_2	17_5	8_1	26_4
12	10_5	19_3	14_4	23_1	29_2
13	15_2	10_3	25_1	32_4	13_5

VI. CONCLUSIONS

In this thesis, Genetic Algorithm and Simulated Annealing Algorithm based heuristic, which uses a Extended Giffler and Thompson Algorithm procedure along with Priority Dispatching Rules, for Flexible Manufacturing System scheduling problems with makespan as the criterion, has been addressed. A comparison in terms of makespan time of the schedules and computation time has been carried out. The comparison indicates that the Genetic Algorithm based heuristic search process is suitable for Flexible Manufacturing System scheduling problems because

- The computational time is reasonable.
- The solution is closer to other methodologies and capable of providing solutions nearer to optimal.
- By changing the evaluation parameter of the genetic search process, solutions can be obtained for other suitable objectives and can be made more flexible.
- The heuristic search process can be regarded as better than simulation in the sense that it guarantees near optimal solutions in actual cases.

Rescheduling with regeneration guarantees the solution quality whatever the state of the system parameters and addresses real-time operation

NOMENCLATURE

m number of machines
n number of jobs
J_i job number
M_j machine number
Tij tool number for the k^{th} operation of the i^{th} job

tik processing time for the k^{th} operation of the i^{th} job
k operation sequence number

REFERENCES

[1]. French. S., "Sequencing and Scheduling", London, Ellis Hordwood, 1982, 5/e.

[2]. Brucker. P., "Scheduling Algorithms", Springer-Verlag Berlin-Heidelberg, 1995, 1/e.

[3]. Jawahar. N., Aravindan. P. and Ponnambalam. S.G., "A Genetic Algorithm for scheduling Flexible Manufacturing Systems", International Journal of Advanced Manufacturing Technology, 1998, Vol. 14,588-607.

[4]. Liu.J, and Macc Carthy. B.L., "General heuristic procedures and solution strategies for FMS scheduling", International Journal of Production Research, 1999, Vol. 37(140), 3305-3333.

[5]. Tiwari, and N.K. Vidyarthi, N.K., "Solving Machine Loading in a flexible manufacturing system using genetic algorithm based heuristic approach", International Journal of Production Research., 2000, Vol. 38(14), 3357-3384.

[6]. Alessandro Agnetics, Arianna Alfieri, Paolo Prinsecchi, "Joint Job/ Tool Scheduling in a Flexible Manufacturing Cell with no On-Board Tool Magazine", Computer Integrated Manufacturing Systems, 1997, Vol. 10(1), 61-68.

[7]. Prabaharan. T., Nakkeeran. P.R., Jawahar.N and Christopher Solomon. S., "Joint Job/Tool scheduling in Automated Job shop system", Proceeding of the International conference on intelligent Flexible Manufacturing Systems., 2000, 443-449.

[8]. Greenberg. H.H., "A branch and bound Solution to the general scheduling Problem". Internal Journal of Operation Research, 1968, Vol. 16, 353-361.

[9]. King. J.R., "Production Planning and control an introduction to quantitative methods", London, 1975, 1/e.

[10]. Michalewicz. Z., "Genetic Algorithms + Data Structures = Evolution Programs", Springer- Verlag, Berlin- Heidelherg, USA, 1992.