International Journal of Engineering Science Invention Research & Development; Vol. I Issue VII January 2015
www.ijesird.com e-ISSN: 2349-6185

MACHINE LOADING IN FLEXIBLE MANUFACTURING SYSTEM

B.B.Ashok Kumar[*1], V.V.S.Kesava Rao[2]
[1]Anil Neerukonda Institute of Technology & Sciences, Visakhapatnam, India
[2]Andhra University, Visakhapatnam, India
*Corresponding Author: Contact No.09704638061, Email: ashokanitsmech@gmail.com

Abstract: **The paper addresses a vital pre-release decision that directly affects the operational effectiveness of a flexible manufacturing system-the machine-loading problem. Flexible manufacturing is a concept that allows manufacturing systems to be built under high customized production requirements. Issues such as cutting down of inventories and shortened product life cycles, reducing the cost of products and services to grab more market shares, etc have made it almost compulsory for many companies to switch over to flexible manufacturing systems (FMSs) as a viable means to accomplish the above goals while producing consistently good quality and cost effective products. The combinatorial and NP-hard nature of this problem makes it arduous to secure the best solutions. The objective is minimization of the system unbalance whereas the system's technological constraints are determined by the availability of machining time and tool slots. Due to the large number of random sequences generated as the number of jobs increase, an eliminator function displays and computes the system unbalance only for a fixed number of sequences, thus improving the quality of the solution and reducing the computational burden. The proposed algorithm is tested on a problem sourced from literatures and shows promising results.**

***Key Words*: flexible manufacturing systems, SPT Algorithm, Integer programming.**

1. INTRODUCTION

Globalization, suddenly changing market requirements and the trends of modern living have thrown several tremendous challenges to manufacturing industries. The success of any manufacturing industry is determined by its ability to respond proactively to the rapidly changing market and produce high quality products at low costs. Product cost is no longer the prevalent agent affecting the manufacturers' production decisions [1-4]. Other equally important factors valid in the present day scenario, such as flexibility, quality, efficient delivery and customer satisfaction are drawing their equal focus. Automation, robotics and other innovative concepts such as just-in-time (JIT), Production planning and control (PPC), enterprise resource planning (ERP) etc. are some of the many concepts that aid manufacturing in industries. Flexible manufacturing is a concept that allows manufacturing systems to be built under high customized production requirements [5-12]. Issues such as cutting down of inventories and shortened product life cycles, reducing the cost of products and services to grab more market shares, etc have made it almost compulsory for many companies to switch over to flexible manufacturing systems (FMSs) as a viable means to accomplish the above goals while producing consistently good quality and cost effective products [13-16]. According to Stecke [1983], an FMS is characterized as "an integrated, computer-controlled complex arrangement of automated material-handling devices and numerically controlled (NC) machine tools that can simultaneously process medium-sized volumes of a variety of part types".

The objectives of the paper are to achieve the optimum loading of machines with minimization of system unbalance in flexible manufacturing system. To accommodate as many jobs as possible from the existing available jobs for their completion in the allotted time considering shortest processing time. To find out sequence of jobs for near optimum loading of machine with minimization of system unbalance in flexible manufacturing system.

2.0 METHODOLOGY

The loading problem in manufacturing deals with selecting a subset of jobs from a set of all jobs to be manufactured and assigning their operations to the relevant machines in a given planning horizon with technological constraints in order to meet certain performance measures, such as the minimization of system unbalance [17-20].

2.1 The assumptions used in this paper are:

1. Initially, all of the jobs and machines are simultaneously available.

2. The processing time required to complete an entire job order is known in prior.
3. The job undertaken for processing is to be completed for all of its operation before considering a new job; this is called non-splitting of the job.
4. The operation of a job once started on a machine is continued until it is completed.
5. The transportation time required to move a job between machines in negligible.
6. The sharing and duplication of tool slots is not allowed.

2.2 Back ground in designing methodology

Let us deliberate and evaluate the number of decision variables and constraints
for a typical machine loading problem. Assuming, say,

Number of jobs (J) = 8
Number of operations for each job (O) = 1…3
Number of machines (M) = 4
Then,
Total number of decision variables = J*((M*O) + 1) = 104
Total number of constraints = J +M + M + J*O = 40

Thus, there can be a fairly large number of combinations in which operations of the part type can be assigned on the different machines while satisfying all the technological and capacity constraints.

These operation-machines allocation combinations are evaluated using a common performance measure: system unbalance

However, the values of system unbalance vary for each assigned job sequence, as some jobs may eliminated in each sequence since they do not satisfy the technological and capacity constraints.

Hence a number of job sequences need to be evaluated to find the optimal job sequence, by considering the minimum System unbalance. Take for instance, a loading problem with 8 jobs.

Number of possible job sequences = 8! = 40320

The computational burden would be too high, and the possibility of finding an optimal solution extremely faint in such a situation.

Thus, while creating the proposed algorithm, the number of iterations was fixed, and could be changed if needed. The computational effort was significantly lessened, and the chance of finding near optimal solution was increased.

2.3 Mathematical formulation

Notations

In order to formulate loading problem of FMS, the following notations are introduced:

j: job index, j: 1,2……J
m: machine index, m = 1,2,……M
$\mathbf{S}_m$: tool slot capacity of machine m
o: number of operations for job, o = 1,2…… O_j
B_j: Batch size of job j
T_m: Length of scheduling period for m^{th} machine
P_{jom}: Processing time of operation o of job j on machine m
S_{jom}: Number of tool slots required for processing operations o of job i on machine m
B(j,o): Set of machines on which operation o of job j can be performed.
SU: System unbalance

$$X_j = \begin{cases} 1, & \text{if job j is selected} \\ 0, & \text{otherwise} \end{cases}$$

$$X_{jom} = \begin{cases} 1, & \text{if operation o of job j is assigned on machine m} \\ 0, & \text{otherwise} \end{cases}$$

Overloading not permitted

Objective of the FMS loading is to minimize total system unbalance and is represented by Eq.(1)
Subjected to the following constraints [21,22].

$$\text{Minimize} \sum_{m=1}^{M} T_m - \sum_{m=1}^{M} \sum_{j=1}^{J} \sum_{o=1}^{O_j} B_j P_{jom} X_{jom} \quad \text{............ (1)}$$

$$\sum_{j=1}^{J} \sum_{o=1}^{O_j} B_j P_{jom} X_{jom} \le T_m \qquad m = 1,2......M \quad \text{............ (2)}$$

$$\sum_{j=1}^{J} \sum_{o=1}^{O_j} S_{jom} X_{jom} \le S_m \qquad m = 1,2......M \quad \text{............ (3)}$$

$$\sum_{j=1}^{J} \sum_{o=1}^{O_j} X_{joG} \le 1 \qquad j = 1,2......J \quad o = 1,2......O_j \quad \text{............ (4)}$$

$$\sum_{o=1}^{O_j} \sum_{m=1}^{M} X_{jom} = X_j O_j \qquad j = 1,2......J \quad \text{............ (5)}$$

The constraint, Eq. (2), ensures that overloading of the machines is not permitted. Eq. (3) is to ensure the jobs will be loaded only when there is availability of tool slots on each machine. Eq. (4) ensures that a particular operation of a job is done on one machine and Eq. (5) ensures that the job cannot be split.

No of jobs considered in this model is 8.no. of jobs. Max no. of operations 3. Tool slots required maximum 5. Machine time max 1000 minutes per machine. The problem is solved using lingo software, the software requirement puts restrictions on the problem size using the integer programming model. The heuristic considered is SPT heuristic and its model is described further.

3. SHORTEST PROCESSING TIME (SPT) ALGORITHM

The following steps are involved in SPT Algorithm.

Step 1: Input the total number of available machines, jobs, batch sizes, tool slots on each machine, operations of all jobs (both essential and optional), and the processing time of each operation of every job.

Step 2: Get the initial set of jobs, machines, processing times for operation one only

Step 3: Get the job whose processing time is shortest and do the following:

a) First, load the essential operation on the machine if and only if the available machining time and available tool slots on the machine is greater than the time and the tool slots required by the essential operation; otherwise, reject the job and terminate

b) Then, load the optional operation on the machine if and only if the available machining time and tool slots on the machine is greater than the time and the tool slots required by the optional operation on the basis of the machine having the maximum available time; otherwise, reject the job and terminate

Step 4: Go to next operation select the machine whose processing time is shortest and do the following

a) First, load the essential operation on the machine if and only if the available machining time and available tool slots on the machine is greater than the time and the tool slots required by the essential operation; otherwise, reject the job and terminate.

b) Then, load the optional operation on the machine if and only if the available machining time and tool slots on the machine is greater than the time and the tool slots required by the optional operation on the basis of the machine having the maximum available time; otherwise, reject the job and terminate.

Step 5: Go to step 4 if further operation exists, else remove the job from the initial set of job list and go to step3.

Step 6 Terminate if all the jobs are assigned.

As there is a deviation observed between integer programming and spt algorithm, a new heuristic is proposed which can give near optimal solution.

4. PROPOSED ALGORITHM

Step 1: Input the total number of available machines, jobs, batch sizes, tool slots on each machine, operations of all jobs (both essential and optional), and the processing time of each operation of every job.

Step 2: Input the number of iterations (n), where (i=1, ... ,n) (the number of job sequences to be generated).

Step 3: Get the initial sequence (i=1) and do the following:

a) First, load the essential operation on the machine if and only if the available machining time and available tool slots on the machine is greater than the time and the tool slots required by the essential operation; otherwise, reject the job.
b) Then, load the optional operation on the machine if and only if the available machining time and tool slots on the machine is greater than the time and the tool slots required by the optional operation on the basis of the machine having the maximum available time; otherwise, reject the job.

Step 4: Terminate if the maximum number of iterations is reached (i=n). Otherwise, go to step 2.

5. CASE STUDY WITH PROPOSED ALGORITHM ON RANDOM SELECTION OF JOBS

First iteration
Sequence of jobs (1,4,8,6,7,5,3,2) ;
Machines = (1,2,3,4); Jobs (1,2,3,4,5,6,7,8); Operations (1,2……)

1. First Job:
For first operation
1a) Processing job i.e (1,3) time = 144 minutes
1b) Compute;

- Job 1starting time on machine 3 = J_1O_1START M_3 = 0 ;
- Job 1completion time on machine 3 = J_1O_1COMPLETION M_3 = J_1O_1START M_3 + 144 = 144 ;
- Machine 3 start job1 operation1 = M_3J_1START O_1 = 0;
- Machine 3 completion job1 operation1 = $M_3J_1O_1$ + 144 = 144 = M_3J_1COMPLETION O_1;
- Tool slot required for machine 3 job1 = TsM_3J_1 =1;
- Total tool slots used on machine 3 = $TTsM_3$ = 0 + TsM_3J_1 = 1;

There is no second operation. Hence job1 is deleted from the sequence.

2. Second Job:
2a) Job 4 is selected. For first operation has to be carried out on machine 1. Processing time of (4,3)=84.
2b) Compute:

- Job 4 operation 1 starting time = max(machine 3 completion time of previous job, job previous operation completion time) = max(144,0) = 144
- Job 4starting time on machine 3 = J_4O_1START M_3 = 144;
- Job 4completion time on machine 3 = J_4O_1COMPLETION M_3 = J_4O_1START M_3 + 84 =228 ;
- Machine3 job 4 operation 1 starting time = max(machine 3 completion time of previous job, job previous operation completion time) = max(144,0) = 144
- Machine 3start job4 operation1 = M_3J_4START O_1 =144;
- machine 3compl job4 operation1 = $M_3J_4O_1$ + 84 = 228 = M_3J_4 COMPLETION O_1;
- tool slot required for machine 3 job4 = TsM_3J_4 =1;
- total tool slots used on machine 3 $TTsM_3$ =1+ TsM_3J_4 = 2;

2c) Operation two for same job .operation two for 4th job will be carried out on machine 4

- job 4 operation 2 starting time = max(machine 4 completion time of previous job,jobprevious operation completion time) = max(0,228)= 228
- Job 4 starting time on machine 4 = J_4O_2START M_4 =228;
- Job 4 completion time on machine 3 operation 2 = J_4O_2COMLETION M_4 = J_4O_2START M_4 + processing time = 228 +114 = 342;
- Machine 4 job 4 operation 2 starting time = max(machine 4 completion time of previous job, same job previous operation completion time) = (0,228) = 228
- Machine 4 start job4 operation 2 = M_4J_4START O_2 = 228;

- Machine 4 completion job4 operation 2= M_4J_4COMPLETION O_2 = M_4J_4START O_2 + 114 = 342;
- Tool slot required for machine 4 job4 = TsM_4J_4 = 1
- Total tool slots used on machine 3 $TTsM_3$ = 0 + TsM_3J_4 = 1;
- Total tool slots used on machine 4 = $TTsM_4$ = 0 + TsM_4J_4 = 1;

2d) Remove job 8 from the list (1,4,8,6,7,5,3,2) job 8 can be performed on any machine (1,2,3) but machine 1 is selected as it has more available time.

3. Third Job:

- For first operation job 8 starting time on machine 1 = J_8O_1START M_1= max(machine 1 completion time of previous job 0, job previous operation completion time) = (0,0) = max(0,0) = 0
- Job 8 completion time on machine 1 = J_8O_1COMPLETION M_1 = 0 +325 = 325 ;
- Machine1start job8 operation 1 = max(machine 1completion time of previous job, job previous operation completion time) = (0,0) = max(0,0) = 0 = M_1J_8START O_1 ;
- Machine 1 completion job8 operation1 = M_1J_8START O_1 + 325 = 325 = M_1J_8COMPLETION O_1;
- Tool slots used on machine 1= $TTsM_1J_8O_1$ = 1;
- Total tool slots used on machine1 = $TTsM_1$ = $TTsM_1J_8O_1$ = 1;
- Total tool slots used on machine 3 $TTsM_3$ = 0 + TsM_3J_4 = 1;
- Total tool slots used on machine 4 = $TTsM_4$ = 0 + TsM_4J_4 = 1;

3a) Operation two for 8th job can be carried out on machine (2,1). But machine availability is high for machine 1. So same job is continued on machine 1

- Job 8 operation 2 starting time machine 1 = max(machine1 completion time of previous job 8, job previous operation completion time) = max(325,325) = 325;
- Job8 starting time on machine1 operation 2 = J_8O_2START M_1 = 325; Job8 completion time on machine 1operation 2 = J_8O_2COMPLETION M_1 = J_8O_2START M_1 + processing time = 325 + 91 = 416;
- Machine1 job 8 operation 2 starting time max(machine1 completion time of previous job8, job previous operation completion time) = max(325,325) = 325;
- Machine1 completion job8 operation2 = M_1J_8START O_2 + 91 = 416 = M_2J_8COMPLETION O_2;
- Tool slots used on machine 1 = $TTsM_1J_8O_2$ = 1 ;
- Total tool slots used on machine 1 = $TTsM_1$ = $TTsM_1J_8O_1$ + $TTsM_1J_8O_2$ = 2;
- Total tool slots used on machine 3 $TTsM_3$ = 0 + TsM_3J_4 = 1;
- Total tool slots used on machine 4 = $TTsM_4$ = 0 + TsM_4J_4 = 1;

3b) Third operation: Job 8 has to be done on machine 1 only as it is essential..

- Job 8 operation 3 starting time machine 1= max(machine1 completion time of previous job, job previous operation completion time) = max(416,416) = 416;
- Job 8 starting time on machine 1 = J_8O_3START M_1 = 416;
- Job 8 comp time on machine 1 operation 3 = J_8O_3COMPLETION M_1 = J_8O_3START M_1 + processing time = 416+312 = 728 ;
- Machine 1 job 8 operation 3 starting time = max(machine 1 completion time of previous job, same job previous operation completion time) = max(416,416) = 416;
- Machine 1 completion job8 operation 3 = M_1J_8START O_3 + 312 = 728 = M_1J_8 COMPLETION O_3;
- Tool slot required for machine 1 job8 = TsM_1J_8 =3;
- Total tool slots used on machine 1 = $TTsM_1$ = $TTsM_1J_8O_1$ + $TTsM_1J_8O_2$ + TsM_1J_8 = 5;
- Total tool slots used on machine 3 $TTsM_3$ = 0 + TsM_3J_4 = 1;

- Total tool slots used on machine 4 = $TTsM_4$ = 0 + TsM_4J_4 = 1;

3c) Delete job from the list

4. Fourth Job:

Job 6 is selected from (6,7,5,3,2).

4a) The job can be carried out on machine 4 only.

4b) Processing time 160 .

4c) Compute;

- For fist operation Job 6 starting time on machine 4 = J_6O_1START M_4 = max(machine 4 completion time of previous job4, job previous operation completion time) = (M_4J_4COMPLETION O_2,0)) = max(342,0) = 342; Job6 completion time on machine 4 = J_4O_1COMPLETION M_4 = J_6O_1START M_4 + 160 = 502 ;
- Machine 4 start job6 operation1 = max(machine4 completion time of previous job, same job previous operation completion time) = max(342,0) = 342 = M_4J_6START O_1 ;
- Machine 4 completion job 6 operation1 = M_4J_6START O_1 + 160 = 502 = M_4J_6COMPLETION O_1;
- Tool slot required for machine 4 job6 = TsM_4J_6 =1;
- Total tool slots used on machine 1 = $TTsM_1$ = $TTsM_1J_8O_1$ + $TTsM_1J_8O_2$ + TsM_1J_8 = 5;
- Total tool slots used on machine 3 $TTsM_3$ = 0 + TsM_3J_4 = 1;
- Total tool slots used on machine 4 = $TTsM_4$ = 0 + TsM_4J_4+ TsM_4J_6 = 2;

4d) Go for next operation.

4e) Operation two for 6th job can be carried out on machine (4,2,3). As there is under usage of machine two it is selected (max available)

- Job 6 operation 2 starting time machine 2 = max(machine 2 completion time of previous job, job previous operation completion time) = max(0,502) = 502;
- Job 6 starting time on machine 2 = J_6O_2START M_2 = 502;
- Job 6 completion time on machine 2 operation 2 = J_6O_2COMPLETION M_2 = J_6O_2START M_2 + processing time = 502 +70 = 572 ;
- Machine 2 job6 operation 2 starting time = max(machine 2 completion time of previous job, same job previous operation completion time) = max(0,502) = 502;
- Machine 2 completion job6 operation 2 = M_2J_6START O_2 + 70 = 572 = M_2J_6COMPLETION O_2;
- Tool slot required for machine 2 job6 = TsM_2J_6 =1;
- Total tool slots used on machine1 = $TTsM_1$ = $TTsM_1J_8O_1$ + $TTsM_1J_8O_2$ + TsM_1J_8 = 5;
- Total tool slots used on machine 3 $TTsM_3$ = 0 + TsM_3J_4 = 1;
- Total tool slots used on machine 4 = $TTsM_4$ = 0 + TsM_4J_4 + TsM_4J_6 =2;
- Total tool slots used on machine 2 = $TTsM_2$ = 0 + TsM_2J_6 = 1;

4f) Check for next operation.

Operation 3 for 6th job can be carried out on machine 2. Machine 2 is selected (maximum available)

- Job 6 operation 3 starting time machine 2 = max(machine 2 completion time of previous job, job previous operation completion time) = max(572,572) = 572;
- Job 6 starting time on machine 2 = J_6O_3START M_2 =572;
- Job 6 completion time on machine 2 operation 3 = J_6O_3COMPLETION M_1 = J_6O_3START M_1 + processing time = 572 + 210 = 882;
- Machine 2 job6 operation 3 starting time = max(machine 2 completion time of previous job, same job previous operation completion time) = max(572,572) = 572;
- Machine 2 completion job6 operation 3 = M_2J_6START O_3 +210 = 882 = M_2J_6COMPLETION O_3;
- Tool slot required for machine 2 job6 = TsM_2J_6 =1;
- Total tool slots used on machine1= $TTsM_1$ = $TTsM_1J_8O_1$ + $TTsM_1J_8O_2$ + TsM_1J_8 = 5;
- Total tool slots used on machine 3 $TTsM_3$ = 0 + TsM_3J_4 = 1;

- Total tool slots used on machine 4 = $TTsM_4$ = 0 + TsM_4J_4+ TsM_4J_6 = 2;
- Total tool slots used on machine 2 = $TTsM_2$ = 0 + $TsM_2J_6O_2$ + $TsM_2J_6O_3$ = 2;

4g) Delete job 6 from the list

5. Fifth Job:
Job 7 is selected from sequence (7,5,3,2).
For first operation
5a) Job 7 can be started on machine (3,2,4.)
5b) Machine 3 available time is more. Hence machine 3 is selected. Processing time 228
5c) Compute;

- For first operation Job 7 starting time on machine 3 = J_7O_1START M_3 = max(machine 3 completion time of previous job1, job previous operation completion time) = (M_3J_7COMPLETION O_1,0) = max(228,0) =228;
- Job 7 completion time on machine 3 = J_7O_1COMPLETION M_3 = J_7O_1START M_3 + 228 = 456 ;
- Machine 3 start job7 operation1 = max(machine 3 completion time of prevous job, same job previous operation completion time) = max(228,0) = 228 = M_3J_7START O_1 ;
- Machine 3 completion job7 operation1 = M_3J_1START O_1 + 228 = 456 = M_2J_7COMPLETION O_1;
- Tool slots required for machine 3 job7 on machine 3 = 1 = TsM_3J_7;
- Total tool slots used on machine 3 = $TTsM_3$ = 0 + TsM_3J_4 + TsM_3J_1 + TsM_3J_5 + TsM_3J_7 = 5;
- Total tool slot required for machine 4 = TsM_4J_1 + TsM_4J_6 = 2;
- Total tool slot required for machine 2 = TsM_2J_6 = 1;
- Total tool slot required for machine 1 = TsM_1J_6 = 1;

5d) Operation 2 for 7th job can be carried out on machine (2,3,1). Machine 1 is selected as machine 3 slots are over. Machine 2 availability is less. Hence machine 1 is selected. Processing time is equal to156.

- Job 7 operation 2 starting time machine 1 = max(machine1 completion time of previous job8, job previous operation completion time) = max(728,456) = 728;
- Job 7 starting time on machine 1 = J_7O_2START M_1 =728;
- Job 7 completion time on machine 1operation 2 = J_7O_2COMPLETION M_1 = J_7O_2START M_1 + processing time = 728+156 = 884;
- Machine 1 job 7 operation 2 starting time = max(machine 1 completion time of previous job, same job previous operation completion time) = max(728,456) = 728;
- Machine 1completion job7 operation 2 = M_1J_7START O_2 + 156 = 884 = M_1J_7COMPLETION O_2;
- Tool slots used on machine 1 for job 7 =1 = TsM_1J_7;
- Total tool slots used on machine 3 = $TTsM_3$ = 0 + TsM_3J_4 + TsM_3J_1 + TsM_3J_5 + TsM_3J_7 = 5;
- Total tool slot required for machine 4 = TsM_4J_1 + TsM_4J_6 = 2;
- Total tool slot required for machine 2 = TsM_2J_6 =1;
- Total tool slot required for machine1 = TsM_1J_6 + TsM_1J_7 = 2;

5e) Go for operation 3

Check for next operation.

- Operation 3 for 7th job can be carried out on machine 4 only.
- Job 7 operation 3 starting time machine 4 = max(machine 4 completion time of previous job6, job previous operation completion time) = max(502,884) = 884;
- Job 7 starting time on machine 4 operation 3 = J_7O_3START M_4 = 884;
- Job 7 completion time on machine 4 operation 3 = J_7O_3COMPLETION M_4 = J_7O_3START M_4 + processing time = 884 + 276 = 1160 which is more than allotted time of the machine ; Hence job 7 cannot be selected.

6. RESULTS AND DISCUSSIONS

The proposed algorithm for the loading problem was coded in Dev-C++ in C language, and the program was used to create .IN and .OUT files displaying the input data and the results. The results include the sequences generated and the system unbalance for each, followed by the minimum system unbalance for the given iterations. The performance of the algorithm is evaluated by using a benchmark problem available in the open literature. The output is displayed by opening the .OUT file using notepad, and exhibiting the screenshot has been demonstrated that the proposed approach is general purpose and can be adopted for any objective function without changing the basic routine

.

S No	Method	Sequence of Jobs	System unbalance(unutilized machine time) in minutes	Throughput (no.s).
1	Integer Programming	1,2,3,6,7,8	908	65
2	SPT sequence	4,1,6,5,2	2164	42
3	Proposed Algorithm	4,7,8,5,6,1	1209	58

a) The results obtained by integer programming are optimum which is 908 minutes and its throughput is 65 units.
b) In SPT sequence we could accommodate only 5 jobs. Its throughput is 42 units where as machine utilized time is 2164 minutes. Here the gap between integer programming and SPT sequence is 1256 minutes. Hence search for another heuristic is necessitated.
c) The proposed new algorithm has unutilized machine time of 1209 minutes after 40 iterations which shows better than SPT sequence.

7.CONCLUSIONS

The primary objective of this paper is to develop an efficient algorithm to solve the machine loading problem of a random FMS. The proposed algorithm reduces the computational burden due to the number of iterations being fixed, and displays the minimum system unbalance achieved within those iterations. Even though this work provides the interesting observations about sequencing flexibility and machine utilization, the result of this study should be interpreted with respect to the considered assumptions and experimental conditions. The results obtained will assist the practitioners in selecting the flexibility strategies and machine loading. The algorithm can be solved using meta heuristics.

REFERENCES

1. Sandhyarani Biswas, S. S. Mahapatra; "*Machine Loading in Flexible Manufacturing System: A Swarm Optimization Approach*"; Eighth Int. Conference on Opers. & Quant. Management October 17-20, 2007.
2. S. K. Mukhopadhyay; M. K. Singh; R. Srivastava; "*FMS machine loading: A simulated annealing approach'*:· International Journal of Production Research.
3. Akhilesh Kumar, Prakash , M.K. Tiwari, Ravi Shankar, Alok Baveja; *asolving machine-loading problem of a nexible manufacturing system with constraint-based genetic algorithm'*:· European Journal of Operational Research 175 (2006) 1043-1069
4. Felix T. S. Chan, Rahul Swamkar and Manoj K. Tiwari,· "*A Random Search Approach to the Machine Loading* Problem of an FMS"; lntemational Symposium on Intelligent Control Taipei, Taiwan, September 2-4,2004.
5. N. K. Vidyarthi; M. K. Tiwari; "*Machine loading problem of FMS: A fuzzybased heuristic approach*"; International Journal of Production Research.14 Nov, 2010
6. Roby Thomas, Marvin D. Troutt; "*Comparison of objective criteria for setup planning in complementary ffexible manufacturing systems*" ;Computers & Industrial Engineering 53 (2007) 17-29.27
7. Tiwari MK, Hazarika B, Vidyarthi NK, Jaggi P, Mukhopadyay SK. *A heuristic solution approach to the machine-loading problem of an FMS and its Petri net model*. lnt J Production Res 1997;35.

8. Shanker K, Tzen YJ. *A loading and dispatching problem in a random flexible manufacturing system*. lnt J Production Res 1985;23
9. Berrada, M and Stecke, K.E. "*A Branch and Bound Approach for Machine Load Balancing in Flexible Manufacturing Systems*." Management Science. 1986. p.1316–1335.
10. Kennedy, J and Eberhart, R.C. "*Particle Swarm Optimization*." Proceedings of IEEE International Conference on Neural Networks. 1995. p.1942-1948.
11. Kumar, N and Shanker, K. "*A Genetic Algorithm for FMS Part Type Selection and Machine Loading*. International Journal of Production Research. 2000. p.3861–3887.
12. Moreno, A.A and Ding, F.Y. *Heuristics for the FMS Loading and Part Type Selection Problems.* International Journal of Flexible Manufacturing System.1993. p.287–300.
13. Mukhopadhyay, S.K, Midha, S and Muralikrishna, V. "*A Heuristic Procedure for Loading Problems in Flexible Manufacturing Systems.*" International Journal of Production Research .1992. p.2213-2228.
14. Nagarjuna, N, Mahesh, O and Rajagopal, K. "A Heuristic based on Multi-Stage Programming Approach for Machine-Loading Problem in a Flexible Manufacturing System." Robotics and Computer Integrated Manufacturing. 2006. p.342-352.
15. Riget, J and Vesterstroem, J.S. "A Diversity Guided Particle Swarm Optimiser - the ARPSO." Dept. of Computer Science, University of Aarhus, Technical Report. No. 2002-02 EVA Life. 2002.
16. Shanker K, Srinivasulu A. "*Some Solution Methodologies for Loading Problems in a Flexible Manufacturing System*." International Journal of Production Research. 1989. p.1019–1034.
17. Stecke, K.E. "*Formulation and Solution of Non-Linear Integer Production Planning Problem for Flexible Manufacturing System*." Management Science. 1983. p.273–288.
18. Swarnkar, R and Tiwari, M.K. "*Modeling Machine Loading Problem of FMSs and its Solution Methodology is using a Hybrid Tabu Search and Simulated Annealing-Based Heuristic Approach*. Robotics and Computer Integrated Manufacturing. 2004. p.199–209.
19. Tiwari, M.K, Hazarika, B, Vidyarthi, N.K, Jaggi, P and Mukhopadhyay, S.K. "*A Heuristic Solution Approach to the Machine Loading Problem of an FMS and its Petri Net Model*." International Journal of Production Research.1997. p.2269-2284.
20. Tiwari, M. K., Saha, J and Mukhopadhyay, S. K. "*Heuristic Solution Approaches for Combined-Job Sequencing and Machine Loading Problem in Flexible Manufacturing Systems.*" International Journal of Advanced Manufacturing Technology. 2007. p.716–730.
21. Vidyarthi, N.K and Tiwari, M.K. "*Machine Loading Problem of FMS: A Fuzzy-Based Heuristic Approach.*" International Journal of Production Research. 2001. P. 953–979.
22. Yoshida, H, Kawata, K, Fukuyama, Y and Nakanishi, Y. "*A Particle Swarm Optimization for Reactive Power and Voltage Control Considering Voltage Security Assessment*." IEEE Transactions on Power Systems. 2000. p.1232-1239.