RESEARCH PAPER | Engineering | Volume : 4 | Issue : 8 | August 2014 | ISSN - 2249-555X

Particle Swarm Optimization Algorithm for Machine Loading Problem in F.M.S

KEYWORDS: Flexible manufacturing systems, PSO algorithm, system unbalance, throughput

Mr. A. Somaiah	Dr. M. Indira Rani	Mr. Ch. Varun Kumar
Department of Mechanical Engineering, Institute of Aeronautical Engineering, Dundigal Village, Quthbillahpur Mandal, Hyderabad, 500043	Department of Mechanical Engineering, JNTUH, Hyderabad 500085	CSE III/I, Institute of Aeronautical Engineering, Dundigal Village, Quthbillahpur Mandal, Hyderabad, 500043

ABSTRACT *FMS operations can be broadly divided into pre-release and post-release decisions. Pre-release decisions include the FMS operational planning problem that deals with the pre-arrangement of jobs and tools before the processing begins whereas post-release decisions deal with the scheduling problems. Pre-release decisions viz., machine grouping, part type selection, production ratio determination, resource allocation and loading problems must be solved while setting up of a FMS. Amongst pre-release decisions, machine loading is considered as one of the most vital production planning problem because performance of FMS largely depends on it. The problem is to assign the machines, the operations of the selected jobs and the tools necessary to perform the operations by satisfying the technological constraints in order to obtain minimum system unbalance and maximum throughput. The machine loading problem addressed in this is that although machine capacity might be sufficient, it may not be possible to process all job orders required in particular planning period due to limited number of tool slots and available machine time. Thus, subsets of job orders are to be processed. In the past, numerous techniques have been suggested and found to be efficient, but they take long computational times when the problem size increases. In order to address the above issues, a meta-heuristic approach based on particle swarm optimization (PSO) has been proposed to improve the solution quality and reduce the computational time.*

1.0 INRODUCTION

FMS is characterized as an integrated, computer controlled complex arrangement of automated material handling devices and computer numerically controlled (CNC) machine tools that can simultaneously process medium sized volumes of a variety of part types. The aim of FMS is to achieve the efficiency of automated high volume mass production while retaining the flexibility of low volume of job shop production.

Loading problem in particular deals with allocation of jobs to various machines under technological constraints with the objective of meeting certain performance measures. The machine loading problem addressed in this is that although machine capacity might be sufficient, it may not be possible to process all job orders required in particular planning period due to limited number of tool slots and available machine time. Thus subset of job orders is to be processed. It is very difficult to evaluate all possible combinations of operation-machine allocation in order to achieve minimum system unbalance and maximum throughput.

As a new approach to enhance the solution quality for machine loading problem, an iterative method using particle swarm optimization (PSO) is proposed. The objective function is to minimize the system unbalance and maximize the throughput.

Particle swarm optimization (PSO) technique developed by Dr. R.C Eberhart and Dr. Kennedy in 1995 inspired by social behavior of bird flocking or fish schooling.

PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. For all iterations, particles are updated by following two "best" values. The first one is the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called pbest. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called gbest.

The particle updates its velocity (1), position (2) and inertia (3) with the following equations.

$$V_{ij}^{t} = w^{t} \cdot V_{ij}^{t-1} + c_1 r_1 (P_{ij}^{t-1} - X_{ij}^{t-1}) + c_2 r_2 (g_j^{t-1} - X_{ij}^{t-1}) \quad \text{--1}$$

$$X_{ij}^{t} = V_{ij}^{t} + X_{ij}^{t-1} \quad \text{--2}$$

$$w^{t} = w^{t-1} \times \alpha \quad \text{--3}$$

Where,

V_{ij}^{t} Is velocity of particle i at iteration t with respect to j^{th} dimension ($j = 1, 2, \ldots\ldots n$).

P_{ij}^{t-1} Is the personal best position in previous iteration of i^{th} particle.

X_{ij}^{t} Is the position value of the i^{th} particle with respect to j^{th} dimension at iteration t.

X_{ij}^{t-1} Is previous position value of the i^{th} particle with respect to j^{th} dimension.

g_j^{t-1} Is global best value of the particle among whole swarm in previous iteration (swarm size is 8).

c_1 and c_2 are positive acceleration parameters, called cognitive and social parameter, respectively and r_1 and r_2 are uniform random numbers between (0,1). w is known as inertia weight, α is a decrement factor. The parameter 'w' controls the impact of the previous velocities on the current velocity.

Using this meta-heuristic, based on position values, by applying SPV rule the job sequence is obtained and then allocation in machines is done. The main advantage of this algorithm is that within few iterations (less than to 20) the required objective function can be obtained, whereas oth-

Volume : 4 | Issue : 8 | August 2014 | ISSN - 2249-555X

er meta-heuristics are considered they require more number of iterations (about 50). So, by reducing number of iterations the computational time reduces and it requires less computational effort too.

2.0 Machine Loading Problem Description

The machine loading problem in a flexible manufacturing system (FMS) is specified as to assign the machine, operations of the selected jobs, and the tools necessary to perform these operations by satisfying the technological constraints in order to ensure the minimum system unbalance and maximum throughput, when the system is in operation. The loading problem addressed in this is that, although machine capacity might be sufficient, it may not be possible to process all the job orders required in particular planning period due to limited number of tool slots and available machining time. Thus subsets of job orders are to be processed. Jobs are available in batches and each job has one or more operations. Each operation can be performed by one or more machines. The processing time and tool slots required for each operation of the job and its batch size are known before hand. Essential and optional types of operations are allied with each job. An essential operation of a job means that this operation can be performed only on a particular machine using a certain number of tool slots. Whereas, optional operations imply that they can be carried out on number of machines with same or varying processing time and tool slots. In this problem, the flexibility lies in the selection of a machine for processing the optional operation of the job. The operation machine allocation combinations are to be evaluated using two common yardsticks, system unbalance and throughput. System unbalance can be defined as the sum of underutilized and over utilized time on all the machines available in the system. Minimization of the system unbalance is same as maximization of machine utilization, whereas throughput refers the sum of batch size for all the selected jobs during the planning horizon.

Job No.	Operation No.	Batch Size	Unit Processing time(min)	Machine No.	Tool Slot needed
1	1	8	18	3	1
2	1	9	25	1,4	1
	2		24	4	1
	3		22	2	1
3	1	13	26	4,1	2
	2		11	3	3
4	1	6	14	3	1
	2		19	4	1
5	1	9	22	2,3	2
	2		25	2	1
6	1	10	16	4	1
	2		7	4,2,3	1
	3		21	2,1	1
7	1	12	19	3,2,4	1
	2		13	2,3,1	1
	3		23	4	3
8	1	13	25	1,2,3	1
	2		7	2,1	1
	3		24	1	3

Table (1) Description of problem

3.0 Formulation of Objective Function and Constraints

The above-described problem is formulated as the bi-criterion objective problem where the two objectives are combined.

The first objective is to minimize the system unbalance:
Minimize

$$\sum_{m=1}^{M}(UT_m - OT_m)$$

This is equivalent to maximize the system utilization:
Maximize

$$F_1 = \frac{M*H - \sum_{m=1}^{M}(UT_m - OT_m)}{M*H} \text{------------(1)}$$

The second objective is maximizing throughput or equivalently maximizing the system efficiency.
Maximize

$$F_2 = \frac{\sum_{j=1}^{J} B_j * X_j}{\sum_{j=1}^{J} B_j} \text{---------------------(2)}$$

Thus the overall objective function is

$$\text{Maximize } F = \frac{M*H - \sum_{m=1}^{M}(UT_m - OT_m)}{M*H} + \frac{\sum_{j=1}^{J} B_j * X_j}{\sum_{j=1}^{J} B_j}$$

$$\text{Thus } F = F_1 + F_2 \qquad \text{-------------(3)}$$

Constraints:

1. Tool Slots:

This constraint guarantees that the number of tool slots needed for the operation of the jobs to be performed on a machine must always be less than or equal to the total tool slots available in that machine. This constraint can be expressed as:

$$\sum_{K=1}^{P_i} T_{jm}\, x_{ijk} \le T_{jm}^{a} \quad i = 1, 2, \ldots N$$

2. Available time on machine:

The available time on each machine should be greater than or equal to the time required by the next job to be assigned to this machine.

$$\sum_{K=1}^{P_i} t_{jm}\; x_{ijk} \le t_{jm}^{a} \quad i = 1,2,\ldots,N$$

3. System Unbalance:

System unbalance equals to the sum of the idle time remaining on machines after allocation of all feasible jobs. The value of system unbalance must either be zero (100% utilization of system) or a positive value.

$$\sum_{m=1}^{M}(UT_m - OT_m) \ge 0$$

4. Non-Splitting of Job:

This constraint implies that once a job is considered for processing, all the operations are to be completed before undertaking a new job.

5. The number of tool slots and remaining time on any machine after any assignment of job should always be positive or zero.

6. Integrity of decision variables:

The decision variables possessing the value of 0 and 1 integers are as follows:

x_i = 1 if job i is selected
0 otherwise, i= 1,2,…,N
x_{ij} =1 if job i is assigned to machine j
0 otherwise, i=1,2,…,N, j=1,2,…,M

Volume : 4 | Issue : 8 | August 2014 | ISSN - 2249-555X

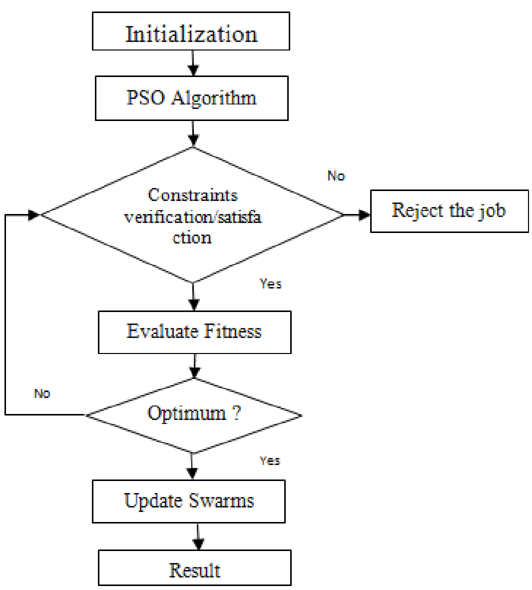
x_{ijk}=1 if operation k of job i is assigned to machine j
0 otherwise, i=1,2,..,N,j=1,2,..,M,k=1,2,..,P

4.0 PSO Algorithm:

t = 0;
for (j=1,2..J); //swarm size
Generate V_{ij}^{t}, X_{ij}^{t}; //X in between 0 & 4, V in between -4 & 4
Search on all particle positions;
Obtain job sequence using SPV rule;
Evaluate F; //objective function value
$P_{ij}^{t} \longrightarrow X_{ij}^{t-1}$; //personal best position is
Previous iterations best position of that particle
$G^{t} \longrightarrow X_{ij}^{t-1}$; //Global best position is
Previous iterations best position among
all particle positions
While ($t<t_{max}$)
{
t = t+1;
for (j=1,2..J);
update velocity V_{ij}^{t}
update position X_{ij}^{t}
search on all particle positions;
obtain job sequence using SPV rule;
Evaluate F;
Update P_{ij}^{t} & G^{t}
}

This paper proposes particle swarm optimization algorithm for machine loading problem in FMS. In this problem; the number of jobs are considered as number of particles and the particles initial positions are selected with in the search space of 0 to 4 and velocities are within the range of -4 to +4. This algorithm uses smallest position value (SPV) rule for sequencing the jobs.

5.0 Procedure to Enumerate System Unbalance and Throughput: Flow chart



6.0 Results

System Unbalance: 14
Throughput: 48
Assigned jobs are: 4 7 3 5 1
Unassigned jobs are: 2 6 8
Best Sequence: 4 7 2 3 5 6 1 8

Total No. of jobs	M.K.Tiwari et al.[3]		N.K.Vidyarthi et al.[2]		M. K. Tiwari & N.K. Vidyarthi,et al.[4]		Akhilesh et al.[5]		Sandhyarani et.al. [6]		Proposed PSO heuristic	
8	SU	TH	SU	TH	SU	TH	SU	TH	SU	TH	SU	TH
	76	42	122	42	14	48	0	36	253	39	14	48
Objective function	1.4854		1.4614		1.5927		1.4500		1.3557		1.5927	

Table (2) Comparison of Results

7.0 CONCLUSIONS

As a new approach to enhance the solution quality for machine loading problem, this thesis proposes an iterative method using particle swarm optimization (PSO). The same machine loading problem (4 machines, 8 jobs) by Sandhyarani Biswas et.al.[6] using Modified PSO algorithm and arrived at SU = 3, TH = 49, by permitting the overloading and SU=253, TH=39 without permitting the overloading. But we attempted the same problem with PSO algorithm without permitting the overloading and arrived at SU = 14, TH = 48. The main advantage of this algorithm is that within few iterations (less than to 20) the required objective function can be obtained, whereas other meta-heuristics are considered they require more number of iterations (about 50). So, by reducing number of iterations the computational time reduces and it requires less computational effort too. The PSO algorithm is coded in C^{++} programming language to obtain the job sequence and based on that sequence the machine loading problem is solved, at every iteration the system unbalance, throughput and objective function value is generated and compared with manual calculations as well as with the results of open literature.

From the comparative study (table 2), it has been observed that the proposed algorithm offers better results.

REFERENCE

[1] Kennedy J, Eberhart RC (1995) "Particle swarm optimization". In: Proceedings of the 1995 IEEE International Conference on Neural Networks (ICNN'95), Perth, Australia, November/December 1995, vol 4, pp 1942–1948. | [2] N.K.Vidyarthi and M.K.Tiwari. "Machine loading problem of FMS: a fuzzy-based heuristic approach", International Journal of Production Research, 2001, 39(5), 953-979. | [3] M.K.Tiwari, B.Hazarika "Heuristic solution approach to the machine loading problem of an FMS and its Petrinet model", International Journal of Production Research,1997, 35(8), 2269-2284. | [4] M. K. Tiwari & N.K. Vidyarthi (2000): Solving machine loading problems in a flexible manufacturing system using a genetic algorithm based heuristic approach, International Journal of Production Research, 38:14, 3357-3384 | [5] Akhilesh kumar, prakash, MK Tiwari (2006), "Solving machine loading problem of a FMS with constraint based GA", European Journal of Operational Research 175(2006) 1043–1069 | [6] Sandhyarani Biswas, S S Mahapatra, "Modified particle swarm optimization for solving machine-loading problems in flexible manufacturing systems". Int J Adv Manuf Technol (2008) 39:931–942. | |