International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 8
August 2017

Solving Machine Loading Problem in Flexible Manufacturing System Using Particle Swarm Optimisation and Gravitational Search Algorithms.

Priyabrata Samal
Indira Gandhi Institute of Technology ,Sarang , Odisha

ABSTRACT:-

In order to sustain in this competitive market every industry has to modify their design and operation to achieve higher rate of customer response. The new trend of modern manufacturing system and Computer integrated manufacturing system has given a flexibility to achieve the target based higher productivity. A flexible manufacturing system is an integrated computer controlled configuration which allows the system to react when any changes occurred. Manufacturing industry extensively using the simulation and trying to model the impact of variability on the hole system and indicating the uncertainty and helps to find the optimal solution to the problems. Basically in FMS the work machine usually CNC machines connected with material handling system and central computer to control the flow of materials movements and machine flow. The concentration is devoted to the integration with auxiliary programs to ensure usability of the developed models by inexperienced users. The feasibility of integration at this stage is demonstrated by designing, developing, and implementing and showing that it can be customized to be used for simulation in FMS. The simulation part is realized by using MAT LAB 10, the performance of the existing flexible manufacturing system under different algorithm rules is discussed, providing a means to apply these rules into computer simulation models of flexible manufacturing systems.

KEYWORD-

INTRODUCTION

The need for flexible processes is to allow rapid low cost switching from one product line to another. This is possible with flexible workers whose multiple skills would develop the ability to switch easily from one kind of task to another. As main resources, flexible processes and flexible workers would create flexible plants which can adapt to changes in real time, using movable equipment, knockdown walls and easily accessible and re-routable utilities. The process of constructing an FMS is costly as it requires heavy capital investment in machinery and equipment. Because of that, the design of FMS's requires an intensive work on planning an efficient and effective system. Simulation shows up at this stage providing managers with a tool that helps to evaluate the results of different configurations of hardware and software for the production of a variety of available products. Simulation helps find optimal solutions to a number of problems at both design and application stages serving to improve the "flexibility" of FMS's.

The advantages of FMS over stand alone NC are that the production of several products can be intermixed, and the production rates are much higher. Instead of batching the products one at a time on a NC machine to meet requirements, the various products can be made simultaneously on the system. The set up time for changeover is minimized with an FMS, so that the economic batch reduces to one at the same time, and the average production rate increases. Intermixing of products on the system permits the output rate of each product to be set at its corresponding demand rate. This reduces the work in- process and final product inventories that are so typical of batch production methods. FMS operational decisions consist of pre-release and post release decisions. FMS planning problems also known as pre-release decisions take into account the pre-arrangement of parts and tools before the process of FMS begins. FMS scheduling problems, which come under the category of post release decisions, deal with the sequencing and routing of the parts when the

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 8
August 2017

system is in operation . The machine loading problem in a FMS is specified as to assign the machine, operations of selected jobs, and the tools necessary to perform these operations by satisfying the technological constraints (available machine time and tool slots constraint) in order to ensure the minimum system unbalance and maximum throughput, when the system is in operation . An attempt has been made to solve the objective function simultaneously to bring the outcomes in close proximity to the real assumption of the FMS environment.

This paper is concerned with Solving Machine Loading Problem in Flexible Manufacturing Systems Using Particle Swarm Optimization and Gravitational Search Algorithm in Flexible Manufacturing (FMS) environment. We discuss the Problem of Flexible Manufacturing Systems is highlighted and how schedule the machines and retrieval machine can provide a solution for a particle swarm optimization (PSO). Algorithm and Gravitational Search Algorithm to solve machine loading problem in flexible manufacturing system (FMS), with bi criterion objectives of minimizing system unbalance and maximizing system throughput in the occurrence of technological constraints such as available machining time and tool slots. A mathematical model is used to select machines, assign operations and the required tools. An attempt has been made to solve the objective function simultaneously by both of the algorithms to bring the outcomes in close proximity to the real assumption of the FMS environment.

MACHINE LOADING PROBLEM

Machine loading problem of a flexible manufacturing system is known for its complexity. The machine loading problem in an FMS is specified as to assign the machines, operations of selected jobs, and the tools necessary to perform these operations by satisfying the technological constraints (available machine time and tool slot constraint) in order to ensure the minimum system imbalance and maximum throughput, when the system is in operation.

Prior to production, careful operational planning is essential to establish how well the system interacts with the operations over time. Hence, successful operation of FMS requires more intense planning as compared to any conventional production system.

The decisions related to 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 , 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. Loading problem, in particular, deals with allocation of jobs to various machines under technological constraints with the objective of meeting certain performance measures.

Generally, the complexity of these problems depends on whether the FMS is of a dedicated type or a random type. A dedicated FMS is designed to produce a rather small family of similar parts with a known and limited variety of processing requirements while in a random-type system a large family of parts having a wide range of characteristics with random elements is produced and the product mix is not completely defined at the time of installing the system.

ASSUMPTIONS

In order to minimize the complexities in analyzing the problem for a practical FMS, the mathematical model is based on the following assumptions:-

- Initially, all the jobs and machines are simultaneously available.
- Processing time required to complete an entire job order is known a priori.
- Job undertaken for processing is to be completed for all its operation before considering a new job.

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 8
August 2017

- Operation of a job once started on a machine is continued until it is completed.
- Transportation time required to move a job between machines is negligible.
- Sharing and duplication of tool slots is not allowed
- Number of pallets and fixtures used in the system are sufficient and readily available.
- Parts are readily available on machines so material handling time is negligible.

The FMS under consideration in this paper consists of a number of multifunctional CNC machines, tools with the potential to execute several operations. The jobs are available in batches and arrive in random sequences with different requirements for processing. The batch size, number of operations, processing time and number of tool slots needed for each job is known initially. There are-two types of operations accessible for a job namely Essential operation – job can be only performed in a particular machine

Optional operation – job can be performed in a number of machines available

Optional operation gives the flexibility in routing of the jobs. The FMS considered has four multifunctional machines with each having 480mins of available processing time (8hrs = 1 shift) and 2 tool slots.

NOTATIONS

- i -index of job; $1 \leq i \leq N$
- j -index of machines; $1 \leq j \leq M$
- N-number of jobs, swarm size
- M- number of machines
- Bi -batch size of job i
- Xi -1 if job i is selected or = 0 otherwise
- H -length of scheduling period
- UTj- underutilized time in machine j
- OTj- over utilized time in machine j
- SU- system unbalance
- TH- Throughput
- RTMj- remaining time on machine j
- RTSj -remaining tool slot on machine j
- S -randomly generated job sequence
- AS- assigned jobs
- UAS- unassigned jobs
- RTMj -Remaining time on machine j
- RTSj -Remaining number of tool slots in machine j
- t -Iteration number
- C_1, C_2, C_3 .....Learning coefficients
- P_k^t- Position of the k^{th} particle at time step t
- $e_{P_k}^t$ Position of the best previous position of the k^{th} particle at time step *t*
- .$Z(e_{P_k}^t)$ - Objective function value of the sequence represented by the position of the particle $e_{P_k}^t$
- G^t The global best particle position at time step t having the minimum objective function value
- v_x^t- velocity of the particle x at time step t

OBJECTIVE

The objective functions and the constraints are discussed in this section..

Minimize system unbalance:

Equals to the sum of the idle time remaining on the machines after allocation of all feasible jobs. The value of system unbalance must be either 0 (100% utilization of the system) or a positive value..

ISSN 2394 – 3386
Volume 4, Issue 8
August 2017

Maximise F1= $\frac{M\times H-\sum_{J=1}^{M}(U\ _J-O\ _J)}{M\times H}$

Maximize throughput: is equals to the sum of batch size for all the selected jobs during the planning horizon.

$$\text{Minimise \_ F2}=\frac{\sum_{i=1}^{N}B_i\times x_i}{\sum_{i=1}^{N}B_i}$$

Thus the overall objective function is

Maximise_ COF = $\frac{M\times H-\sum_{J=1}^{M}(U\ _J-O\ _J)}{M\times H}+\frac{\sum_{i=1}^{N}B_i\times x_i}{\sum_{i=1}^{N}B_i}$

$$X_j = \begin{cases}1 & i\ j\ \ J\ i\ s\\ 0 & O\ he\ \ w\end{cases}$$

$$X_{jc} = \begin{cases}1 & i\ \ o\ \ \ \ U\ o\ j\ \ J\ i\ a\\ 0 & O\ he\ \ w\end{cases}$$

PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) algorithm, originally introduced by Kennedy and Eberhart in 1995[1], is a population-based evolutionary computation technique. It is motivated by the behaviour of organisms such as bird flocking and fish schooling. In PSO, each member is called particle, and each particle moves around in the multidimensional search space with a velocity which is constantly updated by the particle's own experience and the experience of the particle's neighbour or the experience of the whole swarm. The members of the entire population are maintained throughout the search procedure so that information is socially shared among individuals to direct the search towards the best position in the search space. Two variants of the PSO algorithm have been developed, namely PSO with a local neighbourhood, and PSO with a global neighbourhood. According to the global neighbourhood, each particle moves towards its best previous position and towards the best particle in the whole swarm, called the gbest model in the literature. On the other hand, based on the local variant so called the pbest model, each particle moves towards its best previous position and towards the best particle in its restricted neighbourhood. Generally, PSO is characterized as a simple heuristic of well balanced mechanism with flexibility to enhance and adapt to both global and local exploration abilities.

The mathematic description of PSO is , suppose i denotes a particle and the dimension of the searching space is n . The position of the i^{th} particle at iteration t in n dimension space is represented as

$X^t_i = (x_{i1}{}^t\ ,x_{i2}{}^t\text{...........}x_{in}{}^t)$.

The p best of the i^{th} particle at iteration t in the n dimensional space represented as

$P_i{}^t = (\ p_{i1}{}^t, p_{i2}{}^t,\text{...........}p_{in}{}^t)$.

The index of gbest i.e the best pbest among all the particle is represent by the symbol

$G = (\ g_1{}^t, g_2{}^t\text{........}g_n{}^t)$.

The velocity for the i^{th} particle at iteration t in n dimensional space is represented as

$V_i{}^t = (\ v_{il}{}^t + v_{i2}{}^t\text{.......}v_{in}{}^t)$.

PSO is initialized with a population of random solution of the objective function . After finding the personal best and global best values , velocities and positions of each particle are updated using Eq.11 and 12 respectively .

$$V_{ij}{}^t = W^{t-1}V\ _{ij}{}^{t-1} + c_1r_1(p_{ij}{}^{t-1} - x_{ij}{}^{t-1})\text{................(}$$

$$X_{ij}{}^t = x_{ij}{}^{t-1} + v_{ij}{}^t\ \text{...()}$$

Where $v_{ij}{}^t$ represents velocity of particle i at iteration t with respect to j_{th} dimension (j =1,2,……n). $P_{ij}{}^t$ represents the position value of the i^{th} personal best with respect to the j^{th} dimension. $X_{ij}{}^t$ the position value of

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 8
August 2017

the i th particle with respect to jth dimension. c1 and c2 are positive acceleration parameters, called cognitive and social parameter, respectively and r1 and r2 are uniform random numbers between (0,1).

$W^t = w^{t-1} \times \alpha$ .....(13)

where α is a decrement factor. The parameter 'w' controls the impact of the previous velocities on the current velocity. Termination criterion might be a maximum number of iteration or maximum CPU time to terminate the search.

PROPOSED ALGORITHMS

A. Initialization

Parameters such as swarm size, number of generations (t), constants Cl, C2 and C3 must be initialized. The job/machine data (test problem) will be inputted in this phase. The swarm size used is equivalent to the number of jobs. The termination condition and the Cl, C2, C3 are identified after conducting sensitivity analysis.

B. Particle Generation

The seed sequence is created using the widely used SPT rule.. The searching phase is iterated to the predetermined number of iterations. Lastly, the final optimum result will be obtained via the global best variable.

C. Velocity Initialization

After the swarm is initialized, each potential solution is

assigned a velocity randomly. Length of velocity of each particle ||v|| is generated randomly between 0 and n. And the corresponding lists of transpositions $(i_q,j_q)q= 1$, $\|v_k\|$ are generated randomly for each particle. The above formulation .

Velocity and Position Update

Particles velocity is continuously updated using equation and the particles position is continuously updated using equation

$$v_k^{t+1} = C_1U_1v_k^t + C_2U_2\left(e_{P_k^t} - P_k^t\right) + C_3U_3(G - P_k^t).....$$

$$P_k^{t+1} = P_k^t + v_k^{t+1}.....$$

Where $c_1,c_2,c_3,...$ are integers and $U_1,U_2,U_3,.....$ random number between 0 and 1.

Gravitational Search Algorithm (GSA)

In the proposed algorithm, agents are considered as objects and their performance is measured by their masses. All these objects attract each other by the gravity force, and this force causes a global movement of all objects towards the objects with heavier masses. Hence, masses cooperate using a direct form of communication, through gravitational force. The heavy masses- which correspond to good solutions - move more slowly than lighter ones, this guarantees the exploitation step of the algorithm. In GSA, each mass (agent) has four specifications: position, inertial mass, active gravitational mass, and passive gravitational mass. The position of the mass corresponds to a solution of the problem, and its gravitational and inertial masses are determined using a fitness function.

The GSA could be considered as an isolated system of masses. It is like a small artificial world of masses obeying the Newtonian laws of gravitation and motion. More precisely, masses obey the following laws:

Law of gravity: each particle attracts every other particle and the gravitational force between two particles is directly proportional to the product of their masses and inversely proportional to the distance between them, R. We use here R instead of R^2, because according to our experiment results, R provides better results than R^2 in all experimental cases.

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 8
August 2017

Law of motion: the current velocity of any mass is equal to the sum of the fraction of its previous velocity and the variation in the velocity. Variation in the velocity or acceleration of any mass is equal to the force acted on the system divided by mass of inertia.

Now , consider a system with N agents (masses)

$XX_i = \left(x_i^1 \dots\dots x_i^d \dots\dots x_i^n\right)$ f i 1,2, … … … … N.....(6)

Where x_i^d p $the\ p$ o $thei^{th}\ a$ $i\imath$ $the\ d^{th}\ di$

At specific time 't' we defined the force acting on mass i from mass j

$$F_i^d(t) = G\,(t)\frac{M_p\,(t)\times M_a\,(t)}{R_i\,(t)+\varepsilon}\left(x_j^d(t) - x_j^d(t)\right).....(7)$$

Where M_a is the active gravitational mass related to agent j ,M_p the passive G(t) is gravitational constant at time t ,ε is a small constant , and $R_{ij}(t)$ is the Euclidian distance between two agents i and j

$$R_i\,(t) = \left|\left|X_i(t), X_j(t)\right|\right| 2$$

The total force acts on the agent i in a dimension d be a randomly weighted sum of d_{th} components of the forces exerted from other agents

$$F_i^d(t) = \sum_{j=1, j\neq i}^{N} r_{\imath}\quad d_j\, F_i^d(t).....(9)$$

Where $rand_j$ is a random number thin the interval [0,1]

By the law of motion the acceleration of the agent i at time t in direction d^{th},$a_i^d(t)$, is given as

$$a_i^d(t) = \frac{F_i^d(t)}{M_g(t)}$$

Where M_g is the inertial mass of i^{th} agent

The next velocity of an agent is considered as a fraction of its own current velocity added to its acceleration .

$$v_i^d(t+1) = r_{\imath}\quad d_i \times v_i^d(t) + a_i^d(t) \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots..(11)$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \dots\dots\dots\dots\dots\dots\dots(12)$$

Where $rand_i$ is uniform random variable in the interval [0,1]

The gravitational constant G, is initialised at the beginning and will be reduced with time to control the search accuracy in other word G is a function of the initial value G_0 and time t.

G(t)= G(G_0,t).

Gravitational and inertia masses are simply calculated by the fitness evaluation .A heavier masses means more efficient agent , this means that better agent have higher attraction and walk more slowly .Assuming the equality of the gravitational and inertia masses the values of masses are calculated using the map of fitness .

$M_{ai}=M_{pi}=M_{ii}==M_i$, i= 1,2,.....N

$$m_i(t) = \frac{f\quad_{\imath}(t) - w\qquad(t)}{b\quad\ (t) - \ w\qquad(t)}$$

$$m_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)}$$

where fit_i (t) represent the fitness value of the agent i at time t, and, worst(t) and best(t) are defined as follows (for a minimization problem)

best(t)= $m\quad_{j=1,2,\dots N} f\quad_j(t)$

worst(t)= $m\quad_{j\in\{1,2,\dots.n\}} f\quad_j(t)$

for maximization problem this two equation changes to

best(t)= $m\quad_{j\in\{1,2,\dots.n\}} f\quad_j(t)$

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 8
August 2017

worst(t)= $m_{j=1,2,...N} f_j(t)$

Taking the data from the problem a random sequence is formed and utilising SPT rule the iterations are being formed .

Total processing time for job 1 = $B\ \ h\ s\ z\ \times U\ \ P\ \ ti$ Total processing time for job = $o_{\ } 1(B\ \ h\ s_i\ \times U\ \ P\ \ ti_{\ }) + j_{\ } 2(B\ \ h\ s\ \times U\ \ P\ \ ti_{\ })$ Similarly integrating the optional function and compulsory functions we get the job tables are as followings

For batch size 15 and unit processing time 10 the total processing time to complete the job = 15 ×10 = 150

For batch size 10 containing 2 number of operations 1 & 2 simultaneously

For first job it takes 10 ×20 = 200

For the second operation it takes 10 ×35 = 350

So the total time to complete the job = 200+ 350= 550

Similarly all the jobs having the alternate option for processing is sequences in a series and the total processing time is taken in mat lab foe making a number of iteration to determine the minimum tool span and maximum utilising of machine .

The analysis and comparisons made in the matlab 10

JOB	BATCH SIZE	NUMBER OF OPERATION	OPERATION	MACHINE	UNIT PROCESSING TIME	TOOL SLOT	TOTAL PROCESSING TIME
1	15	1	1	4	10	2	150
				2	12	2	180
2	10	2	1	1	20	1	200
			2	3	35	2	350
3	12	1	1	1	22	3	264
4	9	1	1	3,2	25	1	225
5	16	2	1	4	30	2	480
				2	25	1	400
				3	27	2	432
			2	1,4	16	1	256
6	11	1	1	2	21	3	231

Matrix formed initially for 6 no of jobs and 4 number of machine are The unit processing time matrix is as

at=[150 550 264 225 736 231 ;
180 550 264 225 736 231 ;
150 550 264 225 625 231 ;
180 550 264 225 625 231 ;
150 550 264 225 688 231:
180 550 264 225 688 231]

Finding out the Gbest , Lbest , and tool span are

L best value

1.88526654045304 4.40067263334397 4.1524673298814 1.74268358683056 2.57937987501016 2.03852362101989

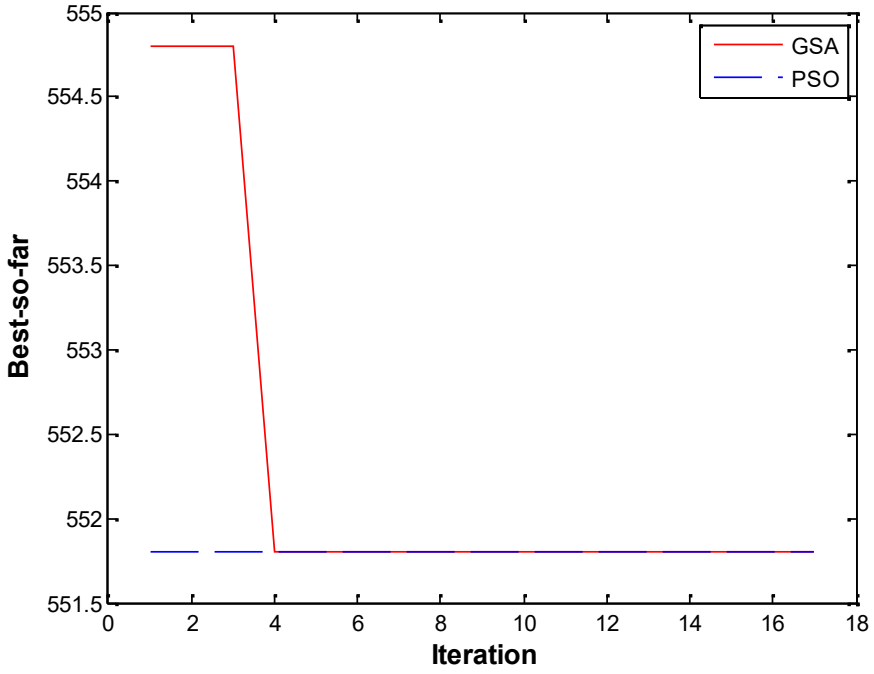
G best value

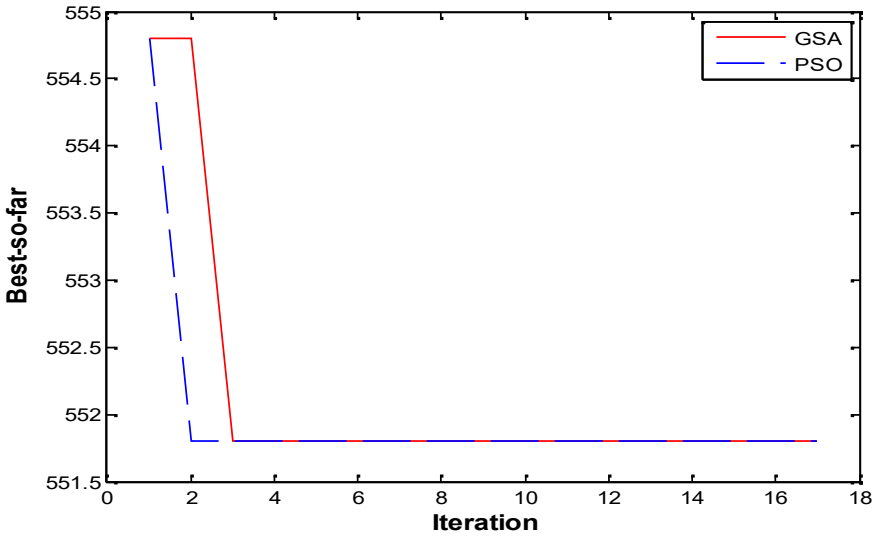
551.800000000000

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 8
August 2017

The results of show that this new proposed algorithm (without considering the number of fitness evaluations which is very low for GSA- PSO in contrast to comparing algorithms) is overall more precise in binary classification considering both classifier parameters optimization and feature subset selection. performs significantly better than for classification Some reasons that the GSA is able to defeat PSO and the GA include:

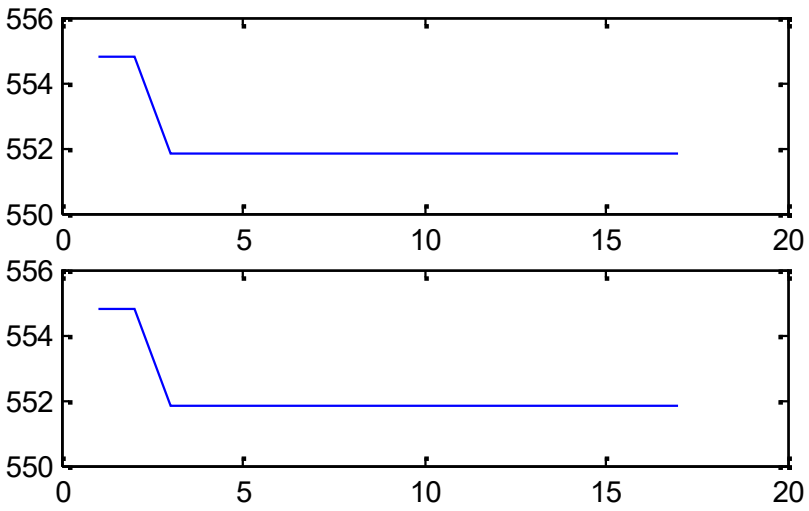
(i) In PSO, the direction of an agent is calculated using only two best positions, personal best position and global best position and in the GA, for the direction of an agent, only two agents called parents are considered but in the GSA, the agent direction is calculated based on the overall force obtained by all other agents. Therefore, the GSA is more capable of exploring the search space.



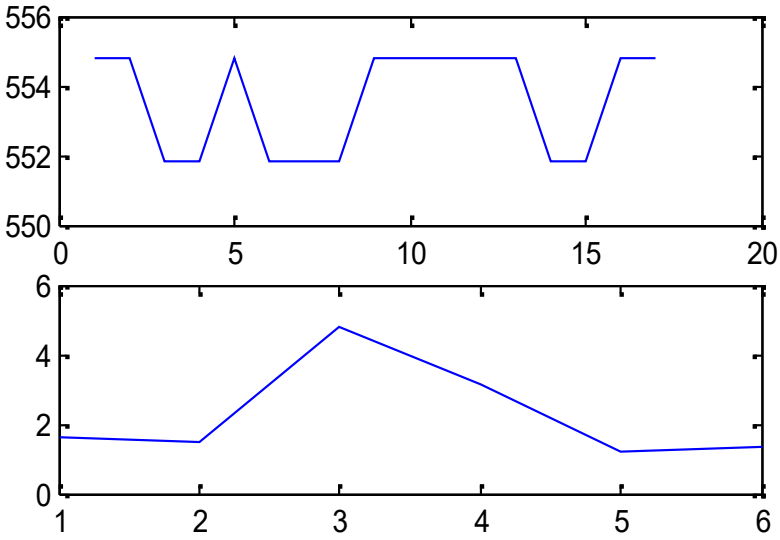


Selecting the best optimum sequence for the loading problem and fining out the route of tool and job allocation

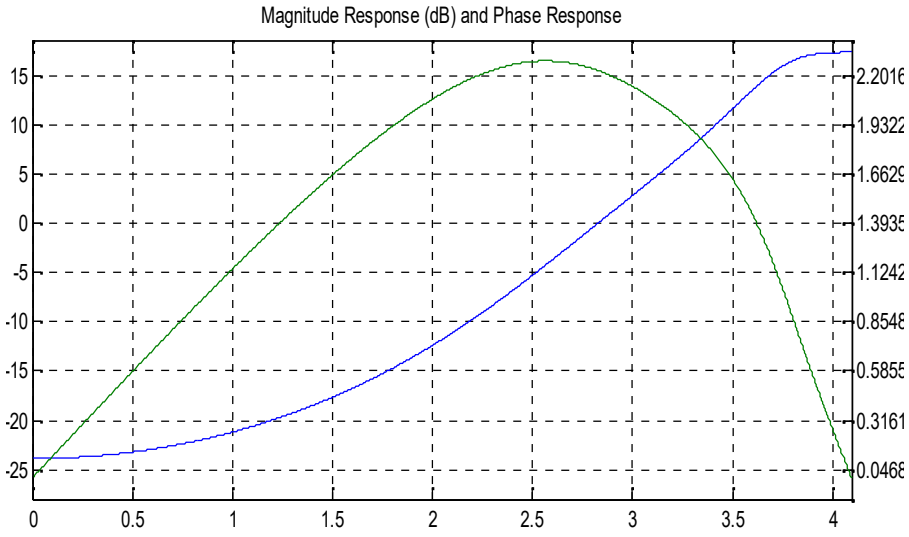
International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 8
August 2017



G best and lbest values of the ittretions

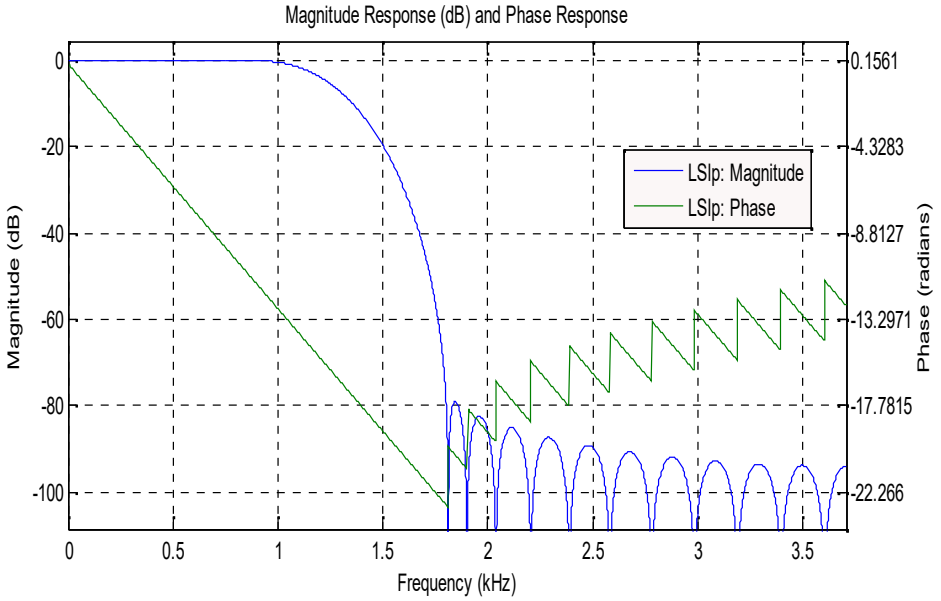


Lbest and mean utilisation curve

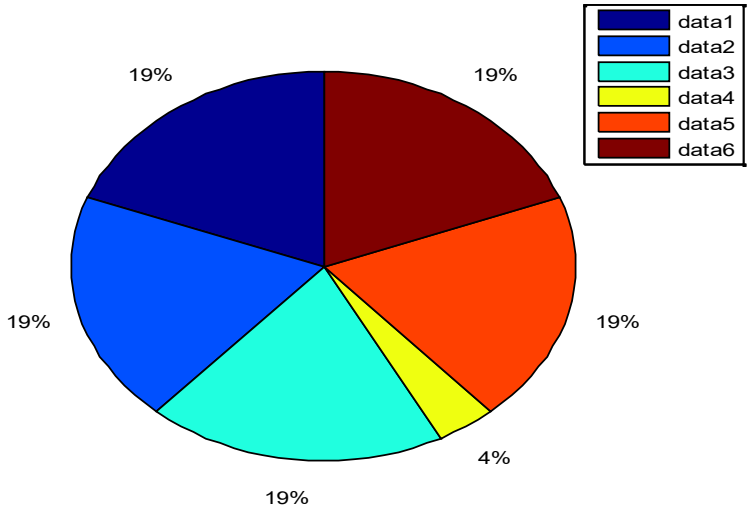


International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 8
August 2017

Frequency of load distribution among the various machine as per there utilisation and available time



Various machine distribution and job sequencing in combined pso and gsa methods .



Job distributions as per the scheduling

CONCLUSIONS

In this paper, we propose a new hybrid GSA-PSO system for binary classification which is a combination of a support vector machine and a combination of Seduling . The main contribution of our work was optimizing both the algorithms and parameter and feature subject selection effectively. We compared our approach with other algorithms including PSO and GSA,. Our experimental results indicate the proposed approach gives better computational performance in comparison with the results of the other methods, it also achieves higher accuracy for selection and making a machine loading

Reference:-

1. Tiwari, M.K. and Vidyarthi, N.K. (2000) Solving Machine Loading Problem in Flexible Manufacturing System Using Genetic Algorithm Based Heuristic Approach. International Journal of Production Research, **38**, 3357-3384.
2. Stecke, K.E. (1986) A Hierarchical Approach to Solving Grouping and Loading Problems of Flexible Manufacturing Systems. European Journal of Operational Research, **24**, 369-378.
3. Modi B. K., Shanker K., Models and solution approaches for part movement minimization and load balancing in FMS with machine, tool and process plan flexibilities. International Journal of Production Reserach, 33, 1994, 1791-1816.
4. Bretthauer K.M., Venkataramanan M.A., Machine loading and alternate routing in a flexible manufacturing system.Computers and Industrial Engineering 18 (3), 1990, pp.341–350
5. Srinivas, Tiwari M. K, Allada V, Solving the machine loading problem in a flexible manufacturing system using a combinatorialauction-based approach, International Journal of Production Research,vol. 42, no. 9, 2004, pp. 1879-189.
6. Prakash, A., Khilwani, N., Tiwari, M.K. and Cohen, Y. (2008) Modified Immune Algorithm for Job Selection and OperationAllocation Problem in Flexible Manufacturing Systems. Advances in Engineering Software, **39**, 219-232.
7. 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.
8. 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.
9. Yogeswaran M, Ponnambalam S.G, Tiwari M.K, An hybrid heuristic using genetic algorithm and simulated annealing algorithm to solve machine loading problem in FMS, IEEE International Conference on Automation Science and Engineering, 2007. CASE 2007. Pp.182-187.
10. Co H.C., Biermann J.S., Chen S.K., A methodical approach to the flexible manufacturing system batching, loading, and tool configuration problems. International Journal of Production Research, 28, 1990, pp. 2171-2186.
11. Zeballos L.J, Quiroga O.D, Henning G.P, A constraint programming model for the scheduling of flexible manufacturing systems with machine and tool limitations, Engineering Applications of Artificial Intelligence Volume 23, Issue 2, March 2010, Pp. 229-248.
12. Nagarjuna N, Mahesh O, Rajagopal K, A heuristic based on multi-stage programming approach for machine-loading problem in a flexible manufacturing system. Robot Comput-Int Manuf 22, 2006, pp.342–352
13. Srivastava B Wun-Hwa Chen, Heuristic solutions for loading in flexible manufacturing systems, IEEE Transactions on Robotics and Automation, vol.2,issue-6, 1996,pp. 858-868.
14. Akhilesh Kumar , Prakash , M.K. Tiwari , Ravi Shankar , Alok Baveja; "Solving machine-loading problem of a flexible manufacturing system with constraint-based genetic algorithm"; European Journal of Operational Research 175 (2006) 1043–1069