International Journal of Current Engineering and Technology
ISSN 2277 - 4106

Available at http://inpressco.com/category/ijcet

Research Article

A MATLAB GUI Tool for Optimization of FMS Scheduling using Conventional and Evolutionary Approaches

A.V.S.Sreedhar Kumar[A*], V.Veeranna[B], B.Durgaprasad[C] and B.Dattatraya Sarma[D]

[A]Professor, Mechanical engg.dept.,Narayana Engg.college.,Nellore, A.P,India
[B]Professor & Dean, Brundavan institute of Technology & science,Kurnool,A.P, India
[C]Professor, Dept. of. Mechanical Engg., JNTUA college of Engineering, Anantapur, A.P, India
[D]Principal, Sri venkateswara college of engineering,Nellore,A.P, India

Accepted 08 November 2013, Available online 01 December 2013, **Vol.3, No.5 (December 2013)**

Abstract

One of the methods to increase competitiveness in manufacturing is by converting existing manufacturing systems in to flexible manufacturing systems (FMS) and by implementing an optimum schedule for its operations.. Scheduling in an FMS environment is more complex and difficult than in a conventional manufacturing environment. To achieve high performance in an FMS, a good scheduling system should make a right decision at a right time according to system conditions. A MATLAB based GUI is designed to provide an automated tool for optimization of scheduling using conventional and evolutionary approaches. The primary objective of this tool is to automate and facilitate scheduling using the best possible approach for a particular job scenario involving multiple machines and jobs. The tool box is implemented using MATLAB version 7.1.The tool enables the user easy access in terms of loading the machine timings and job sequence details. In this work, we have also investigated the suitability of Bacterial Foraging Optimization Algorithm (BFOA) for scheduling in FMS systems. The performance of the system is proposed to be validated against other evolutionary strategies like Genetic Algorithm (GA) and Differential Evolution (DE).Most of the optimization functions proposed in the literature have penalties incorporated in them when the scheduled job is not completed in the specified time. We propose to incorporate a reward for each job if the job is completed in time or ahead of its due date. Such an approach is expected to increase the efficiency in regard to the total machining time.

Keywords: *Flexible manufacturing system, Scheduling, MATLAB GUI Tool, Genetic Algorithm, Differential Evolution, Bacterial Foraging Optimization Algorithm,*

1. Introduction

Flexible manufacturing system has evolved as a solution to efficient mid-volume production of a variety of part types with low setup time, low work-in-process, low inventory, short manufacturing lead time, high machine utilization and high quality (Veeranna.V,2002) . Flexible manufacturing system scheduling could be considered as a static scheduling problem where a fixed set of orders are to be scheduled either using optimization or priority scheduling heuristics. Alternatively, this could also be viewed as dynamic scheduling problem, where orders arrive periodically for scheduling as daily orders are released from a material requirement planning system or as individual customer's orders [Giffler B 1960),

Scheduling in an FMS environment is more complex and difficult than in a conventional manufacturing environment (French, S. 1982), Scheduling of FMS is NP-hard scheduling problems. Therefore, determining an optimal schedule and controlling an FMS is considered as a difficult task. Recently, Several heuristic procedures such as dispatching rules (Durgesh Sharma-2012 , Veeranna.V -2006) ,local search and meta-heuristics involving Genetic Algorithm (GA), Simulated Annealing (SA), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO) and Tabu Search (TS) have been developed for scheduling problems[Brucker,P.(1995]. These, do not guarantee to find an optimal schedule, but have the ability to find near to optimum solutions in a short time.

Metaheuristic optimization algorithms are inspired by biological phenomena or natural phenomena. Some of the newly introduced algorithms include Bacterial Foraging optimization algorithm(BFOA),Biogeography-based optimization (BBO), Firefly optimization algorithm, Cuckoo search optimization galaxy-based search algorithm, and Spiral dynamics inspired optimization (SDA). All these algorithms have gained attention due to their simplicity to program, fast computing time, easy to

*Corresponding author: **A.V.S.Sreedhar Kumar**

implement, and possibility to apply to various applications.

2. Problem description

The problem environment, assumption and aim of the present work are as follows (Jerald.J 2005)

1.The FMS considered in this work has a configuration as shown in Fig. 1. There are four flexible machining cells (FMCs), with two to six computer numerical machines (CNCs) as shown in the table-1, each is provided with independent tool magazines, part program controller, automatic tool changer (ATC) and buffer storage, part-carrying conveyors (input and output), a robot, and an automated storage and retrieval system (AS&RS). All the above are linked by means of host computer. The four FMCs are connected by automated guided vehicles (AGV). This AGV perform the intercell movements between the FMCs, the movement of finished product from any of the FMCs to the unloading station and the movement of semi-finished products between the AS/RS and the FMCs. There is a dedicated robot for loading and unloading AGV.

2. The assumptions made in this work are as follows: There are 40 to 50 varieties of products for a particular combination of tools in the tool magazines. Each type/variety has a particular processing sequence batch size, deadline and penalty cost for not meeting the

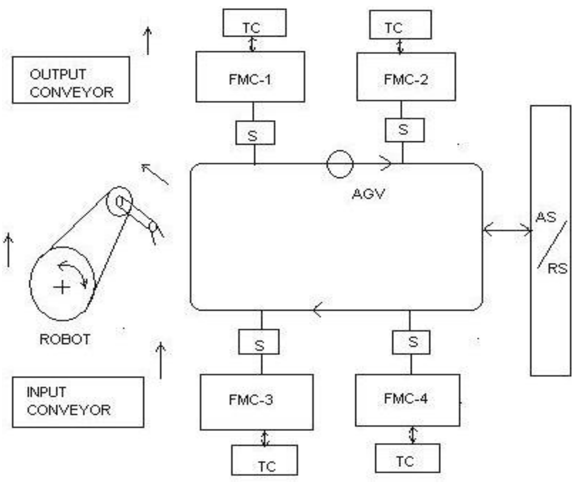

Fig-1 :Configuration of the FMS , FMC(Flexible manufacturing cell),TC(Tool changer),AGV(Automated guided vehical),AS/RS(Automated storage and retrieval ystem),S-shuttles

Table 1 FMCs with CNC machines

FMC number	Machine number
FMC-1	4,7,10
FMC-2	3,14,15
FMC-3	1,11,12,13
FMC-4	2,5,6,8,9,16

deadline and incorporate a reward point for each job if the job is completed ahead of time. Each processing step has a processing time with a specific machine. The objective of the schedule is the combination of minimizing the machine ideal time and minimizing the total penalty cost.

3. Optimization procedure

3.1 Objective function

The objective considered In this work is the combined objective function (COF) of minimizing the machine idle time and minimizing the total penalty cost is considered. However, for computational convenience, the machine setup timings are assumed to be same for all the machines. Feasible schedule is evaluated using the COF for minimizing the total penalty cost and maximizing machine utilization (Choudhury .B.B-2009) . Therefore the objective becomes,

Minimize COF=
$$W1*[(X_p/MPP)/(X_r/MPR)]+(W2)*(X_q/TE)$$
W1=Weight Factor for Customer Satisfaction
W2=Weight Factor for Machine Utilization
X_p=Total Penalty cost Incurred

$$X_p=\sum(CT_i-DD_i)*UPC_i*BS_i$$
X_r=Total Reward Points Incurred

$$X_r=\sum(DD_i- CT_i)*URC_i*BS_i$$

Where,
i=Job Number,
CT_i=Completion time for job i
DD_i=Due Date For job i
UPC_i=Unit Penalty Cost for job i
URC_i=Unit Reward Point for job i
MPP=Maximum Permissible Penalty
MPR=Maximum Permissible Reward
BS_i=Batch Size of job i

X_q=Total Machine Down Time,
$$X_q=\sum MDj$$
$$MD_j=TE-\sum PT_{ji}$$
TE=Total Elapsed Time
PT_{ji}=Processing time of i^{th} job with j^{th} machine
j= Machine Number

In the computation the weight factors W1and W2 are assumed to be equal and hence, W1 = 0.5 and W2 = 0.5. However, different ratios can be applied to them according to the demand of business situation

4. Proposed methodology

4.1 Genetic Algorithm

Genetic algorithm is an approach to optimization and learning based loosely on principles of biological evolution. Genetic algorithms maintain a population of possible solutions to a problem, encoded as chromosomes based on a particular representation scheme. After generating an initial population, new individuals for this population are generated via the process of reproduction. Parents are randomly selected from the current population for reproduction with the better ones (according to the evaluation criteria) more likely to be selected (Guohui Zhang 2011). The genetic operators of mutation and crossover generate children (i.e., new individuals) by random changes to a single parent or combining the information from two parents respectively (Jawahart.N 1998). Genetic algorithms have been applied to scheduling problems in a wide variety of domains. (With a more complete set of references given in (Guohui Zhang 2010).

4.1.2 The GA parameters used in optimization are as mentioned below

Population Size: 100
Scaling Function: Rank
Selection Function: Uniform
Elite Count: 2
Cross over fraction: 0.8
Mutation Function: Adaptive Feasible
Cross Over Function: Single Point.
Generations: 1000
Time limit: -Inf-

4.2 Differential Evolution

DE is a simple evolutionary algorithm that encodes solutions as vectors and uses operations such as vector addition, scalar multiplication and exchange of components (crossover) to construct new solutions from the existing ones. When a new solution, also called candidate, is constructed, it is compared to its parent. If the candidate is better than its parent, it replaces the parent in the population. Otherwise, the candidate is discarded. As a steady-state algorithm, DE implicitly incorporates elitism, i.e. no solution can be deleted from the population unless a better solution is found (Dervis Karabo G.A). While being a very successful optimization method, DE's greatest limitation originates in its encoding. As no vector representation of solution exists for combinatorial problems, DE can only be applied in numerical optimization (Satish Kumar.M.V)

4.2.1 Differential Evolution for Multi-objective Optimization

1. Evaluate the initial population P of random individuals.
2. While stopping criterion not met, do:
2.1. For each individual Pi(i=1,...,pop Size) from P repeat:(a) Create candidate C from parent Pi, (b) Calculate the objectives of the candidate, (c) If the candidate dominates the parent, the candidate replaces the parent, (d) If the parent dominates the candidate, the candidate is discarded. Otherwise, the candidate is added in the population.
2.2. If the population has more than pop size individuals, apply environmental selection to get the best pop Size individuals.
2.3. Randomly enumerate the individuals in P.

3. Return non dominated individuals from P

4.2.2 The parameter settings for DE is as follows

Population Size: 100;
Maximum Iterations: 1000
Mutation Factor: 0.5
Crossover Rate: 0.9

4.3 Bacterial Foraging Optimization Algorithm (BFOA)

Bacterial Foraging Optimization Algorithm (BFOA), proposed by Passino, is a new comer to the family of nature-inspired optimization algorithms (Tai-Chen Chen). Recently natural swarm inspired algorithms like Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO) have found their way into this domain and proved their effectiveness. Following the same trend of swarm-based algorithms, Passino proposed the BFOA. Application of group foraging strategy of a swarm of E.coli bacteria in multi-optimal function optimization is the key idea of the new algorithm. Bacteria search for nutrients is designed in a manner to maximize energy obtained per unit time. Individual bacterium also communicates with others by sending signals. A bacterium takes foraging decisions after considering two previous factors. The process, in which a bacterium moves by taking small steps while searching for nutrients, is called chemotaxis and key idea of BFOA is mimicking chemotactic movement of virtual bacteria in the problem

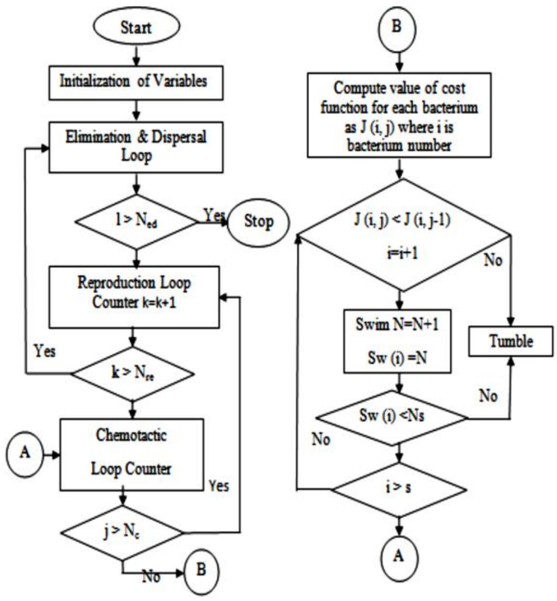

Fig.2: Flowchart of the Bacterial Foraging Algorithm

search space. Since its inception, BFOA has drawn the attention of researchers from diverse fields of knowledge

Table 2 Machining sequence, PT- process time (in min), D.D (due date in days),T (batch size in No's), P.C (penalty cost in Rs/units/day) and RP (Reward Point in Rs/units/day) (43jobs-16machines)

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15	M16	T	P.C	D.D	R.P
P1	0	0	0	0	0	1	1	1	0	2	0	0	0	0	0	0	150	1	17	3
P2	0	1	0	0	0	1	0	2	2	0	0	0	0	4	0	2	200	1	17	2
P3	0	0	0	0	0	0	0	1	0	0	3	0	4	0	0	0	800	1	14	5
P4	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	700	2	26	3
P5	0	0	0	5	3	0	0	0	0	0	0	0	0	0	4	0	150	1	11	5
P6	0	0	0	0	0	5	0	0	0	0	0	0	0	1	0	0	700	1	16	4
P7	0	0	5	0	0	3	0	0	0	0	0	0	0	0	0	5	250	2	26	3
P8	0	0	0	0	4	5	0	1	0	0	0	0	0	0	0	0	850	2	26	5
P9	0	0	0	1	5	0	0	1	0	0	1	0	0	0	0	0	100	0	1	6
P10	0	2	0	0	0	0	0	0	1	0	0	0	0	0	0	4	150	2	20	5
P11	0	0	0	0	0	0	0	4	0	0	0	2	0	0	0	0	250	1	1	3
P12	0	0	0	0	0	2	0	4	0	1	0	0	0	0	0	0	1000	3	19	2
P13	0	0	0	0	0	1	5	0	0	4	0	0	0	0	0	0	700	4	25	1
P14	0	0	0	2	3	2	0	0	0	0	0	0	0	0	2	0	1000	4	22	5
P15	0	0	0	0	4	0	0	3	0	0	0	0	0	0	0	0	700	5	15	4
P16	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	750	3	27	3
P17	0	0	1	0	0	4	0	0	0	0	0	0	0	1	0	0	650	4	20	4
P18	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	3	250	5	24	5
P19	0	0	0	1	5	2	0	2	0	0	0	0	0	0	5	0	450	1	5	5
P20	0	0	0	0	0	0	0	2	0	0	4	0	0	0	0	0	50	5	11	6
P21	0	0	0	5	5	0	0	4	0	0	0	0	0	0	4	0	850	3	16	1
P22	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	200	5	24	2
P23	0	0	0	2	1	5	0	4	0	0	0	0	0	0	0	0	50	4	14	5
P24	0	0	0	0	0	0	0	4	0	0	4	5	4	0	0	0	200	5	7	4
P25	0	0	0	0	0	0	3	0	0	2	0	0	0	0	0	0	350	1	24	3
P26	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	450	0	27	2
P27	0	0	0	0	0	0	0	5	0	0	5	4	0	0	0	0	400	1	22	4
P28	0	1	0	0	0	0	0	1	2	0	0	0	0	0	0	0	950	5	3	5
P29	0	0	0	1	5	0	0	0	0	0	0	0	0	0	0	0	700	1	7	2
P30	0	0	0	0	0	0	0	0	0	0	3	5	0	0	0	0	1000	1	18	1
P31	0	0	0	0	0	0	0	2	0	2	0	0	0	0	0	0	800	2	2	5
P32	0	3	0	0	0	4	0	0	3	0	0	0	0	0	0	0	800	1	15	4
P33	0	0	0	0	4	5	0	0	0	0	0	0	0	0	3	0	500	4	27	5
P34	0	0	2	0	0	2	0	0	0	0	0	0	0	0	0	0	300	4	12	3
P35	0	0	4	0	0	0	0	0	0	0	0	0	0	1	0	0	900	2	9	5
P36	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	700	2	20	5
P37	5	2	0	0	0	3	0	3	2	0	0	0	0	0	0	4	250	4	22	3
P38	0	4	0	0	0	0	0	3	2	0	0	0	0	0	5	0	50	1	8	4
P39	0	0	0	0	0	5	0	0	0	5	0	0	0	0	0	0	500	1	9	2
P40	0	2	0	0	0	4	0	0	4	0	0	0	0	0	0	0	250	5	7	5
P41	0	0	0	0	1	0	0	2	0	0	0	0	0	0	1	0	800	4	22	3
P42	0	5	0	0	0	4	0	0	3	0	0	0	0	0	0	1	400	2	19	6
P43	3	0	0	0	2	2	0	2	0	0	0	0	0	0	3	0	550	3	15	5

especially due to its biological motivation and graceful structure. It has already been applied to many real world problems and proved its effectiveness over many variants of GA and PSO. Figure- 3 depicts how clockwise and counter clockwise movement of a Bacterium take place in a nutrient solution**(**Swagatam Das).flow-chart (Figure 2) of the complete algorithm is presented below

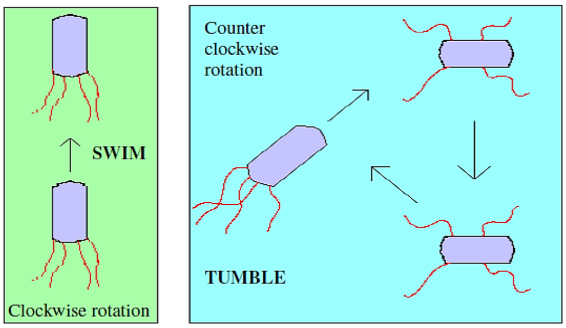

Fig.3 Swim and tumble of a bacterium

4.4 MATLAB GUI with GUIDE

An automated tool for optimization of scheduling using conventional and evolutionary approaches is designed and implemented. The primary objective of this tool is to automate and facilitate scheduling using the best possible approach for a particular job scenario involving multiple machines and jobs. The tool box is implemented using MATLAB version 7.1. The use of MATLAB enables us to solve complex scheduling problems involving different job types and multiple machines. The tool enables the user for an easy access in terms of loading the machining timing and sequence details.

The tool box has the following sections for easy and simple use of interface for the user.

1. Loading the sequence and setup details
2. Visualizing the timing details
3. Interface to run conventional scheduling Techniques
4. Interface to run Evolutionary Scheduling Techniques
5. Display of the optimized schedule, Penalty, Idleness and COF values.

The setup details can be loaded from an Excel work book(table--2) in which various details like timing of each operation, machines involved, sequence , batch size , due date, penalty , reward points are stored .This is a onetime operation and based on this any number of optimization can be done using conventional or evolutionary techniques. Once the data is loaded the number of jobs and the number of machines involved data is displayed in the GUI. Under the conventional Scheduling techniques we have included the Shortest Processing Time (SPT) scheduling rule. Under the evolutionary scheduling approaches we have included Genetic Algorithm (GA),

Table-3 Summary of results by different techniques, 43 jobs- 16 machines problem

S.No	Scheduling Techniques	Job Sequence	Penalty	Idleness	COF
1	SPT(Shortest processing time)	20 23 38 1 9 26 22 10 34 18 36 11 25 5 16 2 40 4 31 41 7 24 28 17 6 29 35 37 15 39 42 27 33 3 43 19 12 13 30 32 8 14 21	0.1733	0.2830	0.4563
2	GA(Genetic algorithm)	9 26 1 2 3 5 6 11 19 25 27 29 30 32 38 39 4 7 8 10 31 35 36 42 12 16 21 43 13 14 17 23 33 34 37 41 15 18 20 22 24 28 40	0.2492	0.3286	0.5779
3	DE(Differential Evolution)	3 4 17 43 1 2 41 15 12 7 39 21 5 25 31 34 28 10 6 23 32 16 27 8 9 24 19 22 13 14 30 35 37 40 18 38 11 29 36 33 20 42 26	0.1491	0.2458	0.3947
4	BFOA(Bacterial foraging optimization algorithm)	9 20 42 3 5 8 10 14 18 19 23 28 31 33 35 36 40 43 6 15 17 24 27 32 38 1 4 7 11 16 25 34 37 41 2 12 22 26 29 39 13 21 30	0.0084	0.1967	0.2051

Differential evolution (DE) and Bacterial Foraging Optimization (BFOA).The tool box is designed using GUIDE interface available in MATLAB. The results of the scheduling like the machine sequence, Penalty value for a particular sequence, Idleness and Combined Objective Function (COF) values are displayed in the command window of the MATLAB (figs- 4,5,6,7). The schedule sequence is also displayed as a stair case plot in GUI. The values of Penalty, Idleness and COF values are also displayed as Bar plot in an individual figure -8.

The bacterial foraging optimization technique is implemented for the combined objective function which includes reward for those works which are completed either in schedule or ahead of schedule. It is observed that the BFOA technique returns the best possible schedule. The GUI provides the user a simple interface tool capable of executing different scheduling techniques and chooses the best technique for the given scenario

5. Results and comparison

The optimization procedures developed in this work are based on the various non-traditional approaches that have been implemented using MATLAB 7.1. Different optimal schedules are obtained for the FMS using the above approaches, and the performances are compared and analyzed. Among the four approaches used in this work, the schedule obtained by the BFOA algorithm gives the optimum COF value, i.e., minimum total penalty cost and minimum machine idleness, as shown in the table-3

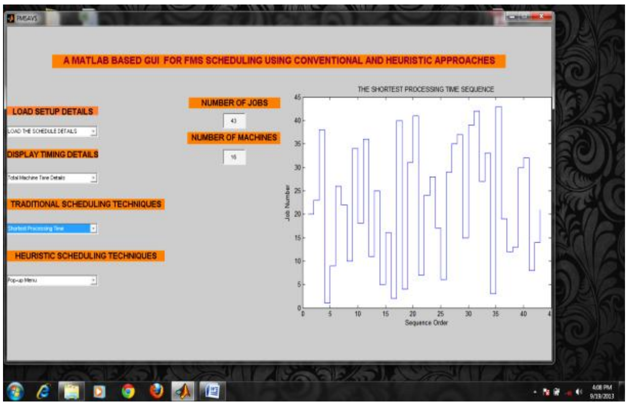

Fig-4 Job sequence using shortest processing time

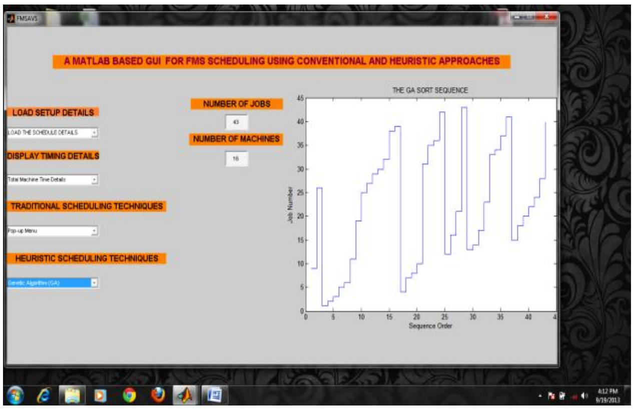

Fig-5 Job sequence using G.A

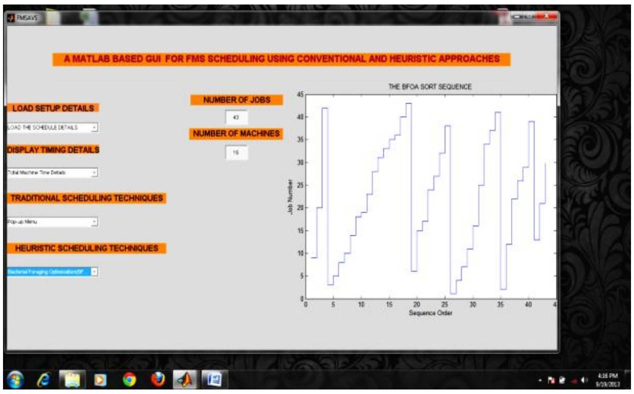

Fig-6 Job sequence using D.E

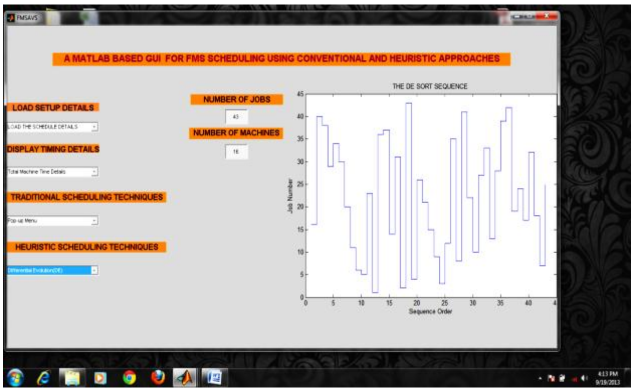

Fig-7 Job sequence using BFOA

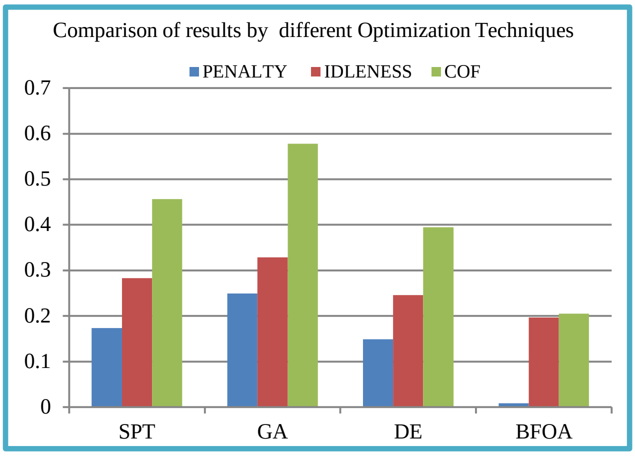

Fig-8 Comparison of results by different Optimization Techniques

Conclusions

In this work, Optimization procedures have been developed based on the three non-traditional approaches, i.e., GA, D.E. and BFOA algorithms .These are implemented successfully for solving the optimization problems of FMS scheduling. A MATLAB based GUI has also been designed to automate the optimization process by providing the user ease of interface. Results are obtained for 43 jobs- 16 machines. Results obtained by the different approaches are compared and the performances are analyzed for the combined objective function of minimizing total penalty cost and minimizing total machine idleness. BFOA algorithm is found to be superior and gives the minimum combined objective function. We have also evaluated the effectiveness of combined objective function in which the penalty value is moderated by the inclusion of reward. The inclusion of reward has improved the convergence of the evolutionary algorithms in finding the optimum schedule.

References

Brucker,P.(1995), Scheduling algorithms *Springer-Verlag*, Berlin-Heidelberg

Choudhury.B.B,Biswal.B.B,Mishra.D,Taskassignment and scheduling in a constrained manufacturing system using GA*International J Agile System & Management*

Choudhury .B.B, Biswal .B.B,Mahapatra.R.N,(2009) Appropriate Evolutionary Algorithm for Scheduling in FMS *IEEE*

Dervi_s Karabo_GA ,Sel_cuk ¨Okdem ,A Simple and Global Optimization Algorithm for Engineering Problems: Differential Evolution Algorithm *Turk J Elec Engin*, Vol.12, No.1

Durgesh Sharma,Suresh Garg,Chitra Sharma (2012) .Effect of Scheduling rules on performance of Semi Automated Flexible Manufacturing System *Global Journal of Enterprise Information System* ,Volume-4 Issue-I

French, S. (1982), Sequencing and Scheduling: An introduction to the mathematics of Job-Shop, *Ellis Horwood Limited*, Chichester.

Giffler B, Thomson GL (1960) Algorithms for solving production scheduling problems. *Int J Oper* ,pg:487–503

Guohui Zhang ,Liang Gao,Yang Shi.(2010) A Genetic Algorithm and Tabu Search for Multi Objective Flexible Job Shop Scheduling Problem, *International Conference on Computing,Control and Industrial Engineering*, 251-254

Guohui Zhang , Liang Gao , Yang Shi (2011)An effective genetic algorithm for the flexible job-shop scheduling problem*Expert Systems with Applications- 3563–3573*

Jerald.J, Asokan.P, Prabaharan.G, Saravanan.R,(2005)
Scheduling optimisation of flexible manufacturing systems using particle swarm optimisation algorithm *Int J Adv Manuf Technol* 25: 964–971

Jawahart.N, Aravindan.P,and Ponnambalam.S.G, (1998) A Genetic Algorithm for Scheduling Flexible Manufacturing Systems *Int J Adv Manuf Technol ,Springer*

Satish Kumar.M.V,Rao.C.S.P,Rangajanardhan.G, A Hybrid Differential Evolution Approach for Simultaneous Scheduling Problems in a Flexible Manufacturing Environment *International Journal of Applied Management and Technology*, Vol 6, Num 3

Veeranna.V,(2002).Some Studies on optimum design of flexible manufacturing system layout ,*P.hd. thesis*

Veeranna.V,Dattatreyasarma.B,Chakraverti.G,((2006)Optimization of FMS Layout by Heuristic Procedure with Scheduling as a Constraint*Industrial Engg.Journal* ,25-28

Tai-Chen Chen, Pei-Wei Tsai, Shu-Chuan Chu*, and Jeng-Shyang Pan A Novel Optimization Approach: Bacterial-GA Foraging

Swagatam Das1, Arijit Biswas1, Sambarta Dasgupta1, and Ajith AbrahamBacterial Foraging Optimization Algorithm: TheoreticalFoundations, Analysis, and Applications

By Shivakumar B L & Amudha T(2012)A Hybrid Bacterial Swarming Methodology for Job ShopScheduling Environment*Global Journal of Computer Science and TechnologyHardware & Computation vol-10,issue12*