RESEARCH ARTICLE OPEN ACCESS

An Automated Tool for Optimization of FMS Scheduling With Meta Heuristic Approach

A. V. S. Sreedhar Kumar*, V. Veeranna**, B. Durgaprasad***, B. Dattatraya Sarma****
*Professor, Mechanical engg. Dept. Narayana Engg. College, Nellore, A.P, India
**Professor & Dean, Brundavan Institute of Technology & Science, Kurnool, A.P, India
***Professor, Dept. of. Mechanical Engg., JNTUA college of Engineering, Anantapur, A.P, India
****Principal, Sri Venkateswara college of Engineering, Nellore, A.P, India

ABSTRACT
The evolutions of manufacturing systems have reflected the need and requirement of the market which varies from time to time. Flexible manufacturing systems have contributed a lot to the development of efficient manufacturing process and production of variety of customized limited volume products as per the market demand based on customer needs. Scheduling of FMS is a crucial operation in maximizing throughput, reducing the wastages and increasing the overall efficiency of the manufacturing process. The dynamic nature of the Flexible Manufacturing Systems makes them unique and hence a generalized solution for scheduling is difficult to be abstracted. Any Solution for optimizing the scheduling should take in to account a multitude of parameters before proposing any solution. The primary objective of the proposed research is to design a tool to automate the optimization of scheduling process by searching for solution in the search spaces using Meta heuristic approaches. The research also validates the use of reward as means for optimizing the scheduling by including it as one of the parameters in the Combined Objective Function. A complex machining operations configured in a FMS having 6 Machines with 3 different setups manufacturing 3 different parts, each having 9 possible routes is used as test case to validate the proposed approach. Three different Meta Heuristic approaches like Genetic Algorithm (GA), Differential Evolution (DE), and Bacterial Foraging Optimization Algorithm (BFOA) are used to optimize the scheduling process. A Graphical User Interface is designed using Matlab to automate the entire approach.

Keywords- Flexible Manufacturing System; Scheduling; MATLAB GUI Tool; Genetic Algorithm; Differential Evolution and Bacterial Foraging Optimization Algorithm

I. INTRODUCTION

During last two decades because of globalization most of the manufacturing industries are required to meet continuously changing customer demands so that Indian industries are transforming their present traditional manufacturing system to flexible manufacturing system. Existing FMS implementation in manufacturing companies have demonstrated a number of benefits by helping minimum unit cost, maximum utilization of the floor area, reduced work in process e.t.c,[1]. The FMS system can be adapted in both internal and external changes and the strategic advantages are improvement in throughput, product quality, information flows, reliability, and others [2]. However appropriate scheduling methodology can better derive these benefits. Some of the newly introduced algorithms like 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) can be beneficially utilized for optimization of FMS scheduling problems[3]. All these algorithms have gained attention due to their simplicity to program, fast computing time, easy to implement, and possibility to apply to various applications. The present work utilizes these powerful approaches like Genetic algorithm (GA) ,Differential evolution and Bacterial foraging algorithm and tries to find out their appropriateness for planning & scheduling of FMS producing variety of parts in batch mode[4,5].

Objectives of the work

The present work is aimed to work out the optimal scheduling process for modular FMS setups [6]. The Scheduling deals with optimizing the cost function in terms of machining time. The search space includes a number of feasible combinations and out of these the best fit solution is derived with help of Genetic Algorithm (GA), Differential evolution and Bacterial foraging algorithm [7]. In order to accomplish the objectives, the methodology is split into the following:

- Detailing the machining processes involved in the manufacture of the jobs
- Application of GA, DE and BFOA for scheduling
- Detailing the automation tool designed using Matlab
- Optimization of scheduling time with alternate assignments within FMS
- Comparison of results obtained from the different Meta heuristic approaches and recommending the appropriate one

II. MODEL FORMULATION

2.1 Description of the jobs

FMS has the capability to manufacturing a verity of jobs However, in this study, the jobs are to be manufactured in the selected setups. These jobs are so chosen that they are having similar in their functions with differences in their design attributes and manufacturing attributes. Under group technology viewpoint,The machining requirements are almost same for all the jobs. The jobs have been chosen keeping in view that they can be manufactured under the set of facilities under consideration without major changes in the setup requirements. The machining requirements for the jobs are Facing,Turning, Drilling, Boring and Thread cutting. The details of machining operations of job-1, job-2 and job-3 (as shown in Figures 1(a),(b)& (c) respectively) are given in Table -1 respectively[6].

(a)

(b)

(c)

Figure .1 (a) Graphical model of job 1 (b) Graphical model of job 2 and (c) Graphical model of job 3

Table.1 Processing operations for job-1,job-2,job-3

Sl .No	***Operations for JOB-1***	***Operations for JOB-2***	***Operations for JOB-3***
1	Facing of face 1 (F11)	Facing of face 1 (F21)	Facing of face 1 (F31)
2	Turning of surface 1 (T11)	Turning of surface 1 (T21)	Turning of surface 1 (T31)
3	Drilling of hole 1 (D11)	Drilling of hole 1 (D21)	Drilling of hole 1 D31)
4	Boring (B11)	Boring (B21)	Drilling of hole 2 (D32)
5	Facing of face 2 (F12)	Facing of face 2 (F22)	Facing of face 2 (F32)
6	Turning of surface 2 (T12)	Turning of surface 2 (T22)	Turning of surface 2 (T32)
7	Thread cutting (TH11)	Thread cutting (TH21)	Thread cutting (TH31)
8	Drilling of hole 2 (D12)	Drilling of hole 2 (D22)	Drilling of hole 3 (D33)
9	---	---	Thread cutting (TH32)

2.2 Description of the setups

The three setups under consideration consist of four Machines (M) to accomplish the desired machining operations on all the three jobs are as follows.

Setup-1 consists of Machine 1(M1)- Lathe Machine 1, Machine 2 (M2)- Machine centre 1, Machine

3(M3) -Lathe Machine 2 : Machine 4(M4)- Machine centre 2
Setup-2 consists of Machine 1(M1)- Lathe Machine 1, Machine 2 (M2)- Machine centre , Machine 3(M3)-Lathe Machine 2 : Machine 4(M4)- CNC Drilling Machine
Setup-3 consists of Machine 1(M1)-Lathe Machine 1, Machine 2(M2)-CNC Drilling Machine 1, Machine 3(M3)-Lathe Machine 2, Machine 4 (M4)-CNC Drilling Machine 2
The three different alternate routes via which the jobs are manufactured are as follows .

In setup-1 are: R1 = M1 → M2 → M3 →M4 ; R2 = M3 → M4→ M2 → M3 : R3 = M1 → M4 → M3 →M4.
The operations performed at all the machines via different routes, for each job in setup-1 are given in Table 2 for Job-1,Job-2 and Job-3 correspondingly.

The routes for setup-2 are: R1= M1 → M2 → M3 → M4, R2 =M3 → M4 → M2 → M3, and R3= M1 → M2 → M3 →M2
The operations performed at all the machines via different routes, for each job in setup-2 are given in Table-3 For Job-1 ,Job-2 and Job-3 correspondingly .
The routes for setup-3 are: R1= M1 → M2 → M3 → M4, R2= M3 → M4 → M2 → M3, and R3= M1 → M2 → M3 → M2.
The operations performed at all the machines via different routes, for each part in setup-3 are given in Table 4 for Job-1 ,Job-2 and Job-3 correspondingly.

Table 2 Machines on routes of Setup-1

Route1:M1(F11,T11)→M2(D11,B11)→M3 (F12,T12) →M4 (D12,TH11)
Route2:M3(F11,T11)→M4(D11,B11)→M2 (F12,T12,D12) →M3 (TH11)
Route3:M1(F11,T11)→M4(D11,B11)→M3 (F12,T12) →M4 (D12,TH11)

Route1:M1(F21,T21)→M2(D21,B21) → M3 (F22,T22) →M4(D22,TH21)
Route2:M3(F21,T21)→M4(D21,B21)→M2 (F22,T22,D22) →M3 (TH21)
Route3:M1(F21,T21)→M4(D21,B21)→M3 (F22,T22) →M4 (D22,TH21)

Route1:M1(F31,T31)→M2(D31,D32,TH31) →M3(F32,T32) →M4(D33,TH32)
Route2:M3(F31,T31)→M4(D31,D32,TH31) →M2(F32,T32,D33) →M3 (TH32)
Route3:M1(F31,T31)→M4(D31,D32,TH31) →M3(F32,T32) →M4 (D33, TH32)

Table 3 Machines on routes of Setup-2

Route1:M1(F11,T11)→M2(D11,B11)→M3 (F_{12},T12, TH11) →M4 (D12)
Route2:M3(F11,T11)→M4(D11,B11)→M2 (F12,T12,D12) →M3 (TH11)
Route3:M1(F11,T11)→M4(D11,B11)→M3 (F12,T12,TH11,) →M4 (D12)

Route1: M1(F21,T21) →M2(D21,B21) →M3 (F22,T22 , TH21) →M4(D22)
Route2:M3(F21,T21)→M4(D21,B21)→M2 (F22,T22,D22) →M3 (TH21)
Route3:M1(F21,T21)→M4(D21,B21)→M3 (F22,T22, TH21) →M4 (D22)

Route1:M1(F31,T31)→M2(D32,D33)→M3 (F32,T32, TH31,TH32) →M4(D31)
Route2:M3(F31,T31,F32,T32)→M4(D31,D32) →M2(D33) →M3 (TH31,TH32)
Route3:M1(F31,T31)→M4(D32,D33,)→M3 (F32,T32, TH31, TH32) →M4 (D31)

Table 4 Machines on routes of Setup-3

Route1:M1(F11,T11)→M2(D11,B11)→M3 (F12,T12, TH11) →M4 (D12)
Route2: M3(F11,T11) →M4(D11,B11) →M2 (D12) →M3 (F12,T12,TH11)
Route3:M1(F11,T11)→M4(D11,B11)→M3 (F12,T12, TH11) →M4 (D12)

Route1:M1(F21,T21)→M2(D21,B21)→M3 (F22,T22, TH21) →M4(D22)
Route2: M3(F21,T21) →M4(D21,B21) →M2(D22) →M3 (TH21, F22,T22)
Route3:M1(F21,T21)→M4(D21,B21)→M3 (F22,T22, TH21) →M4 (D22)

Route1:M1(F31,T31)→M2(D31,D32,)→M3 (F32,T32, TH31, TH32,) →M4(D33)
Route2:M3(F31,T31,F32,T32)→M4(D31,D32) →M2(D33) →M3 (TH31,TH32)
Route3:M1(F31,T31)→M4(D31,D32,)→M3 (F32,T32, TH31, TH32) →M4 (D33)

III. PROPOSED APPROACHES

3.1 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 [10]. 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 [11]. Genetic algorithms have been applied to scheduling problems in a wide variety of domains [12].

3.1.1 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-

3.2 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 and 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 [13]. 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 [14]

3.2.1 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

3.2.2 The parameter settings for DE is as follows
Population Size: 100;
Maximum Iterations: 1000
Mutation Factor: 0.5
Crossover Rate: 0.9

3.3 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 [15]. 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 chemo taxis and key idea of BFOA is mimicking chemo tactic movement of virtual bacteria in the problem search space. Since its inception, BFOA has drawn the attention of researchers from diverse fields of knowledge 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 . Flow-chart (Figure 2) of the complete algorithm is presented below and Figure- 3 depicts how clockwise and counter clockwise movement of a Bacterium take place in a nutrient solution **[15]**.

Fig.2: Flowchart of the Bacterial Foraging Algorithm

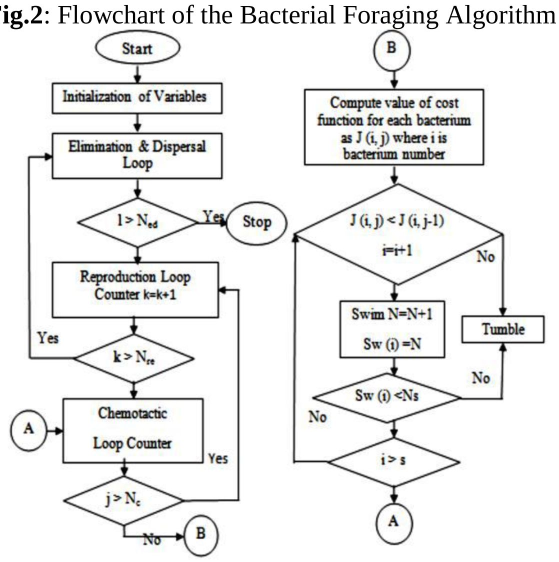
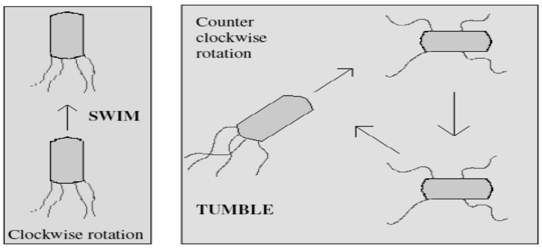


Fig.3 Swim and tumble of a bacterium



3.4 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 setups, maching timings and routing details
2. Visualizing and analyzing the total machining timing details
3. Interface to run conventional scheduling Technique
4. Interface to run Meta heuristic Scheduling Techniques
5. Display of the Optimized Schedule, Total machining timings, Penalty value, Idleness value and COF value.

The setup details can be loaded from an Excel work book(Table-1,2,3,4,5 &6) in which various details like processing operations , machines involved and routes of each setup ,timing of each operation, 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 Meta heuristic techniques. Once the data is loaded different types of analysis involved data is displayed in the GUI. Under the conventional Scheduling techniques we have included the Shortest Processing Time (SPT) scheduling rule. Under the Meta heuristic scheduling approaches we have included Genetic Algorithm (GA).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 Total machining time , Penalty value for a particular sequence, Idleness and Combined Objective Function (COF) values are displayed in the command window of the MATLAB (figures- 4,5,6,7,8,9). The total machining time of each part is also displayed as a plot in GUI.

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

IV. PROBLEM FORMULATION

The problem environment, assumption and aim of the present work are as follows [9]

The FMS scheduling problem can be formulated in this work are as follows. There is a set of N jobs $J = \{J_1, J_2, . . ., Ji, . . ., J_N\}$ and a set of M machines $M = \{M_1, M_2, . . ., M_k, . . ., M_M\}$. Each job Ji consists of a predetermined sequence of operations. Each operation requires one machine selected out of a set of available machines, namely the first sub-problem: the routing sub-problem. In addition, the FMS scheduling problem sets its starting and ending time on each machine, namely the second sub-problem: the scheduling sub-problem. it determine an assignment and a sequence of the operations on the machines so that some criteria are satisfied. However, this problem is more complex and challenging than the classical JSP because it requires a proper selection of a machine from a set of available machines to process each operation of each job[8] .

Table-5 Batch size (in No’s),Duedate (in days), Penalty cost (in Rs/units/day) and Reward point (in Rs/units/day)

Jobs	Batch size	Due date	Penalty cost	Reward point
Job - 1	35	12	5	6
Job - 2	40	10	3	2
Job - 3	50	10	4	5

Illustration of research problem

To illustrate the research problem,the following data sets are taken from literature [9]

Table 6 Machining time (in min) for different operations

Jobs	Operations	L-1	L-2	C-1	C-2	D-1	D-2
job 1	F_{11}	020	030	X	X	X	X
	F_{12}	X	020	020	X	X	X
	T_{11}	060	070	X	X	X	X
	T_{12}	X	040	035	X	X	X
	D_{11}	X	X	100	120	090	100
	D_{12}	X	X	070	080	070	090
	B_{11}	X	X	100	120	090	100
	TH_{11}	X	080	055	060	X	X
job 2	F_{21}	030	040	X	X	X	X
	F_{22}	X	060	050	X	X	X
	T_{21}	050	060	X	X	X	X
	T_{22}	X	080	070	X	X	X
	B_{21}	X	X	120	140	100	110
	D_{21}	X	X	080	100	070	090
	D_{22}	X	X	075	080	070	090
	TH_{21}	X	150	110	120	X	X
job 3	F_{31}	100	110	X	X	X	X
	F_{32}	X	050	040		X	X
	T_{31}	080	100	X	X	X	X
	T_{32}	X	180	160	X	X	X
	D_{31}	X	X	120	140	100	120
	D_{32}	X	X	120	140	100	120
	D_{33}	X	X	020	020	020	025
	TH_{31}	X	200	180	200	X	X
	TH_{32}	X	040	025	030		X

L-1: Lathe-1; L-2: Lathe-2; C-1: Machining Center-1; C-2: Machining Center-2; D-1 : Drilling Machine-1; D-2: Drilling Machine-2.

4.1 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 [9]. 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 Meta heuristic approches in finding the optimum schedule.Therefore the objective becomes,

Minimize COF=

W1*[(Xp/MPP)/(Xr/MPR)]+(W2)*(Xq/TE)

W1=Weight Factor for Customer Satisfaction
W2=Weight Factor for Machine Utilization

Xp=Total Penalty cost Incurred
Xp=Σ(CTi-DDi)*UPCi*BSi

Xr=Total Reward Points Incurred
Xr=Σ(DDi- CTi)*URCi*BSi

Where,
i=Job Number,
CTi=Completion time for job i
DDi=Due Date For job i
UPCi=Unit Penalty Cost for job i
URCi=Unit Reward Point for job i
MPP=Maximum Permissible Penalty
MPR=Maximum Permissible Reward
BSi=Batch Size of job i

Xq=Total Machine Down Time,
Xq=ΣMDj

MDj=TE-ΣPTji
TE=Total Elapsed Time
PTji=Processing time of ith job with jth 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

V. RESULTS AND COMPARISON

The optimization procedures developed in this work are based on the various Meta Heuristic 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 as shown in figures 4,5,6,7,8,9 and tables 7,8,9. Among the three 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-10 and minimum total machining time of three jobs as shown in table-11.a comparision of total machining time obtained from three Meta heuristic approches is presented in figure-10. BFOA algorithm achieves a good efficiency with respect to the total machining time.

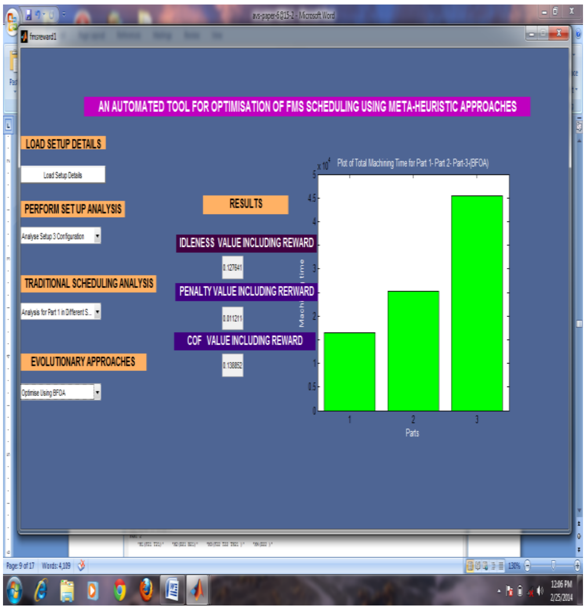


Fig- 4 Total machining time, Idleness value, penalty value and COF value for job-1,job-2 and job-3 using BFOA

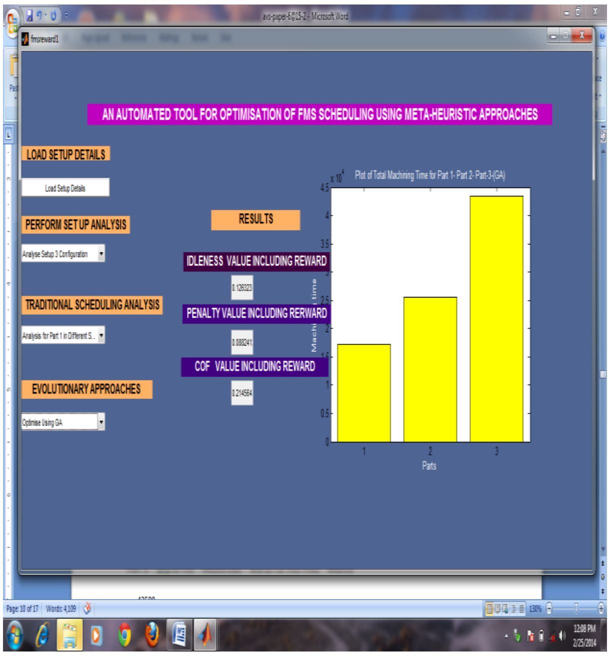


Fig-6 Total machining time, Idleness value, Penalty value and COF value for job-1,job-2 and job-3 using GA

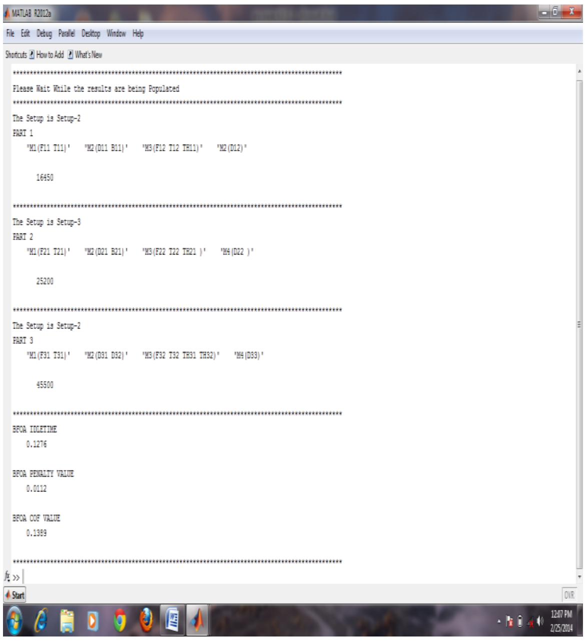


Fig- 5 Operation assignment and Total machining time for job-1,job-2 and job-3 using BFOA

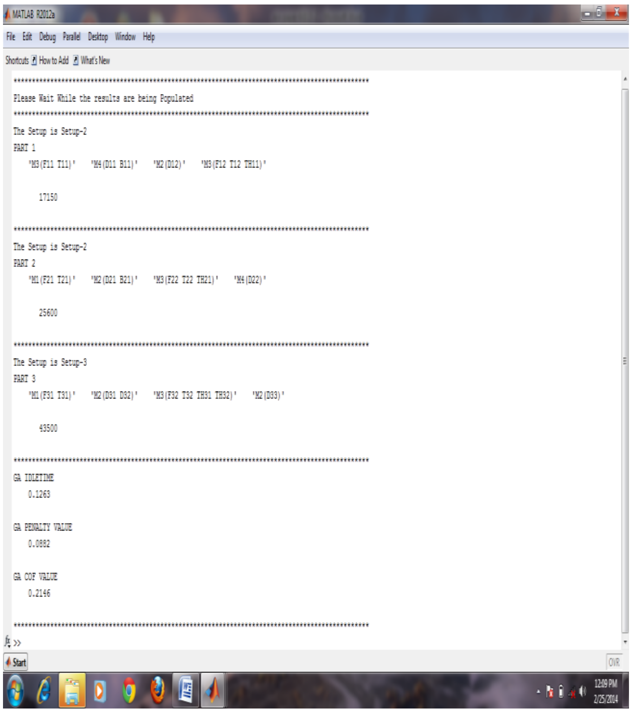


Fig-7 Operation assignment and Total machining time for job-1,job-2 and job-3 using GA

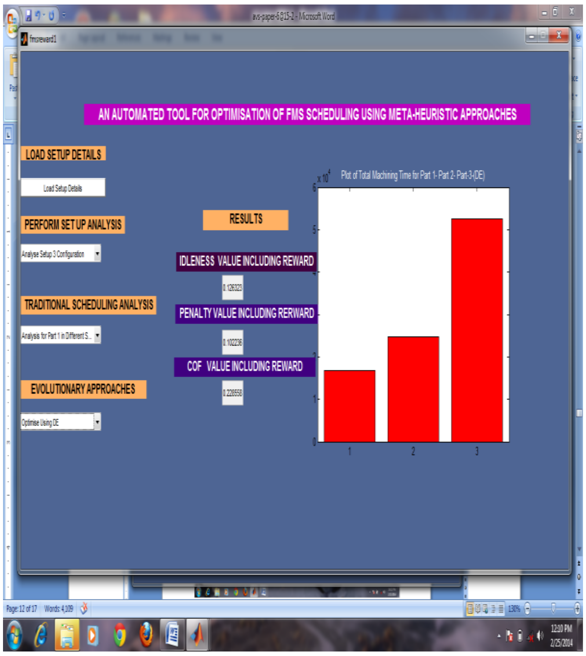


Fig- 8 Total machining time, Idleness value, Penalty value and COF value for job-1,job-2 and job-3 using DE

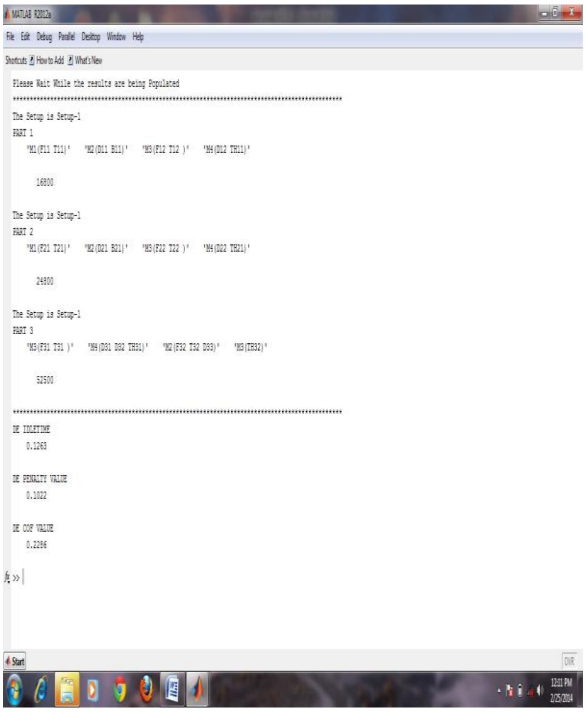


Fig- 9 Operation assignment and Total machining time for job-1, job-2 and job-3 using DE

Table-7 Suggested routes and setups for Job 1 , Job2 and Job3 by BFOA

Job	Desired setup	M1	M2	M3	M4	Total machining time (in min.)
Job 1	Setup -2	F_{11}, T_{11}	D_{11}, B_{11}, D_{12}	F_{12}, T_{12}, TH_{11}	-	16450
Job 2	Setup - 3	F_{21}, T_{21}	D_{21}, B_{21}	F_{22}, T_{22}, TH_{21}	D_{22}	25200
Job 3	Setup - 2	F_{31} T_{31}	D_{31}, D_{32}	F_{32}, T_{32}, TH_{31} TH_{32}	D_{33}	45500

Table-8 Suggested routes and setups for Job 1 , Job2 and Job3 by GA

Job	Desired setup	M1	M2	M3	M4	Total machining time (in min)
Job1	Setup 2	-	D_{12}	F_{11} ,T_{11}, F_{12}, T_{12}, TH_{11}	D_{11}, B_{11}	17150
Job2	Setup 2	F_{21} , T_{21}	D_{21}, B_{21}	F_{22}, T_{22}, TH_{21}	D_{22}	25600
Job3	Setup 3	F_{31} T_{31}	D_{31} ,D_{32}	F_{32}, T_{32}, TH_{31} TH_{32}	D_{33}	43500

Table-9 Suggested routes and setups for Job 1 , Job2 and Job3 by DE

Job	Desired setup	M1	M2	M3	M4	Total machining time (in min)
Job 1	Setup 1	F_{11}, T_{11}	D_{11}, B_{11}	, F_{12}, T_{12},	D_{12} TH_{11}	16800
Job 2	Setup 1	F_{21} T_{21}	D_{21}, B_{21}	F_{22}, T_{22}	TH_{21}, D_{22}	24800
Job 3	Setup 1	--	F_{32}, T_{32}, D_{33}	F_{31} , T_{31}, TH_{33}	TH_{33}, D_{31}, D_{32}	52500

VI. CONCLUSIONS

The research validates use of reward points in optimization of scheduling process, the inclusion of reward along with the penalty value as one of the parameters in the Combined Objective Function has yielded expected results. Among the three Meta heuristic approaches that are considered in this paper from the results it can be observed that the Bacterial Foraging Optimization Algorithm results in better convergence most of the time. The below mentioned table and graph illustrates the efficiency of BFOA in terms of reduced COF value , penalty and machine idle time in providing an optimized solution.

Table-10 Summary of results (Idleness, Penalty, COF) by different Meta heuristic approaches

Scheduling technique	Idleness value	Penalty value	COF value
BFOA	0.1276	0.0112	0.1389
GA	0.1263	0.0882	0.2146
DE	0.1263	0.1022	0.2286

Fig -10 Comparison of results by different Meta -Heuristics

Table-11 Summary of results (Total machining time in min.) by different Meta heuristics

Job	BFOA	GA	DE
Job-1	16450	17150	16800
Job-2	25200	25600	24800
Job-3	45500	43500	52500

The Graphical User Interface aids in easy input of data coded in preformatted Excel sheet. The GUI also has functions that enable to understand the test setup in term of machining time, machine configuration and machining sequence. The GUI provides seamless integration and aid the decision making process by proposing an optimized scheduling approach for the given problem.

REFERENCES

[1] Veeranna.V, "Some Studies on optimum design of flexible manufacturing system layout", P.hd. thesis, JNTU, Hyderabad, 2004
[2] Koren, Y.U.A., "Reconfigurable manufacturing systems:Key to future manufacturing"., Journal of Intelligent Manufacturing, 2000. 11(4)
[3] Swagatam Das1, Arijit Biswas1, Sambarta Dasgupta1, and Ajith Abraham"Bacterial Foraging Optimization Algorithm: Theoretical Foundations, Analysis, and Applications"
[4] French, S, "Sequencing and Scheduling: An introduction to the mathematics of Job-Shop", Ellis Horwood Limited, Chichester. (1982)
[5] Durgesh Sharma, Suresh Garg, Chitra Sharma ".Effect of Scheduling rules on performance of Semi Automated Flexible Manufacturing System" Global Journal of Enterprise Information System, (2012). Volume-4 Issue-I
[6] Choudhury .B.B, Biswal .B.B,"Task assignment and scheduking in a constrained manufacturing system using GA" International Journal Agile System & Management, (2007),
[7] Brucker, P, Scheduling algorithms, Springer-Verlag, Berlin-Heidelberg., (1995)
[8] Jerald. J, Asokan. P, Prabaharan. G, Saravanan. R" Scheduling optimisation of

flexible manufacturing systems using particle swarm optimisation algorithm" Int J Adv Manuf Technol, (2005), 25: 964–971

[9] Choudhury .B.B, Biswal .B.B, Mahapatra.R.N, "Appropriate Evolutionary Algorithm for Scheduling in FMS" IEEE, (2009).

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

[11] Jawahart.N, Aravindan.P, and Ponnambalam.S.G, "A Genetic Algorithm for Scheduling Flexible Manufacturing Systems", Int J Adv Manuf Technol, Springer, (1998) 14:588-607.

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

[13] Dervi_s Karabo_GA, Sel_cuk ¨Okdem, "A Simple and Global Optimization Algorithm for Engineering Problems: Differential Evolution Algorithm" Turk J Elec Engin, (2004), VOL.12, NO.1

[14] 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, (2008), Vol 6, Num 3, 108-130.

[15] Tai-Chen Chen, Pei-Wei Tsai, Shu-Chuan Chu*, and Jeng-Shyang Pan"A Novel Optimization Approach: Bacterial-GA Foraging " ICICIC '07, IEEE Computer Society Washington, DC, USA ©2007 ISBN:0-7695-2882-1