Optimization Procedure by Using Genetic Algorithm

Yedem Muni Ratnam [1], K Mohan Krishna [2], P.Giribabu [3]

[1] PG Scholar, Mechanical Engineering, Chadalawada Ramanamma Engineering college of Engineering,Tirupati, Andhra Pradesh, India
[2] Asst.Professor, Mechanical Engineering, Chadalawada Ramanamma Engineering college of Engineering,Tirupati, Andhra Pradesh, India
[3] Asst.Professor, Mechanical Engineering, Chadalawada Ramanamma Engineering college of Engineering,Tirupati, Andhra Pradesh, India

---***---

Abstract-*The Flexible Manufacturing Systems (FMS) belong to class of productive systems in which the main characteristic is the simultaneous execution of several processes and sharing a finite set of resource. Nowadays, the FMS must attend the demand of the market needs for personalized products. Consequently the product life cycle tends to be shorter and a greater variety of products must be produced in a simultaneous manner. In this paper, we present a Genetic Algorithm based scheduling of Flexible manufacturing system. This work is considering multiple objectives, i.e., minimizing the idle time of the machine and minimizing the total penalty cost for not meeting the deadline concurrently. Software is developed for getting optimum sequence of operation. FMS considered in this work has 16 CNC Machine tools for processing 43 varieties of products. In this paper, various meta-heuristic methods are used for solving same scheduling problems taken from the literature. The results available for the various existing meta-heuristic methods are compared with results obtained by GA. After 1700 generations of GA the global optimum schedule is obtained.*

Keywords *- Flexible manufacturing system, Genetic algorithm, Scheduling Optimization.*

I. INTRODUCTION

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 operation 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 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.

Flexible manufacturing systems (FMS) have been developed to combine the flexibility of job shops and the productivity of flow lines. Such systems consist of three

Today, FMS‟s seem to be a very promising technology as they provide flexibility, which is essential for many manufacturing companies to stay competitive in a highly dynamic and changing manufacturing environment. Existing FMS implementations have already demonstrated a number of benefits in terms of cost reductions, increased utilizations, reduced work-in-process levels, etc. However, there are a number of problems faced during the life cycle of an FMS. These problems are classified into design, planning, scheduling and control problems. In particular, the scheduling task, the control problem during the operation, is important owing to the dynamic nature of the FMS such as flexible parts, tools and automated guided vehicle (AGV) routings.

Scheduling of operations is one of the most critical issues in the planning and managing of manufacturing processes. To find the best schedule can be very easy or very difficult, depending on the shop environment, the process constraints and the performance indicator. One of the most difficult problems in this area the Job-shop Scheduling Problem (JSP) is the most complicated problem, where a set of jobs must be processed on a set of machines.

In scheduling, each job is formed by a sequence of consecutive operations, each operation requires exactly one machine, and machines are continuously available and can process one operation at a time without interruption. Thus, it can be said that it is a very difficult decision making problem which concerns to the given performance indicator to be optimized. JSP is a well-known NP-hard problem. The Scheduling problem in flexible

manufacturing system is more difficult due to the allocation of operations on any among a set of available machines. The intricacy of this system suggests the adoption of heuristic methods producing reasonably good schedules in a reasonable time, instead of looking for an exact solution. In recent years, the adoption of meta-heuristics like GA has led to better results than classical dispatching or greedy heuristic algorithms.

The increased use of flexible manufacturing systems (FMS) to efficiently provides customers with diversified products have created a significant set of operational challenges. The design of these kinds of systems is characterized by massive alternatives of components positions and paths. While in practice there is always the attempt to minimize the cycle time, dealing with a lot of alternatives - in respect to components positioning and paths planning - is necessary.

I.I Earlier research

During the last three decades much research has been done in this area. Many heuristic algorithms have been developed to generate optimum schedule and part-releasing policies. Most of these algorithms include enumerative procedures, mathematical programming and approximation techniques, i.e., linear programming, integer programming, goal programming, dynamic programming, transportation and network analysis, branch and bound, Lagrangian relaxation, priority-rule-based heuristics, local search algorithms (ITS, TA, TS, SA), evolution-ary algorithm (GA), etc. Of these techniques, few are specific to particular objectives, and few are specific to particular problem instances with respect to computational time needed.

Giffler and Thomson [5] developed an enumerative procedure to generate all active schedules for the general „n‟ job „m‟machine problem. Z.X guo and W.K wong [15] presented a comprehensive review of genetic algorithm based optimization model for scheduling flexible assembly lines. In this paper a scheduling problem in the flexible assembly line is investigated and developed a bi-level genetic algorithm is developed to solve the scheduling problem. Tiwari and Vidyarthi [11] proposed a genetic algorithm based heuristic to solve the machine loading problem of a random type FMS. The proposed GA based heuristic determines the part type sequence and the operation machine allocation that guarantee the optimal solution to the problem. In another scheduling paper [1], consider only 6 machines and 6 jobs. Chrisman [2] proposed an analytical model formulated as a traveling salesman problem (TSP) for minimizing total setup time in flow shop production cells. R Kumar, M K Tiwari and R Shankar [9], consider ant colony optimization approach in FMS scheduling. Bu ACO algorithm performs better in problem such as traveling sales problem, the vehicle rooting problem etc. In previous years most research concerning the AGV scheduling has been focused on developing scheduling algorithms for a single objective such as minimizing of setup cost minimizing the loading and unloading time. Toker A, Kondakci S and Erkip N [12] proposed an approximation algorithm for the „n‟ job „m‟ machine resource constraint job shop problem.

Hoitomt et al. [6] explored the use of the Lagrangian relaxation technique to schedule job shops characterised by multiple non-identical machine types, generic procedure constraints and simple routing considerations. Steeke and Soldberg [13] investigated various operating strategies (16 priority rules under 5different loading policies) on a caterpillar FMS by means of deterministic simulation with the number of completed assemblies as a performance criterion (minimization of flow time and minimization of maximum tardiness) scheduling problem associated with parallel identical machines through simulation. Chan and Pak [3] proposed two heuristic algorithms for solving the scheduling problem with the goal of minimizing the total cost of tardiness in a statically loaded FMS. He and Kusiak [4] addressed three different industrial scheduling problems, with heuristic algorithms for each problem. Lee and Dicesare [8] used Petri nets to model the scheduling problems inFMS. Sridhar and Rajendran [10] addressed a GA for part family grouping and scheduling parts within part families in a flow-line-based manufacturing cell. Shnits and Sinreich [10] present the development of a multi-criteria control methodology for FMSs. The control methodology is based on a two-tier decision making mechanism. The first tier is designed to select a dominant decision criterion and a relevant scheduling rule set using a rule-based algorithm. In the second tier, using a look-ahead multi-pass simulation, a scheduling rule that best advances the selected criterion is determined. Yu and Greene [14] use a simulation study to examine the effects of machine selection rules and scheduling rules for a flexible multi-stage pull system. J. Jerald and P. Asokan [7] developed a combined objective based scheduling solution for FMS, but the work was for only 43 parts. M. Saravanan & A. Noorul Haq[16] developed a scatter-search approach for the same problem. But the number of generations size was 100.

Many authors have been trying to emphasize the utility and advantages of GA, SA and other heuristics. In this vein, it has been proposed to use a new evolutionary computative approach such as MA,PS for the scheduling problem in FMS. In this work, a non-conventional optimization procedure - GA has been used to find the optimal schedules for a specific manufacturing environment by considering dual objectives. The procedures is applied to relatively large-size problems of up to 80 part varieties passing through 16 different CNC machine centers, and the results are found to be closer to the global optimum sequence.

II. PROBLEM DESCRIPTIONS

The problem environment, assumption and aim of the present work are as follows:

1. The FMS considered in this work has a configuration as shown in Fig. 1. There are five flexible machining cells (FMCs), each with two to six computer numerical machines (CNCs), an independent and a self-sufficient tool magazine, one automatic tool changer (ATC) and one automatic pallet changer (APC). Each cell is supported by one to three dedicated robots for intra-cell movement of materials between operations. There is a loading station from which parts are released in batches for manufacturing in the FMS. There is an unloading station where the finished parts are collected and conveyed to the finished storage. There is one automatic storage and retrieval system (AS/RS) to store the work in progress. The five FMCs are connected by two identical automated guided vehicles (AGVs). These AGVs perform the inter cell 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.

2. The assumptions made in this work are as follows:

There are 80 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 deadline. Each processing step has a processing time with a specific machine.

3. The objective of the schedule is the combination of minimizing the machine ideal time and minimizing the total penalty cost.

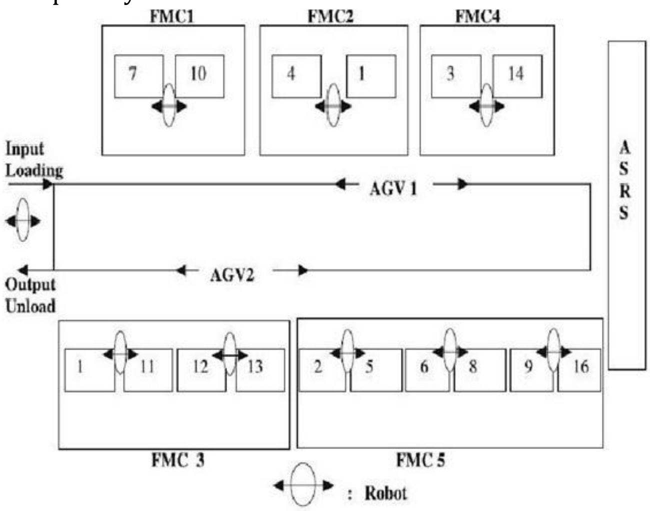

Figure 1.FMS structure

III. PROPOSED METHODOLOGY

III.IGenetic algorithm

A genetic algorithm (GA) is a procedure used to find approximate solutions to search problems through application of the principles of evolutionary biology. Genetic algorithms use biologically inspired techniques such as genetic inheritance, natural selection, mutation, and sexual reproduction (recombination, or crossover). Along with genetic programming (GP), they are one of the main classes of genetic and evolutionary computation (GEC) methodologies.

Genetic algorithms are typically implemented using computer simulations in which an optimization problem is specified. For this problem, members of a space of candidate solutions, called individuals, are represented using abstract representations called chromosomes. The GA consists of an iterative process that evolves a working set of individuals called a population toward an objective function, or fitness function. (Goldberg, 1989; Wikipedia, 2004). Traditionally, solutions are represented using fixed length strings, especially binary strings, but alternative encodings have been developed.).The working of the GA can be understood by the following steps, which is shown in figure 2.

Step 1. Generate the initial population. The size of the population is 100 and the maximum number of the generation is 1500.

Step 2. Calculate the fitness value of each member of the initial population.

Step 3. Calculate the selection probability of each member of the initial population using the ratio of fitness value of that initial.

Step 4. Select a pair of members (parents) that can be used for reproduction using tournament selection probability.

Step 5. Apply the genetic operators such as crossover, mutation, and inversion to the parents. Replace the parents with the new offspring to form a new population. Check the size of the new population. If it is equal to the initial population size, then go to step 6, otherwise go to step 4.

Step 6. If the current generation is equal to the maximumshould d number of the generation then stop, else move to step hav

2.The first paragraph under each heading or subheading should be flush left, and subsequent paragraphs should have

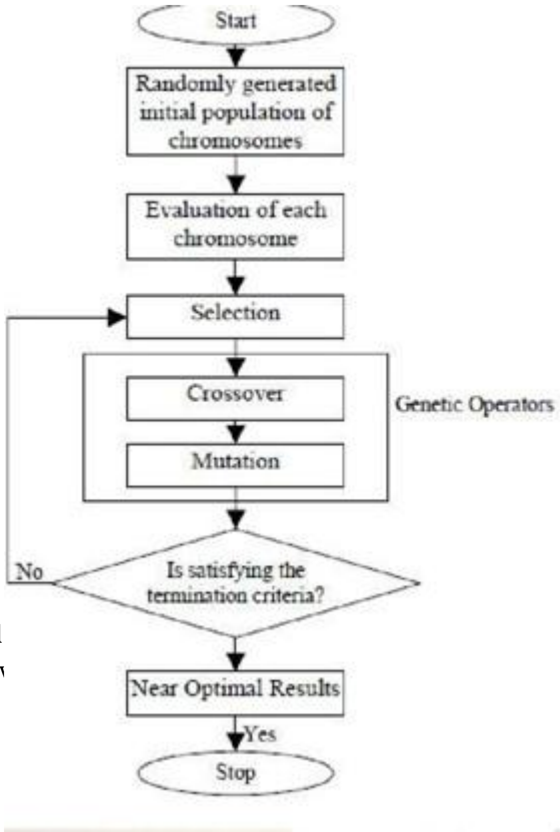
Start
Randomly generated initial population of chromosomes
Evaluation of each chromosome
Selection
Crossover
Mutation
Genetic Operators
No
Is satisfying the termination criteria?
Near Optimal Results
Yes
Stop

Figure 2. Flow chart of Genetic Algorithm

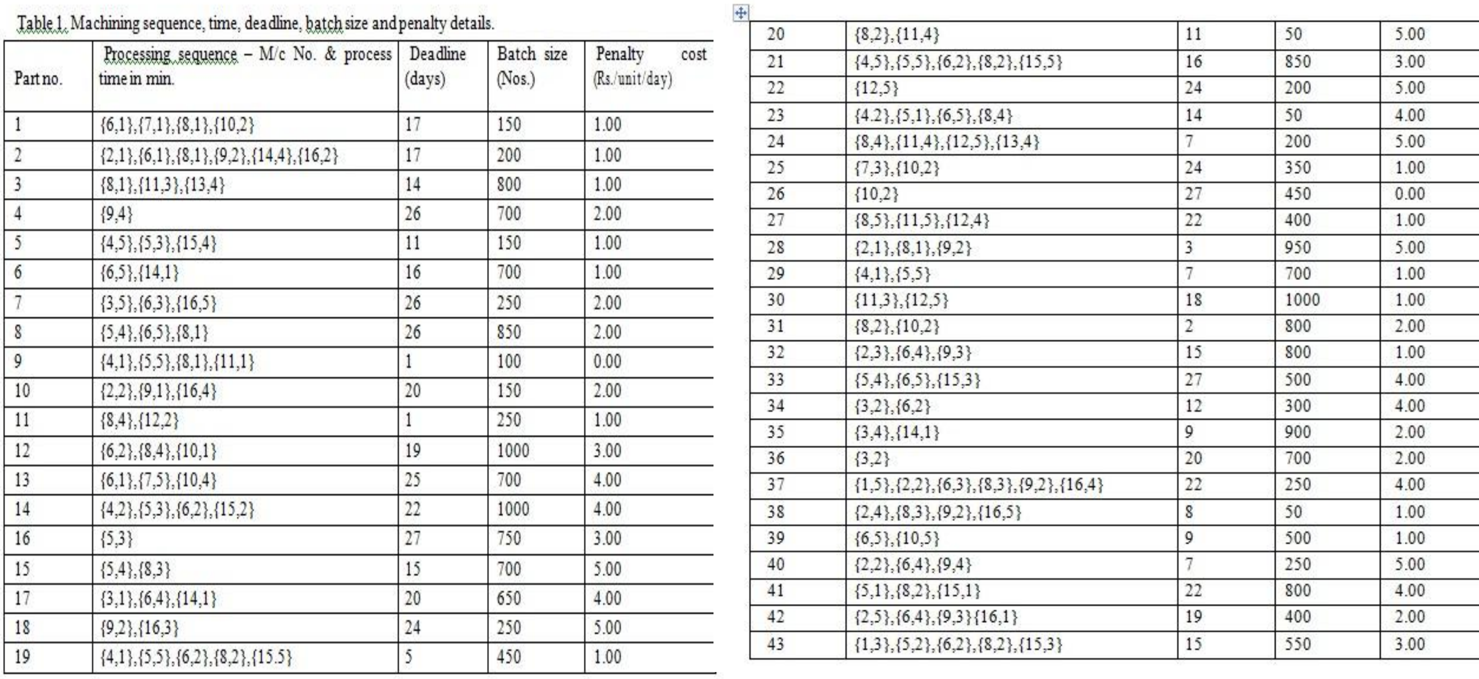

Table 1. Machining sequence, time, deadline, batch size and penalty details.

Part no.	Processing sequence – M/c No. & process time in min.	Deadline (days)	Batch size (Nos.)	Penalty cost (Rs./unit/day)
1	{6,1},{7,1},{8,1},{10,2}	17	150	1.00
2	{2,1},{6,1},{8,1},{9,2},{14,4},{16,2}	17	200	1.00
3	{8,1},{11,3},{13,4}	14	800	1.00
4	{9,4}	26	700	2.00
5	{4,5},{5,3},{15,4}	11	150	1.00
6	{6,5},{14,1}	16	700	1.00
7	{3,5},{6,3},{16,5}	26	250	2.00
8	{5,4},{6,5},{8,1}	26	850	2.00
9	{4,1},{5,5},{8,1},{11,1}	1	100	0.00
10	{2,2},{9,1},{16,4}	20	150	2.00
11	{8,4},{12,2}	1	250	1.00
12	{6,2},{8,4},{10,1}	19	1000	3.00
13	{6,1},{7,5},{10,4}	25	700	4.00
14	{4,2},{5,3},{6,2},{15,2}	22	1000	4.00
16	{5,3}	27	750	3.00
15	{5,4},{8,3}	15	700	5.00
17	{3,1},{6,4},{14,1}	20	650	4.00
18	{9,2},{16,3}	24	250	5.00
19	{4,1},{5,5},{6,2},{8,2},{15.5}	5	450	1.00
20	{8,2},{11,4}	11	50	5.00
21	{4,5},{5,5},{6,2},{8,2},{15,5}	16	850	3.00
22	{12,5}	24	200	5.00
23	{4.2},{5,1},{6,5},{8,4}	14	50	4.00
24	{8,4},{11,4},{12,5},{13,4}	7	200	5.00
25	{7,3},{10,2}	24	350	1.00
26	{10,2}	27	450	0.00
27	{8,5},{11,5},{12,4}	22	400	1.00
28	{2,1},{8,1},{9,2}	3	950	5.00
29	{4,1},{5,5}	7	700	1.00
30	{11,3},{12,5}	18	1000	1.00
31	{8,2},{10,2}	2	800	2.00
32	{2,3},{6,4},{9,3}	15	800	1.00
33	{5,4},{6,5},{15,3}	27	500	4.00
34	{3,2},{6,2}	12	300	4.00
35	{3,4},{14,1}	9	900	2.00
36	{3,2}	20	700	2.00
37	{1,5},{2,2},{6,3},{8,3},{9,2},{16,4}	22	250	4.00
38	{2,4},{8,3},{9,2},{16,5}	8	50	1.00
39	{6,5},{10,5}	9	500	1.00
40	{2,2},{6,4},{9,4}	7	250	5.00
41	{5,1},{8,2},{15,1}	22	800	4.00
42	{2,5},{6,4},{9,3}{16,1}	19	400	2.00
43	{1,3},{5,2},{6,2},{8,2},{15,3}	15	550	3.00

Combined objective function:

As the GA work on coding of parameters, the feasible job sequences (the parameters of the considered problems) are coded in two different ways and separately experimented for the same problem.

(1) Fino style coding

(2) Binary coding

In this work, Fino style coding is considered.

Fino style coding:

In this coding each sequence is coded as 80 sets of two-digit numbers ranging from 01 to 43.

31,28,1,38,18,30,9,3,25,13,23,34,24,7,40,26,5,4,27,2,
20,16,10,36,,9,11,37,35,19,12,41,22,42,29,32,15,43,1

7,21,6,33,14,8

GA parameters

Population size = 100

Reproduction: Tournament selection (Target value – 0.75)

Crossover probability= 0.6

Mutation probability = 0.01

Termination criteria = 1700 number of generations or a satisfactory pre-defined value for COF, whichever occurs first.in your paper cannot be confused with a reference [4] or an equation (3) designation.

IV. GENETIC OPERATIONS

IV.I REPRODUCTION

The tournament selection method is used for reproduction. Tournament selection is one of many methods of selection in genetic algorithms. Tournament selection involves running several "tournaments" among a few individuals chosen at random from the population. The winner of each tournament (the one with the best fitness) is selected for crossover. Selection pressure is easily adjusted by changing the tournament size. If the tournament size is larger, weak individuals have a smaller chance to be selected. Reproduction procedure as follows:

Selection method: tournament selection. (Assume the parameters for comparison as 0.75)

Step 1: select two samples from the population.

Step2: evaluate the population.

Step3: generate random no. in the range (0 to 1)

Step4: if the random number is <= 0.75, select the best one else, select the inferior one.

IV.II CROSSOVER

The strings in the mating pool formed after reproductions are used in the crossover operation. Single-point crossover is used in this work. With a Fino-type coding scheme, two strings are selected at random and crossed at a random site. Since the mating pool contains strings at random, we pick pairs of strings from the top of the list. When two strings are chosen for crossover, first a coin is flipped with a probability Pc = 0.6 check whether or not a crossover is desired. If the outcome of the coin flipping is true, the crossover is performed; otherwise the strings are directly placed in the intermediate population for subsequent genetic operation. Flipping a coin with a probability 0.6 is simulated using the Monte Carlo method. The next step is to find a cross site at random. Total 100 samples and 50 pairs 50 * 0.6 =30 pairs selected for crossover.

IV.III MUTATION

The classic example of a mutation operator involves a probability that an arbitrary bit in a genetic sequence will be changed from its original state. A common method of implementing the mutation operator involves generating a random variable for each bit in a sequence. This random variable tells whether or not a particular bit will be modified. The purpose of mutation in GAs is to allow the algorithm to avoid local minima by preventing the population of chromosomes from becoming too similar to each other, thus slowing or even stopping evolution. This reasoning also explains the fact that most GA systems avoid only taking the fittest of the population in generating the next but rather a random (or semi-random) selection with a weighting toward those that are fitter. In this problem mutation probability is 0.01 (i.e.) 8 bits will be mutated. First generate random number 0 to 1, with 0.01 accuracy. If random number is <= 0.01, perform mutation.

V. RESULTS & DISCUSSIONS

The optimization procedures developed in this work are based on the various non-traditional approaches that have been implemented using .net language. Different optimal schedules are obtained for the FMS using different non-traditional algorithms and compared. Among the approaches used in this work, the schedule obtained by the genetic algorithm give the optimal COF value, i.e. minimum total penalty cost and minimum machine idle time as shown in the table 2.The figure 3 shows the optimization result after performing 1700 generations. Different combinations of genetic operators have been applied and are given in Table 3. The figure 4 shows the Effect of different genetic parameters in a three dimensional graph. Crossover probability 0.6 and mutation probability 0.01 gives the minimum combined objective function. Optimum production sequence is obtained during 1542 th generation at sample no.98. For the optimum sequence, the corresponding combined objective function is 0.113073.

Optimum sequence:
31,28,1,38,18,30,9,3,25,13,23,34,24,7,40,26,5,4,27,2,
20,16,10,36,,9,11,37,35,19,12,41,22,42,29,32,15,43,1
7,21,6,33,14,8.

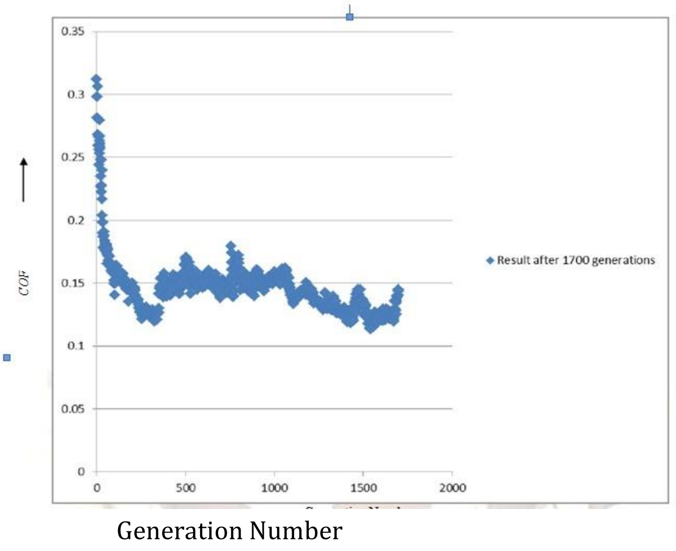

Figure 3.FMS Scheduling Optimization Result Using Genetic algorithm

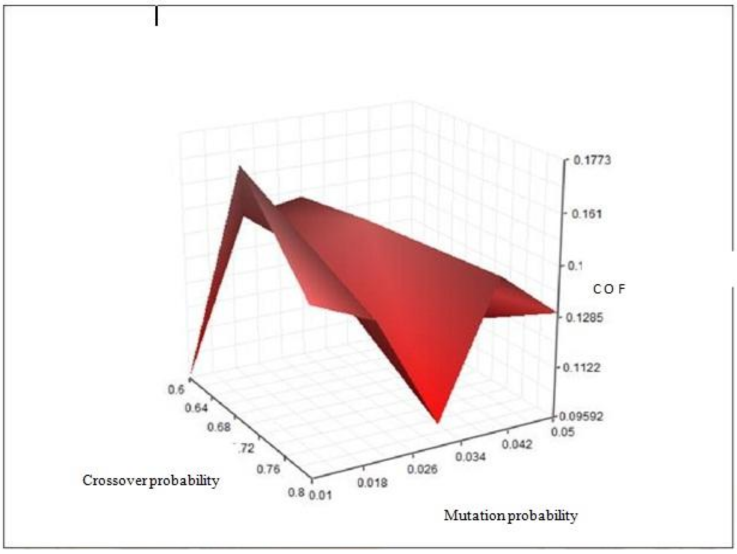

Figure 4 Effect of different genetic parameters

VI. CONCLUSIONS

Optimization procedure has been developed in this work which is based on genetic algorithm and is implemented successfully for solving the scheduling optimization problem of FMS. Software has been written in .net language. Results are obtained for the 43 jobs and 16 machines FMS system. With less computational effort it is possible to obtain the solution for such a large number of jobs (43) and machines (16).This work leads to the conclusion that the procedures developed in this work can be suitably modified to any kind of FMS with a large number of components and machines subject to multi objective functions. Future work will include availability and handling times of loading/unloading stations, robots and AGVs.

References

[1]AVS Sreedhar kumar and Dr.V.Veeranna (2010), "Optimization of FMS scheduling using non-traditional techniques" International Journal of Engineering Science and Technology Vol. 2 (12), 7289-7296.

[2]Chisman JA (1986), Manufacturing cell: analytical setup times and partsequencing., Int J Adv Manuf Technol, 1(5):55–60

[3]Chan TS, Pak HA (1986), Heuristical job allocation in a flexible manufacturing system. Int J Adv Manuf Technol., 1(2):69–90

[4]He W, Kusiak A (1992), Scheduling of manufacturing systems, Int. J Comput Ind., 20:163–175

[5]Giffler B, Thomson GL (1960), Algorithms for solving production scheduling problems. Int. Journ. Of Oper. Res., 8:487–503

[6]Hoitomt DJ, Luh PB, Pattipati KR (1993), A practical approach to job-shop scheduling problems. IEEE Trans Robot Automat., 9(1):1– 13

[7].J. Jerald , P. Asokan , G. Prabaharan , R. Saravanan (2004), Scheduling optimisation of

flexible manufacturing systems using particle swarm optimisation algorithm, Int. Journ. Of Advanced Manufacturing Technology, Vol.25, Numbers 9-10, Pages: 964 – 971

[8]Lee DY, Dicesare F (1994), Scheduling of
flexible manufacturing systems: using Petri nets and heuristic search, IEEE Trans Robot Automat, 10(2):23–132

[9]R Kumar, M K Tiwari and R Shankar (2003), Scheduling of flexible manufacturing systems: an ant colony optimization approach, Proc. Instn Mech. Engrs Vol. 217 Part B: J. Engineering Manufacture

[10]Shnits B, Sinreich D (2006) Controlling flexible manufacturing systems based on a dynamic selection of the appropriate operational criteria and scheduling policy. Int J Flex Manuf Syst 18:1–27

[11]Tiwari M.K, Vidiyarthi N.K(2000), "Solving
Machine loading problem in flexible manufacturing system using genetic algorithm based heuristic approach" International
journal of production research,vol.38 no.141.3357-87.

[12]Toker A, Kondakci S, Erkip N (1994), Job shop Scheduling under a nonrenewable resource constraint. J Oper Res Soc., 45(8):942–947

[13]Steeke KE, Soldberg JJ (1982), Loading and control policies for a flexible manufacturing system. Int J Prod Res., 19(5):481–490

[14]Yu MC, Greene TJ (2006) A simulation analysis of controlling rules for flexible pull systems. Int J Manuf Res 1(3):314–331

[15]Z. X. Guo , W. K. Wong, S. Y. S. Leung, J. T. FanS, F. Chan(2008), A genetic-algorithm-based optimization model for scheduling flexible assembly lines. Int J Adv Manuf Technol 36:156–168.

[16]M. Saravanan,A. Noorul Haq, Evaluation of scatter-search approach for scheduling optimization of flexible manufacturing systems. nt J Adv Manuf Technol (2008) 38:978–986.