Accepted for publication in the
International Journal Agile System & Management (IJASM),
Special Issue on: "Complexity in Manufacturing",
Inderscience Publishers, 2007

Task assignment and scheduling in a constrained manufacturing system using GA

B.B.Choudhury*
National Institute of Technology, Rourkela, Orissa, INDIA
E-Mail:bbcnit@gmail.com
D.Mishra
University College of Engineering, Burla,Orissa, INDIA
E-Mail: dmishrauce@gmail.com
B.B.Biswal
National Institute of Technology, Rourkela, Orissa, INDIA
E-Mail:bibhuti.biswal@gmail.com
*Corresponding author

Abstract: Driven by open global competition and rapidly changing technology, manufacturing organizations come across continuous change and significant amount of uncertainty. With increasing trend in customers' demand for a greater variety, high quality and competitive cost, traditional manufacturing approaches face threats to remain competitive. Flexible Manufacturing Systems (FMS) have brought in significant benefits to manufacturing sectors. The ability of FMS to flex to both internal and external changes gives rise to improvement in throughput, product quality, information flows, reliability, and other strategic advantages. However, the constraints with the individual candidates of FMS must be considered while assigning tasks and making schedules. This makes the scheduling problem complex. Appropriate scheduling methodology can reap better results. In the present work, Genetic algorithm is utilized for optimization of scheduling FMS. This approach is used to determine the best available combination for processing the product and the method has been shown through suitable example.

Keywords: FMS, Task assignment, Scheduling, Constraints, Genetic Algorithm

Reference to this paper should be made as follows: Choudhury, B.B., Mishra, D., and Biswal, B.B. (2008) 'Task assignment and scheduling in a constrained manufacturing system using GA', *Int. J. Agile Systems and Management, Special issue on "Complexity in Manufacturing"*.

Biographical notes: B.B.Choudhury graduated in Mechanical Engineering in Mechanical Engineering Department in the year 1998 From Marathwada University, India. He did his M.Tech Degree in Production Engineering from University College of Engineering Burla, India. Presently he is perusing his Ph.D work at National Institute of Technology, Rourkela as an Institute Scholar in the Department of Mechanical Engineering. His areas research interest includes Multi Robot Task allocation, Flexible manufacturing system & Manufacturing Science.
Dr.D.Mishra graduated in Mechanical Engineering in the year 1988 from College of Engg. & Technology, Bhubaneswar, India. He did his Masters degree & Ph.D from Sambalpur University in 1990 and 2003 respectively. He joined as

a Lecturer in Mechanical Engg. Department at UCE, Burla, India in the year 1991 and subsequently as Reader in Manufacturing Science & Engg in the same college. His research interests include FMS, Robotics & System optimization.
Dr. B.B.Biswal graduated in Mechanical Engineering in 1985 from UCE, Burla. He did his M.Tech. in Production Engineering and subsequently completed Ph.D. in the area of Robotics and CIM from Jadavpur University. After a short stint as GET at Hindustan Aeronautics Limited, he joined as Lecturer in Mechanical Engg at UCE, Burla. His areas research interest includes Robotics FMS and CIM. He was a visiting Professor of MSTU, Stankin and visiting Scientist to G.I.S.T., South Korea. He has published more than 40 papers at National and International levels. Presently he is a Professor at National Institute of Technology. Rourkela, India.

1 Introduction

The global nature of the present manufacturing environment has necessitated an improvement in the way companies manufacture their products. This growth in practical demand has been matched by development in theoretical research. The current state of production scheduling is a mixture of approaches from different areas. With the increasing sophistication of production practices there has been a corresponding increase in the importance and profitability of efficient production scheduling. The intractability of the problem also lends itself to making the developments widely varied. Since the scheduling problem is not amenable to any particular solution, the frontiers of research in this area are vast (Buyurgan et al.,2004; Moon et al., 2004). As a matter of fact scheduling of manufacturing process is treated as NP-hard problem, and it can be treated as a subset of operational research (Lee et al., 2003).

The utility of a Flexible Manufacturing System (FMS) lies in mid -volume and mid-variety part types. FMS is designed to combine the high efficiency of a transfer line and the flexibility of a job shop to best suit to the batch production of mid-volume and mid-variety parts. Today’s manufacturing strategy is to seek benefits from flexibility (Vieira et al.,2003). This is feasible when a production system is under complete control of FMS technology. Having in mind the process-product matrix, it may be realized that for an industry it is possible to reach for high flexibility by making innovative technological and organizational efforts (Lee et al.,2002). There has been a paradigm shift in manufacturing industries over the years which can be attributed to this idea (Sun et al., 2001).

Given the part types and their volume in each batch FMS scheduling is concerned with the real time operation of the system and the allocation of tools to the machine and allocation of operations to machines. In other words FMS scheduling is concerned with the following:

a) Releasing of part types to the system: Only a subset of the part types constitutes a batch. Releasing rule prioritizes the part type of the batch leading to their ordered entry to the system.
b) Assignment of operations of part type to machines: routing flexibility provides alternate machines for an operation of a part type. Operation assignment rule is used to assign an operation to one amongst the alternate machines available for the purpose.
c) Dispatching of part types waiting for processing before a machine: At any given point of time several part types wait in the local buffer for their turn to get service in a machine. Dispatching rules are used to prioritize them.

Production scheduling concerns the efficient allocation of resources over time for the manufacture of goods. The problems in scheduling arise whenever a common set of resources-labor, material and equipment must be used to make a variety of different parts during the same period of time. The objective of the present scheduling problem is to find a way to assign and to sequence the activities of these shared resources such that production constraints are satisfied and production costs are minimized.

2 Objectives of the work

The present work is envisaged to work out the optimal scheduling process for modular FMS setups. 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). Precisely, the objective of the present work is to optimize the scheduling of a typical FMS setup.

In order to accomplish the objective, the following methodology is split into the following:

- Analysis of parts to be produced in an FMS,
- Detailing the machining processes involved in manufacture of the parts,
- Application of GA for scheduling,
- Optimization of scheduling time with alternate assignments within FMS.

The optimization technique has been applied to three example setups.

3 Model formulation

3.1 Description of the parts

FMS has the capability to process large number of part types. However, in the present study the parts to be processed in the selected setups, are so chosen that they are almost similar in their functions with differentiations in their physical and geometrical properties.

The study of the physical properties (design attributes) and manufacturing requirements (manufacturing attributes) of the considered parts put them under one group from group technology viewpoint. The parts are manufactured in batches and depending on the demand there can be variation in batch size as well as the product renewal rate. The machining requirements are almost same for all the parts. The parts 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 parts are: 1) facing, 2) turning, 3) drilling, 4) boring, and 5) thread cutting. The details of machining operations of part-1, part-2 and part-3 (as shown in Figure.1, Figure.2 and Figure.3 respectively) are given in Table1, Table 2 and Table 3 respectively.

3.2 Description of the setups

The three setups under consideration consist of four machines (M) to accomplish the desired machining operations viz. facing, turning, drilling, boring and thread cutting as described before, on all the three parts.

Setup-1 consists of two numbers of lathes, namely lathe-1(M_1) and lathe-2 (M_3) and two numbers of machining centers, machining center-1 (M_2) and machining center-2 (M_4). In setup-2, a CNC drilling machine replaces the machining center-2 of setup-1 as machine M_4. Rest of the machines in the setup is unaltered. In setup-3, another CNC drilling machine replaces the machining center-1 of setup-2 as machine M_2.Rest of the machines in the setup remain same as in setup-2.

Alternate routes have been considered in the setups for all the three parts with machine breakdown and availability fully captured.

Table 1 Processing operations for part-1

Sl.No.	Operation	Tool used
1	Facing of face 1 (F_{11})	Facing tool
2	Turning of φ 75 (T_{11})	Turning tool
3	Drilling of φ 2.5 (D_{11})	Drills
4	Step boring of φ 60 and φ 20 (B_{11})	Boring tool
5	Facing of face 2 (F_{12})	Facing tool
6	Turning of φ 25 (T_{12})	Turning tool
7	Thread cutting M30×2.5 (TH_{11})	Threading tool
8	Drill φ 6, 3 nos. (D_{12})	Drills

Figure 1 Graphical model of part 1

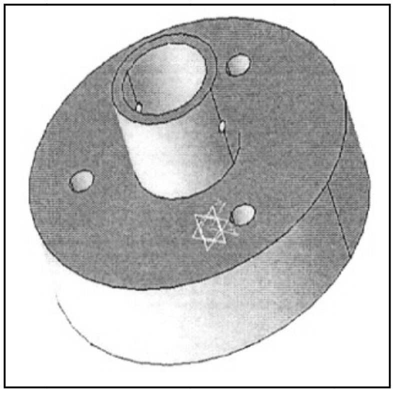


Table 2 Processing operations for part-2

Sl.No.	Operation	Tool used
1	Facing of face 1 (F_{21})	Facing tool
2	Turning of φ 114 (T_{21})	Turning tool
3	Drilling of φ 7,4 nos. (D_{21})	Drills
4	Step boring of φ 76,φ 60 & φ 32 (B_{21})	Boring tool
5	Facing of face 2 (F_{22})	Facing tool
6	Turning of φ 72 (T_{22})	Turning tool
7	Drill φ 5, 2 nos. (D_{22})	Drills
8	Thread cutting M30×5, 2 nos. (TH_{21})	Threading tool

Figure 2 Graphical model of part 2

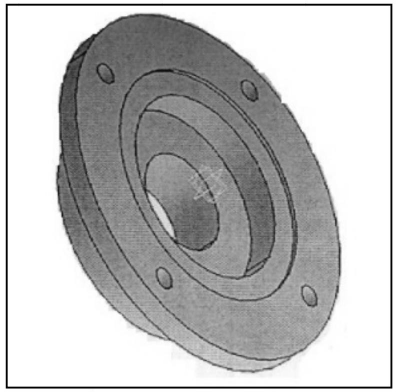
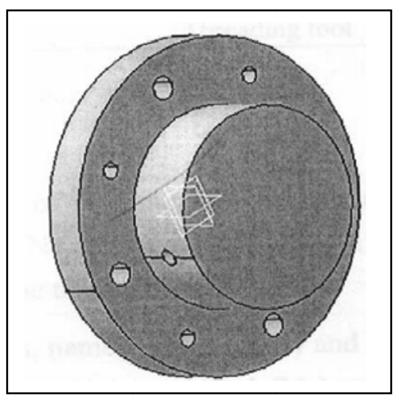


Table 3 Processing operations for part-3

Sl.No.	Operation	Tool used
1	Facing of face 1(F_{31})	Facing tool
2	Turning of φ 100 (T_{31})	Turning tool
3	Drilling of φ 7,4 nos. (D_{31})	Drills
4	Drilling of φ 5,4 nos. (D_{32})	Drills
5	Thread cutting M30×5, 4 nos. (TH_{31})	Threading tool
6	Facing of face 2 (F_{32})	Facing tool
7	Turning of φ 60 (T_{32})	Turning tool
8	Drilling of φ 2.5, (D_{33})	Drills
9	Thread cutting M30×2.5 (TH_{32})	Threading tool

Figure 3 Graphical model of part 3



The three different alternate routes via which the parts are manufactured in setup-1 are:

$R_1 = M_1 \rightarrow M_2 \rightarrow M_3 \rightarrow M_,$

$R_2 = M_3 \rightarrow M_4 \rightarrow M_{2,}$ and

$R_3 = M_1 \rightarrow M_4 \rightarrow M_3 \rightarrow M_4$.

The operations performed, at all the machines via different routes, for each part in setup-1 are given in Table 4.

Table 4 Machines on routes of setup-1

Part 1	Route 1:M_1 (F_{11},T_{11}) $\rightarrow$ M_2 (D_{11},B_{11}) $\rightarrow$ M_3 (F_{12},T_{12}) $\rightarrow$ M_4 (D_{12},TH_{11}) Route 2:M_3 (F_{11},T_{11}) $\rightarrow$ M_4 (D_{11},B_{11}) $\rightarrow$ M_2 (F_{12},T_{12},D_{12}) $\rightarrow$ M_3 (TH_{11}) Route 3:M_1 (F_{11},T_{11}) $\rightarrow$ M_4 (D_{11},B_{11}) $\rightarrow$ M_3 (F_{12},T_{12}) $\rightarrow$ M_4 (D_{12},TH_{11})
Part 2	Route 1:M_1 (F_{21},T_{21}) $\rightarrow$ M_2 (D_{21},B_{21}) $\rightarrow$ M_3 (F_{22},T_{22}) $\rightarrow$ M_4 (D_{22},TH_{21}) Route 2:M_3 (F_{21},T_{21}) $\rightarrow$ M_4 (D_{21},B_{21}) $\rightarrow$ M_2 (F_{22},T_{22},D_{22}) $\rightarrow$ M_3 (TH_{21}) Route 3:M_1 (F_{21},T_{21}) $\rightarrow$ M_4 (D_{21},B_{21}) $\rightarrow$ M_3 (F_{22},T_{22}) $\rightarrow$ M_4 (D_{22},TH_{21})
Part 3	Route 1:M_1(F_{31},T_{31})$\rightarrow$$M_2$($D_{31}$,$D_{32}$,$TH_{31}$)$\rightarrow$$M_3$($F_{32}$,$T_{32}$)$\rightarrow$$M_4$($D_{33}$, TH_{32}) Route2:M_3(F_{31},T_{31})$\rightarrow$$M_4$($D_{31}$,$D_{32}$,$TH_{31}$)$\rightarrow$$M_2$($F_{32}$,$T_{32}$,$D_{33}$)$\rightarrow$ M_3 (TH_{32}) Route3:M_1(F_{31},T_{31})$\rightarrow$$M_4$($D_{31}$,$D_{32}$,$TH_{31}$)$\rightarrow$$M_3$($F_{32}$,$T_{32}$)$\rightarrow$$M_4$ (D_{33}, TH_{32})

The routes for setup-2 are:

$R_1 = M_1 \rightarrow M_2 \rightarrow M_3 \rightarrow M_4$,

$R_2 = M_3 \rightarrow M_4 \rightarrow M_2 \rightarrow M_{3,}$ and

$R_3 = M_1 \rightarrow M_4 \rightarrow M_3 \rightarrow M_4$.

The operations performed, at all the machines via different routes, for each part in setup-2 are given in Table 5.

Table 5 Machines on routes of setup-2

Part 1	Route 1:M_1(F_{11},T_{11})$\rightarrow$ M_2(D_{11},B_{11})$\rightarrow$ M_3(F_{12},T_{12}, TH_{11}) $\rightarrow$ M_4 (D_{12}) Route 2:M_3 (F_{11},T_{11}) $\rightarrow$ M_4 (D_{11},B_{11}) $\rightarrow$ M_2 (F_{12},T_{12},D_{12}) $\rightarrow$ M_3 (TH_{11}) Route 3:M_1 (F_{11},T_{11}) $\rightarrow$ M_2 (D_{11},B_{11}) $\rightarrow$ M_3 (F_{12},T_{12}) $\rightarrow$ M_2 (D_{12},TH_{11})
Part 2	Route 1:M_1 (F_{21},T_{21})$\rightarrow$ M_2 (D_{22},B_{21}) $\rightarrow$ M_3 (F_{22},T_{22}, TH_{21}) $\rightarrow$ M_4 (D_{21}) Route 2:M_3 (F_{21},T_{21}) $\rightarrow$ M_4 (D_{21},B_{21}) $\rightarrow$ M_2 (F_{22},T_{22},D_{22}) $\rightarrow$ M_3 (TH_{21}) Route 3:M_1 (F_{21},T_{21}) $\rightarrow$ M_2 (D_{21},B_{21}) $\rightarrow$ M_3 (F_{22},T_{22}) $\rightarrow$ M_2 (D_{22},TH_{21})
Part 3	Route1:M_1 (F_{31},T_{31})$\rightarrow$$M_2$($D_{32}$,$D_{33}$,$TH_{31}$,$TH_{32}$)$\rightarrow$$M_3$($F_{32}$,$T_{32}$)$\rightarrow$$M_4$ (D_{31}) Route2:M_3(F_{31},T_{31})$\rightarrow$$M_4$($D_{31}$,$D_{32}$,$D_{33}$)$\rightarrow$$M_2$($F_{32}$,$T_{32}$)$\rightarrow$ M_3 (TH_{31},TH_{32}) Route3:M_1(F_{31},T_{31})$\rightarrow$ M_2(D_{31},D_{32},TH_{31})$\rightarrow$$M_3$($F_{32}$,$T_{32}$)$\rightarrow$$M_2$($D_{33}$, TH_{32})

The routes for setup-3 are:

$R_1 = M_1 \rightarrow M_2 \rightarrow M_3 \rightarrow M_4$,

$R_2 = M_3 \rightarrow M_4 \rightarrow M_2 \rightarrow M_{3,}$ and

$R_3 = M_1 \rightarrow M_2 \rightarrow M_3 \rightarrow M_2$.

The operations performed, at all the machines via different routes, for each part in setup-3 are given in Table 6.

Table 6 Machines on routes of setup-3

Part 1	Route1:M_1 (F_{11},T_{11})→ M_2(D_{11},B_{11})→M_3(F_{12},T_{12},TH_{11})→M_4 (D_{12}) Route2:M_3 (F_{11},T_{11}) → M_4 (D_{11},B_{11}) → M_2 (D_{12}) → M_3 (F_{12},T_{12},TH_{11}) Route3:M_1 (F_{11},T_{11}) → M_2 (D_{11},B_{11}) → M_3 (F_{12},T_{12},TH_{11}) → M_2 (D_{12})
Part 2	Route1:M_1 (F_{21},T_{21}) → M_2 (D_{22},B_{21}) → M_3 (F_{22},T_{22},TH_{21}) → M_4 (D_{21}) Route2:M_3 (F_{21},T_{21}) → M_4 (D_{21}, D_{22}) → M_2 (B_{21})→ M_3 (F_{22},T_{22},TH_{21}) Route3:M_1 (F_{21},T_{21}) → M_2 (D_{22},B_{21}) →M_3 (F_{22},T_{22},TH_{21}) → M_2 (D_{21})
Part 3	Route1:M_1(F_{31},T_{31})→M_2(D_{32},D_{33})→M_3(F_{32},T_{32},TH_{31},TH_{32})→ M_4 (D_{31}) Route2:M_3(F_{31},T_{31},F_{32},T_{32})→M_4(D_{31},D_{32})→M_2(D_{33})→ M_3 (TH_{31},TH_{32}) Route3:M_1(F_{31},T_{31})→M_2(D_{32},D_{33})→M_3(F_{32},T_{32},TH_{31},TH_{32})→ M_2 (D_{31})

4 Methodology

A systematic approach is adopted to achieve the objectives of the present work through defining and understanding the manufacturing scenario in FMS. This forms a base to use a mathematical model for optimization of the scheduling time for the systems under consideration. The study of previous literatures reveals that the flexibility measurement provides a better understanding of the ability of a production system. The different approaches made by researchers are mostly theoretical and specific problems are not dealt with. Further, it is observed that a good number of authors have attempted to relative and logical assessing of manufacturing a system. However, the present study is aimed at quantification of manufacturing systems. Scheduling of FMS as a whole is a complex concept influenced by a large number of components with the machines, the flow pattern of the inventories, the processing operations, the parts and the material handling systems being the major ones. The effects of change in these components can be studied accurately by considering an actual production system. However, in the present study, virtual production environments analogous to actual manufacturing facilities in shop floor have been considered which facilitate better manipulation for the purpose of flexibility study in changed situations. The results of design of experiments give rise to concentrating the study on three numbers of setups, each producing same three numbers of parts through three alternate routes. The GA interface interacts with the modules of design process modeling and configuration and prepares the data necessary for the GA-engine. Using the data gathered, the GA-engine searches the near optimal solution among candidates in the population pool. The schedule displayer transforms each candidate solution into a feasible schedule by considering the resource constraints imposed, while the iterative design analyzer is invoked to assess the time span for the schedule. The objective function determines the quality of a schedule. In the following sections, the operators of GA used in this work, namely chromosome encoding, crossover and mutation operators as well as the fitness scaling and selection are addressed in detail. It is important to note that this work focuses on non-interrupted scheduling, where interruption is not allowed when a task is underway. In the present work an attempt has been made to optimize the scheduling of FMS setups using GA.

4.1 Simulation of the setups

The setups have been modeled in QUEST ver. 4.0 for analyzing the manufacturing process visually and obtain useful data for further analysis. QUEST is a 3D graphics based Queuing Event Simulation Tool, for performing graphical simulation of the complete setup. The production scenarios, product mixes and failure responses for machine and labour utilization, throughput bottlenecks and inventory evaluation are efficiently explored in this software. The information and statistics generated by the simulation gives useful inputs for studying the behavior of a manufacturing process and comparing the same with one another. QUEST facilitates the modeling of individual processing elements. The three setups under study are modeled using QUEST to study the operation of the entire setup, regulating the flow of inventory, determination of material handling time, determination of cycle time and getting a complete picture of scheduling and for plotting of the process charts.

Table 7 Machining time for different operations

Machines →		L-1	L-2	C-1	C-2	D-1	D-2
Part 1	F_{11}	020	030	×	×	×	×
	F_{12}	×	020	020	×	×	×
	T_{11}	060	070	×	×	×	×
	T_{12}	×	040	035	×	×	×
	D_{11}	×	×	100	120	090	100
	D_{12}	×	×	070	080	070	090
	B_{11}	×	×	100	120	090	100
	TH_{11}	×	080	055	060	×	×
Part 2	F_{21}	030	040	×	×	×	×
	F_{22}	×	60	050	×	×	×
	T_{21}	050	060	×	×	×	×
	T_{22}	×	080	070	×	×	×
	B_{21}	×	×	120	140	100	110
	D_{21}	×	×	080	100	070	090
	D_{22}	×	×	075	080	070	090
	TH_{21}	×	150	110	120	×	×
Part 3	F_{31}	100	110	×	×	×	×
	F_{32}	×	050	040	×	×	×
	T_{31}	080	100	×	×	×	×
	T_{32}	×	180	160	×	×	×
	D_{31}	×	×	120	140	100	120
	D_{32}	×	×	120	140	100	120
	D_{33}	×	×	020	020	020	25
	TH_{31}	×	200	180	200	×	×
	TH_{32}	×	040	025	030	×	×

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.

However, the individual machining times are obtained from modeling the machines and performing virtual operations in VNC ver. 5.0. VNC is an interactive 3D graphics based real time simulation software. This enables to improve the quality of CNC part programs, eliminate catastrophic program errors and optimize machining process. The fast and real time simulation eliminates the uncertainty about NC programs. It automatically detects collisions and near misses between tool and fixtures, spindle and workpiece and virtually any part in the work cell. The CNC lathe, CNC machining center and CNC drilling machine have been retrieved from the library of VNC and have been modified according to the need of the setups for performing the machining simulation. Virtual NC helps to reduce cycle times by more than 40% and avoid CNC machine down time for dry runs. The outputs of simulation in VNC have been used as inputs to the models in QUEST. The timings (in seconds) for each individual operations (such as F_{11}, T_{11}, F_{22}, T_{22}, D_{11}, D_{12}…etc.) were recorded for different machines on which the operations were actually carried out, from simulation and are presented in Table 7.

In the setups, robots carry out the loading/unloading operations in various machines for different parts. Since the detailed simulation of these operations cannot be performed in QUEST, the modeling and simulation of these operations are done using another simulation tool IGRIP ver. 5.0. IGRIP is an interactive 3D graphics simulation tool for designing, evaluation and off-line programming of robotic work cells. In IGRIP, robotic mechanism can be constructed and analyzed for cycle time, motion planning, collisions, near miss detection, I/O communication and motion constraints. This saves invaluable operator time and boosts productivity by eliminating unnecessary data manipulation. The loading/unloading and part orienting times so determined for different parts on different machines are used in the simulation of the setup in QUEST. However, the material transporting time through conveying elements are directly obtained in QUEST.

5 Genetic Algorithm

The GA is an optimization and search technique based on the principles of genetics and natural selection. A GA allows a population composed of many individuals to evolve under specified selection rules to a state that minimizes the “fitness”. The method was developed by John Holland in 1975 and finally popularized by one of his students; David Goldberg in 1989. A GA begins with a population of initial coded “guesses” (chromosomes) at a solution. An application specific fitness or objective function is applied to each chromosome, and based on the result; the better chromosomes are identified to “survive”. Survivor chromosomes are then spliced together to form a new generation (mating). Occasionally, portions of chromosomes are randomly altered (mutation). This process of fitness determination, mating, and mutation is repeated for a given number of generations, or until the chromosomes improve sufficiently to achieve a predefined goal.

5.1 Applications of GA

Optimization is the art of selecting the best alternative among a given set of options. In any optimization problem there is an objective function or objective that depends on a set of variables. GA is excellent for all tasks requiring optimization and is highly effective in any situation where many inputs (variables) interact to produce a large number of possible outputs (solutions).

A GA starts with a pool of feasible solutions (population) and a set of biologically inspired operators defined over the population itself. In each and every loop (or) cycle a new population of solutions is created by breeding and mutation, with the fitter solutions being more likely to procreate. According to evolutionary theories, only the most suited elements in a population are likely to survive and generate offspring, transmitting their biological inheritance to the next generation. GAs operate through a simple cycle of stages: creation of a population a strings, evaluation of each string, selection of the best strings, and reproduction to create a new population. Individuals are encoded as strings known as chromosomes (or) strings composed over an alphabet.

After reproduction, the cycle is repeated. New individuals are decoded and the objective function evaluated to give their fitness values. Individuals are selected for mating according to fitness and so the process continues. The average performance of individuals in a population is expected to increase as good individuals are preserved and bred, while less fit members die out. The GA is terminated under a given criteria, for example, a certain number of generations have been completed, a level of fitness has been obtained or a point in the search space has been reached. There are several parameters to fine-tune in a GA, such as population size and mutation frequency. These parameters can be chosen with experience or through experiments. An important factor in selecting the string representation for the search nodes is that all of the search nodes in the search space are represented and the representation is unique. It is also desirable, though not necessary, that the strings are in one-to-one correspondence with the search nodes.

The outline of the technique can be stated as follows:

Step1: Set up initial population of chromosomes at random & assign each of them a fitness value.
Step2: Select two chromosomes based on their fitness value.
Step3: Crossover them to produce an offspring.
Step4: Mutate the offspring.
Step5: Repeat steps 2 to 4 until the number of offsprings equal to that of parent‘s generation.
Step6: Replace the parent’s generation with the offsprings and regard it as the new generation.
Step7: Repeat steps 2 to 6 until either all the chromosomes are same or the maximum iterations are reached.

5.2 Operators of GA

The fundamental operators frequently employed in GA are encoding of a chromosome, selection, crossover and mutation (Forrest, 1993). This section describes these parameters in detail as follows:

i. Encoding a chromosome

A chromosome is a string consisting of a set of bits, which represents a point in the solution space. The set of bit symbols is called the alphabets. The chromosome should contain information about the solution it represents. There are different types of encodings in which some are discussed below:

- Binary encoding: The most fundamental and popular presentation of chromosome is the string of binary members, 0 and 1.

Each bit in the string can represent some characteristic of the solution or it could represent whether or not some particular characteristic was present. Considering above example, the problem of purchasing a car, a car can be simply evaluated using appearance (1: luxurious, 0: ordinary), speed (1: high, 0: low), and price (1: low, 0: high). Thus the car with luxurious appearance, high speed, and high price can be expressed with string 110 and so on.

- Permutation encoding*:* Permutation encoding can be used in ordering problems, such as the traveling salesman problem or a task-ordering problem. Every chromosome is a string of numbers, which represents number in a sequence.

- Value encoding: Direct value encoding can be used in problems where some complicated value, such as real numbers, is used and where value encoding is very good for some problems, it is often necessary to develop some specific crossover and mutation techniques for these chromosomes.

Chromosome 1: A B E D B C A E D D
Chromosome 2: N W W N E S S W N N binary encoding would not suffice.

In chromosome 1 above, A could represent a particular task, B another, etc. For chromosome 2 N could be north, S south and thus could be the path through a maze.

- Tree encoding: Tree encoding is used to actually have programs or expressions evolve. In tree encoding every chromosome is a tree of some objects, such as functions or commands in the programming language.

ii. Selection

Selection or Reproduction is a process in which individual strings (chromosomes) are copied according to their fitness. Intuitively one can think of the fitness function as some measure of profit, utility or goodness that is to be maximized. Copying strings according to their fitness or goodness means that strings with a higher value have a higher probability of contributing one or more offspring in the next generation.

This operator is an artificial version of natural selection, a Darwinian survival of the fittest among string creatures. There are many methods for selecting the best chromosomes: Roulette wheel selection, Boltzmann selection, Tournament selection, Rank selection, Steady state selection and others.

- Roulette wheel selection: Simple reproduction allocates offspring strings using a roulette wheel with slots sized according to fitness. This is a way of choosing members from the population of chromosomes in a way that is proportional to their fitness. Parents are selected according to their fitness. The better the fitness of the chromosome, the greater the chance it will be selected; however it is not guaranteed that the fittest member goes to the next generation.

- Steady state selection: This is not a particular method of selecting parents. The main idea of this type of selecting to the new population is that a big part of chromosomes can survive to next generation. The steady-state selection of GA works in the following way. In every generation a few good (with higher fitness) chromosomes are selected for creating new offspring. Then some bad (with lower fitness) chromosomes are removed and the new offspring is placed in their place. The rest of population survives to new generation.

- Elitism: The best chromosome (or a few best chromosomes) is copied to the population in the next generation. The rest are chosen in classical way. Elitism can very rapidly increase performance of GA.

- Rank selection: The roulette method of selection will have problems when the fatnesses differ greatly. For example, if the best chromosome fitness is 90% of the entire roulette wheel then the other chromosomes will have a slim chance of being selected. Rank selection first ranks the population and then every chromosome receives fitness from this ranking. The worst will have fitness 1, second worst 2 etc. and the best will have fitness N (number of chromosomes in population).

iii. Crossover

One of the most important operators in GA is crossover. Crossover is a means for two strings (parent) to produce two offsprings by mixing and matching their desirable qualities through a random process.
After reproduction crossover proceeds in two steps:

-Two strings are selected;
-Segments of each string are chosen at random and the information contained in the two segments is exchanged between two strings.

Several methods can be used for choosing the length and location of the exchange of sites. In this paper single point, two point, uniform and specific crossover methods are represented.

- Single point crossover: Randomly choose a crossover point, then for offspring 1

-Copy everything in parent 1 before the crossover point
-Copies everything in parent 2 after this point to the new generation. For offspring 2 do the reverse. If the crossover point is 3, then the offsprings are as follows:

Example:
Chromosome 1: A B C D E F G H I J
Chromosome 2: 0 1 2 3 4 5 6 7 8 9
Offspring 1: A B C 3 4 5 6 7 8 9
Offspring 2: 0 1 2 D E F G H I J

- Two point crossover: Two-point crossover is similar to the single point crossover but the only difference is that two crossover points are randomly chosen.

Chromosome 1: A B C D E F G H I J
Chromosome 2: 0 1 2 3 4 5 6 7 8 9
Offspring 1: A B C 3 4 5 6 7 8 J
Offspring 2: 0 1 2 D E F G H I 9
Here the crossover points were at 2 and 9.

- Uniform crossover: A certain number of genes are randomly selected to be swapped.

Chromosome 1: A B C D E F G H I J
Chromosome 2: 0 1 2 3 4 5 6 7 8 9
Offspring 1: 0 B C D E 5 G H I 9
Offspring 2: A 1 2 3 4 F 6 7 8 J

- Permutation crossover*:* In position-based crossover bit positions are randomly chosen along with one parent and the jobs in those positions are inherited from the parent to the offspring. The remaining jobs are inherited in the order in which they appear in the other parent the position based crossover in the sequencing problems can be viewed as a kind of uniform crossover as given in (Ishibuchi.,et al 1994).

iv. Mutation

Mutation is intended to prevent falling of all solutions in the population into a local optimum of the solved problem. Mutation operates on a single chromosome with very small probability. With this operation, one or more bits are chosen at random from the chromosomes and are changed into a different symbol in the alphabet. Mutation should not occur very often because GA will become random search when mutation is often. The different types of mutations are:

- Adjacent pair wise interchange: In this method we select an adjacent pair randomly and interchange those two elements in the pair.

Example:
Before Mutation: A B C D E F G H I
After Mutation: A B C E D F G H I

In the above example the pair D E is selected randomly and both elements are interchanged and the result obtained is shown above.

- Swap mutation: Swap mutation works by selecting two tasks and swapping them. The function Swap mutation takes a schedule (string) as an input. It randomly selects two element positions randomly from the string. The elements at those positions are swapped and hence a new string with a slight difference from the parent is obtained.

- Additive mutation: Additive mutation was first created to help in the movement of the substrings in the two-dimensional string. By just using any of the crossovers combined with the swap mutation function, the individual lengths of the substrings never changed.

- Scramble sub-list mutation: In this method two positions are randomly selected for the string and the elements between those points are scrambled (or) randomly placed.

- Shift mutation: The other type of mutation called Shift mutation defined by removing a job at one position and placing it at other position. The two positions are selected randomly and this mutation is also called Position based mutation as given in (Ishibuchi et al.,1994).

Example:
Before Mutation: A B C D E F G H I
After Mutation: A B G C D E F H I

In the above example, the positions selected randomly are 7 and 3. Hence the element ‘G’ at position 7 is shifted to position ‘3’, which is four places left to initial position.

5.3 Parameters of GA

i. Crossover probability: If there is no crossover i.e. crossover probability is 0%, whole new generation is made from exact copies of chromosomes from old population. If crossover is 100% then all the offsprings are made by crossover. But it is good to leave some part of old populations survive to next.

ii. Mutation probability: If there is no mutation, offspring are generated immediately after crossover without any change. If mutation is performed, one or more parts of a chromosome are changed. If mutation probability is 100%, whole chromosome is changed, if it is 0%, nothing is changed. Mutation generally prevents the GA from falling into local extremes. Mutation should not occur very often, because then GA will in fact change to random search.

iii. Population size: If there are too few chromosomes, GA has few possibilities to perform crossover and only a small part of search space is explored. On the other hand, if there are too many chromosomes, GA slows down. Hence a moderate sized population of 20-30 is generally used.

5.4 GA coding scheme

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

(1) Pheno style coding:

In this coding each sequence is coded as 43 sets of two digit numbers ranging from 01 to 08. (e.g.)

01 02 05 08 04 06 07 03.

Decoded sequence

1 2 5 8 4 6 7 3.

(2) Binary coding:

In this coding method each sequence is coded as a pair of strings each having 08 binary digits.(e.g.)

1011101000010001

Decoded sequence

1 3 4 5 7 2 6 8

6 Problem statement

The processing times for various operations are obtained from the graphical simulations of the processes at different machines. The processing times are used to describe the combined objective function of the genetic optimization process. The constraints between the machines in the setups and also the constraints lying within individual machine for the processing operations as presented in Table 8 are considered for the processing of the parts. These are termed as inter-machine restrictions and intra-machine restrictions (Paliouras et al., 2001; Whitley,2001;Bierwirth et al.,1999).For example in setup-1,if the operation F_{11} is carried out in M_1 ,the same operation is not performed by M_2,M_3 and M_4.Further in, the same machine (M_1) the other operations like F_{12} ,T_{12} and TH_{11} cannot be done. The former condition is an example of inter-machine restriction and the later is an example of intra-machine restriction. The inter-machine restrictions take care of the ease in operation for a machine taking into consideration the part orientation. All possible restrictions for all the three setups are found out and put as the rule base in the GA program. The inter-machine restrictions and intra-machine restrictions for setup-1, setup-2 and setup-3 are prepared respectively. The GA programs for various setups are run several times by varying the population size and the number of generations to obtain the optimal scheduling of the FMSs by minimizing the combined objective function i.e. by minimizing the total machining time for realization of the part. However the machine setup times are assumed to be same for all the machines.

After every generation of the GA cycle every individual in the population (i.e feasible schedule) will be evaluated for the combined objective function (COF) of minimizing the total penalty cost and maximizing machine utilization.

Table 8 Machining constraints for various machines in setup-1

Machine	Operations	Intra-machine	Inter-machine
M_1	F_{11},T_{11}, F_{12},T_{12}, TH_{11}	M1(1)=1,M1(3),M1(4)=0, M1(5)=0 M1(2)=1,M1(3)=0,M1(4)=0, M1(5)=0 M1(3)=1,M1(1)=0,M1(2)=0 M1(4)=1,M(1)=0,M1(2)=0 M1(5)=1,M1(1)=0, M1(2)=0	M1(1)=1,M2(1)=0,M3(1)=0,M4(0)=0, M1(2)=1,M2(2)=0,M3(2)=0,M4(2)=0, M1(3)=1,M2(3)=0,M3(3)=0,M4(3)=0 M1(4)=1,M2(4)=0,M3(4)=0,M4(4)=0 M1 (5)=1,M2(8)=0,M3(5) =0,M4(8)=0
M_2	F_{11},T_{11}, F_{12},T_{12}, D_{11},B_{11}, D_{12},TH_{11}	M2(1)=1,M2(3)=0,M2(4)=1, M2(5)=0,M2(8)=0 M2(2)=1,M2(3)=0,M2(4)=1, M2(5)=0, M2(8)=0 M2(3)=1,M2(1)=0,M2(2)=1, M2(6)=0,M2(7)=0 M2(4)=1,M2(1)=0,M2(2)=1, M2(6)=0,M2(7)=0 M2(5)=1,M2(1)=0,M2(2)=1, M2(6)=0,M2(7)=0 M2(6)=1,M2(3)=0,M2(4)=1, M2(5)=0,M2(8)=0 M2(7)=1,M2(3)=0,M2(4)=1, M2(5)=0,M2(8)=0	M2(1)=1,M1(1)=0,M3(1)=1, M4(1)=0 M2(2)=1,M1(2)=0,M3(2)=1,M4(2)=0 M2(3)=1,M1(3)=0,M3(3)=1,M4(3)=0 M3(4)=1,M4(4)=0,M2(5)=1,M4(5)=0 M2(6)=1,M4(6)=0,M2(7)=1,M4(7)=0 M2(8)=1,M1(5)=0,M3(5)=1,M4(8)=0
M_3	F_{11},T_{11}, F_{12},T_{12}, TH_{11}	M3(1)=0,M3(3)=0,M3(4)=0, M3(5) =0 M3(2)=1,M3(3)=0,M3(4)=0, M3(5)=0 M3(3)=1,M3(1)=0,M3(2)=0 M3(4)=1,M3(1)=0,M3(2)=0 M3(5)=1,M3(1)=0,M3(2)=0	M3(1)=1,M1(1)=0,M2(1)=0, M4(1)=0 M3(2)=1,M1(2)=0,M2(2)=0,M4(2)=0 M3(3)=1,M1(3)=0,M2(3)=0,M4(3)=0 M3(4)=1,M1 (4)=0,M2(4)=0,M4(4)=0 M3(5)=1,M1(5)=0,M2(8)=0, M4(8)=0
M_4	F_{11},T_{11}, F_{12},T_{12}, D_{11},B_{11}, D_{12},TH_{11}	M4(1)=1,M4(3)=0,M4(4)=1, M4(5)=0,M2(8)=0 M4(2)=1,M4(3)=0,M4(4)=1, M4(5)=0,M4(8)=0 M4(3)=1,M4(1)=0,M4(2)=1, M4(6)=0,M4(7)=0 M4(4)=1,M4(1)=0,M4(2)=1, M4(6)=0,M4(7)=0 M4(5)=1,M4(1)=0,M4(2)=1, M4(6)=0,M4(7)=0 M4(6)=1,M4(3)=0,M4(4)=1, M4(5)=0, M4(8)=0 M4(7)=1,M4(3)=0,M4(4)=1, M4(5)=0,M4(8)=0 M4(8)=1,M4(1)=0,M4(2)=1, M4(6)=0,M4(7)=0	M4(1)=1,M1(1)=0,M2(4)=1, M3(5)=0, M4(8)=0 M4(2)=1,M1(2)=0,M2(4)=1,M3(5)=0 ,M4(8)=0 M4(3)=1,M1(3)=0 M2(2)=1,M3(6)=0, M4(7)=0 M4(4)=1,M1(4)=0 M2(2)=1,M3(6)=0, M4(7)=0 M4(5)=1,M2(5)=0 M4(6)=1,M2(6)=0 M4(7)=1,M2(7)=0 M4(8)=1,M1(5)=0,M2(8)=0, M3(5)=0

Combined Objective Function (COF):
Minimize COF= $(W_1)\times(X_p \div MPP)+(W_2)\times(X_q \div TE)$

where,
W_1= weight factor for customer satisfaction
X_P=total penalty cost incurred, and
$$X_p = \sum_i (CT_i - DD_i) \times UPC_i \times BS_i$$
i = job number
CT_i=completion time of job i.
DD_i = due date for job i
UPC_i= unit penalty cost for job i
BS_i= batch size of job i
MPP= maximum permissible penalty
W_2 = weight factor for machine utilization
X_q = total machine down time, and
$$X_q = \sum_j MD_j$$
j= machine number, and
$$MD_j = TE - \sum_i PT_{ji}$$
TE= total elapsed time
PT_{ji}= processing time of i^{th} job with j^{th} machine

In the experiment conducted equal weights are given W_1 = 0.5 and W_2 = 0.5. However different ratios can be applied to them according to the demand of business situation. The appropriate values of the GA parameters are arrived at, based on the satisfactory performance of trials conducted for this application with different ranges of values.

The crossover probability was varied from 0.4 to 0.9 and it was found to that the solution was improving faster for a crossover probability of 0.60.Similarly in the range from 0.001 to 0.010 the mutation probability of 0.005 was found to retain more better solutions than worse solutions

Population size (n) = 20 samples
Crossover probability (p_c) = 0.600
Mutation probability (p_m) = 0.005
Termination criteria = 100 generations (or) a satisfactory predefined minimum value for COF, whichever occurs first.

7 Results of the GA model

The proposed approach was coded in MATLAB. The algorithm ran with control parameters obtained from the graphical simulation in QUEST. The proper values of these parameters were determined in a pre-processing phase. The performance of the algorithm was tested over the FMS scheduling benchmarks generated by (Taillard, 1993). Due to the stochastic behavior of the algorithm, and the fact that it does not have a natural termination point, it was decided to run the algorithms for fixed time duration and report the best solution obtained after this time has elapsed. The generated solutions were quantified by the solution quality given in percentage offset from the best known solutions. The optimal schedules as obtained from the GA runoff programs for setup-1, setup-2, setup-3 respectively and results to be presented in the Table 9, Table 10 and Table 11. It is observed from the tables that the total machining time is the lowest in setup-2.This is due to the fact that the operations such as F_{11}, T_{11}, D_{11}, B_{11}, D_{12}, TH_{11} are more effectively carried out in the machines they are assigned to in setup-2.The optimization technique adopted in the present work will always produce the optimal value of the cost function as it is based on the principle of survival of fittest. Any type of layout can be implemented due to the fact that the rule base takes care of the operation sequence. The GA optimization technique depends on the size operational requirement and complexity of the part. The optimized machining times for the three setups as obtained from the GA model.

8 Conclusion

Throughout the previous works Numerous GA approaches to production scheduling are reported by large no. of authors. The approaches differ strongly from each other with respect to the coding, encoding, operations used, the constraints handled and the goals pursued. Despite these differences all approaches have in common that the domain knowledge is required in order to produce competitive schedules. The present approach is aimed towards finding out the global optima in the search space with some restrictions. The results obtained here can be claimed to be the optimal one. The potential of GA for minimizing the makespan in an FMS was explored in this paper. In conclusion, the GA with features from both global and local search techniques, results to a robust optimization tool capable of producing high quality solutions for the FMS scheduling. Future work will examine the performance of the hybrid Simulated Annealing Algorithm on other harder scheduling problems such as the job shop and the open shop scheduling problems. This is a relatively unexplored area of research based on a simple principle: the systematic change of neighborhood within the search.

Moreover, the case of multi-objective optimization will be investigated. Particularly, the research will be focused on scheduling optimization with the aim of simultaneously minimizing objectives like makespan, total flow time, total tardiness, machine utilization, idle time, sum of set-up times, etc. The appropriate combination of these criteria into a single objective function is a difficult task and will constitute a significant subject of future research.

Table 9 Operation assignment and processing time at machines in setup-1

Machines	**M_1**	**M_2**	**M_3**	**M_4**
Operation assigned	F_{12}, T_{12}	B_{11}, D_{12}	F_{11}, T_{11}	D_{11}, TH_{11}
Machining time	42	170	95	180
% of Machine Utilization	8.6	34.9	19.5	37
Total Machining Time:487				

Table 10 Operation assignment and processing time at machines in setup-2

Machines	$\mathbf{M_1}$	$\mathbf{M_2}$	$\mathbf{M_3}$	$\mathbf{M_4}$
Operation assigned	F_{11}, T_{11}	D_{11}, TH_{11}	F_{12}, T_{12}	B_{11}, D_{12}
Machining time	80	155	53	157
% of Machine Utilization	18	34.8	12	35.2
Total Machining Time:445				

Table 11 Operation assignment and processing time at machines in setup-3

Machines	$\mathbf{M_1}$	$\mathbf{M_2}$	$\mathbf{M_3}$	$\mathbf{M_4}$
Operation assigned	F_{11},T_{11}	F_{12},D_{11},TH_{11}	T_{12}	B_{11},D_{12}
Machining time	80	175	45	157
% of Machine Utilization	17.5	38.3	9.8	34.4
Total Machining Time:457				

9 References

Buyurgan, N. Saygin,C. and Kilic,S.E. (2004) Tool Allocation in Flexible Manufacturing Systems with Tool Alternatives, Robotics and Computer Integrated Manufacturing, **20**(4) , pp. 341–349

Bierwirth ,C. and Mattfeld, Dirk C.(1999) Production Scheduling and Rescheduling with Genetic Algorithms, Evolutionary Computation ,**7**(1), pp.1-17

Forrest,S. (1993) Genetic Algorithms: Principles of Natural Selection Applied to Computation, Science, **261**(5123),pp.872-878

Ishibuchi, H. Murata, T. and Tanaka, H.(1994) Genetic Algorithms and neighborhood algorithms for fuzzy flowshop scheduling problems, Fuzzy sets and systems, **67**(1),pp.81-100

Lee,Y.H. Jeong, C.S. and Moon, C. (2002) Advanced Planning and Scheduling with Outsourcing in Manufacturing Supply Chain', Computers and Industrial Engineering **43** (1–2), pp. 351–374

Lee,C.S, Kim, S.S. and Choi, J.S.(2003) Operation Sequence and Tool Selection in Flexible Manufacturing Systems under Dynamic Tool Allocation, Computers in Industry **45**(1),(2003) pp. 61–73

Moon,C, Kim, J.S. and Gen, M. (2004) Advanced Planning and Scheduling based on Precedence and Resource Constraints for E-plant Chains, International Journal of Production Research **42** (15),pp. 2941–2954

Sun, J. and Xue, D.(2001) A Dynamic Reactive Scheduling Mechanism for Responding to Changes of Production Orders and Manufacturing Resources, Computers in Industry **46** (2),pp. 189–207

Paliouras, G. Karkaletsis, V. and Spyropoulos ,C. D.(eds.)(2001) ‘Genetic Algorithms in Machine Learning and Its Applications’, Artificial Intelligence, volume LNAI 2049, pages 146 – 169, Springer.

Taillard, E.(1993)Benchmarks for Basic Scheduling Problems. European journal of operational Research ,**64**(1993),pp.278-285

Vieira, G.E. Herrmann, J.W. and Lin, E. (2003) Rescheduling manufacturing systems: A framework of Strategies, Policies, and Methods, Journal of Scheduling, **6** (1), pp. 39–62

Whitley, D.(2001) An Overview of Evolutionary Algorithms, *Journal of Information and Software Technology* ,**43**,pp.817-831