IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE)
ISSN: 2278-1684 Volume 5, Issue 1 (Jan. - Feb. 2013), PP 01-17
www.iosrjournals.org

"Scheduling in Job Shop Process Industry"

Shantanu Kolharkar[1], Prof. D. R. Zanwar[2]
[1,2](Industrial Engineering, S.R.C.O.E.M. Nagpur, India)

Abstract: *In today's manufacturing scenario Production Scheduling plays a vital role in planning. The scheduling deals directly with the time. The allocation of available resources over time to perform a specific task. Assigning an appropriate number of workers to the jobs during each day, determining when an activity starts and end depending upon the duration of an activity, predecceor activity, resource availability and target completion date. A scheduling problem is to find sequences of jobs on specific machines with an objective to minimize total elapsed time or makespan. Job Shop Scheduling is type where the machine order can be different for each jobs. It becomes complex and agile to evolve optimal solution. JSP is widely defined as a NP hard and combinatorial in nature. In this paper the characteristics of JSP is analyzed and proposed Genetic Algorithm as one of the efficient optimization tool to solve problem like JSP. This research work aim to analyze the JSP at Job shop Process Industry considering static scheduling and also considered due dates as dynamic scheduling. Coding is done in MATLAB Version 7.10 through GUI. Compared the results with and without using GA parameters.*
Key Words*: JSSP (Job shop scheduling problem), Genetic Algorithm, Crossover, Mutation, MATLAB, GUI (Graphical User Interface),*

I. Introduction:

1.1 Nature of Problem:-

Planning and scheduling are distinctly different activities. The plan defines what must be done and restrictions on how to do it; the schedule specifies both how and when it will be done. The plan or schedule refers to the estimates of time and resource for each activity, as well as the precedence relationships between activities and other constraints. The schedule refers to the temporal assignments of tasks and activities required for actual execution of the plan. In addition, any project includes a set of objectives used to measure the performance of the schedule and/or the feasibility of the plan. The objectives determine the overall performance of the plan and schedule. The classical job shop scheduling problem (JSP) involves scheduling '*n*' jobs on "*m*" different machines. Each job is composed of a set of operations and the order of operations on machines for each job is pre-specified, usually by processing and technology constraints. The complexity of this problem is confirmed by the fact that Job shop scheduling falls under NP hard problems. NP hard problems cannot be effectively solved by deterministic methods therefore the most used approach involves heuristics.

- **Job**: A piece of work that goes through series of operations.
- **Shop**: A place for manufacturing or repairing of goods or machinery.
- **Scheduling**: Decision process aiming to deduce the order of processing.

Job Shop Scheduling Problem means jobs to be processed on shop floor within specific time. JSSP is a very tough to solve. It is a very complex problem to solve.JSP is a working area where n jobs to be processed on m Set of Machines with many tasks. In JSP the machine order can be different for each job so it becomes complex and agile to get the optimal solution. With n jobs to be processed on m Machines the number of possible sequences (n!) to the power m. If 5 jobs to be processed on the 2 machines the number of possible sequences are 14400 means when the size of the problem grows up, the time for determining the optimal solution also increases exponentially.JSP follows the concept of forward scheduling when customer needs their orders on as soon as possible basis. Forward scheduling determines start and finish time for next priority jobs by assigning it the earliest available time a lot and from that time determines when the job will be finished on the work station. In this paper Gantt charts are used to represent the total elapsed time or makespan. There are several approximation algorithm have been developed. However the result obtained by Branch and bound methods are not reliable and consumes lot of time. In hill climbing method search procedure stops once it detects no improvement in the next generation or iteration. One of the widely used techniques in industry is local search; GA is one of the local search techniques.

Purpose:-

In the current Manufacturing scenario scheduling of job shop environment is directly deals with the production efficiency. Companies are always focusing to improve the productivity by applying the various new alternatives and techniques. Scheduling is one of the main functions of production planning and control department. There are two types of Scheduling environment, Flow shop and Job shop environment. In Flow shop environment all the jobs are passes through all the machines in same order and in job shop environment the machine order can be different for each job. So because of the complexity and dynamic nature of the JSP it becomes NP-hard problem .Purpose of this thesis is to analyse the characteristics of Job shop scheduling problem considering the constraints and priority of jobs by using Genetic algorithm through MATLAB. In this we implemented GUI as a user interface tool for easy operation. Penalties are provided to control the constraints violation. The objective of this is to minimize the Makespan by using Priority Rule as a fitness function.

Contribution:-

This paper focuses on solving a specific scheduling problem for job shop in industrial environment using Genetic Algorithm. The schedule is planned for Job shop activities in the Industries. This paper provided results in the form of Job sequence, start and finish time for each activity considering the due dates and batch size.

There are 10 jobs, 10 machines 10 different operations of each job and each jobs having different machine sequence along with batch size and due dates. GUI is used to represent the data and edit the data as per the working environment. The input parameters are No. of jobs, No. of Machines, no. of operations pertaining to each jobs ,Batch size, due dates as a completion dates. Output parameters are Job sequence having minimum makespan value, Start and finish time of each operation on each machines considering the technological constraints.

Addition to that computational analysis for JSP with and without using GA parameters is provided in the programme.

In this paper we generate an initial population randomly including the result obtained by priority rule using GA and will go through crossover and mutation to find near optimal solution. We represent the model by using GUI in MATLAB.

II. Literature Review

This chapter will review the relevant literature and background on nurse scheduling problems. A number of researchers have used genetic algorithms to solve the staff-scheduling problem.

Yeli and Yanchon introduced new approach to solve JSP problem. Tow row chromosome structure is adopted based on working procedure and machine distribution. They used job operation and operation machines two row chromosomes to solve agile problem as JSP [1]

Mhanim Adam they used GA to optimize makespan. In this they used two types of initial population to the data. First used the combination of schedules generated using priority dispatching rules and randomly generated schedules as the initial population and find near optimal solution [2]

Kanate Ploydanai and Anan Mungwattana proposed new algorithm for solving JSP by proposingmathematical model by considering machine availability constraints.[3]

A new approach is proposed by Ramezanali Mahdavinejad to minimize makespan by using Ant Colony optimization algorithm using Priority rule.[4]

E. Falkenauer and S. Bouffouix used GA to optimize the cost function to solve JSP problem with many task machines and precedence constraints.[5]

Milos Seda in this paper represents the mathematical models for permutation flow shop and job shop problem are proposed. The first problem based on mixed integer programming model for job shop problem. Mathematical Model and its main representation schemes are presented. In this paper disjunctive graphis used to represents feasible schedule.[6]

Takashi Yamada and Ryochi Nakaano made first attempt rather serious attempt to solve the JSP problemwith an application of GA used bit string representationand conventional operators.used GT algorithmas a basic schedule builder was proposed by Yamada and Nakano and Dondraf and Pesch independently [7].

Shignebu Kobayasch, IsaoOno, masayuki Yamamura.In this Job sequence matrix is constructed and introduces characteristics preserving crossover named the subsequences exchange crossover (SXX). The SXX exchanges subsequences in parents on each machine. When they consist of the same set of Jobs. Schedules generated by SXX are modified by the Giffer Thomson method to transform into active schedule. [8]

M.A.Adibi, M, Zandieh, and M.Amiri introduces Multi objective scheduling of dynamic job shop using variable neighborhood search. In this they used Artificial Neural Network to solve Dynamic JSP problem considering random job arrivals and machine breakdown. Also a Multi objective performance measure is applied as objective function that consists of makespan and tardiness. The proposed method is compared with some common dispatching rules that have widely used in the literature for dynamic job shop scheduling problem.[9].

Shyh-Chang Lin ,Erik D.Goodman and William F. punch,III describes the GA for the dynamic job shop scheduling problem with jobs arriving continually in the shop floor. Both deterministic and stochastic models of the dynamic problem were investigated. The objective function were weighted flow time, maximum tardiness, weighted tardiness, weighted lateness,weigted number of tardy jobs,and weighted earliness plus weighted tardiness.alos they have consider machine workload ,imbalance of machine workload,and due date tightness [10]

III. The Job Shop Scheduling Problem (JSP)

3.1 Introduction to Job-Shop Scheduling

The typical job-shop problem is formulated as a work order that consists of set of n jobs, each of which contains m tasks. Each task has a single predecessor and requires a certain type of resource. Often many resources of a specific type are available, for example five lathe machines and two Milling. Many tasks can be assigned to any one of the available resources, but the resource must be of the right type. Typical objectives include minimizing the makespan for the work order or meeting due dates for specific jobs or tasks in Job shop scheduling problem. The process plan for a single job is typically serial since each job is often associated with a single part. Each task typically requires a single resource. However, more complicated relationships are possible. The order in which jobs are executed is often unimportant in terms of the jobs themselves, but very important in terms of the resources used to do them.

Many variations are possible for either the job-shop scheduling problems. Some of these variations include job-splitting, task pre-emption, multiple execution modes, non-uniform resource availability and usage, and various resource types.

Scaling Issues - The Size of the Problem

The size of a scheduling problem can be approximated by a what-where-when matrix. Scheduling problems consist of asking what must be done where and when. Resources (where) operate on tasks (what) for specific periods of time (when). Using this simple classification, and neglecting precedence and other constraints, a rough approximation of a problem's size can be given by the product of what, where and when for the problem. How many tasks must be completed, by how many resources, over what time intervals?

Many methods have been designed for determining whether parts of a schedule can be feasible given partial knowledge about that schedule, or whether one part of a decision tree can be any better than another part. These methods attempt to reduce the size of the search by taking advantage of problem-specific information. Nevertheless, pruning heuristics are not always available, and rarely are they obvious. The choice of representation also controls the size of the search space. If one chooses a very general representation, more types of problems may be solved at the expense of searching a larger space. Conversely, one may choose a very specific representation that significantly reduces the size of the search, but will work on only a single problem instance.

Uncertainty and the Dynamic Nature of Real Problems

Practically speaking, finding an optimal schedule is often less important than coping with uncertainties during planning and unpredictable disturbances during schedule execution. In some cases, plans are based upon well known processes in which resource behaviours and task requirements are all well known and can be accurately predicted. In many other cases, however, predictions are less accurate due to lack of data or predictive models. In these cases the schedule may be subject to major changes as the plan upon which it is based changes. In either case, unanticipated disturbances to the schedule may occur. Whether a mechanical failure, human error, or inclement weather, disturbances are inevitable. Such disturbances may require only the replacement of a single resource, or they may require complete reformulation of the plan. Any optimization technique should be able to adapt to changes in the problem formulation while maintaining the context of work already completed.

Infeasibility - Sparseness of the Solution Space

Depending on the representation and the modelling assumptions, there may be no feasible solution to a scheduling problem. For example, if all resources are available in constant per period amounts and there are no temporal restrictions on tasks or resources.

Constraints make the search for an optimal solution more difficult by breaking up an otherwise continuous search space. When many constraints are added, traversal of the search space is confounded. In addition, adding constraints typically reduces the number of feasible solutions for a given representation.

Researchers have been focusing on investigating machine scheduling problems in manufacturing and service environments where jobs represent activities and machines represent resources, and each machine can process one job at a time. In this thesis, we will focus on the low volume system also known as job-shop. In this type of environment, products are made to order. The time required to complete the process of all the operation is called as Total processing time or makespan. There are some assumptions considering the JSP problem.

- Each job consists of a finite number of operations.

- The processing time for each operation has been determined.
- Machine Sequence is defined for each job.
- There is a pre-defined sequence of operations that has to be maintained to complete each job.
- Job does not visit the same machine twice.
- Set up time included in the Processing time.
- A machine can process only one job at a time.
- No machine can deal with more than one type of task.
- Operations cannot be interrupted.
- Neither release time nor due dates are specified.
- Each job should be processed through the machines in a particular order or also known as technological Constraints..

In this work we also considered the environment of dynamic job shop scheduling in which the job arrival is random and according to the priority specifies by due dates .We prepare coding and analyze and given new dimensions to the static JSP problem.

We generate initial population randomly including the result obtained by priority rule and will go through crossover and mutation to find near optimal solution. Result will show Job sequence, start and finish time of each operation on each machine considering technological constraints and will give minimum makespan to complete all the set of jobs. We represent the model by using GUI in MATLAB.

IV. Genetic Algorithm Methodology

4.1. Background of Genetic Algorithm

Charles Darwinian evolution in 1859 is intrinsically a so bust search and optimization mechanism. Darwin’s principle “Survival of the fittest” captured the popular imagination. This principle can be used as a starting point in introducing evolutionary computation. Evolutionary computation, describes the field of investigation that concerns all evolutionary algorithms and offers practical advantages to several optimization problems. The advantages include the simplicity of the approach, its robust response to changing circumstances, and its flexibility and so on.GA were formally introduces in the US in the 1970 by John Holland at University of Michigan. This is one of the efficient optimization tool, work very well on mixed (Continuous and discrete) combinatorial problems.

To implement GA it is must to represent a solution to the problem as a genome (Chromosome). The GA then creates populations of solution and applies GA operators such as crossover and mutation to evolve optimal solution. There are three most important aspects of GA: Objective function is to be defined, Genetic representation is defined and implemented and implementation of GA operators.GA differs from conventional heuristic optimization techniques.GA works with the coding, GA uses population of solutions rather than the single solution for searching GA uses fitness function to evaluate rather than derivatives.

The evolutionary process of a GA is a highly simplified and stylized simulation of the biological version. It starts from a population of individuals randomly generated according to some probability distribution, usually uniform and updates this population in steps called generations. Each generation, multiple individuals are randomly selected from the current population based upon some application of fitness, bred using crossover, and modified through mutation to form a new population.

- **Crossover** – exchange of genetic material (substrings) denoting rules, structural components, features of a machine learning, search, or optimization problem .
- **Selection** – the application of the fitness criterion to choose which individuals from a population will go on to reproduce.
- **Replication** – the propagation of individuals from one generation to the next.
- **Mutation** – the modification of chromosomes for single individuals.

Strengths of Genetic Algorithm

1. Genetic algorithms are intrinsically parallel. They can explore the solution space in multiple directions at once. So convergence to an optimal solution does not depend on the chosen initial solution.
2. They are well-suited to solving problems having huge search space.
3. They perform well in problems for which the fitness landscape is complex **-** where the function is Discontinuous, noisy, changes over time, or has many local optima.

Parameters used while encoding for GA

For solving specific type of problems in genetic algorithm encoding is required. Different parameters are needed for the purpose of encoding; some parameters are listed below in the table used for solving JSP,
Elite Count, Number of Initial Population, The length of chromosome, Crossover rate, mutation rate, Maximum number of generation and stall generation.

Genetic Operators

The genetic operators in MATLAB Global Optimization toolbox are used to test this problem and each operator is listed as below:

- **Selection:** Tournament selection is used to select the best individual.
- **Elite Count:** The elitism operator is set to select 1 individual (the best one) for the next generation.
- **Crossover:** Two-point crossover method was used as crossover function.
- **Mutation:** The method which is adaptive feasible was used as mutation function, and this method is used for the problem this contains lower or upper bounds or non linear constraint.
- **Hybrid Function:** Pattern search is used as a hybrid function to increase the efficiency of selection with constraints.

Data Structures and its Representation

- ❖ **Initial Population:** Most of the researchers prefer to generate the initial population randomly but this will consumes lot of time to get an optimal solution. In this we start the initial population with a randomly generated preference matrix, including values involved for generating the nurse schedule.
- ❖ **Parent Selection**: We choose using the MATLAB default value for random selection of parents.
- ❖ **Fitness value**: We use fitness function for finding near optimal solution.
- ➢ Initialize population was generated randomly among the preference matrices, for generating feasible schedule.
- ➢ The fitness was defined by objective of JSP model, and individual adaptive value was evaluated.
- ➢ The crossover was operated in the population according to probability of crossover Pc, so the offspring are generated.
- ➢ The individual was selected randomly according to probability of mutation Pm, so that the offspring are generated.
- ➢ The new individual adaptive value was calculated, parent and offspring are taken part in survival competition together.

Adjusting the termination criterion, then the optimal solution was obtained, otherwise going back to first step.

Genetic Algorithm flow chart:

[
Generate initial population
Evaluate population
While stopping criteria not satisfied **Repeat**
[
Select elements from ***P***t to copy into ***P***t+l
Crossover elements of ***P***t and put into ***P***t+l
Mutation elements of ***P***t and put into ***P***t+l
Evaluate new population ***P***t+l
Pt = ***P***t+l
]
]

V. Work Domain and Approach

5.1 Conceptual Framework:

Normally each industry is facing problem to find the optimal schedules. JSP is such problem it is observed in all type of manufacturing industry where job work is carried out. One of the leading manufacturers of Industrial Hose having end fitting section.

End Fitting is made up of three basic Components, Socket or Ferrule, Nipple or Insert & Nut. Socket / Ferrule is a metallic sleeve which provides the termination between the hose and insert. Nipple / Insert are a component for the connection of Hose assembly with the equipment.

There are three series of End Fittings:

- 1. Standard
- 2. Interlock
- 3. Reusable

It is observed that there are N no. of Variety of Jobs to be manufactured
Each Job having different Machine Sequence, Technological Constraints
Unpredictable Job Orders.
End Fitting Family consist of 6 different jobs Socket, Female nipple, Male Nipple, Brazed Type Nipple, Adapter and Banjo.

5.2 Data structure and representation of JSP

GA process the population of strings. In this paper we construct a population as an array of individual where each individual member contains the artificial chromosome. Artificial chromosome are phenotype and genotype. The objective function (Fitness function) i.e. Minimize Makespan. In Job shop scheduling problem we consider the job order on each machine is phenotype, and the machine schedule as genotype and makespan value is fitness.The representation used in this is permutation representation (Yamada and Nakano,1997) JSSP can be viewed as an ordering problem just like the Travelling Salesman Problem (TSP). A schedule can be represented by the set of permutations of jobs on each machine which are called *job sequence matrix*. Figure 1 shows the *job sequence matrix* for 3× 3 problem. Rows will represent the machines and columns represent the order of jobs. N jobs to be processed on M machines with set of task to be performed and machine sequence for each job is different.

Machining Sequence

Jobs	**Operations**		
J1	M1	M3	M2
J2	M1	M2	M3
J3	M2	M3	M1

Table: 1

Processing time

Jobs	**Operations**		
J1	P11	P12	P13
J2	P21	P22	P23
J3	P31	P32	P33

Table: 2

Table:1 Job sequence Matrix. We considered 3x3 matrix for the for generalization of JSP as a testing parameters. There are three jobs each having three different operations to be performed on three machines. As operations are different the sequence of machines is set accordingly.Table:2 represents the corresponding processing time for each operation on each machine.

5.3 Approach for Solving JSP in working environment:

The solution methods form two distinct classes: exact methods and heuristic methods. These classes may be categorized further into stochastic and deterministic approaches. Exact methods are guaranteed to find a solution if it exists, and typically provide some indication if no solution can be found. Heuristic solutions may have no such guarantee, but typically assure analytically some degree of optimality in their solutions.

Stochastic methods include probabilistic operations so that they may never operate the same way twice on a given problem (but two different runs may result in the same solution). Deterministic methods operate the same way each time for a given problem. Many hybrid methods exist that combine the characteristics of these classes.

Considering the above framework we have construct a 10x10 matrix for formulation of JSP. We use Graphical User Interface in MATLAB R2010a Version7.10.As JSP is a complex problem to solve it by manual method and it take lot of time. In industry time is money. Considering 10X10 matrix illustrate that 10 jobs to be processed on 10 dedicated machines with 10 different operations and for each operation their processing time and machine sequence for each job is determined. Also we have given batch size for each jobs.

In this paper we generated initial population randomly including the results obtained by the Priority dispatching rule as SPT.

5.3.1 Objective Function:

The genome performance measure, often referred to as the *objective function*, consists of two parts, each based upon the schedule the genome represents. The first part is a measure of constraint satisfaction, the second part is based on the schedule performance with respect to the objectives. Since the genome directly represents a schedule, calculation of both measures is straightforward. Some typical constraint and objective measures are outlined in this section, followed by an explanation of how the constraint and objective measures were combined to produce the overall score for each genome.

5.3.2 Constraints:

Most measurements of constraint satisfaction were based upon *resource profiles*. Resource profiles define resource availability or consumption as a function of time.

5.3.2.1 Resource Availability:

Part of the planning stage is the definition of resource availability. For each resource, a profile of availability can be generated to indicate when and how much of that resource will be available. Sample resource availability Note that this representation encompasses both resource quantity and temporal restrictions on resource usage.

(a) Time amount of resource available
(b) Time amount of resource required
(c) Time difference between available and required

The feasibility of a given schedule with respect to a given resource is calculated by comparing the resource availability curve to the resource requirement curve for the resource in question.
This measure of feasibility results in a graduated score that reflects not only whether or not a solution is feasible, but also the degree of infeasibility if it is infeasible.

5.3.2.2 Temporal Constraints

If a task *must* be started at a specific time, then the corresponding start time in the genome is adjusted by the genetic operators so that the task always starts at that time. If a resource is available only at certain times or for a certain duration, this is reflected in the construction of the availability profile for that resource.

5.3.2.3 Precedence Feasibility

Precedence feasibility is enforced by the representation and genetic operators, so precedence infeasible solutions are not possible.

5.4 Objectives

Many different measures of schedule performance exist.
The next three sections highlight some of the more common performance measures.

5.4.1 Due Dates and Tardiness

The performance of many projects is measured in terms of due dates or deviation from projected finish times. These measures are calculated directly from the schedule. For example, if a work order specifies that 80% of the jobs must be completed by their specified finish times, the performance measure can be calculated directly. If each job has a due date, xi, specified in the plan and finish time, fi, determined from the schedule, the tardiness is the difference $d = fi - xi$
where d is truncated to zero (early jobs are not tardy). The mean tardiness for a work order is simply the average of the tardiness scores of each job.

5.4.2 Cost

The total cost of a schedule can be found by adding the individual costs of each activity given the execution mode and resources applied to it. Since the schedule is explicitly defined, any genome can be used to calculate a net-present value or virtually any other cost measurement of performance. If each task has a cost, ci, determined from the scheduled modes, then the total
cost is simply the sum of the costs of each task.

5.4.3 Makespan

The length of time required to complete a schedule is calculated directly from the information in the genome. The makespan is simply the finish time of the last task. Note that a schedule may indicate a makespan when, in fact, that schedule is infeasible due to violations of resource constraints.Objective function is minimize make span or Total Processing Time and also second objective is to Maximize tardiness.

Table 3 . JOB Sequence Matrix with batch size

Jobs	**Machine Sequence**										**Processing Times**										**Batche**
J1	M	M	M	M	M	M	M	M	M	M	1	2	1	8	1	1	4	6	1	1	125
J2	M	M	M	M	M	M	M	M	M	M	1	2	9	7	6	1	1	1	2	2	800
J3	M	M	M	M	M	M	M	M	M	M	8	1	2	5	8	1	1	1	1	1	130
J4	M	M	M	M	M	M	M	M	M	M	1	2	2	1	5	1	1	1	1	2	760
J5	M	M	M	M	M	M	M	M	M	M	9	2	1	1	8	1	7	4	1	1	144
J6	M	M	M	M	M	M	M	M	M	M	1	1	1	1	2	2	2	1	1	1	150
J7	M	M	M	M	M	M	M	M	M	M	7	1	1	2	2	2	2	1	1	1	960
J8	M	M	M	M	M	M	M	M	M	M	1	2	2	2	2	2	3	2	1	1	580
J9	M	M	M	M	M	M	M	M	M	M	2	1	2	2	2	1	2	2	1	1	105
J10	M	M	M	M	M	M	M	M	M	M	1	1	2	1	2	2	2	2	1	9	110

In this research we consider 10x10 matrix along with Batch size for each job. We generate initial population randomly including the result obtained by priority rule and will go through crossover and mutation to find near optimal solution. Result will show Job sequence, start and finish time of each operation on each machine considering technological constraints and will give minimum makespan to complete all the set of jobs. By using GA Operators randomly generate the possible sequences of jobs and evaluate it with the objective function to get the near optimal makespan value and corresponding job sequence and corresponding makespan or total processing time.

Also we have considered the due dates and compare it with the fitness function separately to optimize the total span for completing the each job. In this we have take summation of difference of due date and completion time as a fitness function for priority. It will indicate which sequence of job is having highest priority will give the priority to process the same.

In Scheduling GA represents schedules as individual or a population's of individual. Each member has its own fitness value which is evaluate and measured by the objective function. This procedure works by iteration or generation, and this generation consist of an individual who survive from the previous generation. Here the population size remains constant for next generation.

Gantt Charts are used to represent the solution to the JSP that is to specify the total processing time or completion time for all operations.

Parameters:

No. of Jobs
No. of Machines
No. of Operations
Processing time of each operation of each job on each Machine
Start Time of each operation of each job on each Machine
Finish Time of each operation of each job on each Machine
Total Processing time or Make span
Sequence of operation of jobs depend upon the technological constraints
Job sequence matrix
Batch size
Priority Time for each Job i.e. Due Time
Completion Time for each Job.
Tardiness
Lateness and Earliness

Mannual Method will take lot of time to find the optimal solution by any priority rule. So we have generate the initial population considering the priority rule as SPT and evaluate with the fitness function to find the optimal solution.

5.5 Input and output parameters

Step 1: Input parameters

1.1 No. Of Jobs
1.2 No.Of Machines
1.3 No.of Operations
1.4 Machine Sequence for each jobs

1.5 Processing Time for each operation of each job on each machine.
1.6 Batch Size
1.7 Completion Time of each job
1.8 Due time for each job as per priority

Step 2: Output Parameters.
2.1 Near optimal schedule (Makespan Value)
2.2 Optimal Job sequence
2.3 Tardiness (Earliness and Lateness)

5.6 GA Parameters used while encoding

Lower Bound	
Upper Bound	
Penalty	
Elite Count	1
Number of Initial Population	20
The length of chromosome	100
Crossover rate	0.8
Mutation rate	0.002
Max. Number of generation	500
Stall Generation	100

Lower Bound: It is a lower limit restricted to the random search from population
Upper Bound: It is a Upper limit restricted to the random search from population.
Penalty: Penalties are given to avoid violation in the string i.e. the main objective behind giving penalty is to provide Unique length of the string (1 to 10), the sum of the length which is generated randomly should be as make span and third one is for sum of length should be minimum value. If all this value is violates then penalty will discard the unwanted value and give the unique length as (1 2 3 4 5 6 7 8 9 10)
Elite Count: The elitism operator is set to select 1 individual (the best one) for the next generation.
Crossover: Two-point crossover method was used as crossover function.
Mutation: The method which is adaptive feasible was used as mutation function, and this method is used for the problem which contains lower or upper bounds or non-linear constraint.
Hybrid Function: Pattern search is used as a hybrid function to increase the Efficiency of selection with constraints.

5.9 Design and Modelling:

In working environment not only workers but also executives have no time to spend doing mathematical calculations to get a solution. Here we have used GUI base programming in which the two aspect are taken care one is presentation i.e. layout and accordingly the programming is done.
GUI Graphical User Interface: Provides a set of tools for creating GUI. Process simplifies designing and building GUI's

The main objective of applying GUI in our programming that the values of input parameters can be changed accordingly the requirements and the scope of the problem.
It is flexible to edit the Machine sequence for each job, processing time for each operation on each machine; batch size can be changed as per the conditions.
GUIDE Tools are:
Layout of GUI
Programme the GUI

5.9.1 GUI Layout

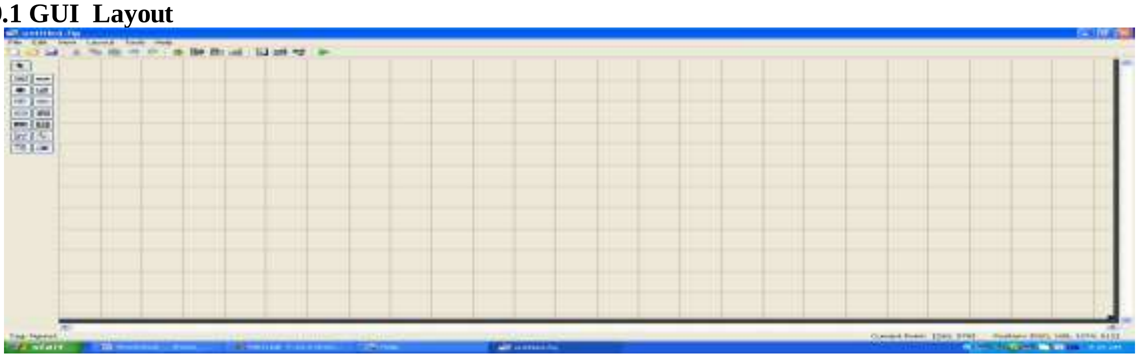

Fig.5.9.1 GUI layout

GUIDE, the MATLAB graphical user interface development environment, provides a set of tools for creating graphical user interfaces (GUIs). These tools simplify the process of laying out and programming GUIs.
Using the GUIDE Layout Editor, you can populate a GUI by clicking and dragging GUI components—such as axes, panels, buttons, text fields, sliders, and so on—into the layout area. You also can create menus and context menus for the GUI. From the Layout Editor, you can size the GUI, modify component look and feel, align components, set tab order, view a hierarchical list of the component objects, and set GUI options.

5.9.2 Designing of GUI for JSP:

We prepare GUI as per the requirement and demand of the problem considering the Input and output parameters.

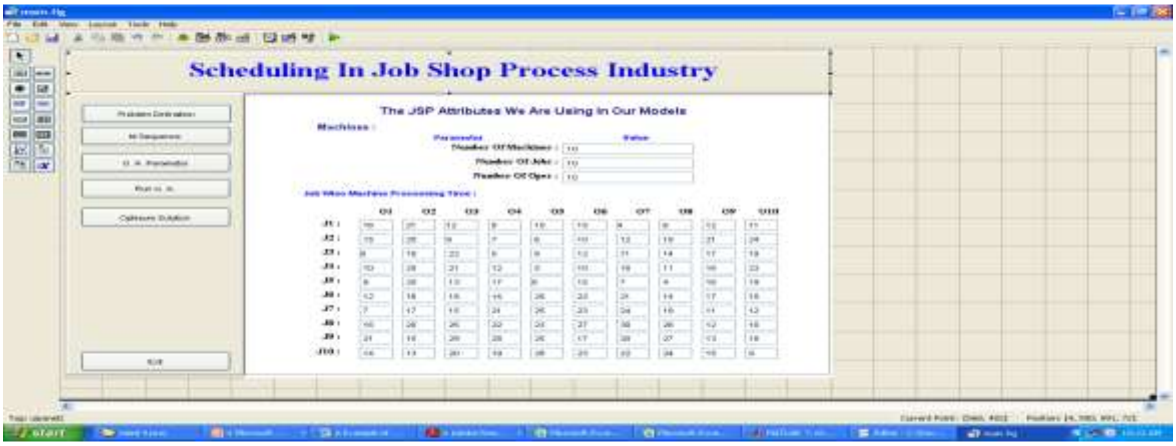

Fig 5.9.2 : Design of GUI for Input parameters of JSP

Figure 5.9.3 : Design of GUI for Input parameters of JSP(Machine Sequence)

5.9.4 GUI Programming

GUIDE automatically generates a program file containing MATLAB functions that controls how the GUI operates. This code file provides code to initialize the GUI and contains a framework for the GUI callbacks—the routines that execute when a user interacts with a GUI component. Use the MATLAB Editor to add code to the callbacks to perform the actions you want the GUI to perform.
Step1: Initially we design the GUI considering the Input and output parameters.
Step2: Create callback function for each parameter in m.file GUIDE initially sets the value of the most commonly used callback properties for each component to %automatic. For example, a push button has five callback properties, ButtonDownFcn, Callback, CreateFcn, DeleteFcn, and KeyPressFcn. GUIDE sets only the Callback property, the most commonly used callback, to %automatic. You can use the Property Inspector to set the other callback properties to %automatic. To do so, click the pencil-and-paper icon next to the callback name. GUIDE immediately replaces %automatic with a MATLAB expression that is the GUI calling sequence for the callback. Within the calling sequence, it constructs the callback name, for example, the subfunction name, from the component Tag property and the name of the callback property.

The following figure shows properties of a push button in the GUIDE Property Inspector prior to saving the GUI. GUIDE set the Tag property to pushbutton1. Before saving the GUI, Callback property displays as %automatic, indicating that GUIDE will generate a name for it when you save the GUI.
Step 3: given string name and Tag for each button in the Inspector

About the Property Inspector

In GUIDE, as in MATLAB generally, you can see and set most components' properties using the Property Inspector. To open it from the GUIDE Layout Editor, do any of the following:
Select the component you want to inspect, or double-click it to open the Property Inspector and bring it to the foreground

- Select Property Inspector from the View menu
- Click the Property Inspector button

The Property Inspector window opens, displaying the properties of the selected component. For example, here is a view of a push button's properties.

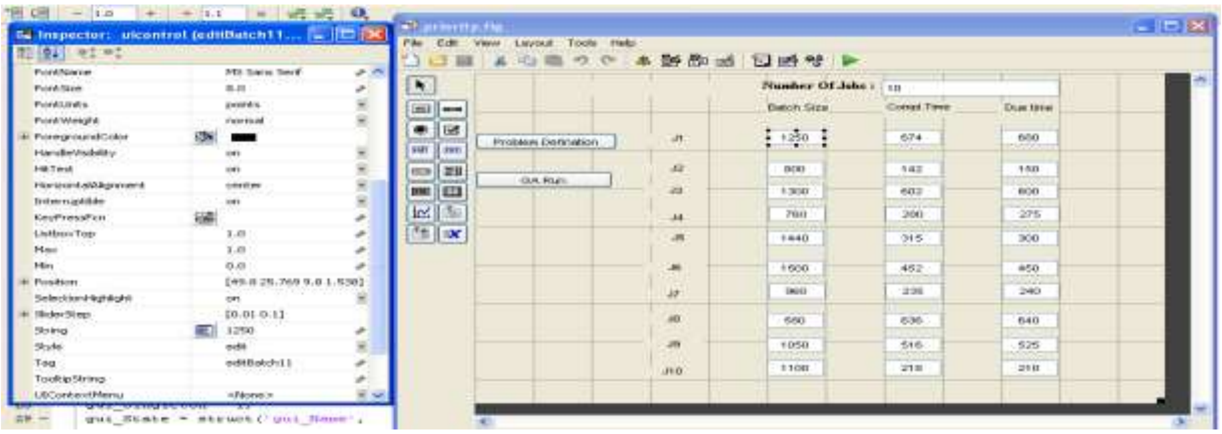

Fig 5.9.4 : Shows Property Inspector

Step 4: Using GA Parameters coding is done

5.10 Programming in Matlab.

Stepwise encoding of JSP in Matlab

- Initialise the Input parameters, No. of Jobs, No. Operations,no.of Machines,Batch Size Due Time
- Initialize Number of Variables defined by the size of Matrix
- Initialize Processing time of each operation on each machine
- Best solution is from the Matrix size
- Generate Population with population size is 100
- The length of the chromosome is 100
- LB = ones(1,u*v); % Lower bound
- UB = ones(1,u*v)*m; % Upper bound
- options.Display= 'iter';
- options.CrossoverFraction=0.8;%CrossRate;
- options.MigrationFraction=0.0002;%MutateRate;
- options.Generations=500;%MaxGane;
- options.StallGenLimit= 100;
- options.TimeLimit=Inf;
- options.Elitecount=1;
- Create Plot Function to plot the graphs of individual fitness value
- Create Fitness function
- Relate it with due time, batch size and other variables.
- Created Gantt chart
- For the best solution created function as start and finish time of each sequence with min makespan value.

VI. Test Problems And Results

6.1 Job shop scheduling problem considering 3X3 Matrix

- The study on GA and job shop scheduling problem provides a rich experience for the constrained combinatorial optimization problems. Application of genetic algorithm gives a good result most of the time.

Although GA takes plenty of time to provide a good result, it provides a flexible framework for evolutionary computation and it can handle varieties of objective function and constraint.

- Consider a 3 jobs and 3 machines problem with the operation sequence and the processing time for each operation have been determined.
- We run the programme for 3 times using the population size = 100, number of Generations is 500. The algorithm was terminated after 101 generations.
- From the result genetic algorithm could provide a result as good as other methods. From solution, we could see that the last job processed is job 1 on machine 2. So, our make span value for this problem is 77.
- The result also gives us the job sequence for each machine to process, the starting time and the finish time for each operation. For example, on machine 1, we start to process job 2 at time 0 and finished at 15. Then we process job 1, followed by job 1, and job 3.
- For Job sequence J2-J3-J1 following is the optimal solution

	M1		M2		M3	
	St	Fi	St	Fi	St	Fi
J2	0	15	15	35	35	44
J1	15	31	65	77	44	65
J3	31	53	0	8	8	26

Table 6: Manually solved example

6.1 Graphical Representation (Results by Implementing GA)

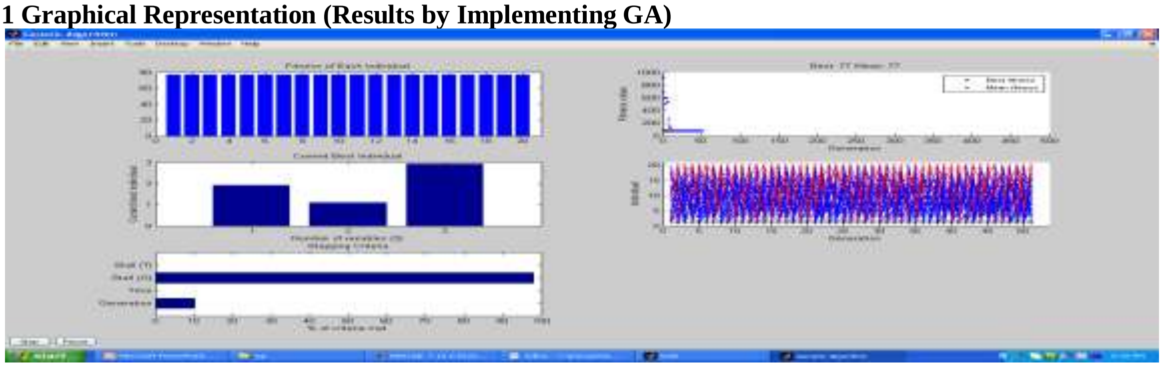

Fig. 6.1: Representation of Genetic Algorithm (Individual Fitness and Best value)

6.2 **Gantt Chart**

Fig. 6.2: Gantt Chart (Machine Vs time Chart)

Solution:
Optimal Job Sequence: 2 1 3

Machine 1 :
Job 2 Start Time : 15 Finish Time : 31
Job 1 Start Time : 0 Finish Time : 15

Job 3 Start Time : 31 Finish Time : 53

Machine 2 :
Job 2 Start Time : 65 Finish Time : 77
Job 1 Start Time : 15 Finish Time : 35
Job 3 Start Time : 0 Finish Time : 8

Machine 3 :
Job 2 Start Time : 44 Finish Time : 65
Job 1 Start Time : 35 Finish Time : 44
Job 3 Start Time : 8 Finish Time : 26

Fig.6.2.a): Optimal solution

If we compared the solution obtained from running GA we have job sequence as J2-J1-J3 with makespan value as 77 and the manually as shown in table no. 6 we have the same value.
GA search randomly the sequence of jobs and evaluate it with the fitness function and give the near optimal makespan value.

6.2 JSP considering Batch size:

In addition Normal Construction of JSP we add one more Input parameter as Batch size for each Job. The results obtained in the following manner. As considering the manufacturing scenario specially JSP there are variety of jobs having large number of quantity to be produced.

Fig. 6.2.1: Input parameters considering Batch size

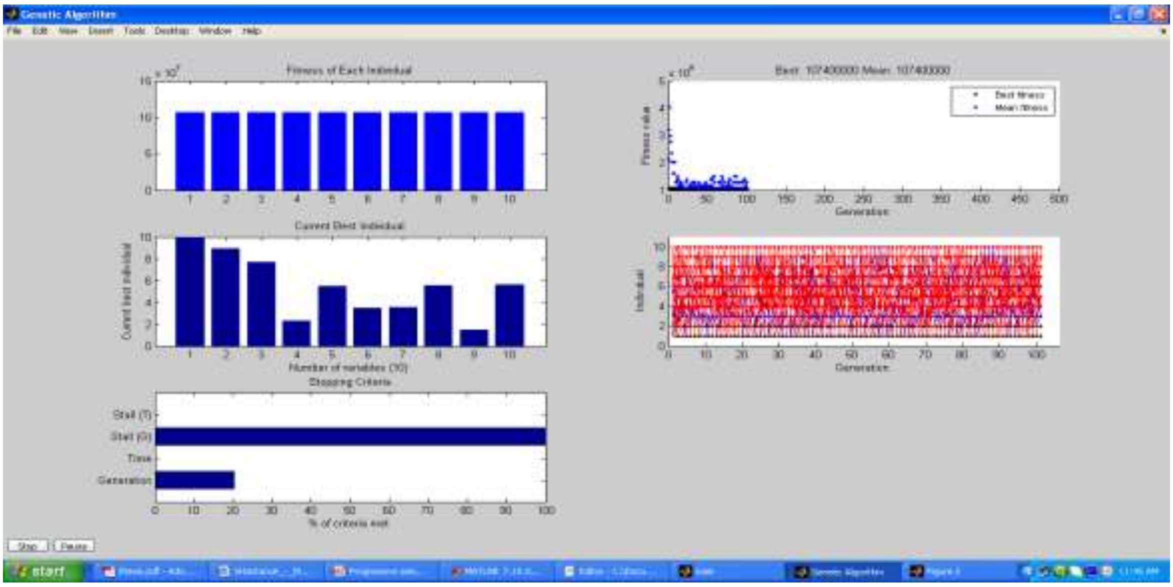

Fig. 6.2.2: Results obtained from GA (For Batch size)

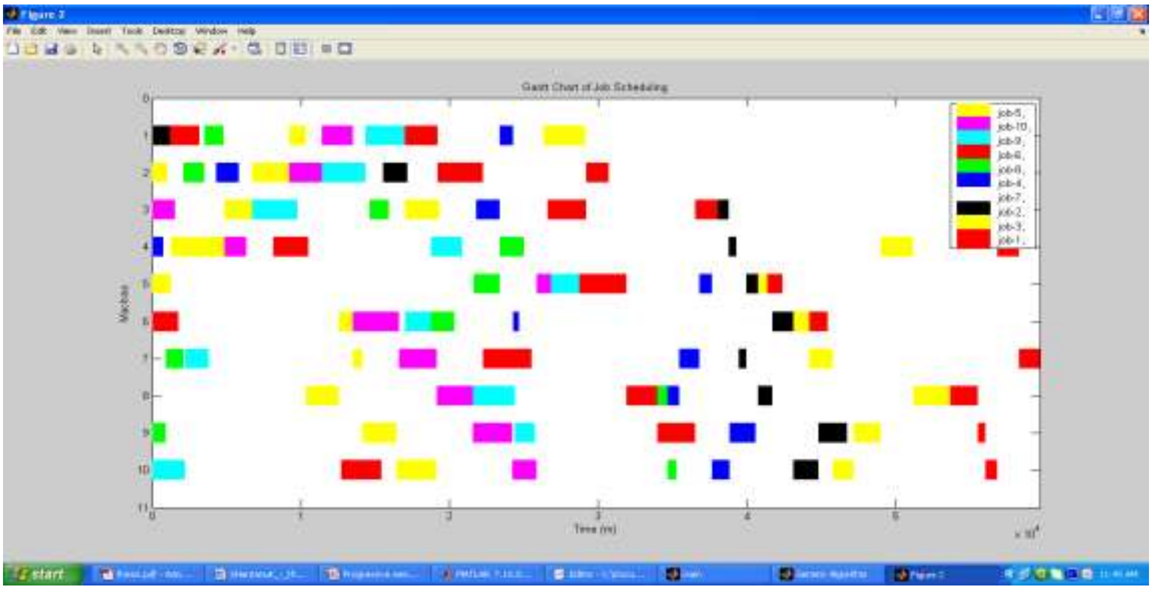

Fig. 6.2.3: Gantt Chart (Batch size)

Results Obtained Machine wise:
Optimal Job Sequence: 5 10 9 6 8 4 7 2 3 1

Machine 1 :
Job 5 Start Time : 35380 Finish Time : 48140
Job 10 Start Time : 12000 Finish Time : 32000
Job 9 Start Time : 263030 Finish Time : 291630
Job 6 Start Time : 233750 Finish Time : 242870
Job 8 Start Time : 0 Finish Time : 12000
Job 4 Start Time : 169810 Finish Time : 192310
Job 7 Start Time : 242870 Finish Time : 263030
Job 2 Start Time : 114160 Finish Time : 135060
Job 3 Start Time : 143560 Finish Time : 169810
Job 1 Start Time : 92160 Finish Time : 103680

Machine 2 :
Job 5 Start Time : 20880 Finish Time : 35380
Job 10 Start Time : 292000 Finish Time : 307000
Job 9 Start Time : 0 Finish Time : 10400

Job 6 Start Time : 42980 Finish Time : 58180
Job 8 Start Time : 155560 Finish Time : 171560
Job 4 Start Time : 192310 Finish Time : 222310
Job 7 Start Time : 222310 Finish Time : 236710
Job 2 Start Time : 92160 Finish Time : 114160
Job 3 Start Time : 114160 Finish Time : 143560
Job 1 Start Time : 67680 Finish Time : 92160

Machine 3 :
Job 5 Start Time : 146270 Finish Time : 159610
Job 10 Start Time : 265750 Finish Time : 292000
Job 9 Start Time : 170010 Finish Time : 193410
Job 6 Start Time : 217790 Finish Time : 233750
Job 8 Start Time : 380360 Finish Time : 387560
Job 4 Start Time : 365360 Finish Time : 380360
Job 7 Start Time : 117810 Finish Time : 134130
Job 2 Start Time : 0 Finish Time : 15400
Job 3 Start Time : 67680 Finish Time : 98130
Job 1 Start Time : 48960 Finish Time : 67680

Machine 4 :
Job 5 Start Time : 233680 Finish Time : 249920
Job 10 Start Time : 568090 Finish Time : 583090
Job 9 Start Time : 490040 Finish Time : 512140
Job 6 Start Time : 0 Finish Time : 7600
Job 8 Start Time : 387560 Finish Time : 393160
Job 4 Start Time : 81260 Finish Time : 105260
Job 7 Start Time : 263030 Finish Time : 287030
Job 2 Start Time : 48960 Finish Time : 63260
Job 3 Start Time : 187660 Finish Time : 208660
Job 1 Start Time : 12960 Finish Time : 48960

Machine 5 :
Job 5 Start Time : 216280 Finish Time : 233680
Job 10 Start Time : 414040 Finish Time : 424040
Job 9 Start Time : 407540 Finish Time : 414040
Job 6 Start Time : 368100 Finish Time : 376460
Job 8 Start Time : 399540 Finish Time : 407540
Job 4 Start Time : 287360 Finish Time : 318860
Job 7 Start Time : 12960 Finish Time : 19680
Job 2 Start Time : 258560 Finish Time : 268460
Job 3 Start Time : 268460 Finish Time : 287360
Job 1 Start Time : 0 Finish Time : 12960

Machine 6 :
Job 5 Start Time : 187660 Finish Time : 203320
Job 10 Start Time : 441940 Finish Time : 454440
Job 9 Start Time : 431540 Finish Time : 441940
Job 6 Start Time : 242870 Finish Time : 246670
Job 8 Start Time : 417140 Finish Time : 431540
Job 4 Start Time : 0 Finish Time : 18000
Job 7 Start Time : 287030 Finish Time : 309110
Job 2 Start Time : 135360 Finish Time : 166160
Job 3 Start Time : 169810 Finish Time : 187660
Job 1 Start Time : 125280 Finish Time : 135360

Machine 7 :
Job 5 Start Time : 9280 Finish Time : 20880
Job 10 Start Time : 583090 Finish Time : 596840

Job 9 Start Time : 441940 Finish Time : 457540
Job 6 Start Time : 354420 Finish Time : 368100
Job 8 Start Time : 394740 Finish Time : 399540
Job 4 Start Time : 222310 Finish Time : 255310
Job 7 Start Time : 377460 Finish Time : 394740
Job 2 Start Time : 166160 Finish Time : 191460
Job 3 Start Time : 22050 Finish Time : 37800
Job 1 Start Time : 135360 Finish Time : 141120

Machine 8 :
Job 5 Start Time : 339860 Finish Time : 346820
Job 10 Start Time : 536840 Finish Time : 555590
Job 9 Start Time : 512140 Finish Time : 536840
Job 6 Start Time : 346820 Finish Time : 354420
Job 8 Start Time : 407540 Finish Time : 417140
Job 4 Start Time : 318860 Finish Time : 339860
Job 7 Start Time : 354420 Finish Time : 377460
Job 2 Start Time : 191460 Finish Time : 215660
Job 3 Start Time : 215660 Finish Time : 244010
Job 1 Start Time : 103680 Finish Time : 125280

Machine 9 :
Job 5 Start Time : 0 Finish Time : 9280
Job 10 Start Time : 555590 Finish Time : 560590
Job 9 Start Time : 471840 Finish Time : 490040
Job 6 Start Time : 388620 Finish Time : 406100
Job 8 Start Time : 448340 Finish Time : 467540
Job 4 Start Time : 339860 Finish Time : 365360
Job 7 Start Time : 406100 Finish Time : 416660
Job 2 Start Time : 215660 Finish Time : 242060
Job 3 Start Time : 244010 Finish Time : 257660
Job 1 Start Time : 141120 Finish Time : 164160

Machine 10 :
Job 5 Start Time : 346820 Finish Time : 352620
Job 10 Start Time : 560590 Finish Time : 568090
Job 9 Start Time : 457540 Finish Time : 471840
Job 6 Start Time : 376460 Finish Time : 388620
Job 8 Start Time : 431540 Finish Time : 448340
Job 4 Start Time : 127310 Finish Time : 154310
Job 7 Start Time : 416660 Finish Time : 428180
Job 2 Start Time : 242060 Finish Time : 258560
Job 3 Start Time : 0 Finish Time : 22050
Job 1 Start Time : 164160 Finish Time : 191520
Maximum value of Completion or total processing time for completing all the jobs indicates makespan value for particular job sequence.

VII. Conclusion And Discussion

The conclusion and results of this study will be presented in this chapter. This chapter has been divided into two parts. This first part summarizes the Job shop problem as well as the result and achievements of the solution. The second part focuses on some potential future work of this study. To get an optimal solution will have to run the GA programme for 3 to 4 times because GA is a random search.

7.1 Conclusion

This work focuses on solving a Job shop scheduling in Job shop environment. This work uses Genetic Algorithm in MAT Lab using GUI to solve this type of problem considering all the constraints as mentioned in previous chapter for getting the best optimal solution in form of optimized schedule for Job shop industry. As shown in the previous chapter, the schedules for JSP are shown in terms of Job sequence and the Min Makespan value. Also it provides the start and finish time of each operation of each job considering the sequence and precedence constraint .The feasibility of the method of Genetic Algorithm was tested by applying it to this Job shop scheduling problem and considering that all the constraints are satisfied. In this flexibility to edit the data or change the input parameters as per the requirement of job shop is provided. Accordingly after running the software it shows the output in the prescribed format.

7.2 Future Scope

The future work for this study is divided into two aspects. The first aspect is on the algorithms. This work uses Genetic Algorithm to solve the schedule optimization problem.

As mentioned previously, many other algorithms like Ant Colony Optimization (ACO), simulated annealing, tabu search and other forms of local search are appropriate for solving this kind of problems. This problem can be used to test other algorithms and compare with Genetic Algorithms.

The second aspect is on the types of Job shop scheduling problem. In every different case, the essence of the JSP remains the same, which is to assign machines to each job as per the sequence of operation. However, the objectives of the JSP range from minimizing total processing time leads to increase the production efficiency.

In view of the dynamic conditions in the Job shop environment like random job arrival, priority of jobs changes as per the customer demand, set up time required for doing specific jobs, breakdown time and all other constraints can be considered.

In today's industrial scenario CNC machines are used for increasing the production efficiency, so in that the single machine can handle various operations. In that case the concept of Flexible manufacturing system can be adopted and it provides bigger search space for GA to give the optimal solutions.

Also, some Job shop scheduling problems includes different skilled of operators can be categorized as skilled, unskilled and semi skilled and with this constraint the schedules can be obtained.

Also same methodology can be applied for other scheduling problems like assignment, transportation etc.

References:

[1] Ye LI and Yan CHEN from Transportation Management College, Dalian Maritime University, Dalian, PR.China in Journal of Software Vol.5, No.3 March2010.A Genetic Algorithm for Job Shop Scheduling.

[2] Mahanim Omar, Adam Baharum, Yahya Abu Hasan School of Mathematical Sciences, in proceedings of the 2nd IMT –GT Regional Conference on Mathematics,Statistics and Applications Universiti Sains Malaysia,Penang,June 13-15,2006,A Job shop Scheduling problem using GA.

[3] Kanate Ploydanai and Anan Mungwattana, IJCSE International journal on Computer Science and Engineering Vol.02,No.05,2010,1919-1925 Algorithm for solving job shop scheduling problem based on machine availability constraints,.

[4] Ramezanali Mahdavinesad of University of Tehran in International journal of Quality researchUniversity of Tehran.UDK-658.524 short Scientific Paper (1.03), A new Approach to Job Shop Scheduling Problem

[5] E.Falkenauer and S.Bouffouix, CRIF - Research Center for Belgian Metalworking Industry published CP 106-P4,50,av.F.D.Roosevelt,B-1050 brussels,Belgium. A Genetic Algorithm for Job Shop.

[6] Milos Seda published a paper in World academy of Science Engineering and technology 31 2007. Mathematical Model of Flow shop and Job shop scheduling problem.

[7] Takeshi Yamada and Ryohei Nakano rom NTT Communication Science Labs in the proceeedings of Modern Heuristic for decision Support,pp. 67-81,UNICOM Seminar,18-19 March 1997,London. Genetic Algorithms for Job-Shop Scheduling Problems.

[8] Shignebu kobayash, Isao and Masayuki Yamamura in Proceedings of ICGA in 1995,pp.506-511(1995) An Efficient Genetic Algorithm for Job Shop Scheduling Problem .

[9] M.A.Adibi,M Zandieh,m.Amiri.in Elsevier Expert System with Applications 37 (2010) 282-287 Multi objective scheduling of dynamic job shop using variable neighborhood search.

[10] Shyh-chang Lin Erik d goodman William F.Punch,III, Genetic algorithm research and Applications group Michigan State University A Genetic Algorithm Approach to dynamic Job Shop Scheduling Problems

[11] S. M. Kamrul Hasan, *Student Member, IEEE, Ruhul Sarker, Member, IEEE, and David Cornforth* GA with Priority Rules for Solving Job-Shop Scheduling Problems.

Books:

- ✓ Introduction to Genetic Algorithm: S. N. Sivanandam and S.N Deepa

Sources :

- ✓ MATLAB Version 7.10