FLEXIBLE JOB SHOP SCHEDULING OPERATION

Submitted In partial fulfillment for the award of the Degree of

Bachelor of Technology

In Department of Mechanical Engineering,

National Institute of Technology, Rourkela,

Submitted By :-

ANSUMAN BARAL

ROLL NO 110ME0280

Under the guidance of :-

Dr. SIBA SANKAR MOHAPATRA

Professor

(Department of Mechanical Engineering, NIT Rourkela)

Department of Mechanical Engineering

NATIONAL INSTITUTE OF TECHNOLOGY

ROURKELA

NATIONAL INSTITUTE OF TECHNOLOGY

ROURKELA

CERTIFICATE

This is to certify that the work in this thesis entitled **Flexible Job shop Scheduling Operation** *by* **Ansuman Baral,** *has been carried out under my supervision in partial Fulfillment of the requirements for the degree of* ***Bachelor of Technology*** *in Mechanical Engineering during session 2013 - 2014 in the Department of Mechanical Engineering, National Institute of Technology, Rourkela.*

To the best of my knowledge, this work has not been submitted to any other University/Institute for the award of any degree or diploma.

Dr. Siba Sankar Mohapatra

(Supervisor**)**

Professor

Dept. Of Mechanical Engg.

National Institute of Technology

Rourkela, 769008

ACKNOWLEDGEMENT

It's a great pleasure to express my deep sense of gratitude and respect to my supervisor **Prof. Siba Sankar Mohapatra** for his excellent supervision, suggestions and support throughout the project. I feel extremely fortunate to work under the direction of such a vibrant personality. I am also thankful to **Prof. K.P. Maity, H.O.D**, Department of Mechanical Engineering, N.I.T. Rourkela for his constant support and inspiration.

Last but not least, I wish to express my sincere thanks to all other faculty members, master research and my friends in the Department of Mechanical Engineering, NIT, Rourkela, for their valuable help and advice at every step the success of this project report.

DATE:

Ansuman Baral
110ME0280
NIT ROURKELA
Dept. Of Mechanical Engineering

CONTENTS

ABSTRACT

When a large number of jobs and machines are taken into account, efficiency in the Job shop scheduling plays an essential role. In case of classical job shop scheduling, an operation is allowed to be processed by any machine from a given set and the FJSP is an extension of the classical job shop scheduling problem. The problem consists of two steps, one is to assign each operation to a machine and the second is to sequence the operations on the machines, such that the maximum completion time (makespan) of all operations is minimized. Many heuristics methods are designed as solutions with a close to optimal solution. This work deals with the job shop scheduling using Genetic algorithm aimed at creating a mathematical model with precedence order of the jobs as constraint. A MATLAB code will be used to generate an algorithm for finding the optimal solution. The input parameters are the operating time and the sequence of operation for each job in the machines provided. The makespan value is used to compare the results.

LIST OF FIGURES AND TABLE

Chapter 1

The Introduction

Chapter-1

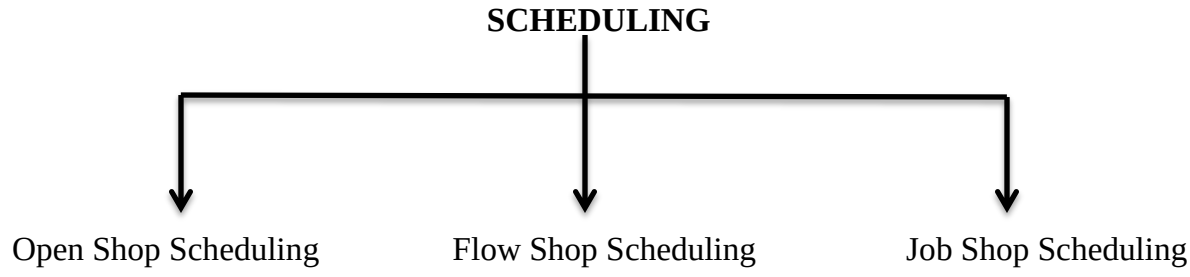
THE INTRODUCTION

Flexible Manufacturing System (FMS):

FMS is a method for producing goods that is readily adaptable to changes in the product being manufactured, in which machines are able to manufacture parts and in the ability to handle varying levels of production. (FMS) is a manufacturing system in which there is a degree of flexibility that allows the system to react in the case of changes, whether planned or unplanned. A flexible manufacturing system (FMS) gives manufacturing firms an advantage in quickly changing manufacturing environment. In this type of production where small batches of a variety of custom products are made.As most of the obtained products require a unique configuration and sequencing of the processing steps of flexible manufacturing systems is usually considered to their more effective in the manufacture of components rather than the finished products.

Scheduling:

As a job is always characterized by its route, processing requirement and priority, scheduling rates the works in order of its priority and then provides for its release to the plant at the proper time and in the correct sequence. Scheduling comes after Routing and the jobs may be scheduled based on various parameters such as lowest processing time, most work remaining etc. Scheduling is categorized into 3 categories.



Open Shop Scheduling:

The **open-shop scheduling problem (OSSP)** is a scheduling problem in which a given "n" jobs must each be processed for given amounts of time at each of a given "m" workstations, in an arbitrary order. Each task must be processed on a workstation at least once and the processing time can be zero in some machines. The order in which this occurs is not relevant.

Flow Shop Scheduling:

Flow shop scheduling problems, are a class of scheduling problems where there are a set of "m" number of jobs and "n" number of machines, where a strict sequence of operations for each job is followed. Especially the maintaining of a continuous flow of processing tasks is desired with a minimum of ***idle time*** and a minimum of ***waiting time***. A minimal downtime and minimal waiting time are the constraints in these kinds of problems. Flow shop scheduling is a special case of the shop scheduling tasks where there is a strict order of all operations performed on all jobs. In this case, each machine can perform more than one operation for a particular job.

.

Job Shop Production:

A job shop is a sort of manufacturing process in which little groups of an assortment of custom items are made. In the job shop process flow, the majority of the items prepared oblige a novel set-up and sequencing of methodology steps. In a job shop problem there are a limited number of jobs are obliged to be handled by a limited number of machines. Each one job comprises of an arrangement of operations which are decided beforehand. For a particular job operations are processed according to their technological sequence and a strict precedence constraint is followed i.e. none of the operations will be able to start processing before the preceding operation is over. The operations on a particular machine are performed without interruption for a period of time. There are no initial and terminal machines i.e. machines those perform inly the initial and final operation of a job.

A viable program is an assignment of operations to time on a violation without restrictions workshops machine. **A makespan** is defined as the maximum time for completion of all work. Our aim is to generate such a program in the process of shop scheduling tasks to minimize the makespan i.e. the time length of the schedule, in which all the operations of each task is completed.

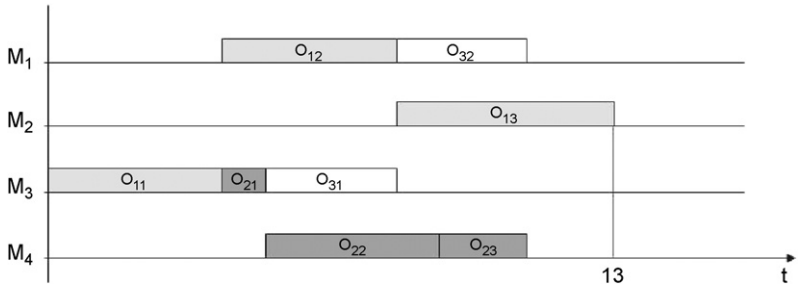
The problem of Job-shop Scheduling flexible (FJSP) is a generalization of the traditional JSP, where there are a set of machines available and each operation is allowed to be processed on any one of the available ones. A FJSP is more troublesome than the established JSP, because it adds a level of decision yet beside that sequencing i.e. job routes. Towards job route decides appropriate machine to process a particular operation among the available machines. The difficulty of FJSP suggests the use of heuristics to produce good enough schedule instead of seeking the exact solution.

Constraints applied in a job shop scheduling problems:

- A job should not visit the same machine more than once.
- There is no precedence constraints on operations of different jobs.
- Operations ones started can't be interrupted.
- Each machine is capable of processing one job at a time
- Each job must be processed through a particular predefined sequence of operations

Each optimization problem must have an objective function to be minimized or maximized is to get some solution. In this case, the objective function is the value of the total execution time or the programming interval length. The makespan value to be set as follows:

The problem has a certain number of jobs and each job is comprised of number of operations. Each operation takes certain amount of time i.e. each operation has a make value. When these operations get arranged in the sequence (according to their precedence order) of the program and then each machine gets a particular make value. **Maximum** of the make values of all the machines is the value of **Makespan (longest path).**



As per the above Gantt chart, it represents a schedule for a 3 job 4 machines FJSP problem, and the schedule represented has a makespan of **13**.

Chapter 2

Literature Review

Chapter-2

LITEARATURE REVIEW:

- Job Shop scheduling problems are NP job hard. Brucker [1] and Garey [2] stated this and obtaining solutions to these problems are a difficult task. A number of heuristic approaches have been developed in recent decades by researchers to optimize scheduling problems scheduling the job shop and some of them are: genetic algorithm, artificial immune systems, simulated annealing, optimization of ant colony, etc.

- Over the years several heuristic processes such as dispatching rules, locol search and a few metaheuristics for example tabu search, GA have been developed to FJSP. These can be classified into two broad categories: the hierarchical approach and integrated approach. The hierarchical approach reduces difficulty by solving the problem by decomposing into a sequence of sub-problems. Brandimarte [4] Paulli [5], rooms and Barnes [3] followed the same approach among others. They all used different dispatching rules to solve the assignment problem and also solved the resulting schedule using different heuristics.

- Integrated approach is much more difficult to solve, for best results, as shown in Vaessens et al [6] Dauzère Fathers and Paulli [7] Hurink et al. [14] And Mastrolilli and Gambardella [8] they all adopted an integrated approach. Among them, Mastrolilli Gambardella showed the results of calculation claiming their tabu search works better than any other heuristics developed so far, in terms of computing time and solution quality.

- Brandimarte (1993) [9] was the first to implement this heuristic method to solve job shop scheduling. And Brucker Carlier and Piso [10] suggested branch and bound methods for the solution to small problems. But for large size problmes Blazewicz [12] developed an effective local search method and the results of the method were found for at least one preferred program.

- n-Chan Choi [13] aimed to develop a local search algorithm to solve the problem of job shop scheduling with an objective to minimize makespan. Local search algorithms reduces the total computation time.Sequence dependent setup status is added to this problem. The total processing time of each job depends on the job sequence in each machine.
- Now a days GAs have been efficiently implemented to solve FJSP. The most significant works are those of Chen et al[15], Jia et al[17], Ho and Tay[18] and, significantly Kacem et al[16]. They all adopted integrated approaches, and vary from each other differen coding methods, initial population pool generation, selection of chromosomes for reproduction and generation strategies for descendants. Chen et al [15] divided the chromosome representation into two parts, the first defining routing policy, and the second sequence of operations on each machine. Jia et al. proposed a modified GA to solve distributed scheduling problems and can be implemented for FJSP

- Kacem et al. [16] suggested a chromosome representation which is a combination of routing and sequencing, and to develop an approach to find the location of favourable initial assignments. To sequence the operations dispatching rules were applied. Once this initial population generated, then crossover and mutation operator to be applied jointly to generate descendants.

- Mastrolilli and Gambardella (2000) [19] worked on neighborhood functions FJSS that might be utilized in Meta heuristic optimization methods. This method is superior to any other method in terms of results and quality of the solution calculation.

- DA Koonce [20] used data mining to find the programming model for problems job shop scheduling. This work aimed at applying the method of data mining to explore the model. Genetic algorithm is used to generate a better solution and Data mining is used to find the relationship between the sequences of the operations and predict the next job in the sequence. The result of data mining can be used to summarize new rule that gives the result as a result of the genetic algorithm

- Chandrasekharan [21] introduced three new dispatching rules for dynamic flow shop problem and the Job shop problem. He compared the performance of these rules, to 13 sequencing rules. The problem is modified by random route. Problems are changed Job shop flow programming problem scheduling problem. The study concluded that the performance of dispatching rules is influenced by the flow of work.

- Chenet al. (1999) [22] utilized an integrated methodology to comprehend the FJSP. Genes chromosomes separately depict a cement assignment operations of each one machine and the sequence of operations on each one machine. Yang (2001) proposed a GA-based discrete element programming methodology. Zhang and Gen (2005) proposed a genetic algorithm on the premise for taking care of the issue from the perspective of dynamic programming

- Jansen [24] scheduling problem solved store work under the assumption that the jobs have a controllable processing time. This means we can reduce the processing time of work by paying a cost. Jansen presented two models and these are the continuous pattern and the reduction model. The test may be clear that the two can solve in polynomial time approximation scheme is fixed when the number of machines and

the number of operations. Problem of job-shop scheduling to minimize makespan is considered.

- Guinet [25] reduces the problem of job shop problems casting jobs limited priority. After that, he used the Johnson rule to solve this problem and he observed that the optimization of rule long Johnson is demonstrated by two state machines and effective for the problems of the shop three or four manual machines.

- Ho et al. (2007) [26] proposed a detail structure for learning and evolution of genetic architecture FJSP called Programmable (from LEGA). LEGA gives a successful incorporation between the development and learning process in an irregular or random search. Pezzella et al. (2007) incorporate different methods for generation of initial population pool and selection of chromosomes for generation of offsprings. Tay and Wibowo (2004) [27] united both GA and a variable neighborhood descent (VND) to solve the Flexible Job shop problem.

- Ganesen[28] solved a special case of job shop scheduling problem by adding a new constraint, Minimum variance time competition restriction (CTV) to the problem. The lower limit of the CTV is developed for the problem to solve this problem using programming approach backwards. The result obtained from the programming approach backwards is compared with those of obtained from the forward approach..

Chapter 3

Methodology Adopted

Chapter-3

Methodology Adopted:-

The FJSP can be defined as follows.

It is given a set $J = \{J_1, J_2, \ldots, J_n\}$ of independent jobs. A job J_i is formed by a sequence $O_{i1}, O_{i2}, \ldots, O_{in_i}$ of operations to be performed one after the other according to the given sequence and also it is given a set $M = \{M_1, M_2, \ldots\ldots, M_m\}$ machines to process these operations. Each operation can be processed on any among given subset of Compatible machines. There exists a case of partial Flexibility if there exists a proper subset for atleast one operation. When each operation can be processed on every available machine there exists a case of Total Flexibility. The make value of each operation is machine-dependent.

- Let $\mathbf{t_{i,j,k}}$ be the processing time of operation $\mathbf{O_{i,j}}$ when processed on machine $\mathbf{M_k}$ and once an operation is started it can't be interrupted. The jobs and machines are available at time 0 and each machine is capable of performing only one operation at a time.The aim of the problem is to assign each operation to an appropriate machine (routing problem), and to sequence the operations on the machines (sequencing problem) in order to minimize the makespan. And we will refer to any solution of the FJSP as a chromosome.

Gantt chart:

Devised by Henry Gantt in 1910s, Gantt chart is the representation type of bar chart used to represent a feasible schedule of a scheduling problem. A Gantt chart is a visual representation of a project schedule. A time-scale is given on the chart's horizontal axis and each activity is shown as a separate horizontal rectangle (bar) whose length is proportional to the time required (or taken) for the activity's completion. Gantt chart also provides the details about the precedence of operations under taken by the Jobs in various machines. It has some advantages as well as disadvantages. It is the best medium for representing the schedule in a small problem but for a problem with large number of activities a Gantt chart can't be used to represent the schedule i.e. it will be very difficult to represent. As it doesn't give any idea about the total project size, it becomes tough to compare two projects with same completion time.

- Problem data is organized in a table, where rows correspond to operations and columns correspond to machines. The entries of the input table are the processing times, as shown in Table 1. In this problem we have total flexibility

	M_1	M	M	M_4
O_{11}	7	6	4	5
O_{12}	4	8	5	6
O_{13}	9	5	4	7
O_{21}	2	5	1	3
O_{22}	4	6	8	4
O_{23}	9	7	2	2
O_{31}	8	6	3	5
O_{32}	3	5	8	3

Table 1: Processing Time table

A GENETIC ALGORITHM FOR FJSP:

A GA is a local search algorithm which follows evolution pattern. The algorithm starts from an Initial solution pool also called as population and applies genetic operators to produce offsprings which are assumed to be more fit than the ancestors. Each new chromosome corresponds to a solution. The process is repeated until a stop criteria is satisfied which may be the maximum number of iterations or a time constraint. In case of GA a more variable search space can be explored at each step.

The overall structure can be described in the following steps:

1. INTIAL POPULATION GENERATION:-

This step includes two processes. One is to assign each operation a machine and second is to sequence the operations keeping in mind the precedence constraint of operations. For assigning we randomly permute the jobs and machines given in the table. By doing this it finds different initial combinations in different run of the algorithm, so it can better explore the search space. And the sequencing is also done randomly with the precedence constraint. For example one of the solutions is as follows.

$$\mathbf{P= (O_{11}, M_1),(O_{12}, M_3),(O_{31}, M_2),(O_{21}, M_4),(O_{32,} M_1),(O_{22}, M_2),(O_{13},M_3),(O_{23}, M_4)}$$

2. CODING:-

For implementation of GA, the solution or the schedule needs to be represented symbolically i.e. by a string. In this case we use the task sequencing list representation proposed by Kacem et al., in which a string is formed by triples (*i*, *j*, *k*), one for each operation, where

i = the job to which the operation belongs

j= the serial number of the operation within the job

k=the machine processing that operation

The above solution can be represented by a string as shown below. And the length of the string will be equal to the number of operations.

(1,1,1)	(1,2,3)	(3,1,2)	(2,1,4)	(3,2,1)	(2,2,2)	(1,3.3)	(2,3,4)

3. FITNESS EVALUATION:-

The fitness evaluation is done to check the fitness of chromosomes at each generation. In this case the fitness evaluation function coincides with the makespan value of the schedule. In each generation all chromosomes are evaluated and the chromosomes with lower values of makespan or fitness value are considered more fit than others and are preferred for genetic evolution.

4. SELECTION:-

The selection operator is used to select chromosomes for reproduction. There are several selection methods to select chromosomes for mating pool, like binary tournament, random selection, linear ranking etc. But in this case we will select a pair of chromosomes randomly from the population. The two selected chromosomes are used for generation of offsprings. And after each iteration the mating pool is renewed.

5. OFFSPRING GENERATION:-

After the chromosomes are selected for reproduction, crossover and mutation operator are used to create offsprings. For crossover operator to be applied two chromosomes are selected and mutation operator is applied to a single chromosome.

In this case we apply Single point crossover that applies sequencing operator to parent chromosomes i.e. the assignment of operations to machines is kept same in offsprings while the sequence of operations is changed with precedence constraint of operations for a single job. In this paper we use **POX (Precedence Preserving Order-based Crossover)** and **PPS (Precedence preserving Shift Mutation)** operators for generation of offspring.

POX:-

In POX, it generates two children starting from two parent chromosomes with a crossover probability of **(P_c = 0.5).** First it selects an operation from the first parent and copies in the first child all the operations of the same job to which the selected operation belongs to. Then the new individual is completed using the remaining operations of the second parent in the same order. The same process is followed to generate the second child also.

PPS:-

In case of PPS, it selects a single operation from a single parent chromosome and shifts it to another position in the string but following the precedence constraints.

The mutation is done with a mutation probability **(P_m= 0.02)** and is applied only if it improves the solution quality.

The offspring generation is continued till a maximum number of offsprings are generated or a new generation is found. In this case the offspring generations stops when the number of offsprings are equal to the initial population. And the algorithm stops when the maximum number of iterations are completed. Here we are taking 300 iterations. And the best individual i.e. the schedule with the lowest makespan value is given as output.

Chapter 4

Results And Discussions

Chapter 4

RESULTS AND DISCUSSION:-

A MATLAB code was generated using the above Genetic algorithm. The program was tested on the following test problem of 3 job 4 machine from F.Pezzella, G.Morganti, G.Ciaschetti[2].

	M_1	M	M	M_4
O_{11}	7	6	4	5
O_{12}	4	8	5	6
O_{13}	9	5	4	7
O_{21}	2	5	1	3
O_{22}	4	6	8	4
O_{23}	9	7	2	2
O_{31}	8	6	3	5
O_{32}	3	5	8	3

Table:2- Problem involving 3 jobs 4 machines

The problem generated in MATLAB was executed and the following output was obtained at the end of 300^{th} iteration.

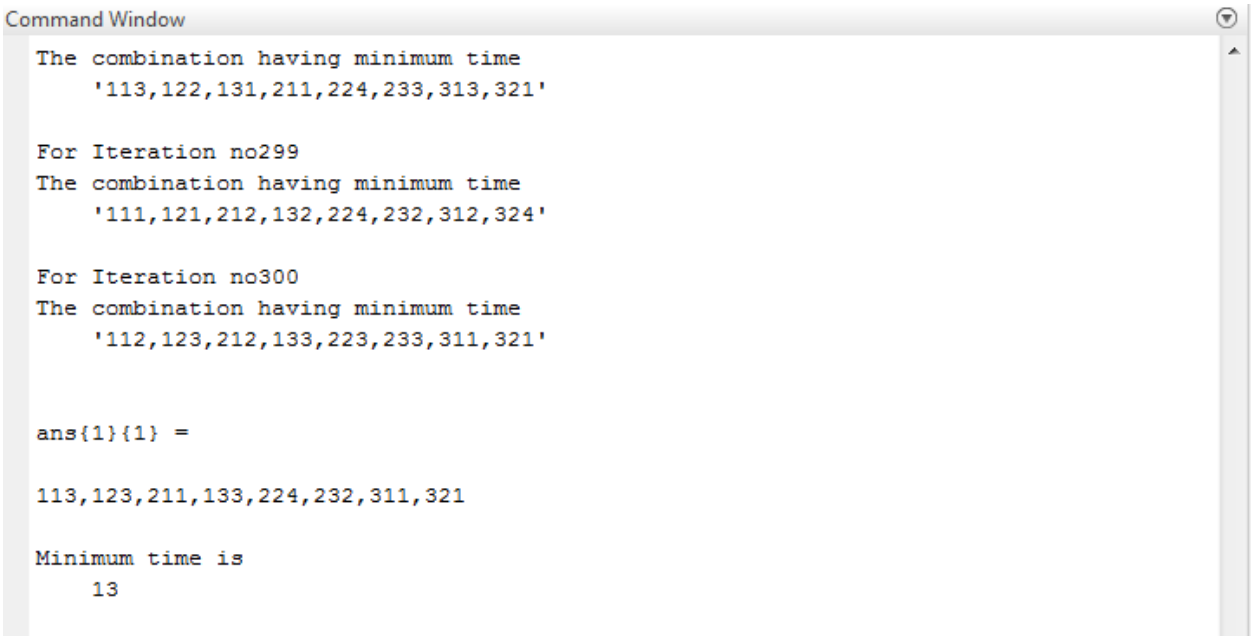


Figure-1: Resultant Schedule

As per the Output obtained, the schedule with minimum makespan value is as follows:

(1,1,3)	**(1,2,3)**	**(2,1,1)**	**(1,3,3)**	**(2,2,4)**	**(2,3,2)**	**(3,1,1)**	**(3,2,1)**

Or

S = (O_{11},M_3), (O_{12},M_3), (O_{21},M_1), (O_{13},M_3), (O_{22},M_4), (O_{23},M_2), (O_{31},M_1), (O_{32},M_1)

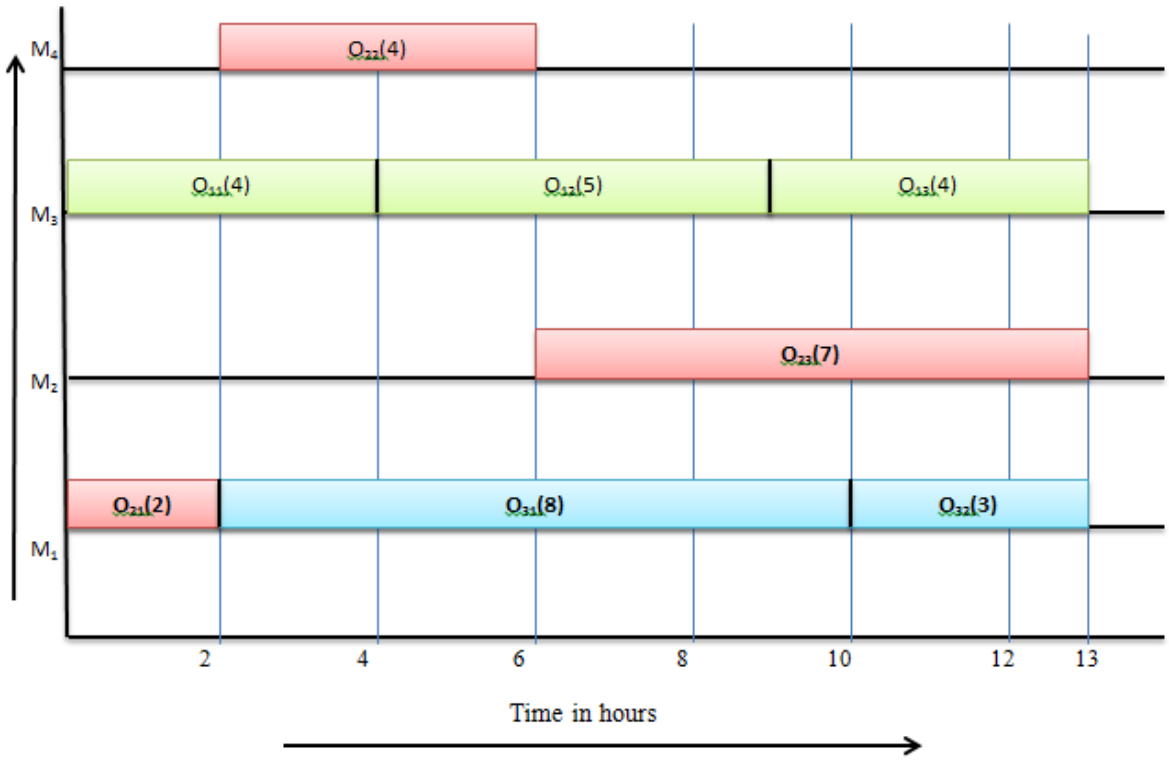


Figure 3 : Gantt Chart for the given schedule

The makespan value of the above schedule was found to be 13hours.

Chapter 5

CONCLUSION

Chapter-5

CONCLUSION:-

This current work focuses on the scheduling of jobs in the job shops and to optimize it by minimizing the makespan value. The genetic algorithm used was aimed at creating a mathematical model without machine availability constraint. The algorithm was coded in MATLAB and the algorithm was effective in many problems. The schedules obtained have makespan value near to optimal.

References :

1. Brucker P., 1995. Scheduling algorithms 2nd edn. Springer. Berlin Heidelberg New York.

2. F.Pezzeella, G.Morganti, G.Ciaschetti, 2007.A genetic algorithm for the Flexible job-Shop Scheduling Problem

3. Garey M., et al, 1976. The complexity of flow shop and Job shop scheduling. Mathematical Operation Research 1, pp. 117-119.

4. Barnes JW, Chambers JB. Flexible Job Shop Scheduling by tabu search. Graduate program in operations research and industrial engineering.Technical Report ORP 9609, University of Texas, Austin; 1996. http://www.cs.utexas.edu/users/jbc/ .

5. Brandimarte P. Routing and scheduling in a flexible job shop by tabu search Annals of Operations Research 1993;41:157–83.

6. Paulli J. A hierarchical approach for the FMS scheduling problem. European Journal of Operational Research 1995;86(1):32–42.

7. Vaessens RJM, Aarts EHL, Lenstra JK. Job Shop Scheduling by local search. COSOR Memorandum 94-05. Eindhoven University; 1994.

8. Dauzére-Pérés S, Paulli J. An integrated approach for modeling and solving the general multiprocessor job-shop scheduling problem using tabu search. Annals of Operations Research 1997;70:281–306.

9. Mastrolilli M, Gambardella LM. Effective neighbourhood functions for the flexible job shop problem. Journal of Scheduling 1996;3:3–20.

10. Brandimarte P., 1993. Routing and scheduling in a flexible job shop by taboo search. Annals of operation research, 41, 157-183

11. Brucker P., 1994 A branch and bound algorithm for Jobshop problem. Discrete Applied Mathematics, 49:107-127

12. Carlier J., Pison E., 1989.An algorithm for solving Job shop problem. Management Science, 35: 164-176

13. Blazewickz J., 1996. The Job shop scheduling Problem: Conventional and new solutions techniques. European Journal of Operational Researchpp, 931-30

14. Choi I. and Choi, D., 2002. "A local search algorithm for job shop scheduling problems with alternative operations and sequence dependent setups". Computer and industrial engineering, 2, 43-58

15. Hurink J, Jurish B, Thole M. Tabu search for the job shop scheduling problem with multi-purpose machines. OR-Spektrum 1994;15:205–15.

16. Chen H, Ihlow J, Lehmann C. A genetic algorithm for flexible Job-shop scheduling. In: IEEE international conference on robotics and automation, Detroit; 1999. p. 1120–5.

17. Kacem I, Hammadi S, Borne P. Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems.IEEE Transactions on Systems, Man, and Cybernetics, Part C 2002;32(1):1–13.

18. Jia HZ, Nee AYC, Fuh JYH, Zhang YF. A modified genetic algorithm for distributed scheduling problems. International Journal of Intelligent Manufacturing 2003;14:351–62.

19. Ho NB, Tay JC. GENACE: an efficient cultural algorithm for solving the Flexible Job-Shop Problem. IEEE international conference on robotics and automation 2004;1759–66

20. Mastrolli M. and Gambardella C. M., 2000. Effective neighborhood functions for the flexible jobshop problem. Journal of scheduling, 3(1), pp. 3-20.

21. Koonce D. A. and Tsai S. C., 2000. "Using data mining to find patterns in genetic algorithm to a job shop schedule", Computer and industrial engineering, 38, pp. 361-374.

22. Holthaus O. and Rajendran C., 1990. "A comparative of study of dispatching rules in dynamic flow shops and job shops". European journal of operational research , 116, pp. 156-170

23. Chen, H., Ihlow, J., & Lehmann, C. (1999). A genetic algorithm for flexible job-shop scheduling. In IEEE international conference on robotics and automation, Detroit 1999 (Vol. 2, pp. 1120–1125).

24. Jia, H. Z., Nee, A. Y. C., Fuh, J. Y. H., & Zhang, Y. F. (2003). A modified genetic algorithm for distributed scheduling problems. International Journal of Intelligent Manufacturing, 14, 351–362.

25. Jansen K., Mastrolilli M. and Oba R.S., 2005. "Approximation scheme for job shop scheduling problem with controllable processing time". European journal of operational research, 1672, pp. 97-319.

26. Guinet A., 2000. "Efficiency of reduction of job shop to flow shop problems". European journal of operational research, 125, pp. 469-485.

27. Ho, N. B., Tay, J. C., Edmund, M., & Lai, K. (2007). An effective architecture for learning and evolving flexible job shop schedules. European Journal of Operational Research, 179, 316–333.

28. Liu, H. B., Abraham, A., & Grosan, C. (2007). A novel variable neighborhood particle swarm optimization for multi-objective flexible job-shop scheduling problems.In Second international conference on digital information management, 2007.ICDIM '07 (pp. 138–145).

29. Ganesen V.K., Sivakumar A. I., and Srinivasan G., 2006. "Hierarchical minimization of completion time variance and makespan in job shops". Computer and operations research, 33, pp. 1345-1367.

30. Jasan C. H. and Chen J. S., 2005. "Mixed binary integer programming formulations for the reentrant job shop scheduling problem". Computer and operational research, 32, pp. 1197-1212.

31. Ganesen V. K. and Sivakumar A. I., 2006. "Scheduling in static job shops for minimizing mean flow time subject to minimum total deviation of job completion times". International journal production economics, 103, pp. 633-647.

32. Watanabe M., Ida K., and Gen M., 2005. "A genetic algorithm with modified crossover operator and search area adaptation for the job-shop scheduling problem". Computer and industrial engineering, 48, pp.743-752.

33. Koonce D.A. and Tsai S.C., 2000. "Using data mining to find patterns in genetic algorithm solutions to a job shop schedule". Computer and industrial engineering, 38, pp. 361-374.

34. Ganesen V. K., Sivakumar A. I., 2006. "Hierarchical minimization of completion time variance and makespan in job shops. Computer and operational research, 33, pp.1345-1367.