JOB SHOP RE- SCHEDULING USING GENETIC ALGORITHM – A CASE STUDY

P.ChithtraiSelvam[1], S.Vignesh[2], K.Mandharasalam[3], Sathiesh kumar[4], C.Sowmya Danalakshmi[5]

1,2,3,4 Department of Mechanical Engineering, P.A.College of Engineering and Technology, Pollachi- 642 002,
Tamil Nadu, India.
5, Assistant Professor, Department of Mechanical Engineering, P.A.College of Engineering and Technology,
Pollachi- 642 002, Tamil Nadu, India.
Corresponding Author Mail-id : sd_kce@yahoo.com

ABSTRACT

Scheduling concerns the allocation of limited resources overtime to perform tasks to fulfill certain criterion. One of the most popular models in scheduling theory is that of the job-shop scheduling, as it has earned a reputation for being notoriously difficult to solve. A job-shop scheduling problem comes under the category of combinatorial optimization problems and is very difficult to solve by conventional optimization techniques. Many scheduling problems from manufacturing industries are complex in nature and are to be best known about the difficult combinatorial optimization problems with a finite number of feasible solutions. A machine shop with four numbers of machines and five numbers of jobs and the following assumptions are taken into account and the machines are available at zero time and there are no breakdowns, machine processes one operation at a time and once an operation initiates processing on a given machine the machine stops only when the operation is over and the processing times are deterministic. In this present work, a rescheduling is done for the existing job-shop in a small scale industry and this problem is solved using the non-traditional optimization technique like Genetic Algorithms. Their performances are depicted graphically to exhibit how they have converged to the optimal solution.

Keywords*: Scheduling, Optimization, Feasible solution, Genetic algorithm

1 INTRODUCTION

Scheduling concerns the allocation of limited resources overtime to perform manufacturing industries are quite complex in nature and are to be best known of the difficult combinatorial optimization problems. The job shop scheduling problem may be described as follows :Given'n' number of jobs, each comprises several operations that must be processed on 'm' number of machines. Each operation uses one of the 'm' machines for a fixed duration. The operations of a given job have to be processed in a given order. The problem consists of finding a schedule of operations on the machines, taking into account the precedence constraints that minimizes the make span. Make span means the finish time of the last operation completed in the schedule. Genetic algorithms are a random search technique, invented by Holland (1975), which can find the global optimal solution by using the principles of natural genetics and natural selection. Genetic algorithm is found to be one of the best and fast converging search techniques when the search span is very large and gives better results when compared to other search procedures. Simulated Annealing Algorithm proposed by Kirkpatricketal (1983) finds the global optimum by cooling phenomenon of molten metals to constitute a search procedure. The cooling phenomenon is simulated by controlling a temperature like parameter introduced with the concept of Boltzmann's probability distribution.

1 a Problem Definition

A Machine shop with four numbers of machines and five numbers of jobs as in Table 1 is considered for scheduling and the following assumptions are taken into account,
a)   All machines are available at time zero
b)   There are no breakdowns
c)   Machine processes one operation at a time and once an operation initiates processing on a given machine, the machine stops only when the operation is over

d) The processing times are deterministic

1 b Feasible Solution

Each operation is coded by a numerical letter called identity number (ID) as shown in table 1. the first operations of all jobs are grouped into category-1 and second operations of all jobs are grouped into the other category and so on. A feasible solution to the problem consists of all categories in such a way that ID's of one category should not be mixed with another category. For example if the occurrence of all the ID numbers is only one time, then it is known as a feasible solution.

1 c Purpose of Coding

The advantage of this coding is that the constraints of scheduling are automatically taken care as all (i+1)th level operations will be carried out after processing (i)th level only.

Sl. No of Job	Machine Type	Processing Times (Min)	Identity Number (ID)
1	1	8	1
	2	7	2
	3	14	3
	4	9	4
2	4	6	5
	1	13	6
	2	10	7
	3	17	8
3	1	18	9
	2	16	10
	3	11	11
	4	12	12
4	3	12	13
	4	9	14
	2	15	15
	1	11	16
5	2	11	17
	1	7	18
	4	16	19

Table. 1. ID Numbers and Processing times

2 SOLUTION APPROACH: GENETIC ALGORITHM

In this work, a single method of population management is taken into account where each generation comprises a combination of immigrant and crossover solutions. Figure 1 represents the flowchart of Genetic Algorithm.
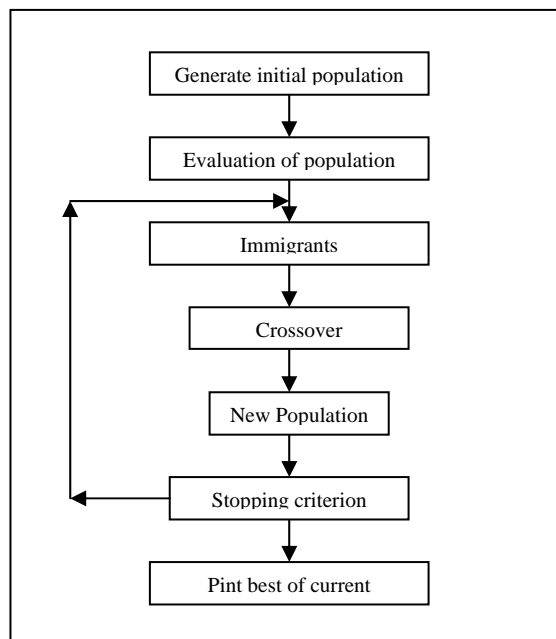
Figure. 1. Genetic Algorithm flow chart

Generally, elite solutions of the previous generation are put in the current population to assure that the best solutions known so far remains in the population and have greater opportunity to produce the offspring. Immigrant solutions maintain diversity as they are added arbitrarily. Crossover solutions constitute the rest of the current population. The sequence of operations is encoded indirectly under random keys as a vector of random numbers. For example, the 20 random vectors decode all the ID numbers as the sequence of operations. This is obtained by aligning the operation 1 with the lowest random component, operation 2 with the next lowest and so on. Crossover on two vectors of the random numbers produces two others. Thus every crossover operation of these random vectors coding produces two new feasible solutions. The following parameters are chosen to run the Genetic Algorithm code.

Table. 2. Parameter for GA Code

Parameters	Size
Population size	20
Cross over	10
Fitness	Make span (to be minimized)
Elites	6
Immigrants	4
Maximum number of iterations	16

3 TABU SEARCH

Tabu Search prevents short term cycling by temporarily forbidding moves that may return solutions recently obtained. For the present problem, a new feasible solution is produced by interchanging a pair of operations ands the first of each two positions interchanged is fixed for a period of 6 iterations. A sample output of Tabu Search algorithm for 5 iterations is given in Table 3. After the first interchange of the
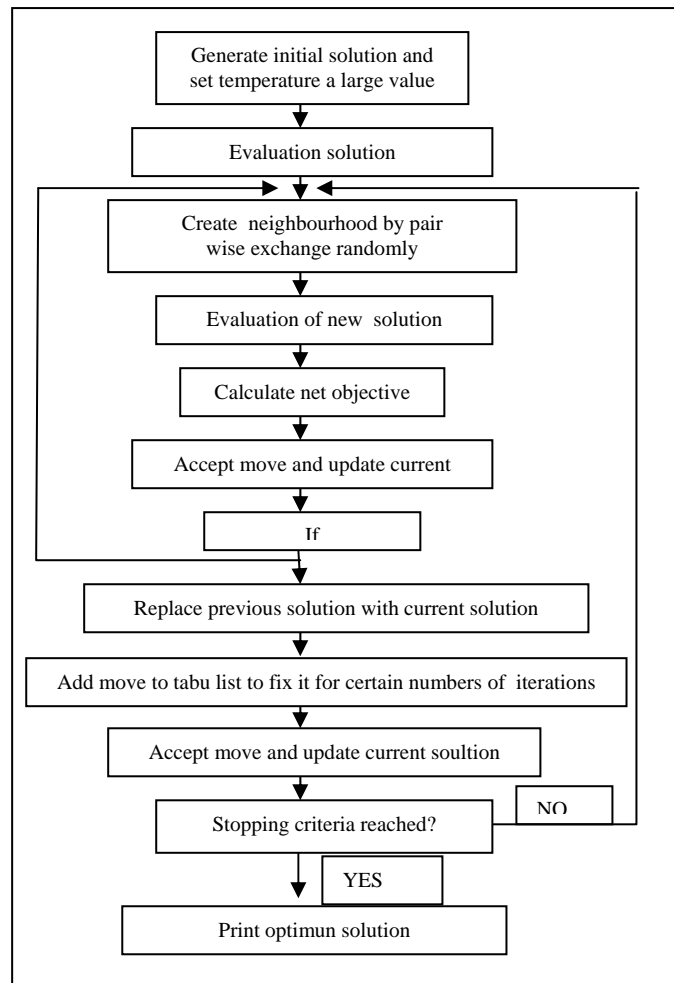
Figure. 2. Flow chart for Tabu Search optimization

12th and 4th operations, no move of the 12th position is allowed for the 6 iterations. This prevents immediate reverses up to 6 iterations.

The above move improves the make span of 89 minutes. All tabu positions are underlined. The optimal solution was found in the iteration number 76. this search is carried out for the maximum of 150 iterations.

Sequence	Make span	Incum bent	Inter change	Deviation or Change
Machine 1 1-9-5-17-13-14-10-18-6-2-3-15-19-11-7-16-20-12-4-8	95	95	12 for 4	------------
Machine 2 1-9-5-17-13-14-10-18-6-2-3-15-19-11-7-16-20-4-12-8	89	89	20 for 8	+6
Machine 3 1-9-5-17-13-14-10-18-6-2-3-15-19-11-7-16-8-4-12-20	89	89	14 for 8	0
Machine 4 1-9-5-17-13-8-10-18-6-2-3-15-19-11-7-16-14-4-12-20	101	89	11 for 15	-12
Machine 5 1-9-5-17-13-8-10-18-6-2-3-11-19-15-7-16-14-4-12-20	101	89	1 for 7	0

Table. 2. Tabu Search Iterations

4 RESULTS AND DISCUSSIONS

The source codes for all the above algorithms are developed using the computer language 'C' and the tests are run on a PC with a Pentium –III processor. All the source codes are generalized in such a way that they can handle any number of machines and any number of jobs. The optimal solution to the present problem by all algorithms is found to be 72 minutes and its sequence is 13-1-9-5-17-18-2-6-14-10-3-15-19-11-7-4-20-12-8-16. Figure 3 and 4 represents graphically the convergence of the optimal solution. The convergence depends upon the parameters chosen.

5 CONCLUSION

In this work, the optimal solution of the job-shop scheduling problem is obtained by using the non-traditional techniques such as Genetic Algorithm and Tabu Search. The major advantage of using these algorithms is that eventhough the number of possible sequences for 20 operations is very high, an optimal solution is obtained within 2 minutes while running on a standard PC. The effectiveness of these algorithms is tested through computer simulation for various real life problems and are found to be very effective.

REFERENCES

[1]   Cheng Runwei & Gen Mitsuo), Genetic Algorithms and Engineering Optimization, John Wiley & Sons, Inc. 2000.
[2]   David E. Goldberg Genetic Algorithms in search, optimization and machine learning. Pearson Education (Singapore) pvt. Ltd. 2002.
[3]   Haupt. L.Randy & Haupt Sue Ellen Practical Genetic Algorithms, John-Wiley & Sons, inc. 1998.

[4]   Karaboga. D. & Pham. D. TIntelligent Optimization Techniques – Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks, Springs-Verlag Berlin Heidelberg, 2000.
[5]   Peter Brucker Scheduling Algorithms, Springs-Verlag Berlin Heidelberg, 1995.
[6]   Rardin L.Ronald Optimization in Operations Research, Pearson Education (Singapore) Pvt. Ltd., 2002.
[7]   Whybark. D.C, William. J.G., Material requirements planning under uncertainty, Decision Sciences 1976, pp.595-606.
[8]   J.J.Kanet, Toward a better understanding of lead times in MRP systems, Journal of Operations Management, 1986,pp.305-315.
[9]   S.Shingo, A study of Toyota Production System from Industrial Engineering Viewpoint, Productivity press, Cambridge, 1989.
[10] E.Porteus, Optimal lot sizing, Process quality improvement and setup cost reduction, Management Science, 1986,pp.137-144.
[11] D.Hofman, The hierarchy of supply chain metrics, Supply Chain Management Review, 2004, pp 28-37.