IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE)
e-ISSN: 2278-1684, p-ISSN: 2320-334X
PP 24-30
www.iosrjournals.org

Job Shop Scheduling Problem by Using Genetic Algorithm

[1]K. Suresh, [2]R. Jeya Girubha
M.E Industrial Engineering Department of Mechanical Engineering Anna university regional office Coimbatore.
Smeie1214@gmail.com
Assistant Professor Department of Mechanical Engineering Anna university regional office Coimbatore.

ABSTRACT: *In Scheduling is one of the most important issues in the planning and operation of manufacturing system, but the generation of consistently good schedules is more difficult. The difficulty of finding the optimal schedule depends on the shop environment, process constraints and performance indicator. One of the most difficult problems in this area is the job shop scheduling problem (JSSP), which has been proved to be a Non-deterministic Polynomial-time (NP)-complete problem. In this paper, a genetic algorithm based solution methodology has been presented and the algorithm is implemented using powerful Matrix Laboratory (MATLAB) environment to solve and to find a schedule that has lowest possible value of makes pan, where the makes pan is the time required for all jobs to be processed according to a given schedule by using Genetic algorithms.*

Index Terms: *Job shop, scheduling, genetic algorithm.*

I. INTRODUCTION

The main driving factors to improve the performance of manufacturing systems are the continuous changing demands of customers and the need to deliver better quality and low-priced products. To achieve these requirements the manufacturing system has to have flexibility that enables it to quickly respond to this changing environment.

The company capacity to deliver their goods in the right time becomes essential. A high variety of product with low quantity comprise between 50% and 75% of all manufactured components, in that way making schedule optimization an crucial step in the overall manufacturing process. Thus, scheduling optimization holds the main role to catch up with market demand.

In general, the scheduling can be described as the allocation of jobs over time when limited resources are available, wherever a number of objectives should be optimized, and several constraints have to be satisfied. A job is determined by a predefined set of operations, and the result of a scheduling algorithm is a schedule that contains the start times and allocation of resources to each operation. It depending on the problem size, represented by the number of tasks and resources, the scheduling task may turn out to be the most time consuming and challenging activity within an enterprise. Each job is to be process on each machine in a predefined sequence and each machine processes only one job at a time. The job shop scheduling problem is one of the combinatorial optimization problems which is belong to the class of NP-hard problems. Since classical optimization methods (branch and bound method, dynamic programming) can be used only for small scale problem, more complex tasks must be solved by heuristic methods such as simulated annealing, tabu search, evolutionary algorithm, particle swarm and genetic algorithm. Here to focuses on developing algorithm to solve job shop scheduling problem. The algorithm is designed by considering machine availability constraint and the transfer time between operations. Next, machine availability constraint is described. The machine availability constraint is used to calculate sensible makespan for company that has breaking period during processing time. These systems provided a balance between mass production, capacity, and high flexibility, wide variety. Accordingly, GA has a better adaptability to variable production plans and goals as well as producing a higher quality and wider variety of products, with minimal machine setups.

For example, an inefficient scheduling results in poor utilization of resources, over-loaded or idle capacity, long production lead time and unreliable due date commitments, which in turn increases production costs and reduces competitiveness in the market place. Furthermore, ineffective scheduling often delays orders and results in unsatisfied customers and may subject the firm to penalties.

II. PROBLEM DEFINITION

This research is focusing on investigating machine scheduling problems in manufacturing environment where jobs represent activities and machines represent resources, and each machine can only process one job at a time. Here, we will focus on the low volume system also known as job shop. We will use the real case data gained from manufacturing company.

To simplify the explanation we will use the following notations throughout the paper:

j= job (j=1,2,….,n)
i= machine (i=1,2,…..,m)
tt= transfer time
P= processing time
W= waiting time
C= completion time job

In this type of environment, where the products are made to order, the job-shop scheduling problem (JSSP) can be described as follow:

1) Job sets
$J = \{J_1, J_2,\ldots,J_i\} \mid j = 1,2,3,\ldots,n$
2) Machine sets
$M = \{M_1, M_2\ldots M_i\} \mid i = 1, 2, 3,\ldots, m$
3) Operations
$O = \{O_1, O_2\ldots O_o\} \mid o = 1, 2, 3,\ldots, k$
4) Processing time for each operation
$P_{ij} = \{P_{11},P_{12},\ldots,P_{ij}\} \mid i = 1,2,3,\ldots,n$; $j = 1, 2, 3,\ldots, m$
5) Transfer time
$t_{ij} = \{t_{11},t_{12},\ldots,t_{ij}\} \mid i = 1,2,3,\ldots,n$; $j = 1, 2, 3,\ldots, m$

Moreover, each job should be processed through the machines in a particular order or also known as technological constraint. The maximal time required for all operations to complete their processes is called makespan while the average time required for all operations is called mean flow time. In this paper, our point is to minimize two objectives, makespan value and mean flow time.

$$C_{\max} = \max\{C_j\} \quad (1)$$

$$F = \frac{\sum_j C_j}{J} \quad (2)$$

When minimizing the make span, at least one of the optimal solutions is a semi-active (no operation can started earlier without violating the technological constraints. For this reason, every time when makespan is optimized, a schedule can be described by the processing orders of operations on the machines. Some assumptions are considers for this research are:

1) The jobs are independent and consist of firmly ordered operation sequences.
2) No priorities are assigned for any job or operation.
3) Jobs pre-emption is not allowed.
4) A given operations be able to performed by one or more non-identical machines (called alternative machines).
5) The time setup is independent of the operation sequence and is included in the processing times.
6) The transfer time between the operations will be occurs when there is a machine changes for each job.

The completion time of each job will be follows the equation (3). The transferring time will be included based on the total transfer time between operation perform in each job.

$$C_j = tt_j + \sum_{i \in I} P_{ij} + \sum_{i \in I} W_{ij} \quad (3)$$

Waiting time will be appears if the job arrive in the machine that still perform another job. Waiting time calculates by subtract the completion time of previous operation with the next operation in certain machine. The equation can be written as:

$$W_{ij} = \left| C_j - M_i \right| \quad (4)$$

With following constraints:

$$C_{ij} \ge C_{hi} + P_{ij} \quad (5)$$

$$C_{ij} - C_{ia} \ge P_{ij} \sqrt{C_{ia} - C_{ij}} \ge P_{ia} \quad (6)$$

if $M_j = M_a$

$$C_{ji} \ge 0, J, a \in M$$

Constraint (5) guarantee the operation can only be started after previous operation in the same machine is done. Constraint (6) required to make sure that each machine can only process a job in a time.

III. PROPOSED METHOD

As the JSSP is an NP-Hard problem, the projected formulation is not applicable to find optimal solutions. Hereafter, the genetic algorithm (GA) is used to attain the result of the objectives. Each chromosome or individual of the GA represents a permutation of the work station. The chromosomes with the heuristic crossover and mutation operators are developed through some replications. The genetic algorithm (GA) is a stochastic search technique that simulators the mechanisms of the Darwinian evolution based on the concept of the survival of the acceptable. The basic components of a GA are the solution representation, popularly recognized as the chromosome or individual, which represents a complete solution of a problem. A projected GA generates a set of permutation as representation solution, where the individual is a result change of the work station to be arranged. The following sub-sections described in detail how the GA is developed to solve the above JSP problem.

A. Initial Solution

The initial solution will be done randomly as a set of job change in each machine. The total generations for each chromosome will be equal to the number of operations performs to finish the products.

		M_1	M_2	M_n
4 2 3 6 5 9 10 11…. 10 11				
8 1 7 6 9 5 10 11…. 11 12	1			
2 3 6 8 1 1 5 10 9 ….14 15	2			
… … … …… …. ….. ……	3			
… … … …. … …. … ……	.			
4 2 3 6 5 9 10 11 ….11 12	.			
	o			

Fig. 1. Solution representation

B. Selection

The function of a selection operator is to form a mating consisting of the above-average chromosomes of the population. The mating pool will be used by the crossover and mutation operators with the expectation for generating good offspring chromosomes. The roulette wheel tournament is applied here for this purpose. It picks up two chromosomes from the population and stores a copy of the best chromosomes (based on objective values) in the mating pool. The process repeated until the mating pool equals that of the populations.

B. Crossover

Using a random procedure, two point cross over is performs. For each couple of parents with single line encoded chromosomes, a random integer is generated to choose the two cross over point. The next step is to swap the range between parent 1 and 2 based on the cross over point. To make sure the feasibility of the solution, this research implemented order cross over.

PARENT 1	1	2	3	4	5	6	7	8	9
PARENT 2	5	4	6	9	2	1	7	8	3

PRE-			3	4	5		7	8		OFFSPRING 1
PRE-				9	2	1	7	8		OFFSPRING 2

PRE-	7	8					3	4	5	OFFSPRING 1
PRE-	7	8					9	2	1	OFFSPRING 2

7	8	6	9	2	1	3	4	5	OFFSPRING 1
7	8	3	4	5	6	9	2	1	OFFSPRING 2

Fig. 2. Two point order cross over

C. Mutation

The mutation operator is applied to create the different new chromosomes and to prevent the population from stagnate in their local optimal solution with a predefined mutation probability. In this research, the mutation is done using swapping mutation. Using this random procedure, the two random integers are generated as the replacing genes.

D. Stopping Criteria

The maximum number of generations (G) is selected as stopping criteria. In this procedure from one generation to next generation, the crossover and mutation is repeated until the maximum number of generation is satisfied. In GA, here are three parameters considered, population size, crossover probability and mutation probability.

IV. EXPERIMENTAL DESIGN

The effect of many different parameters on the performance characteristic in a thick set of experiments can be examined by using the concept of design experiment proposed by Montgomery. If the parameters affecting a process that can be restricted have been determined, the levels at which these parameters must be varied have to be identified. Determining what levels of a variable to test requires an in detail understanding of the process, including the minimum, the maximum, and the current value of the parameter.

TABLE I: Parameters Setting

Factor	Parameter Setting
Population Size	45
Cross Over Probability	0.55
Mutation Probability	0.13

If the difference between the minimum and maximum value of a parameter is huge, the values being tested can be more. If the variety of a parameter is small, then less value be able to be tested or the values tested can be closer together. Some tuning parameter used in the proposed GA will be determined by using DOE method. In GA, there are three parameters considered, population size, crossover probability and mutation probability.

A. Data Set

In this section, the instance is solved by using the proposed approach to evaluate the effectiveness of the projected approach. These instances are based on real case problem in process manufacturing company. We handled the medium instance with 15 jobs and 15 machines. Their optimal solution is unidentified. The only company scheduling data is based on LPT rule that they get used to.

B. Computational Study

Therefore, in order to analyze average performance of the GA, each of the instances is solved 25 times with different sets of such GA parameter values as mention in previous section.

TABLE III: Genetic Algorithm Result

	Mean Flow Time	Makespan
Best	6135	17500
Average	6900.6	18027.41
Standard Deviation	528.8	634.211

TABLE II: Job sequence in each machine

Job no	**Machine sequence**
1	1-3-6-11
2	1-4-7-11-9
3	1-4-7-11-9
4	1-3-6-11-9
5	2-5-8-12-9-7-14
6	1-13
7	1-6-8-13
8	1-6-8-13
9	2-9-10-14
10	2-5-9-10-14-15
11	2-5-9-10-14-15
12	2-5-9-10-14-15
13	2-5-9-10-14-15
14	2-5-9-10-14-15
15	2-5-9-10-14-15

The population size is set 45 in different runs. The crossover probability is put to 55% and the mutation probability is put to 13%. We set the maximal generation in 2000 iterations. Followed by, the performance of the GA is evaluated in terms of standard deviation in objective values over 20 runs and average number of objective function essential in obtaining the best solution of each run. The average objective values, standard deviations, and the best result are given inside Table II, respectively

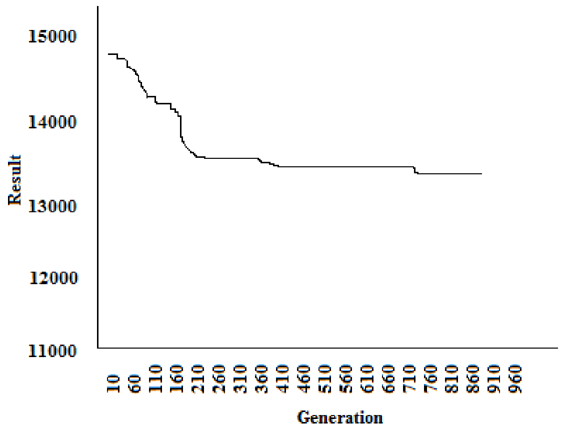

Fig.3. GA Convergence Graph

It is observed in Table II that the proposed GA could obtain the good solution for in every run (out of its 20 runs). However, the standard deviation is still considered huge. Figure 4 shows that the convergence point of the running problem. As declared above, such unavoidable variation in the performance of a metaheuristic is quite common. Since the objective of using the metaheuristic is to improve the company rule result, we have to compare the result with the company actual result which performs using longest processing time approach.

C. Discussions

The optimum objective values obtained by GA for this problem is represent inside Table III row 2, while the result for the actual scheduling is present inside row 3. In this real case problem with 15 jobs, it is observed from the proposed GA could get a better result compared to the company actual schedule.

TABLE III: Comparison with Dispatching Rules

	Mean flow time	Makespan
Genetic Algorithms(GA)	6135	17500
Actual condition-Longest processing time	12870	20420
Shortest processing time	7550	17920
First come and first serve	8750	18180

Compared with the actual solution performed by LPT rule, the proposed GA produces the better solution. It also can be seen that GA outperforms the others procedure in term of solution quality. GA result can obtain 13% improvement of the makespan and 52% of mean flow time. Compare to the other two rules, GA have average gap for makespan is 3% and mean flow time is 33%. The average relative percentage gap obtained by dividing the difference between GA solution and dispatching rule solution by the best known solution based on LPT values.

V. RESULT AND DISCUSSION

The maximum number of generation (G) is selected as the stopping criteria. In this procedure from one generation to the next generation for cross over and mutation is repeated until the maximum number of generation is satisfied. The proposed algorithm is coded in Mat lab. The results are compared with the arrangement of job based on company data using longest processing time rule.

There are many simple dispatching rules (i.e., priority rule) for scheduling. This priority rules are developed to attain a good schedules for number of different objective in different position. They are designed for sequencing many jobs without many effort and time. As the benchmark, another dispatching rule is also performs to solved the problem. Shortest processing time (SPT) and first come first serve (FCFS) rule is chosen as the benchmark rule.

VI. CONCLUSION

In this paper, we proposed genetic algorithm (GA) for solving the job shop scheduling problem, which asks for an arrangement of a sequence of job in certain machine. The proposed GA is investigated on a real case problem, in which found to successful obtain the good solution value for the instances in every run (out of its 20 runs) compare to dispatching rules result.

REFERENCES

[1] El-Bouri, N. Azizi, and S. Zolfaghari,"A comparative study of a new heuristic based on adaptive memory programming and simulated annealing: the case of job shop scheduling," European Journal of Operational Research, vol.177, pp. 1894-1910, March 2007.
[2] J. P. Watson, A. Howe and L. Whitley, "Deconstructing Nowicki and Smutnicki's i-TSAB tabu search algorithm for the job-shop scheduling problem," Computers & Operations Research, vol. 33, pp. 2623-2644, September 2006.
[3] S. Esquivel, S. Ferrero, R. Gallard, C. Salto, H. Alfonso, and M. Schtz, "Enhanced evolutionary algorithms for single and multiobjective optimization in the job shop scheduling problem," Knowledge-Based Systems, vol.15, pp. 13-25, January 2002.
[4] Makoto Asano, Hiroshi Ohta, "A Heuristic for Job Shop Scheduling to Minimize Total Weighted Tardiness", Computers & Industrial Engineering, Vol. 2-4 No 42 pp: 137-147， 2002.
[5] R. Qing and Y. Wang, "A new hybrid genetic algorithm for job shop scheduling problem," Computer &Operations Research, vol. 39, pp.2291-2299, October 2012.
[6] S. French, Sequencing and Scheduling: An Introduction to the Mathematics of the Job Shop, New York, USA: John Willey & Sons Inc., 1982.
[7] M. T. Jensen and T. K. Hansen, "Robust solutions to job shop problems," in Proc. of Congress on Evolutionary Computation, Washington DC, 1999, pp 1138-1144.
[8] D.E.Goldberg, Genetic Algorithms in Search Optimization and Machine Learning, New York, Addison Wesley, 1989, pp. 41
[9] D. J. Hoitomt, P. B. Luh and K. R. Pattipati, "A practical approach to job shop scheduling problems," IEEE Trans. on Robotics and Automation, vol. 9, pp. 1-13**,** February 1993.
[10] T. Bäeck, D. B. Fogel, Z. Michalewicz (Eds.), Handbook of Evolutionary Computation, New York, Bristol: Oxford University Press, Institute of Physics, 1997, ch. F1.5.
[11] A. El-Bouri, N. Azizi, and S. Zolfaghari,"A comparative study of a new heuristic based on adaptive memory programming and simulated annealing: the case of job shop scheduling," European Journal of Operational Research, vol.177, pp. 1894-1910, March 2007.