Flexible Job Shop Scheduling Operation using Genetic Algorithm

Gaurav Kumar
Department of Mechanical Engineering, Sagar Institute of Research & Tech. – Excellence

Trilok Singh Bisoniya
Department of Mechanical Engineering, Sagar Institute of Research & Tech. – Excellence

Abstract - Flexible manufacturing system (FMS) scheduling job shop problems become very complex when it comes to accommodate frequent variations in the part designs of incoming jobs. When a large number of jobs and machines are taken into consider, performance in the Job shop scheduling perform an essential role. In case of job shop scheduling problems, an operation is allowed to be processed by any machine from a given set and the FJSP is an extension of the job shop scheduling problem. The scheduling problem consists of two steps, first is to assign each operation to a machine and the another 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 paper present study of the job shop scheduling using genetic algorithm which is aimed at creating a mathematical model with precedence order of the jobs as constraint. The MATLAB code were used to generate an algorithm for finding the optimal solution. The input parameters were operating time and sequence of operation for the each job in machines provided. The makespan value were used to compare the results.

Key Words: Flexible manufacturing system, Scheduling, Automated Guided Vehicle, Simulation, MATLAB

I. INTRODUCTION

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 routing, processing time 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 proper sequence. Scheduling comes after routing and the jobs may be scheduled based on various parameters such as lowest processing time, most work remaining etc.

Job Shop Production:

In the job shop process flow, the majority of the items prepared oblige a novel set-up and sequencing of methodology. Every 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. no any 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.

A viable program is an assignment of operations to time on a violation without restrictions workshops machine. **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 for jobs and machines is to minimize the makespan i.e. the time distance of the schedule, in which all the operations of every 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.

II. LITEARATURE REVIEW

Brucker [1] and Garey [2] stated for the obtaining solutions to these problems are a difficult task on Job Shop scheduling problems are NP job hard. 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, 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 [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. 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].

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.

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.

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[19].

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.

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.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.

III. METHODOLOGY ADOPTED

The FJSP can be defined as set J = {J_1, J_2 , . . . , J_n } of independent jobs. A job J_i is formed by a sequence Oi_1 , Oi_2 , . . . , Oi_{ni} 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,………, M_m} machines to process these operations. Each operation can be processed on any among given subset of Compatible machines. When each operation can be processed on every available machine there exist a case of total flexibility.

Let $t_{i,j,k}$ be the processing time of operation $O_{i,j}$ when processed on machine M_k and once an operation is started it can not 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.

	***M*1**	***M2***	***M3***	***M*4**
O11	**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

IV. 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 off springs 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:

I. 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.

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

II. 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)

III. 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.

IV. 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.

V.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	**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 300th iteration.

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)

VI.CONCLUSION

This present 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] 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.
[23] 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.
[24] Guinet A., 2000. "Efficiency of reduction of job shop to flow shop problems". European journal of operational research, 125, pp. 469-485.
[25] 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.
[26] 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,
[27] 2007.ICDIM „07 (pp. 138–145).
[28] 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.
[29] 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.
[30] 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.
[31] 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.