A GENETIC ALGORITHM BASED SINGLE MACHINE TARDINESS SCHEDULING PROBLEM

N.S.Mitra

Corresponding author Associate Professor, Production engineering department, Haldia Institute of Technology

A.Mondal

R.Mehta

K.Baitalik

Abstract: This paper presents a new genetic algorithm for the single machine scheduling to minimize the tardiness. Here a code has been developed in C-language for finding the solutions to job shop scheduling problems. The code can be applied to job shop scheduling problem with the following complexity viz. non zero ready times, due dates, job shop or open shop structure, multiple non-identical machines and routing flexibility for jobs, sequence dependent set up times and tooling constraints. The genetic algorithm uses the natural permutation representation of chromosome for encoding simplicity. Heuristic dispatching rules combined with a random method are used to create an initial population for improving the search space, consequently improving searching simplicity. Single row chromosome structure is designed based on working procedure and machine distribution. The relevant cross over and mutation operation is also given.

Keywords: Scheduling, genetic algorithm, optimization, local search.

1. INTRODUCTION:

Scheduling is the allocation of shared resources over time to competing activities. It has been the subject of a significant amount of literature in the operation research field. In the case of shop floor activities like production it has tremendous applicability. Every manufacturing company today is facing tremendous competition to satisfy customers. The first thing that a customer expects from its suppliers is to make the reasonable and the reliable delivery commitments. To keep the commitment requires analysis of many factors including checking of load, finding earliest time of start, availability of factors of production, process planning and modifying the process plans. To perform all these functions production planning is a must. Out of various production planning and control functions routing and scheduling are two important functions. Various techniques of scheduling are job shop scheduling, flow shop scheduling, assignment technique, network analysis and critical ratio scheduling. In a flow shop all orders follow a single routing whereas in job shop scheduling each order is unique with a unique routing. Job shop scheduling is highly complicated with no repetitive pattern. From the beginning various rules has been used in scheduling the machines. Here emphasis has been on investigating the machine scheduling problems where jobs represent activities and machine represent resources; each machine can process one job at a time. A number of jobs are to be processed on a number of machines, where each job has a technological sequence of machines to be processed. The processing of a job on a machine is called operation. An operation requires the exclusive use of a machine for an uninterrupted duration called processing time. A schedule is a set of completion times for each operation. The time required to complete all the jobs is called makespan. The objective when solving this general problem is to determine the schedule which minimizes the makespan. To solve the job shop scheduling problem besides exhaustive search algorithms based on branch and bound methods, several approximation algorithms have been developed. The most popular ones in practice are based on priority rules viz. First Come First Served (FCFS), First In System First Served (FISFS), Earliest Due Date (EDD), Shortest Processing Time (SPT), Weighted Shortest Processing Time (WSPT), Slack Time, Critical Ratio Rule, Hodgson's Algorithm etc. The method of formulation of job shop scheduling problem (JSSP) may be by Disjunctive Graph, by an active schedule generation, by heuristic cross over, by GT cross over and by genetic enumeration. A more sophisticated method called shifting bottleneck has been proved to be very successful. Additional stochastic approaches like simulated annealing, tabusearch and genetic algorithm have recently applied with good success.

Single machine scheduling problem problems can provide help and insight into resolving, understanding, managing and modeling more complex multi-machine scheduling problems and it has received much attention in literature. The single machine total weighted tardiness problem has been solved by branch and bound algorithm, enumerative algorithm and dynamic programming algorithm [1,2,3,4,5] to generate exact solutions that are guaranteed to be optimal. Branch and bound problems are limited by computational time and dynamic programming problems are limited by storage requirements when number of jobs is more than 50 [6]. Thereafter the problem has been extensively studied by heuristics – solution procedures that generate good or even optimal solutions, but do not guarantee optimality. These heuristics include heuristics dispatching rules [7,8,9] and local search heuristics. As there is no single best dispatching rule for all problem environments, in recent years, much attention has been devoted to local search heuristics [10,11,12]. This local search heuristics mainly include neighbourhood search methods such as simulated annealing, threshold accepting, tabu search [3,6,13,14] and genetic algorithm (GA) [6,15,16]. It has been proved that binary encoded GA performs very well and requires comparatively little computation time.

In this paper, we present a genetic algorithm which helps to schedule single machine on a number of jobs and a number of tools which is unique in comparison with other research studies. As input data job number, tool requirement, ready time, due time, processing time and number of machines are taken here and parameters are chosen as total population size, maximum number of generations, cross over probabilities, immigration fraction etc. and finally genetic algorithm has been applied to find out tardiness in a single machine with job and tool interchange.

2. PROBLEM DEFINITION:

The single machine total weighted tardiness problem is defined as follows. Let us consider n jobs to be processed without interruption on a single machine that can handle only one job at a time. Each job j, available for processing at time zero, has positive processing time p_j, a positive weight w_j and a positive due date d_j. For a given sequence of jobs, the tardiness of job j is defined as –

$$Tj = \max \{0, Cj - dj\}.$$

where Cj is the completion time of job j. The objective of the tardiness problem is to find a processing order of all the jobs. This order is a schedule that minimizes the sum of the weighted tardiness of all the jobs.

Sum of the weighted tardiness = $\sum_{j=1}^{n} w_j T_j$

Then the project is to schedule n jobs on a single machine to minimize the sum of weighted tardiness of all the jobs.

3. GENETIC ALGORITHM (GA):-

Genetic algorithm (GA) were originally proposed by John.H.Holland. They are search algorithms that explore a solution space and mimic the biological evolution process. There are many GA implementations successfully applied to a great variety of problems [17]. The procedure for genetic algorithm is divided into the following steps:

(i) Initialization:- Create an initial population. This population is usually randomly generated and can be any desired size, from only a few individuals to thousands.

(ii) Evaluation:- Each member of the population is then evaluated and fitness is calculated for that individual. The fitness value is calculated by how well it fits our desired requirements.

(iii) Selection:- Natural selection of some chromosomes (parents) in the population to generate new members (children).

(iv) Cross over:- Genetic operators applied to those chromosomes whose role is to create new members in the populations by crossing the genes of two chromosomes (cross over operator).

(iv) Mutation:- Natural selection of the member who will survive. It is done by modifying the genes of one chromosome (mutation operator).

4. PARAMETER SELECTION:-

In this study, maximum problem and population sizes are given. They are used for dimensioning arrays, viz maximum number of jobs, maximum number of machines, maximum population size, maximum number of tools. If these limits are violated, the execution of the problem is violated generating an error message. Some of the parameters are specifically problem related viz. total population size, maximum number of generations that GA will run, the number of best solutions from each sub-population (number of clones, obtained by multiplying the sub-population size by clone fraction and a value of 0.04 to 0.05 for clone fraction work very well) etc. A cross over probability is used in cross over operator. A value of 0.7 for cross over probability performs well. The other problem specific parameters are fraction of each sub population that are immigrant, the fraction of immigrants that are not strongly biased, biasing weight assigned to ready time, biasing weight assigned to the due time, fraction of each subpopulation that are initially strongly biased, number of subpopulations, the fraction of each subpopulations to communicate between subpopulations, the number of generations between occurrences of communication between subpopulations, initial random number seed, ending random number seed and the value of lower bound for the problem.

5. METHODOLOGY:-

In the current problem the data are read from two input files. The GA is run for a number of times based on the values for start seed and end seed. The genes of the chromosomes selected includes job number (i), tool requirement (t_j), ready time (r_i), due time (d_i), processing time (p_i), number of machines that can process the jobs (nm_i), the list of machine numbers that can process the job (m_{i1} and m_{i2}). A fraction first initializes the chromosome, evaluates the fitness and performs the necessary ranking. The same function also determines the bias values from their weights. A check is there where the bias values exceed their feasible range that would led to a negative allele value. A second function included in the previous function separately evaluates the three types of chromosomes, which consists of job number, machine number and tool. Random key values for the job are sorted and a semi active schedule is constructed based on the sorted order of the random key. The job number and the machine assignment are determined. Then the starting time and completion time of each job is calculated. At last the tardiness is determined. The population and the sub-population are ranked. A separate function consisting of the total population size, maximum number of generations, clone fraction, cross over probability, immigration fraction, fraction of immigrants that are not strongly biased, biasing weights assigned to ready times, biasing weight assigned to due time, the fraction of sub-population that will be strongly biased initially, number of sub-populations, fraction of each sub-populations to communicate between sub-populations, the number of generations between occurrences of communication between subpopulations, initial random number seed, final random number seed and the lower bound value for minimization problem runs the GA until a stopping criterion is reached. A fourth function which is described in the third one shares chromosomes between sub-populations. A fifth function inside the third function performs the generational change for GA by cross over. The solutions are evaluated for fitness and the best solution is written. Finally a sixth function prints the solution to the screen. A code in C-language in Dev C++ environment is developed using this methodology which was run with realistic data to give the required results.

6. COMPUTATIONAL RESULTS:-

For the above problem the dataset used for 10 jobs and 3 tools to be scheduled for a single machine are given in Table 1. The setup times for tool assignments is presented in Table 2. S_{kl} represent set up time for switching from tool type k to tool type l for k,l = 1 ...... T (where T is total number of tools). The values of parameters selected for the genetic algorithm program are as follows:—

(a) total population size =100
(b) maximum number of generations =250
(c) proportion of best solution in sub-population (clone fraction) = 0.060
(d) cross over probability = 0.70
(e) immigration fraction of sub-population = 0.040
(f) fraction of immigrants that are not strongly biased = 0.60
(g) biasing weight assigned to ready time = 1.00
(h) biasing weight assigned to due time = 0.90
(i) fraction of each population that are strongly biased = 0.70
(j) number of sub populations = 2
(k) fraction of each sub-populations to communicate between sub-populations=0.04

(l) number of generations between occurrence of communication between sub-populations= 20

(m) initial random number seed = 1

(n) final random number seed = 11

(o) lower bound value for minimization = 11.36

The best solution obtained for the generation numbers is 1.13. The optimum sequence obtained in terms of job numbers is 4-7-9-1-2-3-5-6-8-10. The total tardiness (objective value) was determined to be 60.59 hours. The final results are shown in Table 3.

7. CONCLUSION:-

This study resulted in documentation and user manual in C-code developed for finding the solution to job shop scheduling problems. The code can be applied to job shop scheduling problems with the complexities like non zero ready time, due time, open shop structure, multiple non-identical machines, routing flexibility for jobs, sequence dependent setup times, tooling constraints etc. The algorithm developed may be effective in many practical situations where the user can change slightly the conditions and use the values. The schedules obtained have the makespan value near to optimal. Single row chromosome structure is designed based on working procedure and machine distribution. The relevant cross over and mutation operation is also given. Finally the computational results show that GA can obtain better solution. Although GA has several bottlenecks, it provides a flexible framework for evolutionary computation and it can handle varieties of objective functions and constraints.

8. REFERENCES:-

1. Picard, J.C and,Queyranne, M.(1978).The Time-Dependent Travelling Salesman Problem and its Application for the Tardiness Problem in One Machine Scheduling, Operations Research, 26(1), 86-110.
2. Jouglet,A.,Baptiste,P. and Carlier,J.(2002,Oct 6-9),Exact Procedure for Single Machine Total Cost Scheduling, In Proceedings of IEEE International Conference on Systems, Man and Cybernetics.
3. Potts, C.N. and Wassenhove,L.N.V.(1985).A Branch and Bound Algorithm for the Total Tardiness Problems, Operations Research, 33(2),363-377.
4. Shwimer,J.(1972). On the n-job, One-machine, Sequence-independent Scheduling Problem with Tardiness Penalties:A Branch Bound Solution, Management Science, 18(6), B-301-B-313.
5. Schrage, L. and Baker, K.R.(1978). Dynamic Programming Solution of Sequencing Problem with Precedence Constraints,Operations Research, 26, 444-449.
6. Crauwels,H.A.J., Potts, C.N. and Wassenhove,L.N.V.(1998). Local Search Heuristics for the Single Machine Total Weighted Tardiness Scheduling Problem, INFORMS Journal on Computing, 10(3), 341-350.
7. Akturk, M.S.(1999). A New Dominance Rule for the Total Weighted Tardiness Problem, Production Planning and Control, 10(2), 138-149.
8. Kan, A.H.G.R.,Lagweg,B.J. and Lenstra,J.K.(1975). Minimizing Total Costs in One Machine Scheduling, Operations Research, 23, 908-927.

9. Ying,K.C. and Liao, C.J.,(2003). An Ant Colony Systems Approach for Scheduling Problems,Production Planning and Control,14(1), 68-75.

10. Huegler,P.A. and Vasco, F.J.(1997). A Performance Comparison of Heuristics for the Total Weighted Tardiness Problem, Computers and Industrial Engineering, 32(4),753-767.

11. Maheshwaran,R. and Ponnambalam,S.G.(2003). An Investigation on Single Machine Total Weighted Tardiness Scheduling Problems, International Journal of Advanced Manufacturing Technology, 22(3-4), 243-248.

12. Sule,D.R.(1994). Heuristic Method for a Single Machine Scheduling Problem, International Journal for Industrial Engineering,1(2),167-174.

13. Grosso,A.,Croce,F.D. and Tadei,A.(2004). An Enhanced Dynasearch Neighbourhood for the Single Machine Total Weighted Tardiness Scheduling Problem, Operations Research Letters, 32(1),68-72.

14. Matsuo, H.,Suh,C.J. and Sullivan,R.S.(1989). A Controlled Search Simulated Annealing Method for the Single Machine Weighted Tardiness Problem, Annals of Operations Research, 21, 85-108.

15. Avoi,S.,Akturk,M.S. and Storer, R.H.(2003).A Problem Space Algorithm for Single Machine Weighted Tardiness Problems, IIE transactions, 35, 479-486.

16. Madureira,A.,Ramos,C. and Silva,S,D.C.(2001). A GA based Scheduling System for Dynamic Single Machine Problem, In Proceedings of the 4th IEEE International Symposium on Assembly and Task Planning, 262-267, Soft Research Park, Fukuoka,Japan.

17. Goldberg, D.E.(1989), Genetic Algorithms in Search,Optimization and Machine Learning, Massachusetts, Addision-Wesley.

Table 1 (No of jobs =10, no of machines =1, no of tools = 3)

Job.No. (i)	No.of tools reqd,(t_i)	Ready time (r_i),(hrs)	Due time (d_i),(hrs)	Processing time (p_i),(hrs)	No.of machines (nm_i)	List of machine numbers (m_{i_k}1)	Lisdt of machine numbers (m_{i_k}2)
1	1	4.68	2.00	1.12	2	1	2
2	1	1.25	2.50	0.61	2	1	2
3	2	1.50	2.28	1.91	2	1	2
4	2	1.75	4.12	0.43	2	1	2
5	2	1.99	5.13	0.77	2	1	2
6	3	2.24	3.89	1.05	2	1	2
7	3	2.49	4.27	0.49	2	1	2
8	3	2.74	6.74	2.31	2	1	2
9	2	3.12	5.54	1.10	2	1	2
10	1	2.78	4.11	0.41	2	1	2

Table 2 (Set up time for tool change)

$S_{1,1}$=0.00	$S_{1,2}$=0.68	$S_{1,3}$= 1.42
$S_{2,1}$=0.75	$S_{2,2}$=0.00	$S_{2,3}$=0.99
$S_{3,1}$=1.81	$S_{3,2}$=1.12	$S_{3,3}$=0.00

Table 3 (Final Solution)

Job number	Completion time (hrs.)	Tardiness for this job (hrs.)	Total tardines (hrs.)
4	2.18	0.00	0.00
7	3.66	0.00	0.00
9	5.88	0.34	0.34
1	7.75	5.75	6.09
2	8.36	5.86	11.95
3	10.95	8.67	20.62
5	11.72	6.59	27.21
6	13.76	9.87	37.08
8	16.07	9.33	46.41
10	18.29	14.18	60.59