Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 6 (7): 1586-1590, 2006
ISSN 1812-5654

Application of Genetic Algorithms and Rules in the Scheduling of Flexible Job Shops

[1,2]Shahid Ikramullah Butt and [1]Sun Hou-Fang
[1]Department of Manufacturing and Automation (3rd School), Beijing Institute of Technology,
5 South Zhongguancun Street, Haidian District, Beijing 100081, China
[2]National University of Sciences and Technology, Rawalpindi, Pakistan

Abstract: Different methods have been in use for small and large size problems to find out a specific objective function or performance measure such as Makespan. The problem presented in the research is a case study of BIT (Beijing Institute of Technology) Training Workshop, which is the best example of a Flexible Job Shop, considered a special case of job shop scheduling problem. Genetic algorithm is employed in combination with the scheduling rules to solve the scheduling problem with an option of recirculation. Development of software is done to do an offline scheduling. Results of scheduling software are presented for the flexible job shop environment using MT10 and MT20 as benchmark problems. Comparison with LEKIN® software results is also done with the developed software to show that this is practical software and can be used successfully at BIT Training Workshop.

Key words: Flexible job shop, scheduling, genetic algorithms, scheduling rules

INTRODUCTION

One of the important elements of the production systems is scheduling. Many other shop activities are based on scheduling. Various system performance measures can be optimized by properly planning and timing of shop floor activities. There are two key elements in any scheduling system: schedule generation and revisions (monitoring and updating the schedule). The first element that acts as a predictive mechanism determines planned start and completion times of operations of the jobs. The second element which is viewed as the reactive part of the system monitors the execution of the schedule and copes with unexpected events i.e., machine breakdowns, tool failures, order cancellation, due date changes, etc. (Sabuncuoglu and Bayiz, 2000). This study concentrates only on the first element i.e., schedule generation.

The objective in this study is to minimize the completion time of the last job on its last machine i.e., Makespan. The problem is known to be hard when there are more than two jobs or two machines (Lagewag *et al*., 1977). Large sized problems can be solved using dispatching rules, which are used to choose a job to be loaded on a machine from the queue. The static job shop problems assume that all jobs are available at the beginning of the planning period and that the processing times and set up times are deterministic. In such cases, the set up times are added to the processing times. The assumptions in static job shop problems are discussed in Baker (1974) and Kumar and Srinivasan (1996).

THE CASE PROBLEM

The problem in consideration is taken from the Beijing Institute of Technology Training Workshop, which will be used for the training purpose of the undergraduate students. This workshop is a classical example of the flexible job shop containing three work centers.

As BIT Training Workshop is involved in training under-graduate students on CNC machines of three work centers. Different jobs will be allocated to different students as individuals or in groups with the known process plans. Thus, there will be a job of allocating different machines to these students based on their jobs, which is primarily an allocation of jobs to different machines. Thus, this becomes a scheduling problem of n jobs on m number of machines within work centers. A schematic diagram of the BIT workshop is shown in the Fig 1.

Following are the assumptions taken for the said problem.

- Preemptions are not allowed.
- Recirculation will be allowed.

Corresponding Author: Shahid Ikramullah Butt, Department of Manufacturing and Automation (3rd School), Beijing Institute of Technology, 5 South Zhongguancun Street, Haidian District, Beijing 100081, China

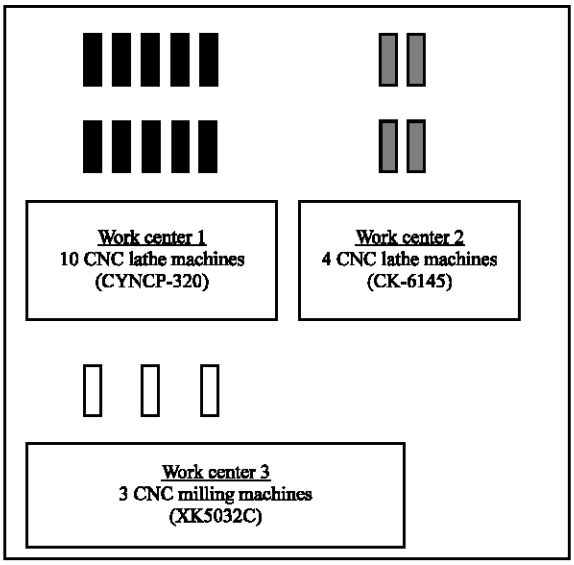

Fig. 1: Schematic diagram of the BIT training workshop

- Any job can be processed on only one machine within a work center.
- Any machine (within a work center) can process any job.
- Machines never breakdown and are available throughout the scheduling period.
- Number of jobs and number of machines are fixed.
- Setup times are zero or are added in the processing times.

USE OF GA AND RULES

Heuristic rules also called scheduling rules, dispatching rules or priority rules in scheduling literature, are used to choose assignment of jobs from a group of jobs waiting to be processed on a particular machine. These rules may choose to schedule the job with the Shortest Processing Time (SPT), Largest Processing Time (LPT) or the least time remaining before the Due Date. The scheduling rules do not examine the outcome of a choice but only choose to start an operation on a given machine according to some predefined quantitative measure.

Heuristics generally do not provide an optimal solution for complex problems but there are many dispatching rules, which may be equally suitable. Panwalkar and Iskander (1977) presented a survey of more than 100 heuristic rules. A broad understanding of the advantages and drawbacks of heuristic rules can be found in Blackstone *et al.* (1982). Heuristic rules have strong advantages in that they are easy to understand, easy to apply and require relatively little computer time. The primary disadvantage is that they alone cannot hope for an optimal solution (David, 1996).

Because of the above mentioned disadvantages, scheduling rules are combined with the genetic algorithm. Genetic Algorithms (GAs) are general purpose optimization algorithms with a probabilistic component that provides a means for searching poorly understood and irregular spaces. The first rigorous description of the genetic algorithms process was given by Holland in 1960 (Goldberg, 1989).

A typical algorithm consists of the following steps:

- A number or population of guesses of the solutions to the problems.
- A way of calculating how good or bad the individual solutions are within the population.
- A method for mixing fragments of the better solutions to form new on average or even better solutions.
- A mutation operator to avoid permanent loss of diversity within the solutions.

SCHEDULING ALGORITHMS

Flexible job shop is a special case of job shop scheduling problem. Unlike job shop scheduling, flexible job shop has more than one work centers and a specific operation of a job can be processed by the work center and any machine in that work center can do that operation. The problem situation in hand is to assign operation to a work center and then to any machine in a way that maximum completion time of all operations (Makespan) is minimized.

In this study the work centers are assigned operations of the jobs, such that each gene of the chromosome represents the machine numbers according to the assignment of the job operations to them. For example, in case of a two jobs, two work centers problem where each work center contains three machines of similar type, situation could be like in Table 1 and 2.

Table 1: An example of two work centers and two jobs

Work centers	Work center 1			Work center 2		
Identical machines in WC1 and WC2	WC 1-1	WC 1-2	WC 1-3	WC 2-1	WC 2-2	WC 2-3
No. of jobs	Job 1			Job 2		
Operations of each job	Job1-1		Job1-2	Job2-1		Job2-2

Table 2: Assignment of real numbers to machines

Machines	WC 1-1	WC 1-2	WC 1-3	WC 2-1	WC 2-2	WC 2-3
Real numbers	1	2	3	4	5	6

The size of chromosome will be number of Work Centers (WCs) multiplied by the Number of Jobs (Ji). In the above situation chromosome will consist of four (4) genes (e.g., 1, 5, 1, 6 or 2, 6, 2, 6 etc.). The total length of Chromosome is taken as;

Chromosome = (Chromosome length-1)

Each gene represents a real numbers for the representation and binary representation is not taken into account as it can create complexity during encoding and decoding process. The machines in each work center are chosen according to their availability and job routing. In this way, the jobs 1, 2, 3 ... are encoded until the operations of all jobs are checked through in turn. The field of the gene is restricted within the Work Center that can process current operation.

Crossover: Two parent chromosomes are chosen randomly from the sample of a selected population size (e.g., 50 or 100) for a single point crossover. Moreover, all the crossover chromosomes remain within the population.

The pairs of individuals selected undergo crossover with probability Pc. A random number R is generated in the range 0-1 and the individual undergo crossover if and only if R<Pc then crossover operation is executed as follows:

$$Pc = \begin{cases} K_1 & \text{where } f_1 \geq f_{avg} \\ 0 & \text{otherwise} \end{cases}$$

Where f_1 is the higher fitness value of the two parents and f_{avg} is the average fitness value of the population. Different values of K_1 were used to find out a near optimal value of the Makespan.

Mutation: After calculating fitness and average values of all the individuals in the population mutation operation is conducted with the probability P_m. Again, a random number R is generated in the range 0-1 and the individual undergoes mutation if and only if $R<P_m$ the mutation operation will take place as follows:

$$P_m = \begin{cases} K_2 & \text{where } f_2 \geq f_{avg} \\ 0 & \text{otherwise} \end{cases}$$

Where f_2 is the fitness value of the mutating chromosome and f_{avg} is the average fitness value of the population. Different values of K_2 can be tried to find out a near optimal value of the Makespan.

Selection: The selection algorithm by the use of roulette wheel selection is as follows:

- Fitness of all the population members is summed up. We can call it f_{sum}.
- A random number R is chosen between 0 and f_{sum}.
- Fitness of the population members is added together one by one until the value of random number R is reached in between the two values of the f_{sum}. The last individual added is the selected individual and copy is added to the new population, which keeps on increasing with the new generations.
- This selection mechanism is continued until N individuals have been selected.

Software is developed on the pattern of LEKIN® that can work for different shop floor situations using heuristics (e.g., Shifting Bottleneck) and rules (e.g., SPT, LPT etc). LEKIN® software does not take into account recirculation of the job, whereas recirculation feature is added in the developed software. Software developed currently combines GA along with the rules namely SPT and LPT with an option to add more rules as required.

RESULTS

Flexible Job Shop (FJS) has more than one instance of a machine type. Thus, the problem becomes that of assigning each operation to the appropriate machine and sequencing the operations on each machine. The major contribution in the area is by Chambers and Barnes, where they developed an adaptive tabu search method to solve the FJS problems (Barnes and Chambers, 1995; (Chambers and Barnes, 1996a, b, c). The test problem was generated based on the classical job shop problem instances. The chosen instances are MT10 and MT20 with seven replications of each instance.

Table 3 and 4 summarize the results for MT10 and MT20 class of FJS problems, respectively. The first four instances were created based on the greatest, second greatest, third greatest and fourth greatest Cumulative Processing Times (CPT). In the last three instances, different machines were replicated based on the respective critical path. Processing times for operations on replicated machines are assumed identical to the original.

The results of MT10 and MT20 of job shop environment in comparison with flexible job shop environment are summarized in the Table 5.

All values of the Makespan were found with the following values of the constants.

$K_1 = 0.8$

$K_2 = 0.3$

Table 3: GA+Rules search results using MT10

Instances	Problem name	Specifications N	Specifications M	Best flexible dispatching solution*	Best know C_{max} in JSSP	In FJSP, using LEKIN	In FJSP, using (GA and rules algo) (C_{max})	(%) Gap (With flexible dispatching solution)	Comparison
1	p1, p1, p1	10	10	1023	930	1247	1058	3.42	Results of
2	p1, p2	10	10	1000	930	1339	1004	0.40	GA+ rule
3	p1, p2, p3	10	10	1008	930	1209	916	9.12 Better	better than
4	p1, p2, p3, p4	10	10	1008 (Using value of instance # 3)	930	1095	904	10.31 Better	LEKIN®
5	c1	10	10	1007	930	1355	1000	0.69 Better	
6	c1, c2	10	10	980	930	1284	982	0.20 Better	
7	c1, c2, c3	10	10	980 (Using value of instance # 6)	930	1264	932	4.89 Better	

*Chambers and Barnes (1996c)

Table 4: GA+Rules search results using MT20

Instances	Problem name	Specifications N	Specifications M	Best know C_{max} in JSSP	In FJSP, using LEKIN® (C_{max})	In FJSP, using (GA and rules Algo) C_{max}	(%) Gap of GA and rules with JSSP	Comparison
1	p1, p1, p1	20	5	1165	1504	1215	4.11	Results of
2	p1, p2	20	5	1165	1452	1174	0.76	GA+ rules
3	p1, p2, p3	20	5	1165	1338	1090	6.88 Better	better than
4	p1, p2, p3,p4	20	5	1165	1298	1084	7.47 Better	LEKIN®
5	c1	20	5	1165	1560	1207	3.47	
6	c1, c2	20	5	1165	1621	1176	0.93	
7	c1, c2, c3	20	5	1165	1455	1178	1.10	

Table 5: Best of the GA and rules algorithm

Problem name	Specifications N	Specifications M	In JSSP best known (C_{max})	In FJSP, using LEKIN®	In FJSP, best of (GA and rules Algo) (C_{max})
MT10	10	10	930	1095	904
MT20	20	5	1165	1298	1084

These values of constants were chosen after trying some other values of constants.

Table 3 and 4 show a satisfactory result of the scheduling software with either better or close values of the Makespan with other standard values. These tables also show that this software can be used satisfactorily in any FJS environment close to the BIT Workshop.

Comparison with LEKIN®(Pinedo 2001) is also shown in Table 3 and 4. These tables show better results of GA and rules algorithm than LEKIN® for the same instances.

CONCLUSION AND FUTURE DIRECTION

Table 3 and 4 present results for the constructed flexible problems. Second column indicates the machine replication model, as described above. Column 5 in Table 5 show the Makespan best achieved by the GA and rules algorithm. The analysis of the results reveals that the best Makespan is achieved when the flexibility level is increased by the replication of machines. Replication of machines with maximum processing times show a better result than replication of machines on the critical path in MT10 and MT20.

This software can be extended to add the rescheduling feature with the machine breakdown. Rescheduling is one of the important features, which can be incorporated with the option of breakdown as this is practical situation which can occur at any shop floor. With the presence of this feature, the management is able to make any decision before time in the case of any machine breakdown after a fixed interval of time.

REFERENCES

Baker, K.R., 1974. Introduction to Sequencing and Scheduling, Wiley, New York.

Barnes, J.W. and J.B. Chambers, 1995. Solving the job shop scheduling problem using tabu search. IIE Transactions, 27: 257-263.

Blackstone, J.H. Jr., T.D. Philips and G.L. Hogg, 1982. A state-of-the-art survey of dispatching rules for manufacturing job shop operations. Intl. J. Prod. Res., 20: 27-45.

Chambers, J.B. and J.W. Barnes, 1996a. New tabu search results for the job shop scheduling problem. Technical Report Series ORP 96-06. Graduate Program in Operations Research and Industrial Engineering, The University of Texas at Austin.

Chambers, J.B. and J.W. Barnes, 1996b. Flexible job shop scheduling by tabu search. Technical Report Series ORP 96-09. Graduate Program in Operations Research and Industrial Engineering, The University of Texas at Austin.

Chambers, J.B. and J.W. Barnes, 1996c. Tabu search for the flexible-routing job shop problem. Technical Report Series ORP 96-10, Graduate Program in Operations Research and Industrial Engineering, The University of Texas at Austin.

David, W.S., 1996. A survey of approaches to the job shop scheduling problem. Proceedings of the Twenty-Eighth Southeastern Symposium on System Theory, 31: 396-400.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley.

Kumar, N.S.H. and G. Srinivasan, 1996. A genetic algorithm for job shop scheduling-A case study. Computers in Industry, 31: 155-160.

Lagewag, B.J., J.K. Lenstra and R. Kan, 1977. Job shop scheduling by implicit enumeration. Manage. Sci., 24: 441-450

Michael P., 2001. Scheduling Theory, Algorithms and Systems. 2nd Edn., Integre Technical Publishing.

Panwalkar, S.S. and W. Iskander, 1977. A survey of scheduling rules. Operat. Res., 25: 45-61.

Sabuncuoglu, I. and M. Bayiz, 2000. Analysis of reactive scheduling problems in a job shop environment. Eur. J. Operat. Res., 126: 567-586.