Journal of Theoretical and Applied Information Technology
31st October 2012. Vol. 44 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A FLEXIBLE JOB SHOP ONLINE SCHEDULING APPROACH BASED ON PROCESS-TREE

XIANGDE LIU, GENBAO ZHANG
Department of mechanical engineering, Chongqing University, Chongqing 400030, China

ABSTRACT

The flexible job shop scheduling problem (FJSP) is one of the most difficult NP-hard combinatorial optimization problems. It is extremely difficult to solve the FJSP with the disturbances of manufacturing environment, which is always regarded as the flexible job shop online scheduling problem. This paper proposes a new method based on Process-Tree to solve this type of scheduling problem. Process sequence of a job can always be described as one of the three types of Process-Trees given in this paper. To manufacture a job, assignment of operations to machines is determined by considering the Process-Tree, together with the status of job shop and appropriateness of machines and auxiliary device for producing the job's features. In this paper dispatching rules are used to generate the scheduling, and simulation results reveal that the presented approach performs well in flexible job shop online scheduling problem.

Keywords: *Process-Tree, Online Scheduling, Flexible Job Shop, Dispatching Rule*

1. INTRODUCTION

Scheduling may broadly be defined as the allocation of resources to tasks over time in such a way that a predefined performance measure is optimized. From the viewpoint of production scheduling, the resources and tasks are commonly referred to as machines and jobs, and the commonly used performance measures are the completion times of jobs. This problem has been extensively researched in the past five decades, and a considerable amount of models and algorithms have been built [1, 2, 3]. The flexible job shop scheduling problem (FJSP) is an extension of JSP in that it incorporates the routing flexibility. Due to complications resulting from the addition of routing flexibility to the already difficult JSP, researchers have focused on new approaches to solve the FJSP [4, 5]. The flexible job shop online scheduling is an extension of FJSP, in which all related tasks information can not be pre-determined at the decision point, and which fully represents the dynamic stochastic characteristic of modern scheduling. In online scheduling, decisions have to be made based only on information regarding the jobs that already have been released and not on information regarding jobs still to be released in the future. In recent years, many researches have been done on these manufacturing problems.

Research on scheduling has been primarily focused on the construction of efficient algorithms to solve different types of scheduling problems: flow shop, job shop, scheduling on parallel machines, and so on. The integration of the scheduling function with other manufacturing functions, has not received adequate attention. Only until recent years, researchers have investigated the advantages to integrate scheduling with different manufacturing functions [6, 8], one of those is to schedule with alternative process plans.

In this paper, flexible job shop online scheduling is taken into consideration. By real-time sampling workshop state information, dispatching rules are dynamically exploited to determine the next operations executed on machines. The remainder of this paper is organized as follows. In section 2, a scheduling system and scheduling algorithm is introduced. The simulation of an industrial case is presented in section 3, conclusion and outlook of this method is put forward in section 4.

2. SCHEDULING SYSTEM DESCRIPTION

Analyzing a scheduling problem and developing a procedure for dealing with the problem on a regular basis is in the real world only part of the system implementation. The procedure has to be embedded in a system that enables the scheduler to actually apply it. The scheduling system has to be incorporated into information system of the plant. The first section gives a general structure of the scheduling system. The second section deals with definitions and notations related to scheduling

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

algorithm. The third section describes the scheduling algorithm based on Process-Tree.

2.1 System Architecture

In manufacturing, the scheduling function has to interact with other modules within the plant. For a flexible job shop online scheduling, the scheduler must interact with the shop floor control system to receive data information from job shop, which concerns availability of machines, progress of jobs, and so on. In opposite direction, operation sequences generated by scheduling algorithm will be changed into operation instructions, and which then is sent to shop floor control system to instruct machines and processing equipments. At the same time, shop floor data collection system provides job shop environment information for scheduling decision in real time. In addition, a schedule generation module also involves the formulation of a suitable model and the formulation of objective functions, constraints and rules, as well as heuristics and algorithms (see Figure 1).

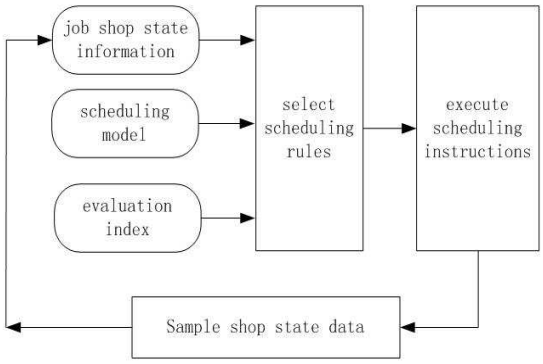

Figure 1: Scheduling System Architecture

In this paper, dispatching rules are used in scheduling algorithm. Because of the complexity of scheduling problems, dispatching rules play an important role in scheduling system. The decision as to which job is to be loaded on a machine, when machines become free, is normally made by the help of dispatching rules. Over the years, many dispatching rules have been proposed by many researchers, and many different rules have been studied in the literature. A simple one, often used in practice, is the Service in Random Order (SIRO) rule. Under this rule, no attempt is made to optimize anything. Another rule often used is the First Come First Served, which is equivalent to the Earliest Release Date first (ERD) rule. This rule attempts to equalize the waiting times of the jobs. Some rules are only applicable under given conditions in certain machine environments.

Dispatching rules are useful when one attempts to find a reasonably good schedule with regard to a single objective, the total completion time or the maximum lateness. However, in real life, objectives are often more complicated. More elaborate dispatching rules that take into account several different parameters can address more complicated objective functions. Some of these elaborate rules are basically a combination of a number of the elementary dispatching rules listed earlier, and which are referred to as composite dispatching rules.

In the scheduling system (see Figure1), scheduler choose dispatching rules in two ways, which are artificial assignment and automatic assignment. The way of artificial assignment usually relies on human experience, the user must carefully consider many cases such as shop model and evaluation index and manufacture information from shop floor, to choose dispatching rules. Obviously, the way of artificially appointing scheduling rules is relatively simple. Furthermore , the way of automatic assignment is more complex than the previous one . Automatic assignment method always depends on a knowledge base, which can be gotten by learning (see Figure 2). New research initiatives are focusing on the design and development of learning mechanisms that enable scheduling systems in daily use to improve their schedule generation capabilities. A number of machine learning methods have been studied with regard to their applicability to scheduling. These methods can be categorized as follows: rot learning, case based reasoning, induction methods and neural networks, classifier systems. The simulation learning system for automatic assignment of dispatching is described as Figure 2.

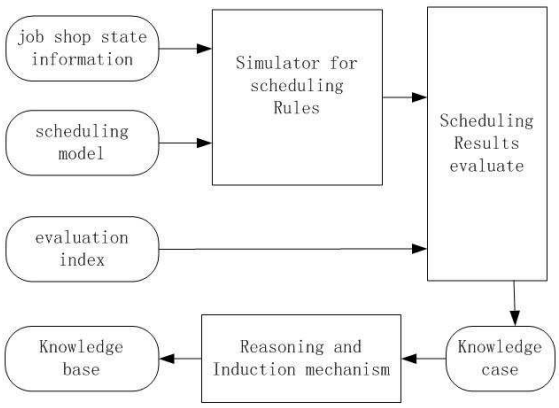

Figure 2: Learning Mechanism Of Dispatching Rule

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2.2 Notation And Terminology

Scheduling deals with the allocation of scarce resources to tasks over time, it is a decision-making process that plays an important role in most manufacturing and production systems. For the sake of convenience, several assumptions are made in this paper.

(i) A machine can only process one job at a time and a job can only be processed by one machine at a time.

(ii) The process plan of each job can be depicted as a Process-Tree.

(iii) Each job has only one feasible process plan.

(iv) Transporting times are ignored.

Following terminology and notations are adopted.

P_j machine j of the shop.

Y_i operation i of the job.

Y_i^{m-n} operation i of the job m-n.

$Y_i^{P_j}$ operation i can be processed by machine j.

$t_{Yi}^{P_j}$ the processing time of operation i processed by machine j.

i-p_j machine j put into operation at the sampling period i.

(i) Process-Tree

Process plan of a job is an ordered list of operations with precedence constraint. A tree structure can characterize this precedence constraint relationship(see Figure 3). In computer science, tree is a widely used data structure that simulates a hierarchical tree structure with a set of linked nodes. A tree can be defined recursively (locally) as a collection of nodes (starting at a root node), where each node is a data structure consisting of a value, together with a list of nodes (the branch). Operations in a Process-Tree can correspond with the root node and branch nodes and leaf nodes of a tree structure , so a Process-Tree consists of root operation and child operation and leaf operations . When a job is processed, leaf operations will firstly be processed , the completed leaf will fall off from the Process-Tree. When the leaf operations being subordinate to a branch operation have totally fallen off, the branch operation then change into a leaf operation . When all of leaf operations have fallen off, that means the job has been completed.

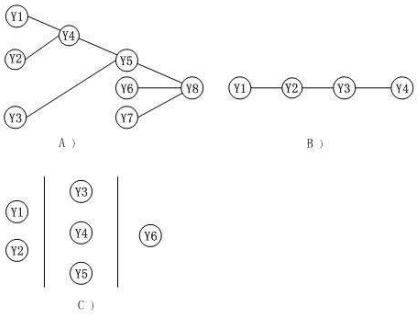

Figure3: Typical Process-Tree

(ii) Sampling period

The scheduling algorithm gets job shop state information within a fixed sampling period T.

(iii) Idle machine operation table

One machine can provide a variety of machining operations, and only one machining operation has been performed completely by a machine, another machining operation could be selected. At each sampling point, all of machining operations provided by a group of machines are recorded in a table, which is the idle machine operation table.

(iv) Leaf table

Leaves of all process- trees can be recorded in a table, this table is called as leaf table.

(v) Job processing table

The information of job being processed is recorded in a special designed table, this table is defined as job processing table, and a job processing table corresponding to a Process-Tree. At the same time, different types of jobs (each type includes a number of jobs) may be processed in a job shop, so that each job can be numbered as job m-n , m represent the type, n represent the serial number. Obviously job m-n is subordinate to a Process-Tree m or a job processing table m . Because a job processing table consists of a group of rows, so the processing information of job m-n can be recorded into the row n of the table m .

2.3 Algorithm Description

The scheduling algorithm of flexible job shop online scheduling, which is based on Process-Tree, can be described as Figure 4.

Journal of Theoretical and Applied Information Technology
31st October 2012. Vol. 44 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

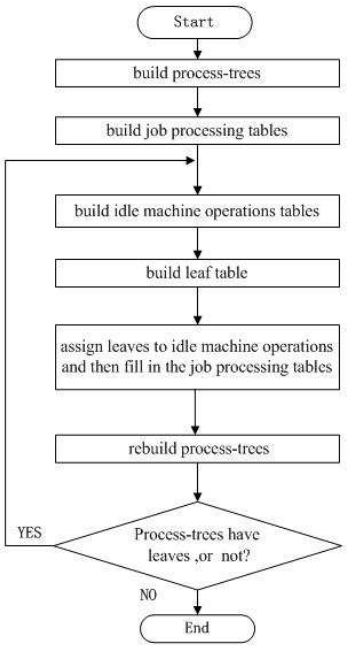

Figure 4: Flowchart Of Scheduling Algorithm

Step1. build Process-Trees for different types of jobs.
Step2. build job processing table corresponding to Process-Trees.
Step3. read job shop state information, and build idle machine operation table.
Step4. scan Process-Trees, and build leaf table.
Step5. according to the dispatching rules to choose leaves from leaf tables, assign leaves to idle machine operations, and then fill in the job processing tables.
Step6. leaves having been selected fall off from Process-Trees, and then rebuild Process-Trees.
Step7. if Process-Trees haven't leaves, algorithm will turn to the end , else turn back to step3.

3. COMPUTATIONAL SIMULATION

Consider the following instance with three machines and four jobs of the same type. The Process-Tree is shown in the Figure 5, and idle machine process table is given as following table 1.

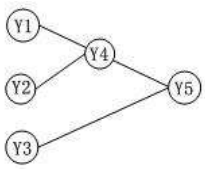

Figure 5: Process-Tree Of Jobs

Table 1: Idle Machine Operation

Idle machine operations	Time of processing
Y_1^{p1}, Y_2^{p1}	t_{Y1}^{p1}, t_{Y2}^{p1}
Y_3^{p2}, Y_4^{p2}	t_{Y3}^{p2}, t_{Y4}^{p2}
Y_5^{p3}, Y_1^{p3}	t_{Y5}^{p3}, t_{Y1}^{p3}

Scheduling cycle 1.
Build Process-Trees for four jobs(see Figure 6)

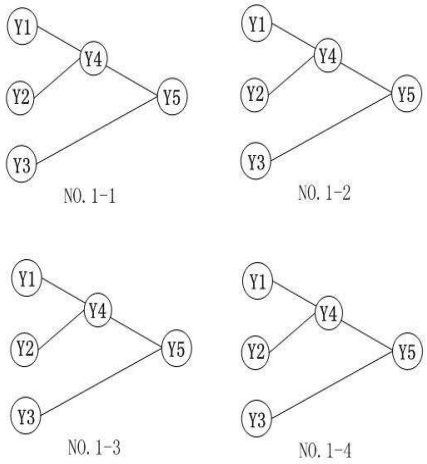

Figure 6: Process-Tree For Each Job

Scan Process-Trees, and build leaf table(see table 2)

Table 2: Leaves Of Each Process-Tree

Job	Leaves of Process-Tree
1-1	$Y_1^{1-1}, Y_2^{1-1}, Y_3^{1-1}$
1-2	$Y_1^{1-2}, Y_2^{1-2}, Y_3^{1-2}$
1-3	$Y_1^{1-3}, Y_2^{1-3}, Y_3^{1-3}$
1-4	$Y_1^{1-4}, Y_2^{1-4}, Y_3^{1-4}$

Fill in the job processing table (see table 3) , according to the random dispatching rule.

Table 3: Operation Process

Job	Process sequence				
	Y_1	Y_2	Y_3	Y_4	Y_5
1-1	$1\text{-}p_1$	–	–	–	–
1-2	–	–	$1\text{-}p_2$	–	–
1-3	$1\text{-}p_3$	–	–	–	–
1-4	–	–	–	–	–

Leaves fall off from Process-Trees, and then rebuild Process-Trees(see Figure 7).

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

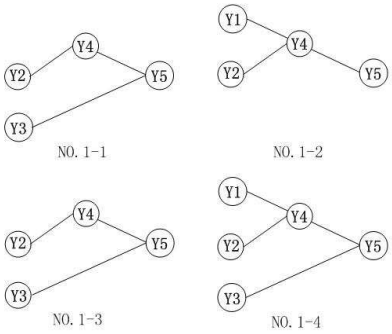

Figure 7: Process-Tree For Each Job

Scheduling cycle 2.

Scan Process-Trees, and build leaf table(see table 4).

Table 4: Leaves Of Each Process-Tree

Job	Leaves of Process-Tree
1-1	Y_2^{1-1}, Y_3^{1-1}
1-2	Y_1^{1-2}, Y_2^{1-2}
1-3	Y_2^{1-3}, Y_3^{1-3}
1-4	$Y_1^{1-4}, Y_2^{1-4}, Y_3^{1-4}$

Read job shop state information, and suppose can build idle machine operations table(see table 5).

Table 5: Idle Machine Operation

Idle machine operations	Time of processing
Y_1^{p1}, Y_2^{p1}	t_{Y1}^{p1}, t_{Y2}^{p1}
Y_3^{p2}, Y_4^{p2}	t_{Y3}^{p2}, t_{Y4}^{p2}

Fill in the job processing table (see table 6) , according to the random dispatching rule.

Table 6: Operation Process

Job	Process sequence				
	Y_1	Y_2	Y_3	Y_4	Y_5
1-1	$1\text{-}p_1$	$2\text{-}p_1$	–	–	–
1-2	–	–	$1\text{-}p_2$	–	–
1-3	$1\text{-}p_3$	–	–	–	–
1-4	$2\text{-}p_3$	–	–	–	–

Leaves fall off from Process-Trees, and then rebuild Process-Trees(see Figure 8).

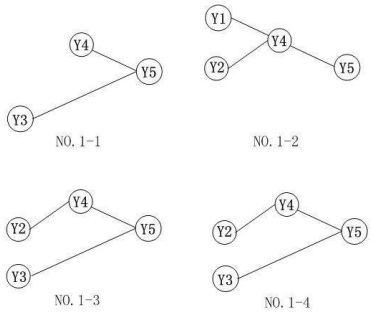

Figure 8: Process-Tree For Each Job

Repeat the above scheduling cycles, after cycle 9, algorithm will turn to the end, an algorithm simulation result is depicted in table 7 (see table 7 and Figure 9).

Table 7: Operation Process

Job	Process sequence				
	Y_1	Y_2	Y_3	Y_4	Y_5
1-1	$1\text{-}p_1$	$2\text{-}p_1$	$3\text{-}p_2$	$4\text{-}p_2$	$5\text{-}p_3$
1-2	$3\text{-}p_1$	$4\text{-}p_1$	$1\text{-}p_2$	$5\text{-}p_2$	$6\text{-}p_3$
1-3	$1\text{-}p_3$	$5\text{-}p_1$	$7\text{-}p_2$	$6\text{-}p_2$	$8\text{-}p_3$
1-4	$2\text{-}p_3$	$6\text{-}p_1$	$9\text{-}p_2$	$8\text{-}p_2$	$7\text{-}p_3$

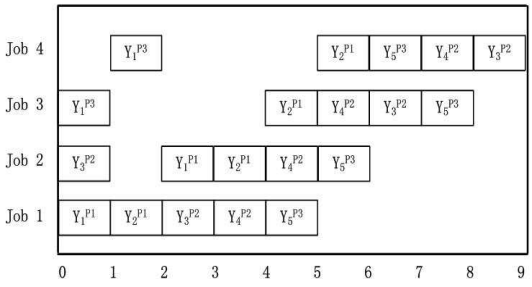

Figure 9: Gantt Chart For The Scheduling

Such results can be concluded from table 7 and Figure 9 .

(i)Processing time t_i of each job is provided as follows.

Job 1 $t_1=t_{Y_1}^{P1}+t_{Y_2}^{P1}+t_{Y_3}^{P2}+t_{Y_4}^{P2}+t_{Y_5}^{P3}$

Job 2 $t_2=t_{Y_1}^{P1}+t_{Y_2}^{P1}+t_{Y_3}^{P2}+t_{Y_4}^{P2}+t_{Y_5}^{P3}$

Job 3 $t_3=t_{Y_1}^{P3}+t_{Y_2}^{P1}+t_{Y_3}^{P2}+t_{Y_4}^{P2}+t_{Y_5}^{P3}$

Job 4 $t_4=t_{Y_1}^{P3}+t_{Y_2}^{P1}+t_{Y_3}^{P2}+t_{Y_4}^{P2}+t_{Y_5}^{P3}$

(ii)The number of operations processed on each machine are given with variables Ni.

Machine1 $N_1 = 6$

Machine2 $N_2 = 8$

Machine3 $N_3 = 6$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

(iii) The first completed job is job 1, and completion time is $(5-1)\times T + t_{Y5}^{P3}$, the last completed job is job 4, and completion time is $(9-1)\times T + t_{Y3}^{P2}$, T is sampling period , t_{Y5}^{P3} is the last operations of job 1 on the machine3, t_{Y3}^{P2} is the last operations of job 4 on the machine2 (see Figure 9).

4. CONCLUSION AND OUTLOOK

A new approach based on Process-Tree has been proposed to solve the flexible job shop online scheduling problem. The work of this paper will contribute to construct better method to this kind of scheduling problem. In the future, one direction for further research is to develop some effective approaches to pattern the form of process sequence with precedence constraint after a Process-Tree. When process precedence constraint of a job can not be depicted as a Process-Tree, the approach is essential to scheduling algorithm, and which will involve the alternative process planning(one of the most popular practices). Another direction for future research to improve scheduling performance, is to provide more efficient machine learning algorithms , and apply these learning algorithms to improve knowledge base which is used to dynamically select proper dispatching rules.

REFERENCES:

[1] S.C.Graves, "A review of production scheduling, Operations" Research, Vol.29, No.4, 1981, pp. 646-675.

[2] W.Lawrence, C.Philippe B, "Broader view of the job shop scheduling problem", Management Science, Vol.38, No.7, 1992, pp. 1018-1033.

[3] P.Jihyung, K.Mujin, "Intelligent operations scheduling system in a job shop", International Journal of Advanced Manufacturing Technology, Vol.11, No.2, 1996, pp.111-119.

[4] F.Parviz, J.Fariborz, A.Jamal, "Flexible job shop scheduling with overlapping in operations",Applied Mathematical Modelling, Vol.33, No.7, 2009, pp.3076-3087.

[5] A.Nasr, Elmekkawy, "Robust and stable flexible job shop scheduling with random machine breakdowns using a hybrid genetic algorithm", International Journal of Production Economics, Vol.132, No.2, 2011, pp.279-281.

[6] K.Marnix, H.Han, "Identifying and exploiting commonalities for the job-shop scheduling problem", Computers and Operations Research, Vol.38, No.11, 2011, pp.1556-1561.

[7] N.Yoni, W.Gideon, "A fluid approach to large volume job shop scheduling", Journal of Scheduling, Vol.13, No.5, 2010, pp.509-529.

[8] G.Thomas, R.Martin, "Distributed policy search reinforcement learning for job-shop scheduling tasks", International Journal of Production Research, Vol.50, No.1, 2012, pp.41-61.

[9]G.Selcuk, S.Ihsan, K.Utku, "Optimization of schedule stability and efficiency under processing time variability and random machine breakdowns in a job shop environment", Naval Research Logistics, Vol.59, No.1, 2012, pp.26-38.

[10]K.Voratas, S.Siriwan, "A two-stage genetic algorithm for multi-objective job shop scheduling problems", Journal of Intelligent Manufacturing, Vol.22, No.3, 2011, pp.355-365.

[11]H.Myoungsoo, L.Young Hoon, K.Sun Hoon, "Real-time scheduling of multi-stage flexible job shop floor", International Journal of Production Research, Vol.49, No.12, 2011, pp.3715-3730.

[12]Y.Rubiyah, K.Marzuki, "Solving job shop scheduling problem using a hybrid parallel micro genetic algorithm", Applied Soft Computing Journal, Vol.11, No.8, 2011, pp.5782-5792.