Chapter 13
Study on an Affected Operations Rescheduling Method Responding to Stochastic Disturbances

Ya-qin Zhou, Bei-zhi Li, Jian-guo Yang, and Peng Yang

Abstract Literature on job shop scheduling has primarily focused on the development of predictive schedules in static environment. However, when the schedule is carried out in job shop, it may affected by varied disturbances. These disturbances will make the original schedule worse, even invalid. Rescheduling is the process of finding a new schedule to respond to the stochastic disturbances. In this paper, an affected operations rescheduling method (AORM) is studied to respond to disturbances. First, the basic theory of the method is given. Then the algorithmic procedure is introduced. The objective functions evaluating the rescheduling method are given. At last, an illustration example is tested and analyzed. The result shows that the rescheduling method proposed can produce new optimal schedules to respond to stochastic disturbances.

Keywords Affected operations rescheduling • Dynamic scheduling • Repair method • Rescheduling method

Introduction

Most of the literature on production scheduling focuses on how to generate a schedule to guide production assuming some specialty of the problem known. The schedule generated can solve the conflict between resources by controlling the release time of job to workshop. It can ensure the raw material needed to be ordered on time and the job to be delivered on time to satisfy the due date. Scheduling problem is a typical NP-complete problem (Garey et al. 1976). For static job shop scheduling problem, many optimization methods to obtain the optimal schedule have been proposed, such as neural network (Ibrahim et al.

Y.-q. Zhou (✉) • B.-z. Li • J.-g. Yang • P. Yang
College of Mechanical Engineering, Donghua University, Shanghai, China
e-mail: zhouyaqin@dhu.edu.cn

R. Dou (ed.), *Proceedings of 2012 3rd International Asia Conference on Industrial Engineering and Management Innovation(IEMI2012)*,
DOI 10.1007/978-3-642-33012-4_13, © Springer-Verlag Berlin Heidelberg 2013

2009; Zhao et al. 2010), simulated annealing algorithm (Jamili et al. 2011; Zhang and Wu 2010), tabu search (Eswaramurthy and Tamilarasi 2007; Geyik and Cedimoglu 2004), and genetic algorithm (Ritwik and Deb 2011; Wang et al. 2009). In the dynamic job shop environment, some random disturbances will disturb or interrupt the implement of the original optimal schedule to make it invalid and rescheduling is needed. The varied disturbances which always appear in workshop include the machine failure, the delay of raw material, the insertion of new job, and the cancellation of the order etc. Most of the disturbances can be modeled as machine failure, which may make the machine unavailable for a period of time.

Rescheduling is a process to generate a new feasible schedule when the disturbance occurs. Most job shop is dynamic, so it is more important to study on rescheduling method, which attracts many researchers' attention recently (Abumaizar and Svestka 1997; Adibi et al. 2010; Cao and Du 2011). In this paper, an affected operations rescheduling method is studied for job shop scheduling problem. Firstly, the basic principle of the affected operations rescheduling method is given, and then its algorithm process is introduced, with the indicator function measuring rescheduling method's performance given. Finally, an illustration is analyzed to testify that the rescheduling method proposed can generate a new schedule to direct production responding to random disturbance.

The Basic Principle of the Affected Operations Rescheduling Method

The affected operations rescheduling only reschedule the operations which are affected directly or indirectly by the disturbance during the execution of the schedule. This algorithm is based on a binary tree algorithm (Li et al. 1993). The affected operations rescheduling is a heuristic rescheduling method. Its objective is to obtain a new schedule with a smallest makespan increment, while keeping the job's processing sequence as in the original schedule.

In most manufacturing systems scheduling, dispatching the process sequence of jobs is based on jobs' processing route. Any feasible schedule, satisfying the jobs' processing route constraints, must determine the processing sequence, start and end time of each task (job) on each equipment (machine) to get the optimal or near optimal value of the selected objective.

The sequence of job and machine processing can be represented by a binary tree. The tree is a graph starting from a single (root) node (the first one of the affected processes). The binary tree is a tree with nodes connected without cycles, that is, each node has at most two branches (as shown in Fig. 13.1). Each node represents a operation, the left branch of each node represents the job branch, that is, the next operation of the job (*noj*), which can be obtained from the job's processing route constraints, while the right branch represents the machine branch, that is, the next

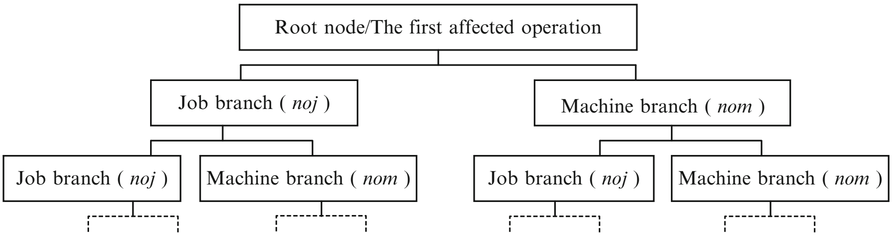

Fig. 13.1 The binary tree description of a schedule

operation to be processed on the machine of current operation (*nom*), which can be get from the processing sequence of the operations on the machine in the original schedule.

The basic principle of the affected operation rescheduling is to delay the start processing time of certain operations with a minimum amount of time to respond to any disturbance. The minimum amount of time must: (1) make the jobs' processing route constraints satisfied; (2) maintain the initial processing sequence of operations on each machine as in the original schedule.

The effect of the first affected operation (the first operation disturbed by the disturbance) on its job and machine branch and the effect of these branched on their corresponding job and machine branches are studied to solve the effect of the disturbance. That is to track the spread of the effect of the disturbance on the "rescheduling" binary tree branches, and re-update the binary tree. The first affected operation is the root node with each successor node representing one of the affected operations.

The Algorithm of the Affect Operation Rescheduling Method

Symbol Definition

The definition of the symbols used in the affected operation rescheduling algorithm is given as follows:

R: The set of the remaining operations needed to be rescheduling when the disturbance occurs. (Including the disturbed operations and the operations not yet started when the disturbance occurs).

O: The set of the probably affected operations.

A: The set of the affected operations after rescheduling (with new starting time and ending time).

noj: The next operation of the job (job branch).

nom: The next operation on the machine (machine branch).

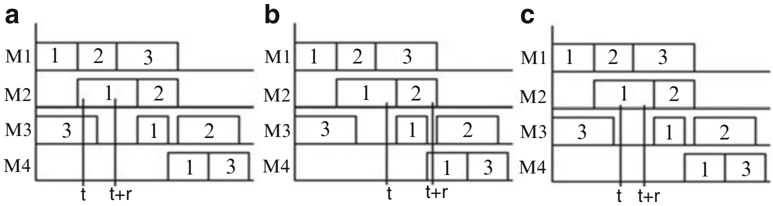

Fig. 13.2 The effect of machine failure on the original schedule

jST: The starting time of the operation, restricted by the former operation of the job.
mST: The starting time of the operation, restricted by the former operation on the machine.
ST: The starting time of the operation in the original schedule.
ET: The ending time of the operation in the original schedule.
newST: The starting time of the operation in the new schedule.
newET: The ending time of the operation in the new schedule.
devST: The deviation of the starting time of the operations between the original schedule and the new schedule.

readyTime: The ready time of the machine.
noR: The number of the set *R* (The number of the remaining operations).
noA: The number of the set *R* (The number of the affected operations).
i: The index of the set *A*.
g: The index of the set *O*.

Abbreviations and Acronyms

Step 1: Set $i = 1$, $g = 1$, $devST = 0$, $O = \varnothing$, $A = \varnothing$, *jST* and *mST* of all the operations equal to 0. Determine the first affected operation $O[1]$. The effect of the disturbance (machine failure) on the original schedule has three cases, as shown in Fig. 13.2. At time t, there is a disturbance occurring on machine M3 with duration time of r, thus, $O[1]$ belongs to one of the three cases as following:

(a) The interrupted operation, the effect of the interruption is as shown in Fig. 13.2a.

(b) If there is no operation being interrupted, select the first remaining operation on the machine if the operation exists and its ST is smaller than the *readyTime* of the shut down machine (The *readyTime* of the shut down machine equals to the stopping time t plus duration time r), as shown in Fig. 13.2b.

(c) Otherwise, the algorithm stops, there is no operation affected (Rescheduling is not needed), as shown in Fig. 13.2c.

Step 2: $O[1]$ is as the current operation, its *mST* equals to the *readyTime* of the shut down machine, g pluses 1.

Step 3: The *newST* of the current operation equals to Max (*jST* of the current operation, *mST* of the current operation). The *newET* of the current operation equals to *newST* of the current operation pluses the processing time of the current operation.

Step 4: If the current operation can not be matched to any affected operation v in set A , go to step 5, otherwise, reset the attribute of $A[v]$ as following, then go to step 7.

devST = *devST* + Max(*newET* of the current operation-*newET* of $A[v]$, 0).
newST of $A[v]$ = Max(newST of the current operation, *newST* of $A[v]$).
newET of $A[v]$ = *newST* of $A[v]$ + the processing time of $A[v]$.

Step 5: Set the i-th affected operation $A[i]$ as the current operation, i pluses 1.

Step 6: *devST* = *devST* + (*newET* of $A[i]$-*ET* of $A[i]$).

Step 7: Get the job branch *noj* of the current operation from the of process route of the job, if *noj* exists and its *ST* is less than *newET* of the current operation. Then set $O[g]$ = *noj*, *jST* of $O[g]$ = *newET* of the current operation, g pluses 1.

Step 8: Get the machine branch *nom* of the current operation from the original schedule if *nom* exists and its *ST* is less than *newET* of the current operation. Then set $O[g]$ = *nom*, *mST* of $O[g]$ = *newET* of the current operation, g pluses 1.

Step 9: Remove the current operation from the set O. Add the new operations of step 7 and 8 into the set.

Step 10: If $O = \emptyset$, end. Otherwise, select randomly an operation from set O as the current operation, then go to step 3.

Performance Measurement

Three performance measurement indexes are used to evaluate the new schedule generated by the affected operation rescheduling method, in order to evaluate the performance of this rescheduling method: the change rate of Makespan, the starting time deviation of each operation, and the sequence deviation between the new schedule and the original schedule.

1. The change rate of Makespan (M_p)
   The change rate of Makespan can be defined as following:

$$M_p = \left[\frac{M_m - M_0}{M_0}\right] * 100\% \tag{13.1}$$

   where, M_0 is the Makespan of the original schedule, M_m is the Makespan of the new schedule generated by the rescheduling method.
2. The starting time deviation (*devST*)

The starting time deviation is a valid measurement of the efficiency of rescheduling method, especially in the job environment where the auxiliary resources (such as tools and fixtures) is delivered to machine based on the original schedule. Obviously, if the material is delivered earlier than the demand, the change of jobs' starting time may incur storage costs and, more importantly, if the demand for cutting tools and materials is earlier than the original schedule, it will incur emergency ordering cost. The start time deviation is calculated by calculating the sum of the absolute operation ending time difference value between the new schedule and the original schedule. The starting time deviation consists of two parts, the delay time, which equals to the sum of the positive ending time difference, and the ahead of time, which equals to sum of the absolute value of the negative ending time difference

$$devST = \sum_{i=1}^{n} \sum_{j=1}^{hi} \left| newET_{ij} - ET_{ij} \right| \tag{13.2}$$

where n is the number of job, hi is the operation number of job i.

3. Sequence deviation ($devSQ$)
If the adjustment is prepared in advance based on the initial operation sequence on the machine, then this measurement of sequence deviation is critical. It is measured using the following method.

For each operation j on machine k in the new schedule, define S_1 as the set of the operations processed before operation j in the original schedule, S_2 as the set of the operations processed after operation j, $S = S_1 \cap S_2$, $N_{jk} =$ the cardinal number of set S (the capacity of the set). The sequence deviation can be obtained by calculating the sum of the sequence deviation on each machine:

$$devSQ = \sum_{k} \sum_{j} N_{jk} \tag{13.3}$$

For affected operation rescheduling method, the change rate of Makespan and the starting time deviation is very small. It has no sequence deviation because it only reschedules the affected operations.

Illustration

The case La31 (30 jobs and 10 machines) proposed by Lawrence (1984) is used to test. The original schedule with the optimal makespan 1784 is obtained by the biological intelligent method (Zhou et al. 2006), with the Gantt chart shown in Fig. 13.3. In Fig. 13.3, the processing sequence of jobs on each machine is given, with the number in the box representing job. If there is no enough places, then the job is shown in the top of the box.

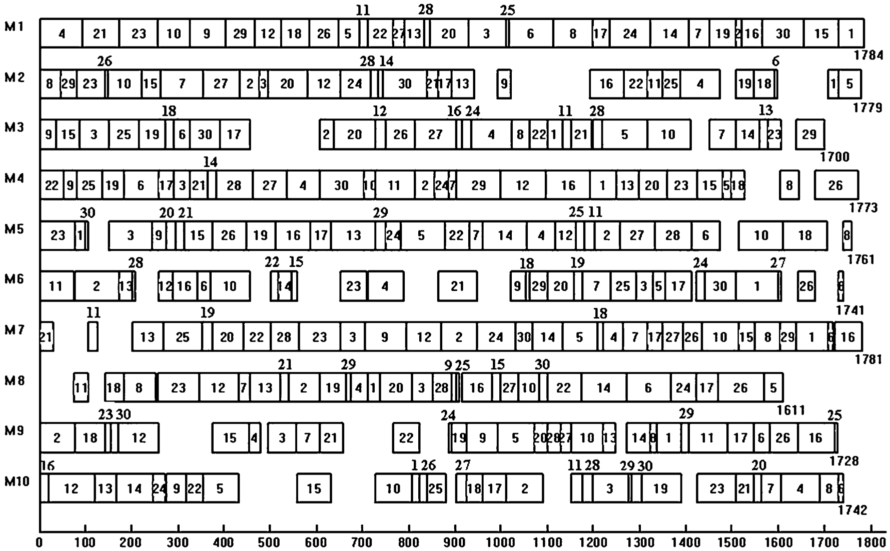

Fig. 13.3 The Gantt chart of the original schedule of La3

The majority of disturbance occurring during production can be mapped to machine failure. The disturbance of machine failure is tested. Assume that when the original schedule shown in Fig. 13.3 is carrying out, there is a breakdown occurring at time 800 on machine M9, with the duration time of 80 time units.

For this disturbance, a new schedule is generated by using the affected operation rescheduling method proposed in this paper, with the Gantt chart shown in Fig. 13.4. In the figure, the dashed line on the machine M9 means the interruption time of the machine. In the new schedule, the processing of the operations before the interruption is not changed, only the operations after the disturbance in the original schedule are rescheduled. The new starting time and the ending time of each affected operations are calculated to get the Makespan of the schedule 1791. For Figs. 13.3 and 13.4, it is shown that only the affected operations have new starting time and ending time. By comparing the new schedule generated by the AORM and the original schedule, it is shown that the Mapespan only increases 7 time units while the machine M8 is invalid for 80 time units, the change rate of Mapespan is 0.4% and the processing sequence of each operation is not changed, there is no sequence deviation.

Conclusion

During the schedule execution in workshop, the schedule will be interrupted by a variety of random disturbances, and therefore a new schedule proposed by rescheduling to respond to random disturbances has a good guide to the actual

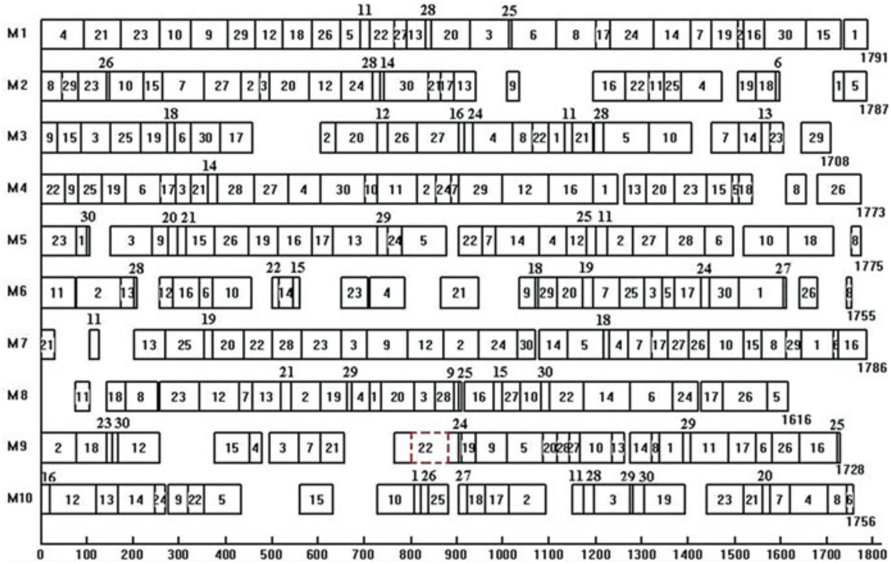

Fig. 13.4 The Gantt chart of the new schedule generated by AORM

production. Affected operations rescheduling method is proposed in this paper, with the principle and algorithm process. The result of the illustration shows that the proposed rescheduling method can produce a new optimal schedule to respond to disturbance.

Acknowledgement This research was partially supported by the National Science Foundation of China (70371040), Shanghai Leading Academic Discipline Project (B602) and the Fundamental Research Funds for the Central Universities.

References

Abumaizar RJ, Svestka JA (1997) Rescheduling job shops under random disruptions. Int J Prod Res 35(7):2065–2082

Adibi MA, Zandieh M, Amiri M (2010) Multi-objective scheduling of dynamic job shop using variable neighborhood search. Expert Syst Appl 37(1):282–287

Cao Y, Du J (2011) An improved Simulated Annealing algorithm for real-time dynamic job-shop scheduling. Adv Mater Res 186:636–639

Eswaramurthy VP, Tamilarasi A (2007) Tabu search strategies for solving job shop scheduling problems. J Adv Manuf Syst 6(1):59–75

Garey EL, Johnson DS, Sethi R (1976) The complexity of flowshop and jobshop scheduling. Math Oper Res 1:117–129

Geyik F, Cedimoglu IH (2004) The strategies and parameters of tabu search for job-shop scheduling. J Intell Manuf 15(4):439–448

Ibrahim S, El-Baz MA, Esh E (2009) Neural networks for job shop scheduling problems. J Eng Appl Sci 56(2):169–183

Jamili A, Mohammad A, Tavakkoli-Moghaddam R (2011) A hybrid algorithm based on particle swarm optimization and simulated annealing for a periodic job shop scheduling problem. Int J Adv Manuf Technol 54(1–4):309–322

Lawrence S (1984) Resource constrained project scheduling: an experimental investigation of heuristic scheduling techniques (Supplement). Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh

Li R, Shyu Y, Adiga S (1993) A heuristic rescheduling algorithm for computer-based production scheduling systems. Int J Prod Res 31(8):1815–1826

Ritwik K, Deb S (2011) A genetic algorithm-based approach for optimization of scheduling in job shop environment. J Adv Manuf Syst 10(2):223–240

Wang YM, Yin HL, Wang J (2009) Genetic algorithm with new encoding scheme for job shop scheduling. Int J Adv Manuf Technol 44(9–10):977–984

Zhang R, Wu C (2010) A hybrid immune simulated annealing algorithm for the job shop scheduling problem. Appl Soft Comput J 10(1):79–89

Zhao FQ, Zou JH, Yang YH (2010) A hybrid approach based on Artificial Neural Network (ANN) and Differential Evolution (DE) for job-shop scheduling problem. Appl Mech Mater 26–28:754–757

Zhou YQ, Li BZ, Yang JG (2006) Intelligent scheduling algorithm for problems with batch and non cutting time. Jixie Gongcheng Xuebao/Chin J Mech Eng 42(1):52–566 (Chinese)