A rescheduling approach based on genetic algorithm for flexible scheduling problem subject to machine breakdown

Li Nie*1, Xiaogang Wang1, Kai Liu1 and Yuewei Bai1

1 School of Intelligent Manufacturing and Control Engineering, Shanghai Polytechnic University, Shanghai, 201029, P. R. of China

*Corresponding author’s e-mail: nieli@sspu.edu.cn

Abstract. In this paper, a rescheduling approach based on genetic algorithm (GA) for solving the flexible scheduling problem subject to machine breakdown is proposed. In the proposed approach, event-driven rolling horizon rescheduling policy is employed to trigger the rescheduling procedure. Computational experiments are conducted on several benchmark data to prove the performance of the proposed approach. The results show that the proposed approach combines the rescheduling strategies of right-shift rescheduling and routing changing rescheduling to optimize the robustness and stability of rescheduling solution simultaneously.

1. Introduction

Scheduling task is one of the most time consuming and challenging activities within an enterprise. An effective production scheduling approach plays a particularly important role to improve the manufacturing performance in such large, complex manufacturing systems as automobile industry, electronics industry, aerospace industry and so on [1]. In practices, these manufacturing systems usually undergo the delays of various jobs/tasks due to such unpredictable interruptions as machine breakdowns, rush orders, order cancellations, due date changes, etc. These random events significantly interrupt the smooth progress of the whole schedule and result in low productivity of the manufacturing systems. Therefore, to improve the ability to cope with these random interruptions in the manufacturing systems is becoming an increasingly important issue [2].

The machine breakdown is regarded as one of the most popular and typical random events in manufacturing systems. The uncertainties about the machine breakdown in the manufacturing environments are classified into two types: partially unknown uncertainties or completely unknown uncertainties [3]. Partially unknown uncertainties mean some information relevant to the machine status is available when the predictive schedule is generated. Scheduling algorithms can take advantage of the information to develop the predictive schedule. For example, Mehta and Uzsoy [4] proposed a predictive schedule into which idle times were inserted according to the machine breakdown distributions as well as the structure of the predictive schedule. On the other hand, completely unknown uncertainties mean that no information about machine status is available in advance. As for completely unknown uncertainty problem such as scheduling problem subject to machine breakdown randomly, the approaches are divided into two types: pre-rescheduling and completely reactive scheduling [5]. In pre-rescheduling approach, firstly a schedule without considering machine breakdowns is generated and implemented until machine breakdown occurs, and a rescheduling procedure is triggered to react to the machine breakdown. In completely reactive scheduling, the scheduler generates a response to breakdown in real time. Usually the pre-rescheduling
approach is better to be adopted to deal with machine breakdown interruptions [6]. To develop an effective schedule for the manufacturing systems with this type of uncertainty is difficult but attractive for researchers. For example, Al-hinai and ElMekkawy [5], He and Sun [6], Sun et al. [7], Ahmadi et al. [8] developed effective scheduling approaches for scheduling problem with unpredictable machine breakdown.

The strategies which are mostly applied in the rescheduling procedure to adjust the initial scheduling when machine breakdown happens are: right-shift rescheduling and routing changing rescheduling. For example, Kamoun and Sriskandarajah [9], Kutanoglu and Sabuncuoglu [10], He and Sun [6], Ahmadi et al. [8] adopted the two strategies to deal with machine breakdown, however, the performance of robustness or stability cannot be guaranteed perpetually. To apply the evolutionary algorithms such as genetic algorithm (GA) to flexible scheduling problem is a potential method. Al-Hinai and ElMekkawy [5, 11], Lei [12], Jalivand-Nejad and Fattahi [13], Li and Gao [14], Wu et al. [15] achieved many good results through designing effective GAs for flexible scheduling problem.

This paper is organized as follows: Section 2 gives the definitions and assumptions of the flexible scheduling problem. Rescheduling strategy and proposed approach based on GA are given in Section 3. Section 4 presents experiments and the results. Lastly, the conclusion is drawn in Section 5.

2. Problem formulation

The flexible scheduling problem with machine breakdown is described as follows. A number of jobs are to be processed on a number of machines. The sequence of operations for each job is fixed. Each operation of a job can be operated on any of its alterative machines. The processing time of each operation on each of its alterative machine is known and deterministic. Machines may breakdown randomly and the maintain operation spends a deterministic time duration. In the prescheduling stage, to minimize makespan is the main objective (Eq (1)), whereas in the rescheduling stage, the schedule’s performance is measured by the weighted sum of robustness and stability (Eq (2)).

\[
F_1 = \min\{C_i, i = 1, ..., n\} \quad (1)
\]

\[
F_2 = w_1 * RM + w_2 * SM \quad (2)
\]

Where, \(C_i\) denotes the complete time of Job \(i\), \(RM = \frac{MS' - MS^p}{MS^p} \times 100\%\) and \(SM = \sum_{i=1}^{n} \sum_{j=1}^{m} |C_{i,j}^p - C_{i,j}^r| / \sum_{i=1}^{n} h_i\) are the robustness and stability of the reschedule, respectively. \(MS^p\) and \(MS'\) are the makespan of the prescheduling solution and rescheduling solution, respectively. \(C_{i,j}^p\) and \(C_{i,j}^r\) are the complete time of operation \(O_{i,j}\) in prescheduling solution and rescheduling solution, respectively.

3. Rescheduling strategy and proposed approach

An event-driven rolling horizon rescheduling policy proposed by Wang et al. [16] is employed and a GA based approach to solve the flexible scheduling problem subject to machine breakdown is proposed.

3.1. Event-driven rolling horizon rescheduling policy

The event-driven rescheduling policy is used and the critical events that trigger the rescheduling procedure are defined as a machine breakdown. In other word, once a machine breakdown occurs, the rescheduling procedure begins.

In rolling horizon scheduling method, operation set in current window determines the initial state in prescheduling and rescheduling stage. At the beginning, assuming all jobs are available and ready for processing, the operation in the current window is defined as \(OW = O\). When the rescheduling procedure is triggered at time \(t\) because of the breakdown of machine \(j\), the operation set is updated by removing the finished operations, i.e. \(OW = OW - FO\), where \(FO\) includes the operations that have
finished at time $t$ (In the paper, it is assumed that there are no new jobs arrival. Therefore, there are no new operations added into $OW$). $FO$ is defined as $FO = \{O_{i,j} | i = 1,...,n; j = 1,...,h; C_{i,j} < t \}$.

It is noticeable that the directly affected operation, i.e. the operation that is being operated on the breakdown machine, can be operated continually on the machine after the machine is available again or change its planned routing and move to another alternative machine to be operated continually.

### 3.2. Genetic algorithm for flexible scheduling problem

#### 3.2.1. Procedure of the mGA

The overall evolutionary process of mGA is described as follows: Step 1: Set the parameters of mGA, such as size of population $PopSize = 200$, probability of crossover and mutation $p_c = 0.8$ and $p_m = 0.2$, maximum iteration number $niter = 100$, and set viable $gen = 0$; Step 2: Initialize the population; Step 3: Evaluate every individual in the population according their fitness; Step 4: If termination criteria is satisfied, i.e. if $gen >= niter$, then go to Step 6; otherwise, $gen = gen+1$, and go to Step 5; Step 5: Generate a new generation of population through applying the genetic operators (including selection, crossover and mutation operators) to the current generation of population; then go to step 3; Step 6: Out put the best solution.

#### 3.2.2. Encoding scheme and decoding scheme

Considering these two sub-problems in flexible scheduling problem, routing and sequencing sub-problem, the chromosome structure in mGA is designed to comprise of two parts. The one is the routing string and the other is the sequencing string. They have different encoding methods, respectively. The algorithm that translate a chromosome into an active schedule in [14] is used in decoding scheme.

#### 3.2.3. Genetic operators

Roulette method is adopted to select the individuals according to the fitness. The individual with better fitness is selected with a high probability. In additional, different crossover and mutation operators are employed to the routing string and sequencing string of the chromosome respectively. The two-point crossover and neighborhood mutation are applied on the routing string, and precedence operation crossover and swapping mutation are used on the sequencing string of the chromosome.

### 4. Experiments and results

In this section, the proposed approach mGA is tested on 5 FJSP instances which are taken from [17]. For each datum, preschedules without considering machine breakdown are generated by mGA and the results are compared with the best solutions obtained from start-of-the-art reported algorithms. In order to test the performance of mGA on the rescheduling procedure, machine for failure and the time for breaking down are randomly generated based on the 5 FJSP data. Three tuple $(a, b, c)$ is used to express the data. $a$ denotes the machine that breakdown occur, $b$ denotes when the machine breakdown and $c$ denotes the time needed to repair the machine.

Table 1 shows the experimental results. AIA is selected from Yuan et al. [18], HA represents the algorithm proposed by Li and Gao [14] and mGA represents the proposed GA-based approach in the paper. The results marked by * are the best results without considering the breakdown of machine among all these algorithms. From the results in Table 1, it is found that the proposed algorithm mGA found a better solution than existing best algorithms on MFJS04, a suboptimal solution on MFJS02, and also found the best solution of the rest 3 instances in the prescheduling without considering machine breakdown. It is concluded that mGA is not worse than the existing algorithms in the respect of search capability. From Table 1, it is also found that mGA can optimize the robustness and stability simultaneously under random breakdown. The Gantt chart of the best solution of MFJS05 is shown in Fig. 1. The shaded area in Fig. 1 (b) denotes the machine breakdown occurred on $M_6$ between time 154 and 244. The comparison of the arrangement for all operations between the preschedule and reschedule is listed in Table 2. The data shown in bold means the corresponding operation changes its original routing; the data shown in italics means the corresponding operation right shifts. From Fig 1
and Table 2, it is found that the rescheduling solution obtained by mGA combined the rescheduling strategies of right-shift rescheduling and routing changing rescheduling intelligently. For example, O_{2,1} is being operated on M_6 when M_6 breakdown. In order to maintain the robustness of the reschedule, O_{2,1} changes its routing from M_6 to M_3 and continues to be operated before O_{6,1} on M_3. Since O_{2,1} inserts before O_{6,1} on M_3, O_{6,1} and its successor O_{0,2} have to be right shifted. Whereas the rest operations such as O_{1,0}, O_{0,1}, O_{1,1} and O_{4,2} don’t need to right shift or change their routings. Eventually, the robustness and stability of the rescheduling solution on MFJS05 is 8.75% and 15.24.

Table 1. Experimental results on test sets.

| No | Problem Type | Prescheduling | (a,b,c) | Rescheduling | robustness | stability |
|----|--------------|---------------|---------|--------------|------------|-----------|
| 1  | MFJS01 5*6*3*3 | 468* 468* 468* | (M_1, 110, 120) | 25.64% | 24 |
| 2  | MFJS02 5*7*3*3 | 448 446* 448 | (M_6, 87, 90) | 16.29% | 19.07 |
| 3  | MFJS03 6*7*3*3 | 468 466* 466* | (M_2, 102, 120) | 6.22% | 20.17 |
| 4  | MFJS04 7*7*3*3 | 554 554 541* | (M_3, 219, 100) | 0% | 1.86 |
| 5  | MFJS05 7*7*3*3 | 527 514* 514* | (M_6, 154, 90) | 8.75% | 15.24 |

Table 2. Comparison between the preschedule and reschedule on MFJS05.

| OPer | Preschedule | Reschedule | OPer | Preschedule | Reschedule |
|------|-------------|------------|------|-------------|------------|
| Mach | Start       | Finish     | Mach | Start       | Finish     |
| O_{0,0} | 2 0 100 | - - - | O_{1,2} | 5 238 374 | 5 238 374 |
| O_{0,1} | 1 188 318 | 1 188 318 | O_{4,0} | 1 65 188 | 1 65 188 |
| O_{0,2} | 3 318 468 | 3 409 559 | O_{4,1} | 6 232 318 | 6 244 330 |
| O_{1,0} | 0 87 301 | 0 87 301 | O_{4,2} | 1 384 484 | 1 384 484 |
| O_{1,1} | 1 318 384 | 1 318 384 | O_{5,0} | 3 0 154 | - - - |
| O_{1,2} | 5 384 479 | 5 384 479 | O_{5,1} | 3 154 304 | 3 259 409 |
| O_{2,0} | 0 0 87 | - - - | O_{5,2} | 6 318 498 | 4 409 529 |
| O_{2,1} | 6 87 232 | 3 154 259 | O_{6,0} | 2 100 245 | 2 100 245 |
| O_{2,2} | 4 238 338 | 4 259 359 | O_{6,1} | 2 245 369 | 2 245 369 |
| O_{3,0} | 1 0 65 | - - - | O_{6,2} | 4 369 514 | 6 369 547 |
| O_{3,1} | 4 65 238 | 4 65 238 | |

5. Conclusion and future work

In this paper we study the flexible scheduling problem with machine breakdown. An event-driven rolling horizon scheduling policy is applied and a GA-based algorithm mGA is proposed. Simulation results show that mGA with the event-driven rolling horizon scheduling policy responds to the changes in the shop floor immediately, and achieves high efficient scheduling results which improve shop performance even under the effects of disruptions. However, the proposed scheduling approach
based on event-driven rolling horizon rescheduling strategy is not suitable in the manufacturing system in which the critical events occur high frequently. To develop a hybrid approach which incorporated the advantage of pre-rescheduling approach and the completely reactive scheduling approach for manufacturing system disturbed by critical events which occur high frequently deserves to further research.

Acknowledgments
This research is supported by the National Key R&D Program of China under Grant No. 2017YFE0118700, the National Natural Science Foundation of China under Grant No. 51605273 and U1537110, and Shanghai Polytechnic University - Key Discipline Construction (Mechanical Engineering, XXKZD1603)

References
[1] Liu, M. (2009) A Survey of Data-based Production Scheduling Methods. Acta Automatica Sinica, 35: 785-806.
[2] Goren, S., Sabuncuoglu, I. (2009) Optimization of Schedule Robustness and Stability under Random Machine Breakdowns and Processing Time Variability. IIE Trans., 42: 203–220.
[3] Herrmann, J.W. (2006) Handbook of Production Scheduling. Springer Science & Business Media, New York.
[4] Mehta, S.V., Uzsoy, R.M. (1998) Predictable Scheduling of a Job Shop Subject to Breakdowns. IEEE Trans. Robot. Autom., 14(3): 365–378.
[5] Ai-Hinai, N., Eimekkawy, T.Y. (2011) Robust and Stable Flexible Job Shop Scheduling with Random Machine Breakdowns Using a Hybrid Genetic Algorithm. Int. J. Prod. Econ., 132: 279–291.
[6] He, W., Sun, D. (2013) Scheduling Flexible Job Shop Problem Subject to Machine Breakdown with Route Changing and Right Shift Strategies. Int. J. Adv. Manuf. Tech., 66: 501–514.
[7] Sun, D., He, W., Zheng, L., Liao, X.Y. (2013) Scheduling Flexible Job Shop Problem Subject to Machine Breakdown with Game Theory. Int. J. Prod. Res., 52: 3858-3876.
[8] Ahmadi, E., Zandieh, M., Farrokh, M. (2016) A Multi Objective Optimization Approach for Flexible Job Shop Scheduling Problem under Random Machine Breakdown. Comput. Oper. Res., 73: 56-66.
[9] Kamoun, H., Sriskanadarajah, C. (1993) The Complexity of Scheduling Jobs in Repetitive Manufacturing Systems. Eur. J. Oper. Res., 70 (3): 350–364.
[10] Kutanoglu, E., Sabuncuoglu, I. (2001) Routing-based Reactive Scheduling Policies for Machine Failures in Dynamic Job Shops. Int. J. Prod. Res., 39: 3141–3158.
[11] Ai-Hinai, N., Eimekkawy, T.Y. (2012) An Efficient Hybridized Genetic Algorithm Architecture for the Flexible Job Shop Scheduling Problem. Flex. Serv. Manuf. J., 23: 64–85.
[12] Lei, D. (2011) Scheduling Stochastic Job Shop Subject to Random Breakdown to Minimize Makespan. Int. J. Adv. Manuf. Tech., 55: 1183–1192.
[13] Jalivand-Nejad, A., Fattahi, P. (2015) A Mathematical Model and Genetic Algorithm to Cyclic Flexible Job Shop Scheduling Problem. J. Intell. Manuf., 26:1085–1098.
[14] Li, X., Gao, L. (2016) An Effective Hybrid Genetic Algorithm and Tabu Search for Flexible Job Shop Scheduling Problem. Int. J. Prod. Econ., 174: 93–110.
[15] Wu, R., Li, Y.B., Guo, S.S. (2018) Solving the dual-resource constrained flexible job shop scheduling problem with learning effect by a hybrid genetic algorithm. Adv. Mech. Eng., 10: 1-14.
[16] Wang, B., Xi, Y.G., Gu, H. (2005) Terminal Penalty Rolling Scheduling Based on An Initial Schedule for Single-machine Scheduling. Comput. Oper. Res., 32: 3059-3072.
[17] Fattahi, P., Mehrabad, M.S., Jolai, F. (2007) Mathematical Modeling and Heuristic Approaches to Flexible Job Shop Scheduling Problems. J. Intell. Manuf., 18: 331–342.
[18] Yuan, Y., Xu, H., Yang, J.D. (2013) A Hybrid Harmony Search Algorithm for the Flexible Job Shop Scheduling Problem. Appl. Soft. Comput., 13: 3259–3272.