FJS Problem Under Machine Breakdowns

Rim Zarrouk[1](✉), Imed Bennour[1], Abderrazak Jemai[2], and Abdelghani Bekrar[3]

[1] Labo NOCCS, National Engineering School of Sousse, Sousse, Tunisia
rima.zarrouk@gmail.com
[2] Labo LIP2, Faculty of Sciences of Tunis, University of Tunis El Manar, Tunis, Tunisia
[3] LAMIH, University of Valenciennes and Hainaut-Cambrésis, UVHC, Valenciennes, France

Abstract. One of the most challenging problems in manufacturing field is to solve the flexible job shop (FJS) problem subject to machines breakdown. In this paper, we propose two rescheduling solutions to handle machine breakdowns: a PSO-based solution and a shifting-based solution. The first solution aims to improve the robustness while the second solution aims to improve the stability.

Keywords: Flexible job shop · Machine breakdowns · Particle swarm optimization · Scheduling

1 Introduction

The FJS Problem consists in scheduling a set of operations forming jobs on a limited set of machines such that the maximal completion time of all operations is minimized. FJS problem is a strongly NP-hard problem and handling random machine breakdowns further complicates the problem. In the context of machines breakdown, there are two phases: a prescheduling phase (before the breakdown) and a rescheduling phase (after the breakdown). The quality of a rescheduling solution is often measured by three criteria: completion time of all jobs, robustness and stability comparing to the prescheduling solution [2]. Machines breakdown can be handled at priori (preventive) [1], at posteriori (curative) [5–7] or at both stages [3]. In [1] a genetic algorithm with idle-time insertions is proposed. In [5, 6], a genetic algorithm combining right shift strategy (RSS) and route is proposed as a curative solution. In [3] authors use the particle swarm optimization (PSO) and RSS to handle machine breakdowns.

In this paper, two curative solutions are presented: PSO historic route changing (PSO-HRC) and modified shifting strategy (MSS). The first aims to improve the robustness while the second aims to improve the stability. Two assumptions are used: a single machine breakdown and non-resumable mode (i.e. affected operations have to be restarted).

S. Benferhat et al. (Eds.): IEA/AIE 2017, Part I, LNAI 10350, pp. 124–130, 2017.
DOI: 10.1007/978-3-319-60042-0_14

This paper is organized as follows. Sections 2 and 3 define respectively the FJS problem under breakdown machine, and the PSO meta-heuristic. Sections 4 and 5 present respectively the proposed curative solutions and the experimental results. Section 6 concludes the paper.

2 The FJS Problem with Stability and Robustness Criteria

The FJS problem is defined by:

$J = \{J_1, J_2 \ldots J_n\}$ a set of n independent jobs, $M = \{m_1, m_2 \ldots m_k\}$ a set of machines and $O = \{(O_{11}, O_{12}, \ldots), (O_{21}, O_{22}, \ldots) \ldots (O_{n1}, O_{n2}, \ldots)\}$ the set of operations, where O_{ji} is operation i of job j.

The goal is to find a schedule of operations that minimizes the completion times of all jobs (MakeSpan of the schedule), where, C_j is the completion time of job J:

$$MS = Minimize\left[Max\left(C_1, C_2, \ldots .C_j\right)\right] \tag{1}$$

We will use the same definitions of robustness and stability of the rescheduling solution as in [1, 8]. Two formulas are used to measure robustness:

$$RM1 = \frac{MS_r - MS_p}{MS_p} \times 100\% \tag{2}$$

where, MS_p is the makespan of the prescheduling solution and MS_r is the makespan of the rescheduling solution. A schedule is robust if RM1 is low.

$$RM2 = \sum\nolimits_{\mathrm{i}=1}^{O} \frac{Load_m}{Load_{tot}} Pt_i \tag{3}$$

$$Load_{tot} = \sum\nolimits_{\mathrm{m=i}}^{\mathrm{k}} Load_m \tag{4}$$

where, Pt_i is the processing time of the i and $Load_m$ is the workload of theMachine handling operation i. A schedule is robust if RM2 is high.

Three formulas are used to measure stability:

$$SM1 = \sum\nolimits_{j=1}^{n} \sum\nolimits_{i=1}^{qj} \left|c_{O_{jip}} - c_{O_{jir}}\right| \tag{5}$$

$$SM2 = \frac{\sum_{\mathrm{j}=1}^{n} \sum_{i=1}^{qj} \left|C_{O_{jir}} - C_{O_{jir}}\right|}{\sum_{\mathrm{j}=1}^{n} Oj} \tag{6}$$

$$SM3 = \frac{\sum_{j=1}^{n} \sum_{i=1}^{qj} \left| C_{O_{jio}} - C_{O_{jir}} \right|}{\sum_{i=1}^{n} \sum_{j=1}^{qi} AO_j} \tag{7}$$

where, n is the number of jobs, q_j is the number of operations in job j, $c_{O_{ijP}}$ the completion time of O_{ji} in the pre-schedule, $c_{O_{ijr}}$ the completion time of O_{ji} in the re-schedule, and O_j the total number of operations of job j and, AO_i the total number of operations in jobs j affected by the breakdown.

3 The PSO Meta-heuristic

The PSO [3] works by having a population of candidate solutions that are moving around in the search space in order to improve their current solutions. The movements of particles are guided by their own best-known position in the search-space as well as the entire swarm's best-known position. At each instant, each particle p takes a new position vector noted $X_p(t)$, and new velocity vector noted $V_p(t)$, are computed using:

$$V_{p,d}(t+1) = w.V_{p,d}(t) + K_1.r_1\left(Xbest_{p,d}(t) - X_{p,d}(t)\right) + K_2.r_2\left(Xgbest_{p,d}(t) - X_{p,d}(t)\right) \tag{8}$$

$$X_{p,d}(t+1) = X_{p,d}(t) + V_{p,d}(t+1) \tag{9}$$

Where, d is the dimension of vectors, $Xbest_p(t-1)$ is the best position reached by the particle up to time $t-1, Xgbest_p(t)$ is the best position ever found by the whole swarm. r_1 and r_2 are random numbers in the interval [0, 1], K_1 and K_2 are positive constant called respectively the coefficient of the self-recognition component, and the coefficient of the social component. w a dynamic inertia coefficient varying over time [7].

4 The Proposed Rescheduling Solutions

This section presents the two rescheduling approaches (MSS and PSO-HRC). We use the PSO metaheuristic to determine the prescheduling solution that minimizes the total workload and the makespan.

Let X_{pt} be the set of operations that are already triggered before the breakdown and X_{pr} the set of operations not triggered yet. The MSS approach starts by determining: the operation AO_{jim} directly affected by the breakdown, the indirectly affected operations $O_{j(i+1)m'}$ of the same job, and the indirectly affected operations $O_{j'i'm}$ mapped to machine m. Then the MSS performs a guided right shift according to Algorithm_1.

```
Algorithm_1: MSS
```

```
-Prescheduling: run the PSO for solving FJS problem.
-Occurrence of a breakdown on machine m.
-Rescheduling
   For each operation affected AO_jim in X_pr  do
     a. Move AO_jim right to a time period = repair time
     b. Update the start time and the end time of AO_jim
     c. If (O_j'i'm exist) then
          If (there is an idle time after AO_jim that can
          absorb the repair time) then go to (d)
          Else AO_jim <- O_j'i'm , go to (a) End if
        End if
     d. If (O_j(i+1)m' exist) then
          Get the start time denote as Next_st of O_j(i+1)m'
          If (completion of AO_jim >  Next_st) then
            AO_jim <- O_j(i+1)m' , go to (a)
          End if
        End if
   End for
```

In the PSO-HRC approach, a leader historic table is maintained during the prescheduling phase. This table contains the best scheduling solutions reached by leading particles. Once a machine is broken down, the position vector X_p is divided into two parts X_{pt} and X_{pr}, then a search of X_{pt} in the historic_table is performed followed by an update of X_{pr}. Algorithm_2 presents the main steps of PSO-HRC.

```
Algorithm_2: PSO-HRC
```

```
-Prescheduling: run the PSO for solving FJS problem and
save the best schedules in the historic_table.
-Occurrence of a breakdown on machine m.
-Rescheduling
  For each position vector HX_p in the historic_table do
    Split HX_p into 2 vectors HX_pt and HX_pr
    if (HX_pt == X_pt) then
      X_pr <- HX_pr ;
      Calculate the new MS.
      if (new MS < old MS) then update the solution End if
    End if
  End for
```

5 Experimental Results

In this section we present the simulation results of the proposed rescheduling algorithms then we compare them to those obtained by the idle time insertion (ITI) [1] and the random route changing (RRC). The comparison criteria are robustness RM1 (Eq. 2) and RM2 (Eq. 3), stability SM1, SM2 and SM3 (Eqs. 5, 6 and 7), the makespan MS (Eq. 1) and the total workload (Eq. 4). Due to lack of space we present the experimental results only for two FJS problem instances: an instance of 10 machines and 10 jobs with a total flexibility, and an instance of 8 machines and 8 jobs with a partial flexibility. The probability of machine breakdowns, the machine repair period, the occurrence time of the breakdown and the four disruption scenarios are chosen in a similar manner as in [1]. These scenarios are listed in Table 1. Figure 1 shows the prescheduling solutions (i.e. before breakdown) obtained by the PSO algorithm after 20 trials using 500 particles and 500 iterations.

Tables 2, 3 and 5 compare the fitness, the workload and the robustness of the four scheduling algorithms. We notice the out performance of PSO-HRC. RRC and ITI give best result if there is only one affected operation with a short repair time (as in SN2 and SN4). Tables 4 and 6 compares the stability of the four rescheduling algorithms. We notice that MSS gives the best stability but not the best fitness.

Table 1. Breakdown scenarios

FJS problem instance	Problem1: 8*8				Problem2: 10*10	
Broken machine *m*	1				2	
Disruption scenarios	SN_1	SN_2	SN_3	SN_4	SN_1	SN_2
Occurrence time	1	5	3	5	2	2
Repair period	2	2	5	5	1	4
Directed AO_{jim}	$O_{8,1,1}$	$O_{6,3,1}$	$O_{5,1,1}$	$O_{6,3,1}$	$O_{4,2,2}$	$O_{4,2,2}$
Number of AO_{ij}	13	1	8	1	16	16

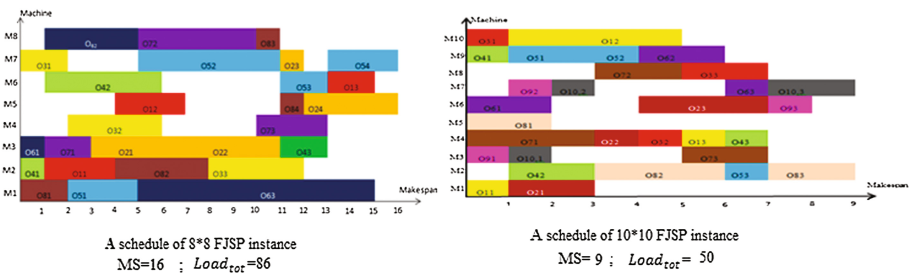

Fig. 1. Prescheduling schedules of the FJS instances

Table 2. Problem 8*8, SN_1 & SN_2: fitness, robustness, workload and slack time

	SN_1				SN_2			
	MS	$Load_{tot}$	RM1	RM2	MS	$Load_{tot}$	RM1	RM2
PSO-HRC	**17**	**73**	**6.25**	**9.84**	**16**	**75**	**0**	**9.84**
RRCA	**17**	**73**	**6.25**	**9.84**	17	73	6.25	9.84
MSS	19	86	18.75	11.3	17	86	6.25	11.3
ITI	27	86	68.75	11.3	17	86	6.25	11.3

Table 3. Problem 8*8, SN_3 & SN_4: fitness, robustness, workload and slack time

	SN_3				SN_4			
	MS	$Load_{tot}$	RM1	RM2	MS	$Load_{tot}$	RM1	RM2
PSO-HRC	**20**	**77**	**25**	**9.84**	**16**	**76**	**0**	**9.84**
RRCA	21	73	31.25	9.84	17	73	6.25	9.84
MSS	22	86	37.5	11.3	20	86	25	11.3
ITI	37	86	131.25	11.3	20	86	25	11.3

Table 4. Problem 8*8: stability

	SN_1			SN_2			SN_3			SN_4		
	SM1	SM2	SM3	SM1	SM2	SM3	SM1	SM2	SM3	SM1	SM2	SM3
PSO-HRC	60	2.22	4.61	20	0.74	20	59	2.18	7.37	12	0.44	12
RRCA	60	2.22	4.01	28	1.03	28	71	2.62	7.88	25	0.92	25
MSS	29	1.07	1.93	2	0.07	2	48	1.77	5.33	5	0.18	5
ITI	86	3.18	5.73	2	0.07	2	123	4.55	13.66	5	0.18	5

Table 5. Problem 10*10, SN_1 & SN_2: fitness, robustness, workload and slack time

	SN_1				SN_2			
	MS	$Load_{tot}$	RM1	RM2	MS	$Load_{tot}$	RM1	RM2
PSO-HRC	**9**	**41**	11.11	5.5	**11**	**47**	**11.11**	**5.5**
RRCA	15	41	66.66	6.9	12	41	20	6.9
MSS	10	50	22.22	4.7	18	50	80	4.7
ITI	13	50	30	4.7	28	50	211.11	4.7

Table 6. Problem 10*10: stability

	SN_1			SN_2		
	SM1	SM2	SM3	SM1	SM2	SM3
PSO-HRC	23	0.76	3.27	31	1.03	4.42
RRCA	46	1.53	7.66	68	2.26	8.5
MSS	8	0.86	1.33	24	0.8	3
ITI	15	0.5	2.5	76	2.53	9.5

6 Conclusions

In this paper, we proposed two rescheduling solutions (PSO-HRC and MSS) to solve the FJS problem under machine breakdowns. Comparing to other solutions, the PSO-HRC provides better makespan, better workload and better robustness. While the MSS offers a better stability. In our future work we will target other types of machine breakdowns such as the partial breakdowns and the breakdowns causing changes on the durations of operations.

References

1. Nasr, A.-H., El Mekkawy, T.Y.: Robust and stable flexible job shop scheduling with random machine breakdowns using a hybrid genetic algorithm. Int. J. Prod. Economics **132**, 279–291 (2011)
2. Xiong, J., Xing, L., Chen, Y.: Robust scheduling for multi-objective flexible job-shop problems with random machine breakdowns. Int. J. Prod. Economics **141**, 112–126 (2013)
3. Singh, M.R., Mahapatra, S.S.: Robust scheduling for flexible job shop problems with random machine breakdowns using a quantum behaved particle swarm optimization. Int. J. Serv. Oper. Manage. **20**(1), 1–20 (2015)
4. He, W., Sun, D.: Scheduling flexible job shop problem subject to machine breakdown with route changing and right-shift strategies. Int. J. Adv. Manuf. Technol. **66**, 501–514 (2012)
5. He, W., Sun, D.: Scheduling flexible job shop problem subject to machine breakdown with game theory. Int. J. Prod. Res. **52**(13), 3858–3876 (2013)
6. Yahyaoui, A., Fnaiech, N., Fnaiech, F.: New shifting method for job shop scheduling subject to invariant constraints of resources availability. In: 35th Annual Conference of IEEE Industrial Electronics, pp. 3211–3216 (2009)
7. Modares, H., Alfi, A., Sistani, M.B.N.: Parameter estimation of bilinear systems based on an adaptive particle swarm optimization. Eng. Appl. Artif. Intell. **23**(7), 1105–1111 (2010)
8. Xiong, J., Xing, L.N., Chen, Y.W.: Robust scheduling for multi-objective flexible job-shop problems with random machine breakdowns. Int. J. Prod. Economics **141**(1), 112–126 (2013)