

Job-shop scheduling with machine breakdown prediction under completion time constraint

Łukasz Sobaszek
Lublin University of Technology,
Nadbystrzycka 38 D,
20-618 Lublin, Poland
Email: l.sobaszek@pollub.pl

Arkadiusz Gola
Lublin University of Technology,
Nadbystrzycka 38 D,
20-618 Lublin, Poland
Email: a.gola@pollub.pl

Edward Kozłowski
Lublin University of Technology,
Nadbystrzycka 38 D,
20-618 Lublin, Poland
Email: e.kozlovski@pollub.pl

□ **Abstract** — This paper discusses the problem of time-constrained job-shop scheduling with technological machine breakdown prediction. The first section gives short characteristics of the field of research, and describes the effects of machine failure on job completion times. Secondly, the discussed problem is represented by means of mathematical equations and solved with original algorithms for machine failure prediction and implementation of redundant service times, and finally, the proposed solutions are verified by means of simulation. In the computational experiment stage a typical production case with taking into account 3 machines failure was considered. The last part of this paper draws conclusions from the study and presents directions of future research work.

I. INTRODUCTION

Job scheduling has been receiving considerable attention of researchers [1]–[2]. Scheduling problems are approached from the perspective of production systems [3]–[4], dynamic character and randomness [5], time-dependence [6]–[7], and its relevance to industrial conditions [7]. In each of its aspects scheduling is governed by a variety of constraints that must be accounted for in scheduling.

We may distinguish two broad categories of constraints considered in scheduling: Resource-Constrained Scheduling and Time-Constrained Scheduling [8]–[9]. Although this division concerns mainly the field of project scheduling [10]–[11], it may be nonetheless found in scheduling of production as well [12]–[14].

Numerous constraints are discussed in the job-shop environment, yet they tend to be included as limiting constraints, for the sake of simplification of given scheduling problems [15]. Therefore, Robust Scheduling has been attracting an increasing amount of focus as it addresses the presence and the negative effect of various conditions and factors influencing the execution of production jobs [16].

II. COMPLETION TIME AND MACHINE FAILURE EFFECT

Scheduling production jobs is most frequently considered under the objective function C_{max} – makespan/completion time of all jobs. This parameter is implemented in both

analysis of test problems (aimed at evaluating the effectiveness of solutions) [17], as well as in determining job completion times [2], [6]. In practical industrial applications scheduling must strictly follow the C_{max} as exceeding these times and missing order delivery times may enforce contractual fines and lead to losing customers [2]. It is therefore a typical time constraint.

Apart from time constraints, there is a wide variety of factors that may only be addressed and resolved by implementing suitable methods and solutions. Scheduling Under Machine Failure is one of the most frequently considered factors, which is of considerable importance to observing order deadlines and therefore to time-constrained scheduling, as downtime of one technological machine may negatively affect the schedule through delaying jobs (Fig. 1).

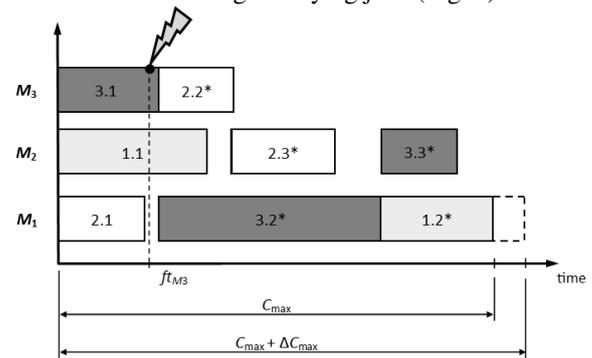


Fig. 1 M_3 machine failure and its impact on makespan (* – delayed jobs)

III. PROBLEM FORMULATION

The job-shop scheduling problem consists in assigning jobs from the set of jobs $J = \{J_1, J_2, \dots, J_n\}$ to the set of available machines $M = \{M_1, M_2, \dots, M_m\}$ so that the schedule is optimised according to the objective function. Processing job J_j on machine M_i is described as operation. Simultaneously, we must consider the technology of processes, which is described by the matrix of machine orders $MO = [o_{ij}]$. Times of particular operations are described by the matrix of processing times $PT = [pt_{ij}]$. The size of matrices MO and PT is $m \times n$ [15].

□ This work was not supported by any organization

In order to implement the machine failure constraint, each machine must be described by means of the following sets:

- $FT_{Mi} = \{ft_{Mi1}, ft_{Mi2}, \dots, ft_{Min}\}$, describing potential failure times of machines (expressed in hours),
- $P_{Mi} = \{p_{Mi1}, p_{Mi2}, \dots, p_{Min}\}$ describing the probability of machine failure,
- $TB_{Mi} = \{tb_{Mi1}, tb_{Mi2}, \dots, tb_{Min}\}$ defining time buffers (length of potential service times in minutes).

Moreover, implementing job completion time constraints requires that the schedule based on the set of feasible solutions accounts for the machine failure and adheres to the objective $C_{max} = C_{max} + \min(\Delta C_{max})$.

IV. SUGGESTED PROBLEM SOLUTION

The solution we propose to the scheduling problem under time and machine failure constraints employs algorithms based on actual historical data, which are recorded in the sets of failure times $T_{Mi} = \{t_1, t_2, \dots, t_n\}$ and repair times $RT_{Mi} = \{rt_1, rt_2, \dots, rt_n\}$. Each set is specified individually for each machine.

A. Failure prediction algorithm

The first algorithm defines the key times in the schedule where failure may occur. Executed algorithm produces information regarding failure times (FT_{Mi}) and failure occurrence probability (P_{Mi}). The subsequent steps of the algorithm are as follows:

1. Define machine M_i and load data from set T_{Mi} .
2. Determine sequence and sort observations in an increasing order:

$$\{(t_i, d_i)\}_{1 \leq k \leq n} \quad (1)$$

where: t_i – failure times,
 d_i – number of occurrences.

3. Filter data – delete outliers.
4. Estimate the survival function based on:

$$\hat{S}(t) = \begin{cases} 1, & \text{dla } t < t_1 \\ \prod_{t_i \leq t} \frac{r_i - d_i}{r_i}, & \text{dla } t_1 < t \end{cases} \quad (2)$$

where: r_i – the total number of failures expressed by:

$$r_i = \sum_{j=i}^k d_j \quad (3)$$

5. From the survival function determine failure times and failure probability (Fig. 2).

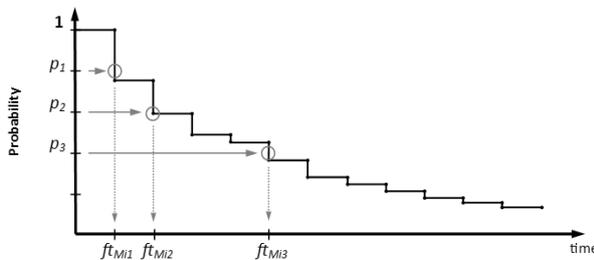


Fig. 2 Determining failure times from survival probability

6. Determine probability of failure:

$$p_{Mij} = 1 - p_i \quad (4)$$

7. Save data in sets FT_{Mi} and P_{Mi} .

Execution of this algorithm enables to pinpoint the times in the schedule where redundant time buffers should be implemented.

B. Algorithm for estimation and implementation of redundant time buffers

The second proposed algorithm determines the span of the service times and their implementation in order to make the schedule robust to failures. The algorithm is composed of the following steps:

1. Load observations from RT_{Mi} and sort in an increasing order.
2. Filter data – delete outliers.
3. Divide into two subsets according to:

$$RT_{Mi} = \{RT_{Mi1}, RT_{Mi2}, \dots, RT_{Mi8}\} \quad (5)$$

where: RT_{Mij} – subset of repair times, and:

$$RT_{Mi1} = \{rt_i \in (0; 60>\}$$

$$RT_{Mi2} = \{rt_i \in (60; 120>\}$$

...

$$RT_{Mi8} = \{rt_i \in (420; 480>\}$$

4. Define service buffer times – determine subset of RT_{Mi} of maximum weight – determine auxiliary sets RT'_{Mi} and TB'_{Mi} :

$$RT'_{Mi} = \{\overline{\overline{RT_{Mi1}}}, \overline{\overline{RT_{Mi2}}}, \dots, \overline{\overline{RT_{Mi8}}}\} \quad (6)$$

$$\bigvee_{\max(RT'_{Mi})} TB'_{Mi} = RT_{Mij} \quad (7)$$

$$TB'_{Mi} = \{rt_i \in \langle \min(RT_{Mij}); \max(RT_{Mij}) \rangle\} \quad (8)$$

where: RT_{Mij} – maximum weighted subset.

5. Estimate service time buffers:

$$TB_{Mi} = \{tb_{Mi1}, tb_{Mi2}, \dots, tb_{Min}\} \quad (9)$$

$$tb_{Mij} = \frac{\max(TB'_{Mi})}{n_p} \cdot j \quad (10)$$

where: j – number of element of set TB_{Mi} ,
 n_p – number of considered levels of probability (elements of set P_{Mi}).

6. Take (p_{Mij}, ft_{Mij}) from sets FT_{Mi} and P_{Mi} .

7. Determine control probability:

$$p_{kMi1} = \frac{l}{n_p} \quad (11)$$

where: $l = \langle 1, 2, \dots, n_p \rangle$.

8. Select buffer according to:

IF $p_{Mij} \leq p_{kMi1}$
 select buffer tb_{Mi1} (minimal)
ELSE IF $p_{Mij} > p_{kMi1}$ and $p_{Mij} \leq p_{kMi2}$
 select buffer tb_{Mi2} (where $tb_{Mi2} > tb_{Mi1}$)
 ...

```

ELSE IF  $p_{Mi j} > p_{kMi (n-1)}$  and  $p_{mi} \leq p_{kMin}$ 
    select buffer  $tb_{Min}$  (where  $tb_{Min} > tb_{Mi (n-1)}$ )
END IF
    
```

- Implement selected buffers of set TB_{Mi} in points defined by elements of set FT_{Mi} – in case buffers double – select higher.

The presented algorithm determines time buffers which make the schedule robust to failure of technological machines.

C. Schedule optimisation for makespan

As a result of execution of proposed algorithms the obtained schedule becomes robust. However, in order to meet the specified deadlines for orders, the most suitable schedule should be selected. Hence, the schedule selected from the group of solutions will be the one that offers the optimum solution to $\min(\Delta C_{max})$ constraint. In the following case then, the best scheduling methods will be the exact ones and methods based on expert knowledge.

V. SIMULATION TESTS

The proposed solutions were verified by executing the scheduling process in job-shop conditions. The scheduling concerned 8 production orders processed on a stock of 5 machine tools. The processing times, the job order, and the number of jobs assigned to machines were randomly generated with available software, on the basis of the following assumptions:

- processing times cannot exceed one shift (maximum processing time is 7.5 h),
- machine loading is specified at 75%,
- machine routing is predetermined and not subject to change.

Moreover, the schedule allowed for the failure of 3 technological machines. The machine failure data was obtained from actual historical data of a production company. The source of information about machines failure times were electronic forms collected by maintenance department. In the data-treatment process authors used RStudio software with selected libraries. 3 failure probability levels were considered: $p_{Mi1} = 0.25$, $p_{Mi2} = 0.5$ and $p_{Mi3} = 0.75$. The results of the executed prediction algorithm are shown in Table 1.

TABLE I. RESULTS OF MACHINE FAILURE PREDICTION ALGORITHM EXECUTION

Probability level	Predicted failure times [h]		
	M_1	M_2	M_3
$p_{Mi1} = 0.25$	$\hat{f}_{M11} = 8$	$\hat{f}_{M21} = 8$	$\hat{f}_{M31} = 8$
$p_{Mi2} = 0.50$	$\hat{f}_{M12} = 16$	$\hat{f}_{M22} = 16$	$\hat{f}_{M32} = 24$
$p_{Mi3} = 0.75$	$\hat{f}_{M13} = 40$	$\hat{f}_{M23} = 32$	$\hat{f}_{M33} = 48$

Execution of time buffer algorithm determined redundant service buffers and their implementation times (Table 2).

TABLE II. DETERMINED SERVICE BUFFERS

Implementation time	Buffer length [min.]		
	M_1	M_2	M_3
after 8 h	$tb_{M11} = 20$	$tb_{M21} = 20$	$tb_{M31} = 20$
after 16 h	$tb_{M12} = 40$	$tb_{M22} = 40$	$tb_{M31} = 20$
after 24 h	$tb_{M11} = 20$	$tb_{M21} = 20$	$tb_{M32} = 40$
after 32 h	$tb_{M12} = 40$	$tb_{M23} = 60$	$tb_{M31} = 20$
after 40 h	$tb_{M13} = 60$	$tb_{M21} = 20$	$tb_{M31} = 20$
after 48 h	$tb_{M12} = 20$	$tb_{M22} = 40$	$tb_{M33} = 60$

The effectiveness of the proposed solutions was assessed by means of the following criteria:

- criterion C_{max} ,
- criterion y_j – the number of critical operations in reference to technology (operations of one job),
- criterion y_M – the number of critical operations in reference to machines.

The conducted verification tests implemented the following popular dispatching rules: FCFS, EDD, LPT and SPT. In the computational experiment stage LiSA software (*Library of Scheduling Algorithms*) was used. The tests were carried out on a typical PC-class computer. Subsequently, the obtained robust schedule was optimised under the specified objective function. The results of the calculations are shown in Tables 3–4.

TABLE III. VALUES OF CRITERION C_{MAX}

Dispatching rules	C_{max} value [h]		
	Nominal schedule	Robust schedule	Optimized schedule
LPT	48	67	47.83
SPT	46	54.50	
FCFS	43.5	56.50	
EDD	49	63.17	

Implementation of service time buffers leads to delaying the makespan (Table 3). The mean delay is 14.77 h, which amounts to approx. 2 shifts. The delay in completion time of all jobs, however, increases the stability of production and indicates the feasible production completion time, simultaneously accounting for technological machine failure.

Determination of the actual production completion time allows preventing potential contractual fines. In the case of strictly specified order delivery deadline, implementation of the proposed algorithms and optimisation with the use of expert knowledge produces a robust algorithm under the specified completion time constraint. In the presented problem the schedule was optimised according to the defined objective function, in which case the makespan generally became slightly delayed, and in 2 cases was shortened; therefore the obtained schedules did meet the constraint $\min(\Delta C_{max})$.

TABLE IV. VALUES OF CRITICAL OPERATIONS

Dispatching rules	Number of critical operations [-]					
	Nominal schedule		Robust schedule		Optimized schedule	
	y_J	y_M	y_J	y_M	y_J	y_M
<i>LPT</i>	17	20	3	12	4	12
<i>SPT</i>	11	17	6	9	5	10
<i>FCFS</i>	8	20	4	11	4	10
<i>EDD</i>	9	18	4	10	4	10

The robust schedule was characterised by the decreased number of critical operations – both in terms of processed jobs and operations on particular machines (Table 4). Both robust and optimised schedules showed a nearly 50% drop in the number of such operations, compared to the nominal schedule. The abovementioned leads to increasing the stability of production and reducing uncertainty and nervousness, as even if the failure of the machine should occur, the allocated service time absorbs the potential negative impact of failure on processing subsequent operations.

VI. CONCLUSION

It must be remarked that scheduling production jobs in industrial applications is inherently connected with a wide range of potentially disruptive factors, which ought to be treated as constraints. This paper presented scheduling under constraint of completion time of all jobs and machine failure. The proposed algorithms were verified in simulation, which proved their effectiveness and applicability. Future research works should be continued and could incorporate other factors occurring in job scheduling, such as alternating processing times, transport between workstations, *etc*). The proposed algorithms and methods could be implemented in the real production environments – at the managerial and production planning departments for instance.

REFERENCES

- [1] M. T. Jensen, "Robust and Flexible Scheduling with Evolutionary Computation," Aarhus, 2001.
- [2] L. Sobaszek, A. Gola, A. Świć, "Predictive scheduling as a part of intelligent job scheduling system," in Intelligent Systems in Production Engineering and Maintenance – ISPEM 2017: proceedings of the First International Conference on Intelligent Systems in Production Engineering and Maintenance ISPEM 2017, D. Mazurkiewicz, A. Burduk, Ed. Switzerland, 2018, pp. 358–367.
- [3] J. Louis, Zhijie Xu, "Genetic Algorithms for Open Shop Scheduling and Re-Scheduling," Departament of Computer Science, University of Nevada, 1999.
- [4] In-Chan Choi, Dae-Sik Choi, "A Local Search Algorithm for Jobshop Scheduling Problems with Alternative Operations and Sequence-Dependent Setups," Computers & Industrial Engineering, 42 (2002), pp. 43–58.
- [5] P. Sharma, A. Jain, "Performance analysis of dispatching rules in a stochastic dynamic job shop manufacturing system with sequence-dependent setup times: Simulation approach," CIRP Journal of Manufacturing Science and Technology, Vol. 10, 2015, pp. 110–119.
- [6] M. Pawlak, "Algorytmy ewolucyjne jako narzędzie harmonogramowania produkcji," Wydawnictwo Naukowe PWN, Warszawa, 1999.
- [7] Yu. N. Sotskov, N. Yu. Sotskova, Lai T.-C., F. Werner, "Scheduling under Uncertainty – Theory and Algorithms," Belorusskaya nauka, Minsk, 2010.
- [8] J. S. Norwood, "An evaluation of the time constrained and resource constrained scheduling features of commercially available project management software," Naval Postgraduate School, Monterey, California, 1996.
- [9] M. Klimek, "Priority algorithms for the problem of financial optimisation of a multi stage project," Applied Computer Science, vol. 13, no. 4, 2017, pp. 20–34.
- [10] P. Peczynski, "Scheduling Constraints," Department of Computer Science, Saarland University, Germany, 2005.
- [11] S. Van de Vonder, E. Demeulemeester, W. Herroelen, "Proactive Heuristic Procedures for Robust Project Scheduling: An Experimental Analysis," European Journal of Operational Research, 189 (2008), pp. 723–733.
- [12] Hung-Kai Wang, Chen-Fu Chien, Che-Wei Chou, "An empirical study of bio manufacturing for the scheduling of hepatitis in vitro diagnostic device with constrained process time window," Computers & Industrial Engineering, Volume 114, 2017, pp. 31–44.
- [13] A. M. Aguirre, L. G. Papageorgiou, "Resource-constrained formulation for production scheduling and maintenance," in Computer Aided Chemical Engineering, A. Espuña, M. Graells, L. Puigjaner, Ed. Elsevier, vol. 40, 2017, pp. 1375–1380.
- [14] E. Gebennini, L. Zeppetella, A. Grassi, B. Rimini, "Production scheduling to optimize the product assortment in case of constrained capacity and customer-driven substitution," IFAC – PapersOnLine, vol. 48(3), 2015, pp. 1954–1959.
- [15] L. Sobaszek, A. Gola, E. Kozłowski, "Application of survival function in robust scheduling of production jobs," 2017 Federated Conference on Computer Science and Information Systems (FedCSIS), Prague, 2017, pp. 575–578.
- [16] Jian Xiong, Li-ning Xing, Ying-wu Chen, "Robust Scheduling for Multi-Objective Flexible Job-Shop Problems with Random Machine Breakdowns," International Journal of Production Economics, vol. 141(1), 2013, pp. 112–126.
- [17] M. T. Jensen, T. K. Hansen, "Robust solutions to Job Shop problems," The 1999 Congress on Evolutionary Computation, July 1999, pp. 1138–1144.