Operations Research Forum (2023) 4:52
https://doi.org/10.1007/s43069-023-00233-4

ORIGINAL RESEARCH

Sensitivity Analysis of the Unrelated Parallel Machine Scheduling Problem with Rework Processes and Machine Eligibility Restrictions

Javad Rezaeian[1] · Reza Alizadeh Foroutan[1] · Toraj Mojibi[2] · Yacob Khojasteh[3]

Received: 14 January 2022 / Accepted: 2 June 2023 / Published online: 21 June 2023

Abstract

This paper deals with an unrelated parallel machine scheduling problem considering machine eligibility restrictions and release times. Setup times are both operation sequence and machine dependent, and defective items are possible to occur. In order to retrieve imperfect items and to improve them to meet an acceptance level of quality, rework processes are considered. A developed optimization model is used to minimize the makespan. Since the considered system involves several parameters, to understand how changes in parameters may affect the output of the model and to address the types of questions arise from parameter variations, it is necessary to perform sensitivity analysis. It is a hard task to track the effect of all parameters and their interactions. Therefore, this paper focuses on parameters which are recognized the most appropriate to perform sensitivity analysis. Hence, several scenarios and cases including some test problems are designed and the computational results are analyzed.

Keywords Sensitivity analysis · Unrelated parallel machines · Rework processes · Sequence- and machine-dependent setup times · Machine eligibility

1 Introduction

Scheduling problems are an inseparable part of every manufacturing and service system. As one of the most practical problems in today's world, scheduling problems have a considerable impact on increasing the productivity. Parallel machine

✉ Yacob Khojasteh
khojast@sophia.ac.jp

[1] Department of Industrial Engineering, Mazandaran University of Science and Technology, Babol, Iran

[2] Department of Management, Firoozkooh Branch, Islamic Azad University, Firoozkuh, Iran

[3] Graduate School of Global Studies, Sophia University, Tokyo, Japan

scheduling problems are generally categorized into three classes: (1) identical machines (P_m), where all machines operate at the same processing time on a job; (2) uniform machines (Q_m), where there are machines with different speeds but a constant rate for all jobs per machine; and (3) unrelated machines (R_m), where each machine is able to operate at different rates and each job can be processed with different times on each machine; i.e., the processing time of a job is both machine and job dependent [1]. This paper deals with the concept of sensitivity analysis. In this regard, unrelated parallel machine scheduling problem with operation sequence– and machine-dependent setup times where rework processes are allowed and machine eligibility restrictions and release time are considered is investigated.

The remainder of this paper is organized as follows. In Sect. 2, the relevant literature is reviewed. In Sect. 3, the mathematical model is described. Section 4 explains the solution representation method employed in this study. The numerical examples used are given in Sect. 5. In Sect. 6, sensitivity analysis procedure is discussed and the analysis of the sensitivity of the model with regard to some parameters is presented. Eventually, conclusions and future work suggestions are presented in Sect. 7.

2 Literature Review

Unrelated parallel machine scheduling problem has been a favorite field of research during the last decade and is widely applicable in all manufacturing and service providing systems. In such case, a set of n jobs are to be processed using a set of m parallel machines. Each job can only be processed on one machine and each machine can only process one job at a time. Machines are able to operate at different rates and the processing time of each job on each machine may vary; i.e., the processing time of a job is both machine and job dependent [1]. As one of the recent studies in the field of unrelated parallel machines, the study by Pei et al. [2] can be referred. They provided a concise guide to scheduling problems with learning and deteriorating effects. They designed a new classification scheme based on levels of different domains, namely, effects, processing ways, processing time functions, and manufacturing environments. According to their proposed scheme, at the first stage, the scheduling problems are distinguished into three categories: learning effects, deteriorating effects, and combined effects. In the next stage, models are refined along three lines in each category, which are general processing way, batch scheduling, and group scheduling. Then, they analyzed the evolvement of related scheduling models and conducted a critical analysis on the proposed algorithms by combining the attributes of actual processing time functions and manufacturing environments [2].

Many studies in the literature ignore setup time or include it in processing time [1]. That is considered when setup time is slight compared to processing time, or sequence of jobs is not dependent on setup times. Even though this assumption simplifies the analysis of the scheduling problem, it affects the solution quality in a negative manner and makes the model inapplicable for real-life implementations, since many applications need explicit treatment of setup times [3]. Pei et al. addressed a serial-batching scheduling problem with arbitrary release times to minimize the makespan, where the position of the current job and the sum of previous jobs' processing times determine each job's actual processing time. They took the effects of learning and deterioration into consideration at the same time. By combining society and civilization (SC) algorithm with variable neighborhood search (VNS), they presented a novel hybrid SC-VNS algorithm to solve the studied problem. Computational experiments indicated the superiority of the proposed solution approach compared to the other algorithms examined [4].

Sequence- and machine-dependent setup times are dependent on both the job currently being scheduled and the last scheduled job. They usually occur while switching between different types of jobs on machines. In this situation, setup time depends on the type of job that is processed on the machine and also the type of previous job which has been already processed on the machine. This kind of setup time can be found in some manufacturing processes and service provider systems. Some of its industrial applications are drilling operations for printed circuit board fabrication, dicing operations for semiconductor wafer manufacturing, and some main operations in textile manufacturing like warp making, weaving, dyeing, and cloth cutting [5]. Despite the importance of unrelated parallel machine problem with sequence- and machine-dependent setup times, it has not been thoroughly investigated in the literature. Among those who dealt with this problem, Lin and Ying [5] can be mentioned. They proposed a hybrid artificial bee colony algorithm to solve the manufacturing unrelated parallel machine scheduling problem with machine-dependent and job sequence–dependent setup times trying to minimize the makespan. Manupati et al. [6] developed a multi-objective-based evolutionary artificial immune non-dominated sorting genetic algorithm (AI-NSGA-II) for handling unrelated parallel machine scheduling problem with release dates and sequence- and machine-dependent setup times to minimize the total completion time, the number of tardy jobs, the total flow time, and the machine load variation. The quality of their proposed method was ensured in comparison with two other evolutionary algorithms. Zhang et al. [7] proposed a combinatorial evolutionary algorithm to solve the unrelated parallel machine scheduling problem with worker resources, learning effect, and sequence- and machine-dependent setup times.

Lots of scheduling problems are dedicated to the case where all machines are allowed to process all jobs with no restriction, whereas the real-world conditions are in conflict with this assumption in most cases. Generally, a subset of machines is allowed to process an individual job, not all of them. This constraint is usually referred to as processing set or machine eligibility restriction in the literature of scheduling problems. Food processing plants, wireless communication networks, hospital operating room management, and steel production industries are some examples of practical applications of scheduling problems with machine eligibility restrictions in manufacturing environments. Bektur and Saraç [8] studied unrelated parallel machine scheduling problem with sequence-dependent setup times, common server, and machine eligibility. Tabu search (TS) and simulated annealing (SA) were used to minimize the total weighted tardiness. Attempting to minimize the makespan, Afzalirad and Rezaeian [9] addressed a scheduling problem of unrelated parallel machine in which resource constraints, machine eligibility, precedence constraints, release dates, and sequence-dependent setup times were taken into consideration. Genetic algorithm (GA) and artificial immune system (AIS) were developed to find optimal or near optimal solutions. Yunusoglu and Topaloglu Yildiz [10] investigated multi-resource-constrained unrelated parallel machine scheduling problem with the aim of minimizing maximum completion time and developed an exact solution approach based on constraint programming (CP). Sequence-dependent setup times, precedence relations, machine eligibility restrictions, and release dates were the restrictions involved. Foroutan et al. [11] investigated unrelated parallel machine scheduling problem to minimize costs of tardiness and hiring workers. Sequence- and machine-dependent setup times, precedence constraints, and machine eligibility were considered. Multi-objective simulated annealing (MOSA), multi-objective tabu search (MOTS), and a hybrid evolutionary algorithm were employed to deal with large size instances.

In manufacturing environments, many reasons such as machine incompetence, instability of manufacturing systems, inadequacy of preventative maintenance, and human errors make it reasonable to consider the possibility of producing defective jobs. Hence, rework processes are taken into account in order to retrieve imperfect jobs. Regarding rework procedures, semiconductor, steel, and glass are proper examples of process industries where many practical applications can be found [12]. Rework assumption in unrelated parallel machine scheduling problems has been considered by many researchers in recent years. Ramezanian and Saidi-Mehrabad [13] addressed a multi-product unrelated parallel machine scheduling problem considering the probability of producing imperfect jobs. They formulated a mixed-integer non-linear programming model with the aim of minimizing makespan. Considering random reworks, Wang et al. [14] dealt with unrelated parallel machine scheduling to minimize expected total weighted tardiness and total weighted completion time. Approximate methods based on priority rule, GA, and sensitivity analysis (SA) were proposed to handle medium- and large-scale problems. Rambod and Rezaeian [15] addressed an unrelated parallel machine scheduling problem with rework processes, setup times, and machine eligibility restrictions. They formulated the problem mathematically and presented metaheuristic algorithms including a genetic algorithm and two bee algorithms (BA1 and BA2) to find optimal/suboptimal solutions.

Every statistical model consists of a series of equations, input factors, parameters, and variables aiming to describe the process under investigation. The input depends on a number of uncertainty sources such as absence of information, errors of measurement, and poor or partial understanding of the driving forces and mechanisms. Such uncertainty limits the confidence in the output of the model. In addition, models may have to deal with the natural inherent variableness of the system, like the happening of stochastic occurrences [16]. Sensitivity analysis is utilized to clarify how the output of any given model depends on its input parameters. Therefore, SA is a significant technique to test the quality of given models. It is also an effective tool to check the reliability and robustness of any solution [17]. In other words, sensitivity analysis is necessary to understand the effect of a given parameter variation on the output of a model. Nevertheless, there are few studies on sensitivity analysis of scheduling problems. Kolen et al. [18] studied sensitivity of some dispatching rules to changes of the operation processing times in a parallel machine environment.

Penz et al. [19] studied the performance of static scheduling policies under the influence of online disturbances. They proved that the independence of tasks can assure the sensitivity not to exceed the square root of the magnitude of the perturbation. Hall and Posner [20] carried out a review on sensitivity analysis applied in scheduling problems. They tried to devise a new optimal scheduling for problems with modified parameters by applying the optimal schedule on the original problem. To the authors' knowledge, there have been no studies focusing on sensitivity analysis of unrelated parallel machines. Therefore, in this research, sensitivity analysis is performed on the scheduling problem of unrelated parallel machines considering the possibility of rework, sequence- and machine-dependent setup times, release time, and machine eligibility constraints with the aim of minimizing makespan. To do so, the mathematical model suggested by Rambod and Rezaeian [15] is employed. This analysis tries to make a better understanding of the effect of parameter variations and their interactions on the output of the model. Injection molding departments and LCD manufacturing processes are two well-known applications of the model under investigation in production environments, where unrelated parallel machines are applied to process different components and sequence- and machine-dependent setup times are frequently observed [21].

3 Problem Formulation

There are n independent jobs to schedule on a set of m unrelated parallel machines. Each machine can process only one job at a time and the process of each job is allowed to be done on only one machine. At time zero, not all jobs are ready to be processed. The probability of imperfect items is assumed. Hence, a finite number of rework activities are employed to improve imperfect items to meet an acceptance level of quality. For each job, the first operation is deterministic and referred to as the work process, whereas the next ones are probabilistic and mentioned as rework processes. At any time, only one operation per job is permitted. Processing times, including inspection times, are considered deterministic. There are setup actions between successive operations such that setup times are dependent on both operation sequence and machine. Each machine can process some of the jobs; therefore, for any given job, only a subset of them is capable of processing. Machine breakdown is not considered, so all machines can operate any time. Also, job preemption is not permitted (see Fig. 1).

3.1 Indices

n number of jobs

m number of machines

i index of machines ($i = 1, \dots, m$)

j, h indices of jobs ($j, h = 1, \dots, n + m$)

l, l' indices of process. $l,\ l' = \begin{cases} 1 & \text{main process} \\ 2, \ldots,\ L_j & \text{rework process} \end{cases}$

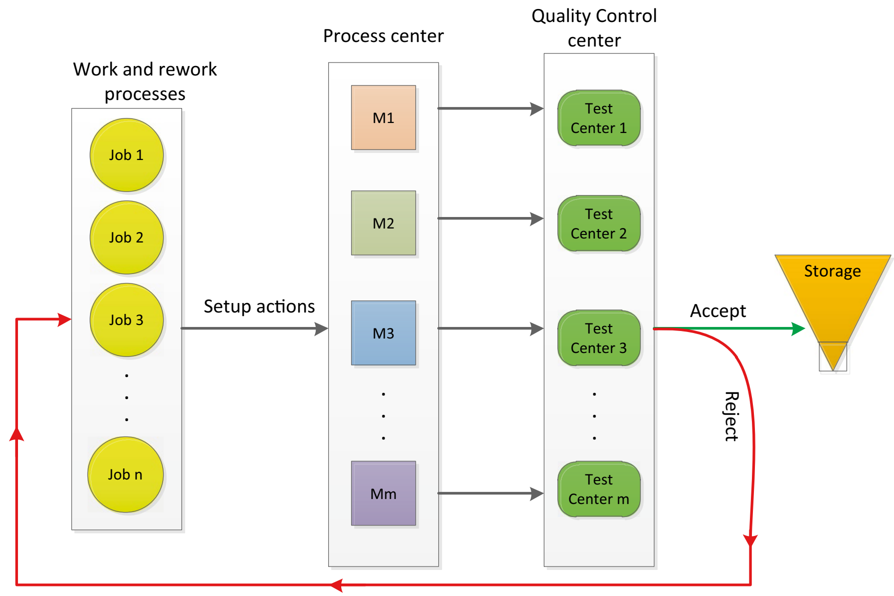


Fig. 1 The illustration of the proposed problem

3.2 Parameters

p_{ijl} Defective probability of operation l of job j on machine $i(p_{ijL_j} = 0)$. Job j is permitted for processing (both main and rework processes) at most L_j times
t_{ij} main processing time of job j on machine
$i\alpha_{ijl}$ decreasing coefficient of processing time of rework process $l(0 \le \alpha_{ijl} \le 1$
pt_{ijl} processing time of process l of job j on machine$(pt_{ij1} = t_{ij},\ pt_{ijl} = pt_{ij(l-1)}a_{ijl})$
$su_{ihl'jl}$ setup time required to switch from operation l' of job h to operation l of job j on machine i

$$Z_{ijl} = \begin{cases} 1 \text{ if operation } l \text{ of job } j \text{ can be processed onmachine } i \\ 0 \qquad\qquad\qquad\qquad \text{otherwise} \end{cases}$$

R_j release time of job j
M an arbitrary large number

3.3 Decision Variables

S_{jl} Start time of process l of job j
C_{jl} Completion time of process l of job j
C_j Completion time of job j

$$x_{ijl} = \begin{cases} 1 & \begin{array}{l}\text{if operation } l \text{ of job } j \text{ is} \\ \text{processed on machine } i\end{array} \\ 0 & \text{otherwise} \end{cases}$$

$$y_{hl'jl} = \begin{cases} 1 & \begin{array}{l}\text{if operation } l \text{ of } j \text{ is processed directly} \\ \text{after operation } l' \text{ of job } h \text{ on machince } i\end{array} \\ 0 & \text{otherwise} \end{cases}$$

3.4 Problem Formulation

$$Min\ Z = C_{max} \tag{1}$$

$$C_{\max} \geq E\left(C_j\right)\ \forall j \tag{2}$$

$$E\left(C_j\right) = \sum_{i=1}^{m} C_{j1}.\left(1 - p_{ij1}\right).\ x_{ij1} + \sum_{i-2}^{L_j}\left(\sum_{i-1}^{m} C_{jl}\ .\ \prod_{l'}^{l-1}\left(\sum_{i-1}^{m} p_{ijl'}.\ x_{ijl'}\right).\ \left(1 - p_{ijl}\right).\ x_{ijl}\right)\ \forall j \tag{3}$$

$$\sum_{i=1}^{m} x_{ijl} = 1\ \forall l,\ j = m+1, \dots, m+n \tag{4}$$

$$C_{jl} = S_{jl} + \sum_{i=1}^{m} pt_{ijl}.x_{ijl}\ \forall j, l \tag{5}$$

$$S_{jl} \geq C_{j(l-1)} + \sum_{i=1}^{m} su_{ihl'jl}.x_{ihl'}\ .y_{hl'jl}\ \forall h, l^{'}, j = m+1, \dots, m+n, l = 2, \dots, L_j \tag{6}$$

$$S_{j1} \geq C_{hl'} + \left[\sum_{i=1}^{m} su_{ihl'jl}\ .\ x_{ihl'} - (1 - y_{hl'jl}).\ M\right]\ \forall j,\ h,\ l,\ l' \tag{7}$$

$$S_{j1} \geq R_j + \sum_{i=1}^{m} su_{ihl'jl}.x_{ihl'}\ .y_{hl'j1}\ \forall h, l^{'}, j = m+1, \dots, m+n \tag{8}$$

$$\sum_{h=1}^{m+n} \sum_{l'=1}^{L_j} y_{hl'jl} = 1\ \forall l, j = m+1, \dots, m+n \tag{9}$$

$$\sum_{j=1}^{m+n} \sum_{l'=1}^{L_j} Y_{hl'jl} \leq 1\ \forall h,\ l' \tag{10}$$

$$Y_{hl'jl} \leq \sum_{j=1}^{m+n} x_{ihl'} \cdot x_{ijl} \ \forall j, h, l, l' \tag{11}$$

$$x_{ijl} \leq Z_{ijl} \ \forall j, j, l \tag{12}$$

$$x_{jj1} = 1 \ j = 1, \ldots, m \tag{13}$$

$$S_{j1} = 0 \ j = 1, \ldots, m \tag{14}$$

$$X_{ijl'} Y_{hl'jl} = \{0, 1\} \ \forall i, j, l, h, l' \tag{15}$$

Equation (1) minimizes the expected maximum completion time. Constraint (2) calculates maximum completion time. Constraint (3) is the expected value function $E(C_j)$ by which the expected completion time of each job j is determined. This constraint is formulated based on Bayes' theorem as follows. O_{ijl} represents the order of operations.

$$\begin{aligned} E(C_j) = & \sum_{i=1}^{m} (C_{j1} \Big| \text{CompletedatO}_{ij1}).(1 - p_{ij1}).x_{ij1} \\ & + \sum_{i=1}^{m} (C_{j2} \Big| \text{CompletedatO}_{ij2}) \times [p_{ij1}.(1 - p_{ij2})].x_{ij1} + \ldots \\ & + \sum_{i=1}^{m} (C_{jL_j} \Big| \text{CompletedatO}_{ijL_j}) \times \prod_{l'=1}^{l-1} (\sum_{i=1}^{m} p_{ijl'}.x_{ijl'}).(1 - p_{ijL_j}).x_{ijl}) \end{aligned} \tag{16}$$

Constraint (4) guarantees that only one machine processes each operation of each job. Constraint (5) calculates the completion time of each operation of each job. Constraint (6) limits the start time of each operation of each job to be greater than the sum of completion time of previous operation of the same job and the setup time that is required to switch between previous and current operations on the same machine. Constraint (7) assures that a successor operation is allowed to start only when its predecessor is completed. The last two constraints are related to observing the primacy between the operations pertaining to the same job and respecting the priority between the operations of different jobs, respectively. Constraint (8) limits the start time of the first operation of each job to be greater than the sum of job release time and the setup time that is required to switch from previous operation to current operation on the same machine. Constraints (9) and (10) state predecessor/successor restrictions between two consecutive operations. Constraint (11) indicates that a valid sequence is the one with operations processed on the same machine. Constraint (12) restricts machine eligibilities. Constraint (13) assigns a dummy job to each machine. Constraint (14) indicates that dummy jobs must start at time zero on each machine; i.e., dummy jobs are the first assigned job on each machine. Finally, constraint (15) defines the binary decision variables.

4 Solution Representation Description

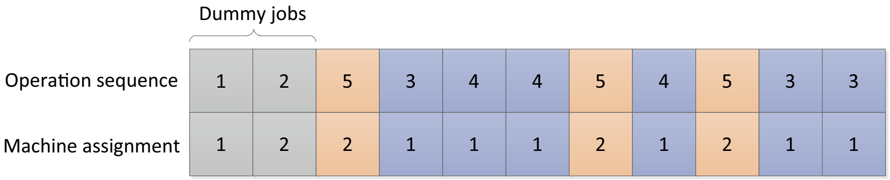
In this study, a solution representation method which is proposed by Ramezanian and Saidi-Mehrabad [13] is applied to indicate a feasible solution. Two types of elements are included in the schematic structure of problem representation method. Operations and their assigned machines are contained in the first and second row respectively, so the solution is entirely represented as a vector. The same symbol is used to name all work and rework processes of a job and for any given solution vector. The order of occurrence in operation sequence is used to interpret them. Therefore, L_j is defined as the exact number of the appearances of the j th job in the operation sequence in which the first appearance is identified as the main process and the next ones are recognized as rework processes.

An example is described in the following. An unrelated parallel machine scheduling problem with two machines, three jobs, and two dummy jobs is considered. Three operations are required per job. For each job, the first operation is referred to as the main process and the next ones as rework processes which are probabilistic. Therefore, to compute the completion time, an expected value approach is applied. It should be noted that generating the base sequence is done in accord with minimization of the expected makespan. However, in case that some jobs are successfully able to pass the test center, the related rework processes are deleted and the operation sequence is updated consequently. The real makespan can be determined only when all jobs pass the test center. Below are the basic operation sequences for the described instance where all main and rework processes appeared in the solution vector are depicted in Fig. 2. They can be converted to a list of the order of operations (O_{ijl}) as follows:

Machine 1: $O_{111} \rightarrow O_{131} \rightarrow O_{141} \rightarrow O_{142} \rightarrow O_{143} \rightarrow O_{132} \rightarrow O_{133}$
Machine 2: $O_{221} \rightarrow O_{251} \rightarrow O_{252} \rightarrow O_{253}$

5 Experiments and Results

In order to explain the scheduling problem, a two-machine example including three jobs and three operations with one main and two rework activities is solved using the data given in Tables 1, 2, and 3 and illustrated in Figs. 3 and 4. The data described in [15] is used. For each problem, job processing times are generated from the uniform [10, 100]. In the same way, release times are produced via [5, 20]. The probability



Operation sequence	1	2	5	3	4	4	5	4	5	3	3
Machine assignment	1	2	2	1	1	1	2	1	2	1	1

Fig. 2 The solution representation method

Table 1 Data generated for operation 1

Machine	Job 3			Job 4			Job 5		
	t	*α*	*p*	*t*	*α*	*p*	*t*	*α*	*p*
1	12	1	0.23	27	1	0.48	49	1	0.20
2	17	1	0.42	43	1	0.42	35	1	0.22
Rj		7			15			10	

of jobs being defective is a random real number between 0.2 and 0.4, and decreasing coefficients of rework processing times are between 0.25 and 0.35 [11, 13].

The first operations loaded on machines 1 and 2 are dummy jobs j_1 and j_2 respectively. Setup, processing, and release times required for dummy jobs are assumed to be zero. As a matter of fact, the only reason to consider dummy jobs is to determine the first operation in each machine's sequence, so they have no real impact on the objective function. The sequences of operation on available machines are illustrated in a schematic structure in Figs. 3 and 4 in which the number of machines and dummy jobs is equal. To interpret the figures, a description of an arbitrarily chosen operation sequences is presented in the following. For instance, the first operation assigned to machine 1 is the dummy job j_1 that precedes the first job j_3. The release time of j_3 is equal to 7; therefore, the start of the processing of j_3 cannot be earlier than this time. Setup action for preparing the process of j_3 starts at time 7, since setup time is required. The setup time between successive operations of a same job is assumed to be zero.

Figures 3 and 4 show the same problem in two different structures: when (1) unforced idleness is not allowed and when (2) unforced idleness is allowed. Since the objective function is to minimize the makespan ($C_{\max}$) (Eq. 1) and the makespan is greater equal to the maximum expected value for completion times ($\max E\left(C_j\right), \forall j$) (Eq. 2), the other completion times except for the largest one can be delayed as long as they do not exceed $C_{\max}$, so the objective value will remain unchanged. This analysis leads to a better understanding of the objective function by measuring the effect of unforced idleness as an input factor on the objective function as the output. In the first condition, machines work continuously and unforced idleness is not allowed. The expected completion times are as follows:

Table 2 Data generated data for operation 2

Machine	Job 3		Job 4		Job 5	
	α	*p*	*α*	*p*	*α*	*p*
1	0.18	0.43	0.20	0.25	0.23	0.38
2	0.15	0.36	0.25	0.30	0.19	0.38

Table 3 Data generated for operation 3

Machine	Job 3		Job 4		Job 5	
	α	*p*	*α*	*p*	*α*	*p*
1	0.25	0	0.23	0	0.27	0
2	0.34	0	0.26	0	0.27	0

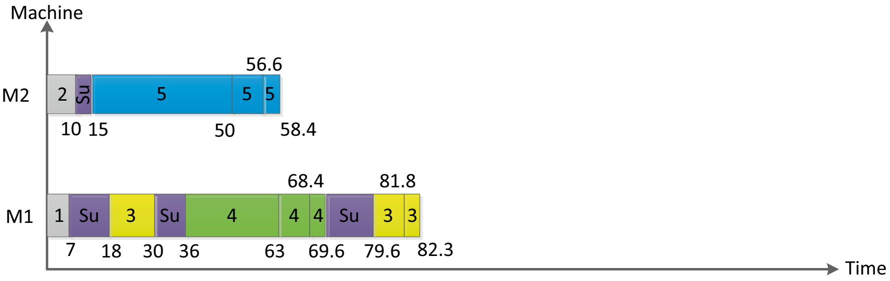


Fig. 3 Depiction of $m = 2$, $n = 3$ and $L_j = 3$ when the unforced idleness is not allowed

$$E(C_3) = 41.96\ ;\ E(C_4) = 65.74\ ;\ E(C_5) = 51.61$$

Consider job 3 as an example to clarify how these values are determined. Since $x_{131}, x_{132}, x_{133} = 1$, Eq. (3) which is used to calculate the expected completion times becomes $C_{31} \times (1 - p_{131}) + C_{32} \times p_{131} \times (1 - p_{132}) + C_{33} \times p_{131} \times p_{132} \times (1 - p_{133})$. By inserting the required values from Tables 1, 2, and 3 and Fig. 3 into this equation, $E(C_3)$ is determined as $30 \times (1 - 0.23) + 81.8 \times 0.23 \times (1 - 0.43) + 82.3 \times 0.23 \times 0.43 \times (1 - 0) = 41.96$. The same goes for $E(C_4)$ and $E(C_5)$. According to the expected completion times obtained from the first condition, it is clear that j_4 has the greatest expected value among jobs. Hence, the operations of j_3 and j_5 can be moved along the scheduling horizon by adding the unforced idleness while the expected completion times of j_3 and j_5 do not exceed the expected value of $j_4(E(C_3), E(C_5) \le E(C_4))$. Thus, in the second condition, it is assumed that unforced idleness is allowed and the expected completion times are as follows:

$$E(C_3) = E(C_4) = E(C_5) = 65.74$$

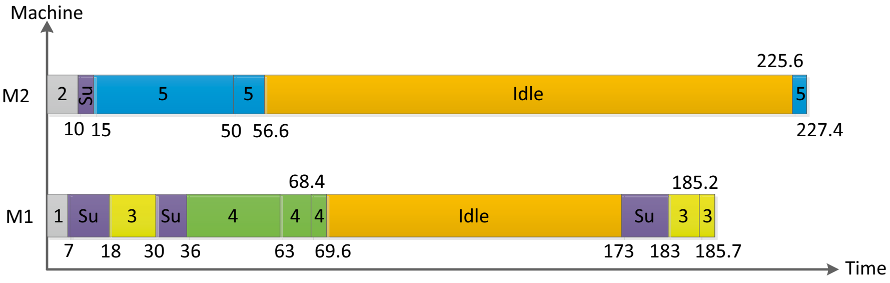


Fig. 4 Depiction of $m = 2$, $n = 3$ and $L_j = 3$ when the unforced idleness is allowed

6 Sensitivity Analysis

Understanding how a model responds to input changes has a fundamental importance to ensure the correct use of the model. Sensitivity analysis (SA) is the study that specifies how an uncertainty in the output of a mathematical model can be attributed to different origins of uncertainty in its inputs [22]; i.e., it is a method of systematic changes in variables of a model to determine the impacts of such variations. In an overall view, it can be noted that SA addresses the types of questions that arise from parameter changes.

The system under study involves several parameters and its complexity is attributable to the wide range of variability of those parameters. However, it is a hard task to track the impacts of all the parameters along with their interactions. Therefore, this paper focuses on parameters distinguished to be the most appropriate to perform sensitivity analysis. The outcome of the sensitivity analysis is shown in Table 4. In this data set, six different scenarios are assumed. In scenarios 1 to 6, the effects of an increase in rework process, jobs, machines, machine eligibility restrictions, setup times, and decreasing coefficients on the objective function value are investigated, respectively. To reach the exact optimal solutions for small-size problems, the exact software LINGO was utilized and for large-size ones, the genetic algorithm is utilized [15].

6.1 Analysis of Scenario (1) Results

By increasing the number of rework processes, the objective function value can be increased, but this increase is not tangible because the processing time of each operation is a small percentage of prior operation of the same job. By increasing the number of rework processes, the processing times of rework activities are shorter than their previous activities. Therefore, increasing the number of rework processes has a negligible effect on the objective function value. The above analysis is shown clearly in Fig. 5. However, when the size of problem grows, the impact of increasing the number of rework processes on the objective value becomes significant as Fig. 6 illustrates.

6.2 Analysis of Scenario (2) Results

Increasing the number of jobs has a significant impact on the objective function value, because in this case, at least one job with one or more than one rework process is allocated to the existing machines. A two-machine example which includes four jobs and two operations with one main and one rework process is illustrated in Fig. 7. The expected completion times are as follows:

$$E\left(C_3\right) = E\left(C_4\right) = E\left(C_5\right) = 76.51$$

$$E\left(C_6\right) = 72.18$$

Table 4 Sensitivity analysis results

Scenario No	Parameter	Cases	Problem size m, n, L	Objective value
(1)	Rework process	1	2, 3, 2	65.59
		2	2, 3, 3	65.74
		3	2, 3, 4	65.75
		4	2, 3, 5	65.75
		5	5, 10, 2	84.34
		6	5, 10, 3	87.23
		7	7, 25, 2	125.61
		8	7, 25, 3	167.56
(2)	Job	1	2, 3, 2	65.59
		2	2, 4, 2	76.51
		3	2, 5, 2	90.59
		4	2, 6, 2	114.62
		5	2, 3, 3	65.74
		6	2, 4, 3	76.57
		7	2, 5, 3	90.67
		8	2, 6, 3	117.16
		9	5, 10, 2	84.34
		10	5, 12, 2	87.56
		11	5, 14, 2	92.49
		12	5, 16, 2	99.82
		13	7, 30, 2	133.48
		14	7, 35, 2	157.56
(3)	Machine	1	2, 3, 3	65.74
		2	3, 3, 3	51.61
		3	4, 3, 3	48.28
		4	5, 3, 3	48.28
		5	2, 3, 3	65.74
		6	3, 4, 3	51.61
		7	3, 5, 3	72.27
		8	4, 5, 3	51.61
		9	2, 6, 2	114.60
		10	3, 6, 2	76.37
		11	4, 6, 2	61.10
		12	5, 6, 2	56.37
(4)	Machine eligibility restriction	1	2, 3, 3	65.74
		2	2, 3, 3	88.48
		3	2, 3, 3	71.55
(5)	Setup time	1	2, 3, 3	65.74
		2	2, 3, 3	71.55
		3	2, 3, 3	73.74
(6)	Decreasing coefficient	1	2, 3, 3	65.74
		2	2, 3, 3	65.79
		3	2, 3, 3	67.96

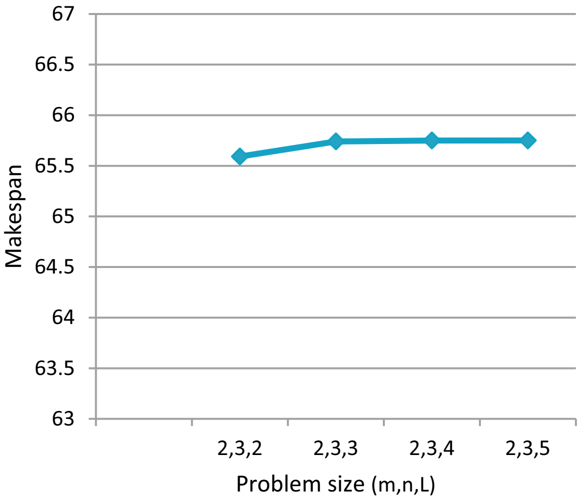


Fig. 5 The effect of increasing the number of rework processes on the objective value

Each job has a long processing time and its rework processing time is a small percentage of the previous operation of the job. The results of cases from scenario (2) show that if the number of jobs increases, the objective value can be deteriorated drastically. The sharp increase in objective value is shown in Fig. 8. However, when the size of problem grows, the impact of increasing the number of jobs on the objective value becomes less intense as Fig. 9 illustrates.

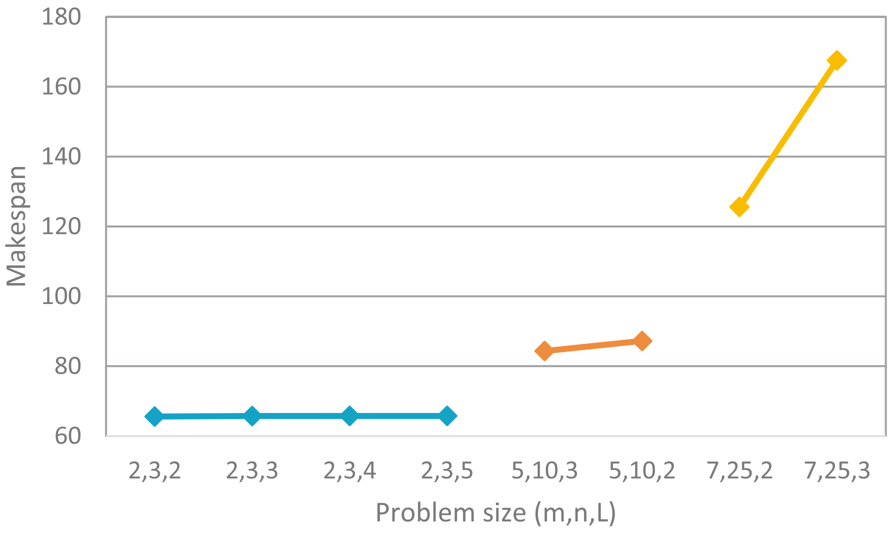


Fig. 6 The effect of increasing the number of rework processes on the objective value as the size of problem grows

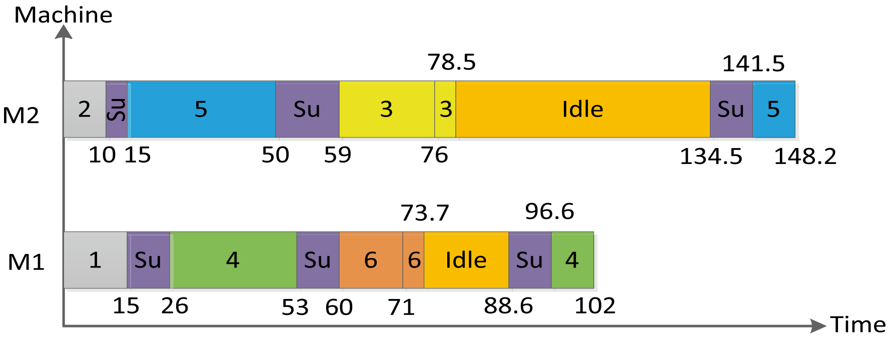


Fig. 7 Obtained result for $m = 2$, $n = 4$ and $L_j = 2$

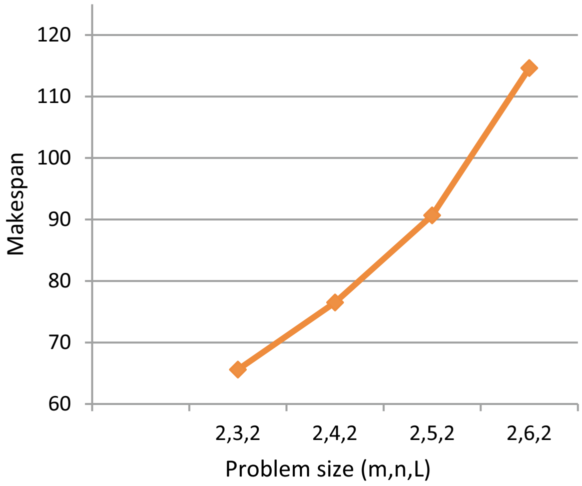
6.3 Analysis of Scenario (3) Results

In this scenario, the results of increasing the number of machines on the objective value are provided. In order to clarify these results, a three-machine example which includes three jobs and three operations for each job is illustrated in Fig. 10. The expected completion times are as follows:

$$E\left(C_4\right) = 39.56; E\left(C_5\right) = 49.66; E\left(C_6\right) = 51.61$$

As the results indicate, by increasing the number of machines, the objective value is decreased significantly. The reason is that in this case, operations can be allocated to more machines with better assignment and the length of the operation sequence on each machine can be shortened and finally, the objective function value can be improved. It should be noticed that increasing the number of

Fig. 8 The impact of increasing the number of jobs on the objective value



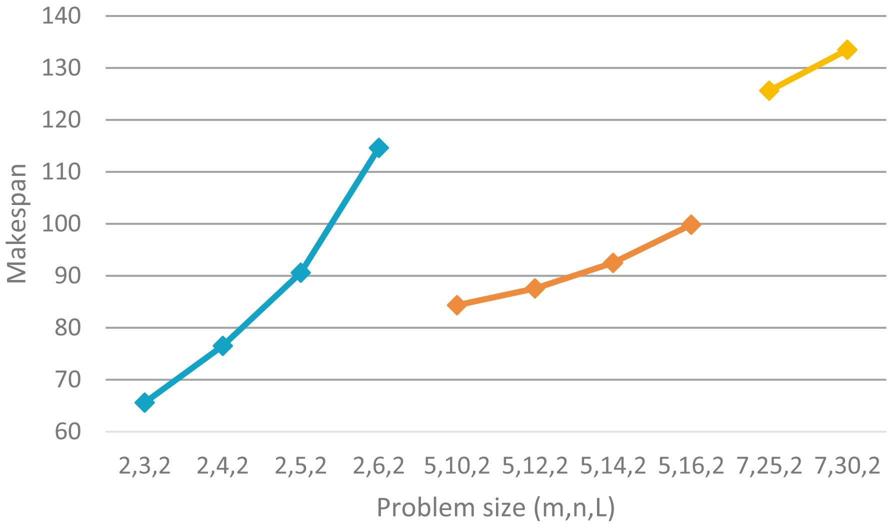


Fig. 9 The impact of increasing the number of jobs on the objective value as the size of problem grows

machines up to a specific number can improve the objective value. In other words, adding extra machines more than this threshold has no effect on the objective value. This analysis is shown graphically in Fig. 11.

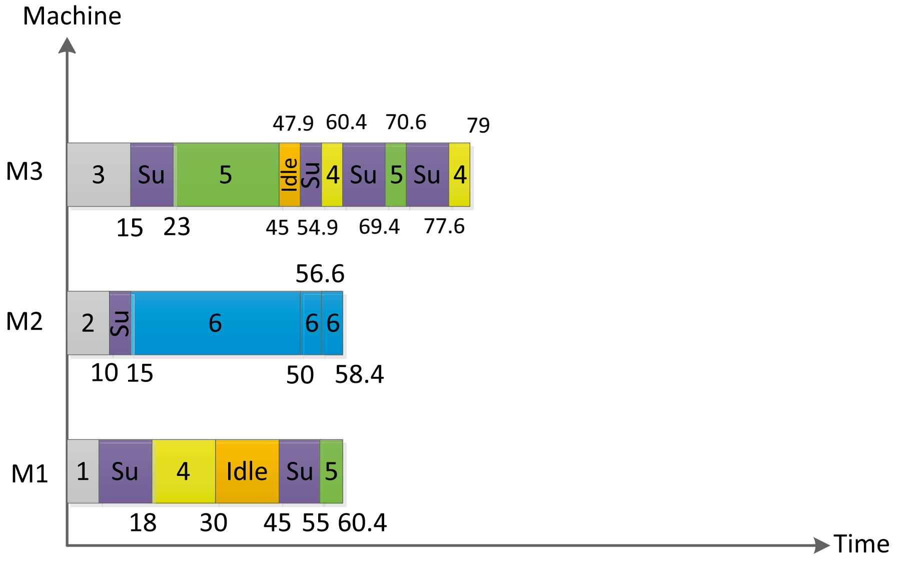


Fig. 10 Obtained result for $m = 3$, $n = 3$ and $L_j = 3$)

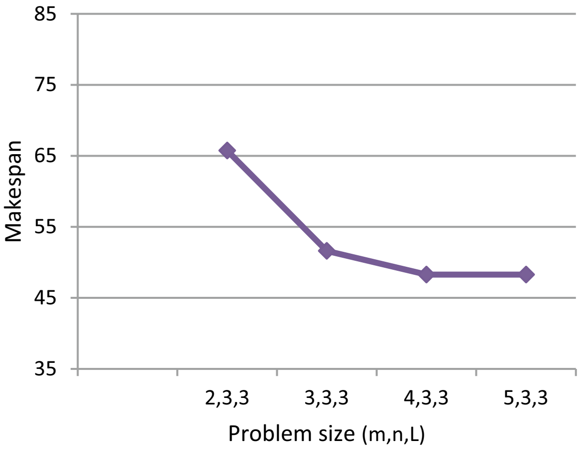


Fig. 11 The impact of increasing the number of machines on the objective value

6.4 Analysis of Scenario (4) Results

The majority of scheduling studies address the problems where all jobs are allowed to be processed on all machines with no restriction. This assumption is considered in case 1 where there is no machine eligibility restriction for jobs. In most cases, this assumption conflicts with the real-world environments. Hence, the machine eligibility restrictions are imposed in cases 2 and 3. Figure 12 illustrates case 2 of scenario (4) where the main process and rework activities for each job have to be processed on the same machine. The processing sets of operations are as follows:

$$M^{31}, M^{32}, M^{33}, M^{51}, M^{52}, M^{53} = \{M_1\}$$

$$M^{41}, M^{42}, M^{43} = \{M_2\}$$

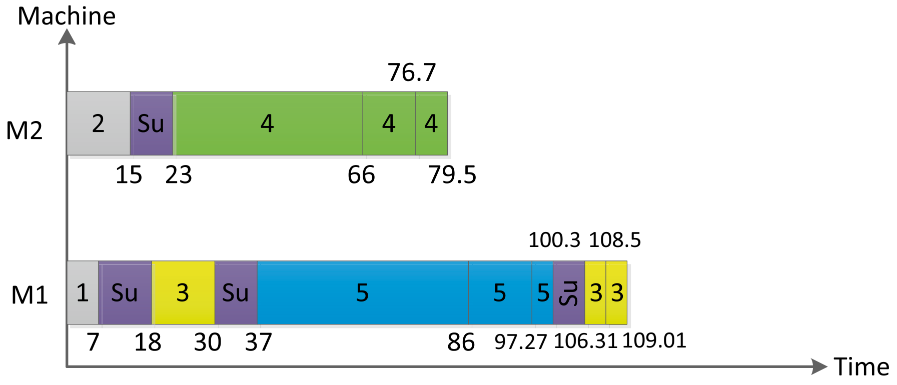


Fig. 12 Obtained result for $m = 2$, $n = 3$ and $L_j = 3$ with machine eligibility restrictions

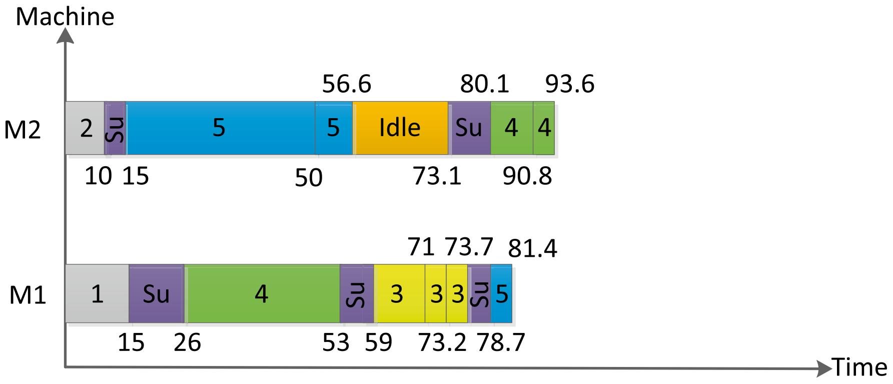


Fig. 13 Obtained result for $m = 2$, $n = 3$ and $L_j = 3$ with machine eligibility restrictions

Figure 13 illustrates case 3 of scenario (4) where the machine eligibility restrictions for all operations are imposed randomly. The processing sets of operations are as follows:

$$M^{31}, M^{33}, M^{41}, M^{43}, M^{51}, M^{52} = \{M_1, M_2\}$$

$$M^{32} = \{M_1\};\ M^{42} = \{M_2\};\ M^{53} = \{M_1\}$$

Clearly in both cases, the objective function value is worsened. The results of scenario (4) cases show that by imposing the machine eligibility restrictions, the objective value cannot be improved. In most cases, imposing the eligibility constraints makes the objective value deteriorated clearly.

6.5 Analysis of Scenario (5) Results

This section discusses the effect of increasing the setup times on the objective function value and operation sequences are investigated. It is necessary to note that setup times depend on operation sequence as well as machine. Figure 14 illustrates case 2 of scenario (5) in which the required setup time to prepare j_3 for processing on machine 1 is increased 10 time units. As is clear from Fig. 14, the operation sequence on machine 1 is changed and the objective function value is deteriorated. In case 3, by adding 4 time units to all the setup times, the objective function value is increased more than case 2, but the operation sequence on both machines is not changed. The results indicate that the objective value can be worsened by increasing the setup times between two consecutive operations. Also, change in operation sequence can occur under certain circumstances.

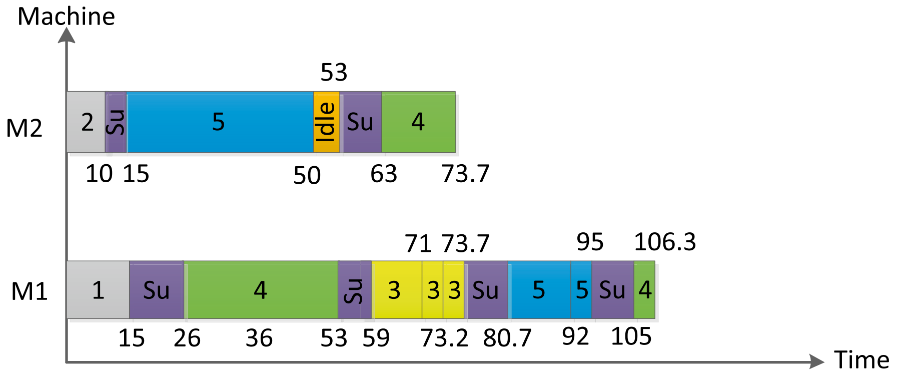


Fig. 14 Obtained result for $m = 2$, $n = 3$ and $L_j = 3$ in case 2 of scenario (5.)

6.6 Analysis of Scenario (6) Results

The processing time of each rework activity is a percentage of previous operation processing time of the same job, and this percentage is known as a decreasing coefficient named $0 \leq \alpha_{ijl} \leq 1$ in this study. The results of cases from scenario (6) show the effect of increasing these coefficients on the objective function value and operation sequences. In case 2, decreasing coefficient of the second operation of j_4 on machine 2 is increased 0.15, and in case 3, the same coefficients of all operations on all machines are increased 0.15. The schematic structure of case 2 and case 3 is illustrated in Figs. 15 and 16, respectively. As is clear from these figures, by increasing these coefficients, the objective function value is increased in the expected direction and the operation sequences on machines are changed. The results of cases from scenario (6) show that by increasing the decreasing coefficients, the objective function value can get worse and the operation sequences on machines can be changed.

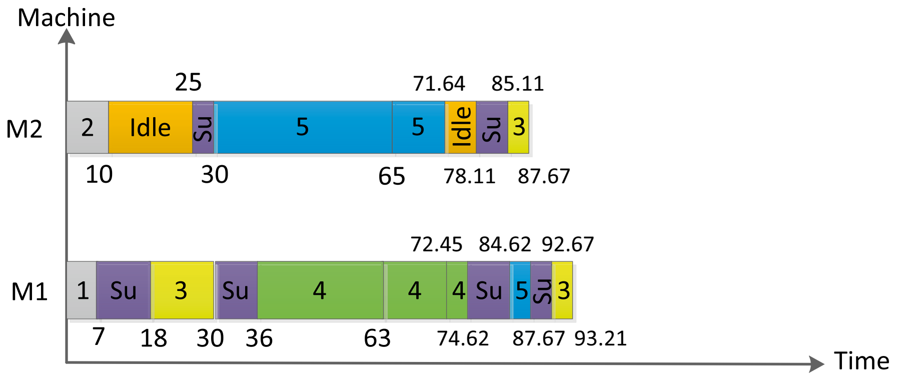


Fig. 15 Obtained result for $m = 2$, $n = 3$ and $L_j = 3$ in case 2 of scenario (6)

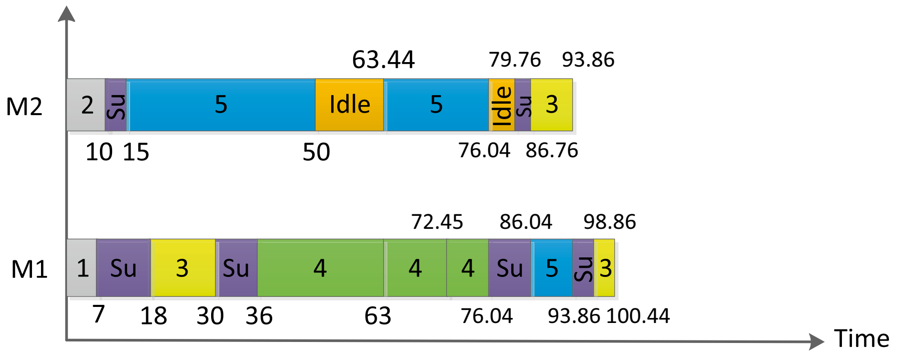


Fig. 16 Obtained result for $m = 2$, $n = 3$ and $L_j = 3$ in case 3 of scenario (6)

As mentioned previously, it is a hard task to track interactions of all parameters which are analyzed in this study. For this reason, only the interactions between the number of rework processes, jobs, and machines are considered here.

6.7 Analysis of Interaction Between the Number of Rework Process and Job

Consider a two-machine example with three jobs and three operations containing one main and two rework operations ($m = 2$, $n = 3$, $L_j = 2$). By adding one rework process to all jobs ($m = 2$, $n = 3$, $L_j = 3$), no tangible effect on the objective function value is observed. On the other hand, increasing one job to the mentioned problem ($m = 2$, $n = 4$, $L_j = 2$) can intensely make the objective value worse. The results show that increasing the number of jobs has more negative impact on the objective function value than rework processes.

6.8 Analysis of Interaction Between the Number of Rework Process and Machine

It is necessary to note that the effects of increasing the number of rework processes and machines on the objective value are in conflict with each other. In other words, increasing the number of rework processes can deteriorate and increasing the number of machines can improve the objective value. As it was stated in the previous section, the effect of increasing the number of rework processes on the objective value is negligible. On the other side, increasing the number of machines has a clear impact on the objective value where by increasing one machine to the mentioned problem ($m = 3$, $n = 3$, $L_j = 3$), the objective value is improved significantly. As a result, the sensitivity of model with regard to increasing the number of machines is greater than the number of rework processes.

6.9 Analysis of Interaction Between the Number of Job and Machine

In this section, the interaction between the number of jobs and machines is discussed. Consider a two-machine example with three jobs and three operations containing one main and two rework processes ($m = 2$, $n = 3$, $L_j = 3$). By adding one job to the mentioned problem ($m = 2$, $n = 4$, $L_j = 3$), the objective function value would be deteriorated. In order to understand the impact of increasing the number of machines on the new problem, one machine is added ($m = 3$, $n = 4$, $L_j = 3$). By increasing one machine, the objective value is improved significantly. It seems that the positive impact of increasing the number of machines on the objective value is greater than the negative effect of increasing the number of main jobs in the expected direction. In order to have a better understanding of interaction between increasing the number of jobs and machines, consider the computational results of cases 1 to 4 of scenario (2) and cases 9 to 12 of scenario (3). The results indicate that the effect of increasing the number of machines on the objective value is more tangible than the number of jobs. The results can be seen in Fig. 17.

6.10 Practical Insights

Unrelated parallel machines with sequence- and machine-dependent setup times are largely applied in production environments to process different components, e.g., injection molding departments or LCD manufacturing processes [21]. In such industries, a series of jobs must be completed, the output of which is the final product that can be delivered to the customer. Since the completion time of the last job is

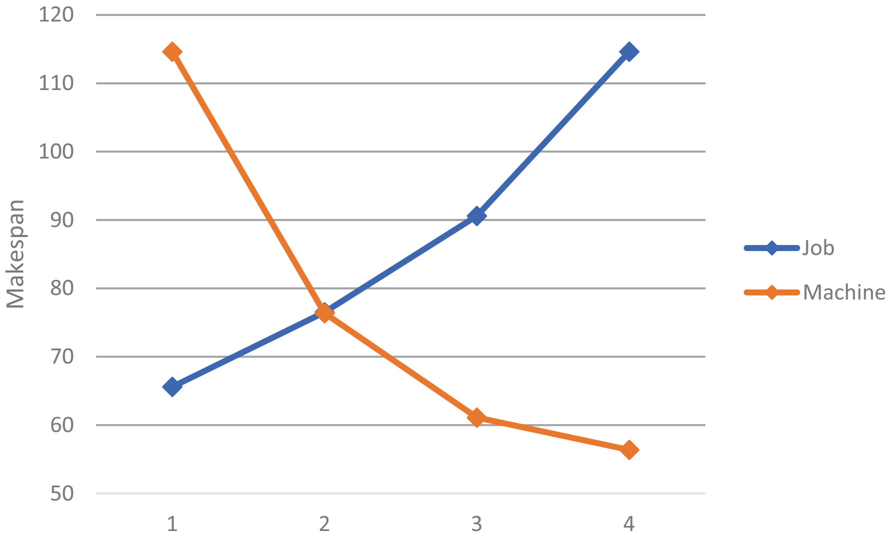


Fig. 17 The interaction between the number of jobs and machines

the product delivery time, in such cases, $C_{\max}$ is used. Sensitivity analysis helps the manufacturer's decision-making process by evaluating the impact of input parameters on the objective function value. For example, according to the obtained results, if the intention is to keep the objective function constant while increasing the number of jobs, this goal can be achieved by increasing the number of machines with a ratio lower than the rate of increase in the number of jobs. On the other hand, it can be said that achieving this goal by reducing the number of reworks requires extra effort, because increasing the number of jobs has a more negative effect on the value of the objective function than rework processes. In another scenario, if the goal is to reduce $C_{\max}$ while increasing the number of jobs, it is attainable with a combination of increasing the number of machines while reducing the number of reworks at appropriate rates.

7 Conclusions and Future Work

In this paper, an unrelated parallel machine scheduling problem was considered where the defective items are allowed and machine eligibility restrictions and release times are assumed. Rework process is applied to improve the imperfect items to meet an acceptance level of quality. There are setup actions between successive operations where setup times are dependent on both operation sequences and machines. To evaluate the presented model, the expected makespan was employed as the performance criterion. In order to understand the impact of parameter variation on the output of the model, sensitivity analysis was performed for parameters distinguished as the most appropriate. To perform the sensitivity analysis, numerical experiments were conducted by considering six different scenarios, each investigating the effects of changes in the six essential parameters on the objective function value. Those parameters were the number of rework processes, number of jobs, number of machines, machine eligibility restrictions, setup times, and decreasing coefficients. Also, the results of the analysis of each scenario were discussed. In addition, the interactions between three parameters including the number of rework processes, jobs, and machines were tracked.

For future research, assumptions like precedence constraints, machines breakdown, and transportation constraints can be considered to improve the proposed model. Furthermore, for each operation, instead of using historical data to estimate the parameters of decreasing coefficient and defective probability which was applied in this study, a probability distribution can be employed.

Data Availability Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Competing Interests The authors declare no competing interests.

References

1. Allahverdi A, Ng CT, Cheng TE, Kovalyov MY (2008) A survey of scheduling problems with setup times or costs. Eur J Oper Res 187(3):985–1032
2. Pei J, Zhou Y, Yan P, Pardalos PM (2023) A concise guide to scheduling with learning and deteriorating effects. Int J Prod Res 61(6):2010–2031
3. Tavakkoli-Moghaddam R, Taheri F, Bazzazi M, Izadi M, Sassani F (2009) Design of a genetic *algorithm* for bi-objective unrelated parallel machines scheduling with sequence-dependent setup times and precedence constraints. Comput Oper Res 36(12):3224–3230
4. Pei J, Song Q, Liao B, Liu X, Pardalos PM (2021) Parallel-machine serial-batching scheduling with release times under the effects of position-dependent learning and time-dependent deterioration. Ann Oper Res 298(1–2):407–444
5. Lin SW, Ying KC (2014) ABC-based manufacturing scheduling for unrelated parallel machines with machine-dependent and job sequence-dependent setup times. Comput Oper Res 51:172–181
6. Manupati VK, Rajyalakshmi G, Chan FT, Thakkar JJ (2017) A hybrid multi-objective evolutionary algorithm approach for handling sequence-and machine-dependent set-up times in unrelated parallel machine scheduling problem. Sādhanā 42:391–403
7. Zhang L, Deng Q, Lin R, Gong G, Han W (2021) A combinatorial evolutionary algorithm for unrelated parallel machine scheduling problem with sequence and machine-dependent setup times, limited worker resources and learning effect. Expert Syst Appl 175:114843
8. Bektur G, Saraç T (2019) A mathematical model and heuristic algorithms for an unrelated parallel machine scheduling problem with sequence-dependent setup times, machine eligibility restrictions and a common server. Comput Oper Res 103:46–63
9. Afzalirad M, Rezaeian J (2016) Resource-constrained unrelated parallel machine scheduling problem with sequence dependent setup times, precedence constraints and machine eligibility restrictions. Comput Ind Eng 98:40–52
10. Yunusoglu P, Topaloglu Yildiz S (2022) Constraint programming approach for multi-resource-constrained unrelated parallel machine scheduling problem with sequence-dependent setup times. Int J Prod Res 60(7):2212–2229
11. Foroutan RA, Rezaeian J, Shafipour M (2022) Bi-objective unrelated parallel machines scheduling problem with worker allocation and sequence dependent setup times considering machine eligibility and precedence constraints. J Ind Manag Optim 19(1):402–436
12. Flapper SDP, Fransoo JC, Broekmeulen RA, Inderfurth K (2002) Planning and control of rework in the process industries: a review. Prod Plan Control 13(1):26–34
13. Ramezanian R, Saidi-Mehrabad M (2012) Multi-product unrelated parallel machines scheduling problem with rework processes. Scientia Iranica 19(6):1887–1893
14. Wang X, Li Z, Chen Q, Mao N (2020) Meta-heuristics for unrelated parallel machines scheduling with random rework to minimize expected total weighted tardiness. Comput Ind Eng 145:106505
15. Rambod M, Rezaeian J (2014) Robust meta-heuristics implementation for unrelated parallel machines scheduling problem with rework processes and machine eligibility restrictions. Comput Ind Eng 77:15–28
16. Der Kiureghian A, Ditlevsen O (2009) Aleatory or epistemic? Does it matter? Struct Saf 31(2):105–112
17. Li Z, Ierapetritou M (2008) Process scheduling under uncertainty: review and challenges. Comput Chem Eng 32(4–5):715–727
18. Kolen AW, Kan AR, Van Hoesel CPM, Wagelmans AP (1994) Sensitivity analysis of list scheduling heuristics. Discret Appl Math 55(2):145–162
19. Penz B, Rapine C, Trystram D (2001) Sensitivity analysis of scheduling algorithms. Eur J Oper Res 134(3):606–615

20. Hall NG, Posner ME (2004) Sensitivity analysis for scheduling problems. J Sched 7:49–83
21. Chen JF (2009) Scheduling on unrelated parallel machines with sequence-and machine-dependent setup times and due-date constraints. Int J Adv Manuf Technol 44:1204–1212
22. Saltelli A, Ratto M, Andres T, Campolongo F, Cariboni J, Gatelli D, Tarantola S (2008) Global sensitivity analysis: the primer. John Wiley & Sons

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.