Vol. 07, No. 01, January 2023, pp. 41-54

INTERNATIONAL JOURNAL OF INNOVATION IN ENTERPRISE SYSTEM

Journal homepage: https://ijies.sie.telkomuniversity.ac.id
e-ISSN: 2580-3050, DOI: https://doi.org/10.25124/ijies.v7i01.190

Scheduling Splittable Jobs on Identical Parallel Machines to Minimize Makespan using Mixed Integer Linear Programming

Ayudita Oktafiani[1*], Muhammad Nashir Ardiansyah[2]

[1]School of Industrial and System Engineering
Telkom University, Bandung, Indonesia
ayuditaoktafiani@telkomuniversity.ac.id

[2]School of Industrial and System Engineering
Telkom University, Bandung, Indonesia
nashirardiansyah@telkomuniversity.ac.id

*ayuditaoktafiani@telkomuniversity.ac.id

ARTICLE INFO

Article history:
Received 26 December 2022
Accepted 29 January 2023
Published 31 January 2023

ABSTRACT

The scheduling of parallel machines with and without a job-splitting property, deterministic demand, and sequence-independent setup time with the goal of minimizing makespan is examined in this work. For simultaneous processing by multiple machines, single-stage splitable jobs are broken into random (job) sections. When a job starts to be processed on a machine, an operator has to setup the machine for an hour. By creating two Mixed Integer Linear Programming models, this work proposes a mathematical programming strategy (MILP). A MILP model takes the job-splitting property into account. Another model, however, does not include the job-splitting property. This study investigates the performance of the proposed models using Gurobi solver. These programs' numerical calculations are based on actual problems in the Indonesian city of Bandung's plastics industry. On four identical parallel injection molding machines, 318 jobs must be finished in 22 periods. The real scheduling method is contrasted with these two MILP models. The maximum workload imbalance, the maximum relative percentage of imbalance, and the makespan of these three scheduling systems are used to evaluate their effectiveness. Without the job-splitting property, MILP can handle the real issue of scheduling identical parallel machines on injection molding machines to reduce makespan, resulting in a 36% average decrease. The MILP model's job-splitting property can reduce makespan by an additional 2.40%. The order of relative ranking is MILP with job-splitting property, MILP without job-splitting property, and actual scheduling based on the makespan minimization, workload imbalance, and relative percentage of imbalance.

Keywords:
Identical Parallel Machine; Makespan; Job-splitting Property; Injection Molding; Workload Balancing

1. INTRODUCTION

Due to their wide range of applications in computer and manufacturing systems, parallel machine scheduling problems have been the subject of a great deal of research [1]–[5]. Many early works on makespan minimization-based parallel machine scheduling have examined the worst-case boundaries of priority-based rules. Ovacik and Uzsoy introduced rolling horizon heuristics for parallel machine scheduling with sequence-dependent setup times to reduce maximum lateness. To reduce the total weighted tardiness, Lee and Pinedo suggested a three-phase heuristic strategy for parallel machine scheduling with sequence-dependent setup durations. The approach assigns a job index based on sequence-dependent setup times and tightness of the due date, and then applies a dispatching rule based on the indexes.

To reduce makespan, Behnamian et al. suggested a hybrid metaheuristic for scheduling parallel machines with sequence-dependent setup delays [6]. The study of parallel machine scheduling problem (PMSP) with the goal of minimizing makespan which required setup times before executing various jobs is a NP-hard problem. The previous study by Karp in 1972 which is limited to two machines demonstrated that the jobs scheduling on these two identical parallel machines without preemption to reduce makespan is NP-hard [7], [8].

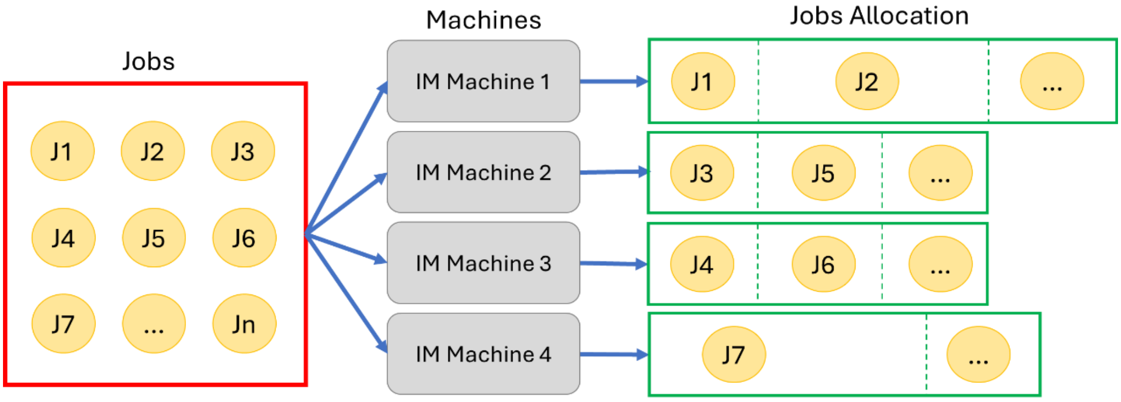

Figure 1. Concept model for injection molding scheduling

This work addresses a parallel machine scheduling problem for makespan minimization, where jobs can be divided into many portions that can be handled concurrently on machines. On a machine, setup activities are necessary between the sequential processes of several tasks or job portions, and the number of setup operations that may be carried out at once is constrained if there are few setup operators (or resources). Jobs' setup times are thought to be independent of sequence. In contrast to job preemption, job splitting allows job parts to be processed concurrently on various parallel machines [9]–[13].

This study is inspired by the injection molding process used in an Indonesian plant that manufactures spring guides and undercase. Polymers are used to injection molding process in a number of identical machines, each of which must be outfitted with the proper tools by a setup operator in order to process different types of spring guides and undercase. There are enough operators to set up all the machines at once, and the setup time is solely dependent on the type of spring guides and undercase that will be produced next. The study of identical parallel machines has been conducted by other researchers in Korea. The case studies which motivated the study are coming from casting processes in an automotive piston manufacturing factory [9] and printed circuit board (PCB) manufacturing system [14].

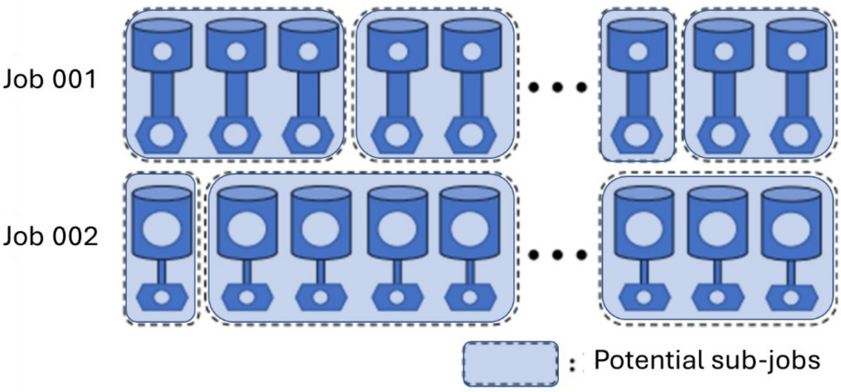

Figure 2. Job-splitting in piston manufacturing, adapted from [9]

Even though there have been many studies on parallel machine scheduling, there is still limited study that evaluate the performance of both approaches, with and without job-splitting property and setup times that independent on operation sequence, which is in the injection molding process used in an Indonesian plant. This work first suggests a mathematical formulation to precisely model the issue. Afterward, perform a numerical calculation to demonstrate its performance. **Figure 1** illustrates the concept of injection molding scheduling process. There are N-jobs which should be processed in M-machines. As a result, the jobs allocation can be obtained. It gives the sequence of jobs being processed on each machine. As mentioned previously, the jobs which are considered here can possibly be split into a number of sub-jobs.

A good illustration of job-splitting is depicted in **Figure 2**. It shows a possible example of work splitting in the manufacturing of pistons. In this context, a job section denotes a (small) number of pistons that are sequentially cast on a machine, while a job designates a collection of pistons of the same type.

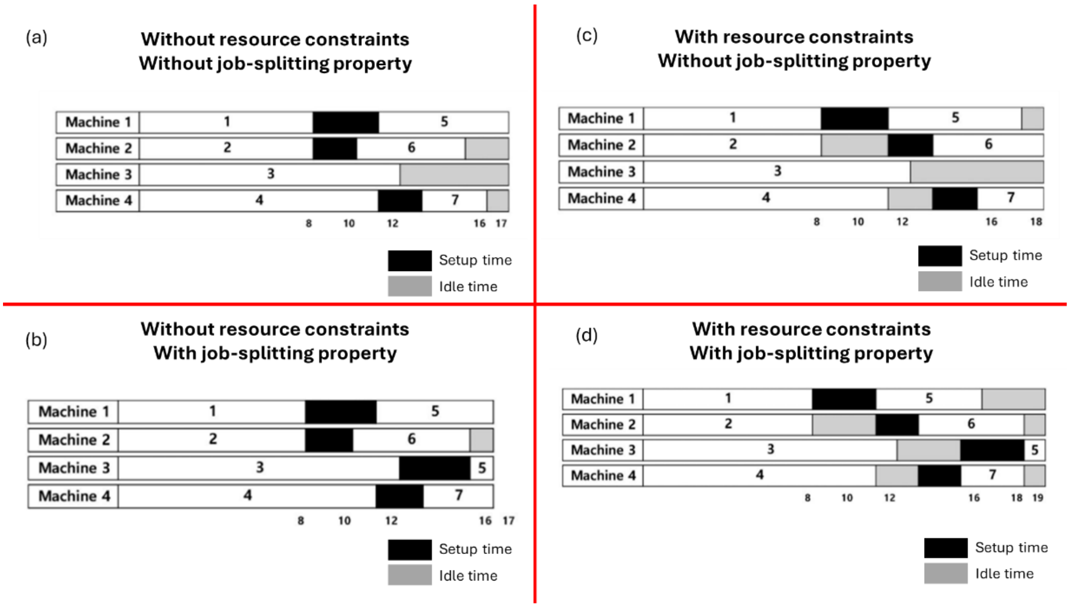

Figure 3. Gantt Chart of Identical Parallel Machine Scheduling with different constraints, adapted from [9]

In scheduling identical parallel machines, there are various constraints that can be considered. **Figure 3** illustrates the Gantt Chart obtained from scheduling identical parallel machine with and without resource constraints as well as the job-splitting property. The examined case from injection molding operation is best described by left column of **Figure 3**. The case does not take into account the resource constraints. Therefore, the setup operations can be performed simultaneously on different machines. Because there are sufficient operators to do the setup. **Figure 3a** shows a longer makespan because the job is not possible to be split. If the job-splitting property is considered in the scheduling strategy, the obtained makespan is shortened as shown in **Figure 3b**. These two strategies will be further elaborated in the next sections. Whereas **Figure 3c** and **Figure 3d** will not be discussed into more detail in this study.

2. METHOD

Because scheduling jobs on parallel machines offers a significant operational difficulty for operating a time-sharing computer system, the family of parallel-machine scheduling problems has long been the focus of intensive research by computer scientists. McNaughton is the author of the earliest study on parallel machine scheduling. He established the three performance criteria of makespan, total weighted tardiness, and total weighted flowtime for parallel-identical machine scheduling [1]. Over decades, the identical parallel machine scheduling is still an interesting research topic. **Table *1*** summarizes the previous research that relates to this study. For scheduling the identical parallel machine there are several objective functions that can be used. Besides, several different constraints can also be applied to obtain the optimality of the objective functions. Among five studies listed in **Table *1***, only two are motivated by real case in the industry. The remaining studies obtain the numerical example by generating computational instances. The real case studies also include the job-splitting property. Although there are a number of objective functions and constraints that can be used to schedule the identical parallel machine, the study to compare of scheduling approach with and without job-splitting property is still limited.

Table 1. Previous Study in Identical Parallel Machine Scheduling

No	Objective Function	Numerical example	Constraints	Reference
1	Minimize total tardiness	M=2-5, N=10-200	Sequence dependent jobs	[2]
2	Minimize total weighted completion time	M=2-9, N=10-80	Family dependent setup times	[3]
3	Minimize total tardy jobs	M=2-4, N=15-20	Maintenance time	[4]

4	Minimize makespan	M= 5-20, N=40-100, Real case from casting procedures in a Korean facility that makes car pistons	Job-splitting property; setup operator constraints	[9]
5	Minimizing total tardiness	M= 5-20, N=40-100, Real case from system for manufacturing printed circuit boards (PCBs)	Job-splitting property; varying setup time constraints	[14]

We take into account a scheduling issue for parallel machines with *M* identical machines and *N* jobs that can possibly be divided into any number of portions. A job's processing time is indicated by the symbol p_i. Jobs can be processed in sections on many computers at the same time. After processing a part of job *i* on machine *j*, a setup operation is carried out by a setup operator, requiring setup time units. Section processing times are expressed as integers. Two decisions must be made: (1) the lengths of the work portions processed on the machines, and (2) the sequencing of those sections on each machine by taking setup times and processing time that depend on the demand of the job into account. The goal is to reduce makespan. The three-field notation of the scheduling problem can be represented as Pm| |Cmax [15]. The following are the problem assumptions: Jobs are available at the beginning of the period; Jobs have their own due dates; Jobs can be divided into sub-jobs; one machine cannot process multiple sub-jobs at once; jobs (and sub-jobs) have specific setup times that are unaffected by job sequences; jobs are completed once all sub-jobs have been completed; the first job being processed on each machine requires setup; and there is no resource constraints in terms of operators as well as the mold being used to run the jobs.

Mixed-integer linear programming (MILP) is utilized for system analysis and optimization of identical parallel machine scheduling because it is versatile and powerful for handling huge and complicated [16]. After developing MILP-based models, the next step is to validate the models. Validation testing evaluates a model's performance. MILP models can be validated by construct and results. Validation by construct uses "sensible" methods inspired by real-world data to build a model that is considered valid. Validation by results involves methodically comparing model results to real-world observations and performing association tests to determine degree of association [17]. In this study, the proposed MILP-based models are validated by results. The results of two proposed models are constrated to the actual scheduling result.

2.1 MILP Model without job-splitting property

The MILP models listed below do not employ work splitting. As a result, each job that is processed on a machine is not divided into smaller projects that can be completed simultaneously on another machine. Furthermore, this model is referred to as the MILP Model 1. The mathematical model developed is as follows:

Parameters

M	Number of machines
N	Number of jobs
P_i	Processing time of job $i, i \in N$
D_i	Demand of job $i, i \in N$
S	Set up time

Variables

x_{ij}	Binary variable indicating job $i, i \in N$ is scheduled in machine $j, j \in M$
y_j	Total processing time of machine $j, j \in M$
C_{max}	Makespan

Objective Function

$$\min C_{max} \tag{1}$$

Subject to:

$$\sum_{j \in M} x_{ij} = 1 \qquad \forall i \in N \tag{2}$$

$$y_j \geq \sum_{i \in N} x_{ij} * (D_i P_i + S) \qquad \forall j \in M \tag{3}$$

$$C_{max} \geq y_j \qquad \forall j \in M \tag{4}$$

$$x_{ij} \in \{0,1\} \qquad \forall i \in N, j \in M \tag{5}$$

$$y_j \geq 0 \qquad \forall j \in M \tag{6}$$

$$C_{max} \geq 0 \tag{7}$$

Model descriptions:
Objective function is to minimize the makespan in (1). Equation (2) defines that every job $i, i \in N$ must be scheduled in a machine $j, j \in M$. Definition of total processing time of a machine $j, j \in M$ consists of total processing time of jobs that is scheduled in that machine in (3). Makespan is defined as the latest total processing time of any machine $j, j \in M$ in (4). Variables definition is stated in equations (4) - (7).

2.2 MILP Model with job-splitting property

The setup operator constraints may prolong idle times on machines and delay the completion of jobs. However, if sub-jobs are correctly assigned to machines, job splitting can minimize such delays[6]. Jobs can be divided into smaller ones in the MILP models that follow so that they can be processed simultaneously on several machines. In fact, if a job can be divided, it can be thought of as a collection of smaller jobs whose combined sizes equal the size of the original job. Alternately, one may think of the job as being made up of tiny jobs of similar sizes. The issue thus resembles the situation in which jobs have unit sizes and equal processing times [10]. Furthermore, this model is called the MILP Model 2. The mathematical model developed is as follows:

Parameters

M	Number of machines
N	Number of jobs
T	Number of split jobs
P_i	Processing time of job $i, i \in N$
D_i	Demand of job $i, i \in N$
S	Set up time
Z	Sufficient big number

Variables

x_{ilj}	Binary variable indicating job $i, i \in N$ which is split in sub-job $l, l \in T$ in machine $j, j \in M$
w_{ilj}	Number of portion job $i, i \in N$ split in sub-job $l, l \in T$
y_j	Total processing time of machine $j, j \in M$
C_{max}	Makespan

Objective Function

$$\min C_{max} \tag{8}$$

Subject to:

$$\sum_{j\in M}\sum_{l\in T} w_{ilj} = D_i \qquad \forall i \in N \tag{9}$$

$$w_{ilj} \le x_{ilj}Z \qquad \forall i \in N, j \in M, l \in T \tag{10}$$

$$\sum_{i\in N}\sum_{j\in M} x_{ilj} \le 1 \qquad \forall l \in T \tag{11}$$

$$y_j \ge \sum_{i\in N}\sum_{l\in T} w_{ilj}P_i + x_{ilv}S \qquad \forall j \in M \tag{12}$$

$$C_{max} \ge y_j \qquad \forall j \in M \tag{13}$$

$$x_{ilj} \in \{0,1\} \qquad \forall i \in N, j \in M, l \in T \tag{14}$$

$$w_{ilj} \ge 0 \qquad \forall i \in N, j \in M, l \in T \tag{15}$$

$$y_j \ge 0 \qquad \forall j \in M \tag{16}$$

$$C_{max} \ge 0 \tag{17}$$

Model description
Objective function is to minimize the make span in (8). Total sub-job j of job $i, i \in N$ must satisfied each demand job i, D_i in (9). Equation (10) defines variable x_{ilj} must be one if variable w_{ilj} is greater than zero. Definition of total processing time of a machine $j, j \in M$ consists of total processing time of jobs that is scheduled in that machine in (12). Make span is defined as the latest total processing time of any machine $j, j \in M$ in (13). Variables definition is stated in equations (14) - (17).

A real case study for identical parallel machine scheduling is the data from a plastics firm in Bandung. These data indicate that 318 jobs were completed on four injection molding machines during 22 periods. The demand quantity from each job executed on each machine varies every period. The longest overall processing time on a machine can be found after scheduling in accordance with this scenario. This will reveal the makespan of scheduling.

Based on the demand data, processing time, and setup time, the two MILP models mentioned above are executed using Gurobi software. **Table 2** contains a list of the parameter settings used when running the models. A time limit of 60 seconds is set for computing a solution of the two MILP models. If no solution is identified after 60 seconds, the solution search is stopped, and the final result is taken into consideration as a MILP model solution. The two MILP models' makespans are then contrasted with one another.

Table 2.Parameter Setting for MILP Models

No	Parameter	Level
1	Number of machines	3; 4; 5
2	Setup Time (second)	900; 1,800; 2,700; 3,600

3. RESULT AND DISCUSSION

This section will describe the results of the scheduling simulation with the actual scheduling scenario, the MILP scheduling scenario without job splitting property, and the MILP scheduling scenario with job splitting property. The results obtained from this study include a comparison of the Gantt chart which shows the sequencing of jobs on each machine in the actual scheduling, MILP model 1, and MILP model 2. From the Gantt chart one can determine the completion time of each machine. It will be used to calculate the makespan. The makespan from three different strategies is compared and analyzed. Furthermore, with the help of completion time, the workload imbalance and relative percentage of imbalance can be computed. All these three components measure how well the performance of scheduling strategies.

3.1. Comparison of actual scheduling with the proposed MILP-based models

The machine scheduling Gantt chart shows a visual representation of the sequencing of jobs on each machine. The Gantt chart shows which jobs were done on machine 1, machine 2, machine 3, and machine 4. The Gantt chart also provides information regarding makespan and loading on each machine for a certain period. Gantt chart results of actual scheduling, MILP model 1, and MILP model 2 in period 1 will be shown as an example of a comparison of scheduling results.

The actual scheduling results for the four injection molding machines are shown in **Figure *4***. There were 18 jobs done by the four machines in that period. In actual scheduling, machine 1 and machine 2 each process five jobs. While machine 3 and machine 4 each execute four jobs. The sequence of jobs processed on machine 1 is job 001, 005, 009, 013, and 017. Machine 2 works on five jobs in the following order job 002, 006, 010, 014, and 018. Machine 3 is doing jobs 003, 007, 011, and 015. The remaining four jobs, namely jobs 004, 008, 012, and 016 are done on the fourth machine. The job sequencing gives a makespan of 274,500 seconds.

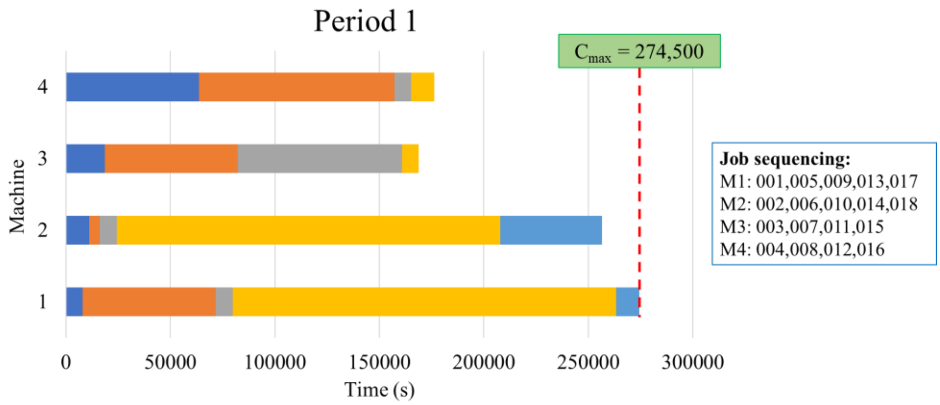

Figure 4. Actual Scheduling of four identical parallel injection molding machines

The MILP model without job splitting property gives different results from the actual scheduling, in terms of job sequencing and makespan as shown in **Figure *5***. Machines 1,2,3 and 4 do 3, 5, 4, and 6 jobs respectively. The first machine runs jobs 004, 005, and 008. Jobs 009, 012, 013, 015, and 017 are performed on the second machine. The third machine processes jobs 003, 006, 014, and 016. The last machine processes jobs with a relatively short processing time, such as jobs 001, 002, 007, 010, 011, and 018. The makespan obtained from this job sequencing is 220,800 seconds. When compared with the makespan of the actual scheduling, there is a significant difference. Makespan results from scheduling jobs using MILP without job splitting property in period 1 decreased by 20% from the actual scheduling makespan.

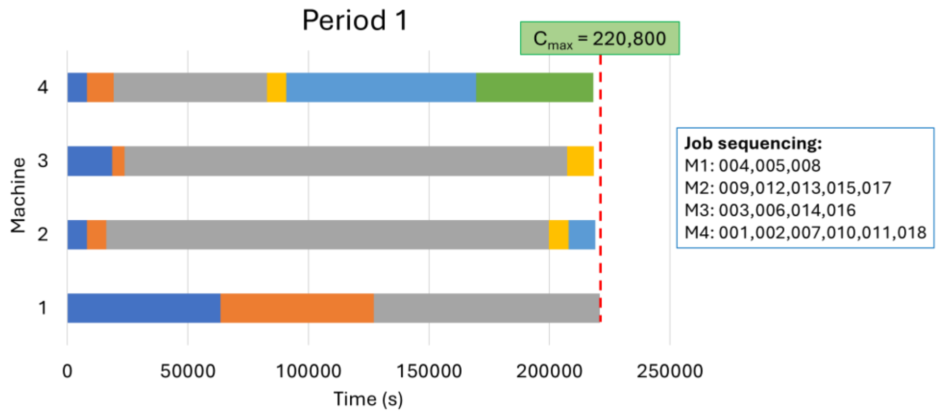

Figure 5. Scheduling of four identical parallel injection molding using MILP Model 1

The sequencing of jobs and makespan in **Figure *6*** illustrates how the MILP model with the job splitting property yields different outcomes from the actual scheduling. Job 008 is split into two sub-jobs, 008S1 and 008S2. Machines 1, 2, 3, and 4 each perform 6, 5, 5, and 3 jobs. Jobs 010, 013, 001, 006, 016, and sub-job 008 are run by machine number one. The second machine is used for jobs 014, 012, 015, 002, and 009. Processed on the third machine are jobs 017, 007, 018, 011, and 003. Jobs 0041, 005, and another sub-job 008 are processed by the last machine. The jobs were sequenced to provide a makespan of 220,500 seconds. There is a substantial gap between this and the actual scheduling's makespan.

The results of scheduling MILP without a job-splitting property in period 1 showed a 20% reduction in makespan compared to the actual scheduling makespan.

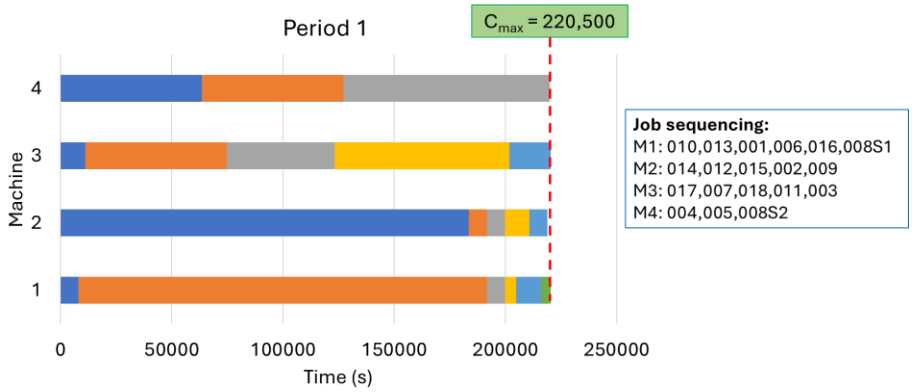

Figure 6. Scheduling of four identical parallel injection molding using MILP Model 2

Table 3 summarizes the makespan between actual scheduling, MILP Model 1, and MILP Model 2 for 22 scheduling periods of four identical parallel machines. In general, MILP model 1 performs 36% better in terms of makespan. It means that scheduling jobs with MILP model 1 leads to shorter makespan. Similar to the first proposed model, MILP model 2 also gives a makespan improvement of 38% on average. The lowest makespan reduction is shown by the first period and it is followed by the fifth period. Both periods show a makespan improvement of less than 25%. On the contrary, the highest makespan improvement of 49% is depicted by the 18th period. When the company can compress the makespan, the production runs more efficiently and the difference in completion time of each machine is not significant. Therefore, the workload of operators from M1 to M4 is relatively similar. Since both proposed models do not differ significantly in terms of makespan minimization, the scheduling performance is further analyzed by using workload imbalance and relative percentage of imbalance. The results of this analysis are discussed in Section 3.3 and Section 3.4.

Table 3. Makespan comparison of actual scheduling, MILP 1 model and MILP 2 model

Period	Makespan (s)*			Improvement**	
	Actual	MILP Model 1	MILP Model 2	MILP Model 1	MILP Model 2
1	274,500	220,800	220,500	20%	20%
2	312,000	210,900	210,900	32%	32%
3	282,000	210,900	210,900	25%	25%
4	296,400	188,400	187,800	36%	37%
5	273,900	207,000	207,000	24%	24%
6	374,400	210,000	209,550	44%	44%
7	359,400	210,300	210,300	41%	41%
8	314,400	198,000	198,000	37%	37%
9	325,800	183,600	178,350	44%	45%
10	374,400	216,900	216,600	42%	42%
11	336,900	213,600	213,600	37%	37%
12	336,900	201,600	200,100	40%	41%
13	288,300	190,200	188,310	34%	35%
14	344,400	209,100	209,100	39%	39%
15	305,400	194,700	194700	36%	36%
16	325,800	183,600	168,300	44%	48%
17	285,300	183,600	170,100	36%	40%
18	325,800	183,600	167,100	44%	49%
19	325,800	183,600	168,600	44%	48%
20	285,300	183,600	167,550	36%	41%

Period	Makespan (s)*			Improvement**	
	Actual	MILP Model 1	MILP Model 2	MILP Model 1	MILP Model 2
21	277,200	183,600	165,600	34%	40%
22	296,400	196,500	196,500	34%	34%
Average improvement				36%	38%

* Number of machine (M) = 4, setup time (S) = 3600 s

** concerning actual scheduling

By reducing the makespan by an average of above 30% in the two proposed MILP models, the company can streamline the use of machines by reducing the number of injection molding machines from four to three machines. By using three machines to process all the jobs in 22 periods, identical parallel machine scheduling of the two proposed MILP models provides an average makespan reduction of 18%. However, not all periods can be scheduled with just three machines. There are two periods, such as period 1 and period 5, which are still scheduled with four machines. Scheduling all jobs in those periods with three machines is not possible. It is caused by the makespan improvement of the two periods is less than 25% as shown in **Table 3**. Therefore, if the company would like to run the jobs on three machines only, it has to be fully aware that in periods 1 and 5, it should provide four machines instead of three. This result can be seen in **Figure** *7* below. It is clearly seen in periods 1 and 5 that the makespan improvement has negative value. It shows that the makespan jobs processed in those periods is higher than the actual makespan.

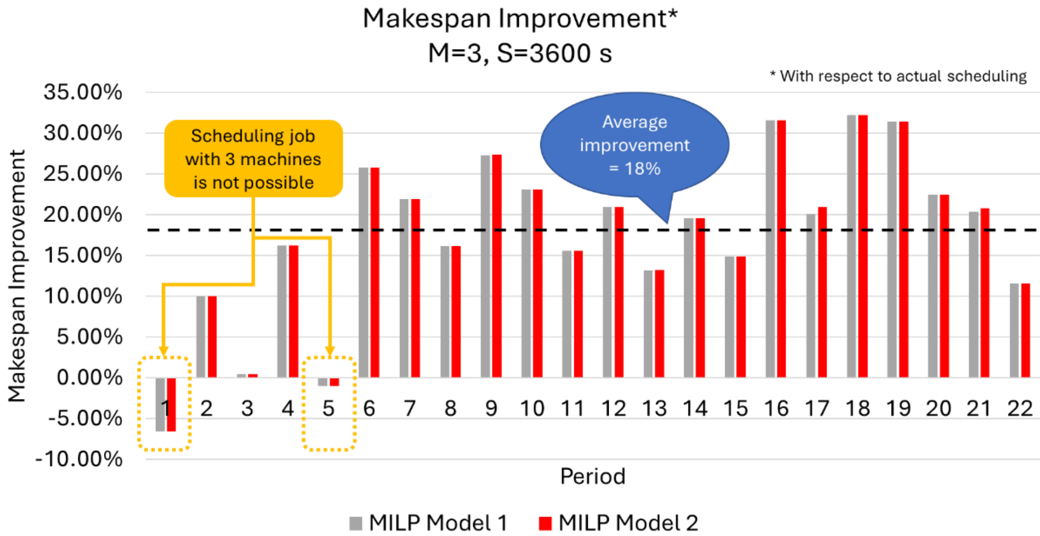

Figure 7. The makespan reduction from the two proposed MILP models by using three injection molding machines

This study also examines the effect of the number of machines on the makespan improvement as shown in **Table 4**. Initially, the company runs its business with four machines. When the jobs are scheduled with MILP model 1 and 2, the company can run the jobs on three machines only. The average improvement is 18%, half of the improvement from four machines run simultaneously. This makespan improvement can have a higher value if the company add another machine to process the jobs. With five machines, the makespan can be shortened 50% when the company applies the MILP model 2 scheduling strategy.

Table 4. The effect of the number of machines on the makespan improvement

Number of machines	Makespan Improvement*	
	MILP Model 1	MILP Model 2
3	18%	18%
4	36%	38%
5	41%	50%

* Concerning actual scheduling

Another effort to minimize makespan is to reduce setup time. The results of computer calculations show that the smaller the setup time, the lower the makespan will be. **Table 5** shows the results obtained from scheduling 318 jobs on 4 identical parallel machines for 22 scheduling periods. When the setup time is reduced to a quarter of initial value, 900 seconds, the makespan of both MILP models is improved 40% compared to the actual makespan. But this effort produces unsignificant improvement, only 3-4% difference if it is compared to the initial setup time, 3600 seconds. Compressing the setup time in practice requires skillful operators. They need to be able to change the mold as well as loading the polymer pellets into the injection molding machine in a relatively short time.

Table 5. Effect of setup time on makespan reduction

Setup Time, S (s)	Makespan Improvement*	
	MILP Model 1	MILP Model 2
3,600	37%	38%
2,700	38%	40%
1,800	39%	41%
900	40%	42%

* Concerning actual scheduling

3.2. Comparison of makespan reduction in MILP Model 1 and MILP Model 2

The MILP scheduling model without the job splitting property that was previously run on the four machines is then run on a new set of machines. Additionally, the model is run using various setup times following **Table 2**. The same procedure is also carried out using the job-splitting property of the MILP scheduling model. The results of the two models' makespans at various parameter settings are then contrasted and assessed. **Table 6** provides an overview of the outcomes of these comparisons.

According to these findings, the difference in the makespan value between the MILP model with job splitting property and MILP without job splitting property is relatively small, 2.40%, if the company utlizes four injection molding machines concurrently with the number of jobs and processing time per unit as the actual conditions and the setup time is 3600 seconds. If the setup time can be reduced, the makespan values will differ even more.

Based on the results summarized in **Table 6** the scheduling that formerly required four injection molding machines may be scheduled with just three of them. The difference in makespan between the MILP model 1 and MILP model 2 is about 0.08% when considering the number of jobs and processing time per unit under actual conditions and setup time of 3600 seconds. But if the setup time can be decreased, the makespan percentage reduction will be much greater. Therefore, the relation between setup time as well as the number of machines with the makespan reduction can be drawn as follows: the more machines, the shorter the makespan and the shorter the setup time, the more makespan reduction.

Table 6. The makespan reduction matrix of MILP Model 1 and MILP Model 2

Number of Machines	Setup Time, S (s)			
	900	1800	2700	3600
3	0.43%	0.24%	0.15%	0.08%
4	3.77%	3.25%	2.77%	2.40%
5	18.95%	17.70%	16.49%	15.26%

3.3. Workload Imbalance

The workload imbalance is a different scheduling performance from makespan which is also covered in this study. The maximum workload imbalance is determined for 22 periods and three different scheduling strategies. This calculates the difference between each machine's maximum and minimum completion times for each period. The workload imbalance checks to see if each machine has a similar workload. The maximum workload imbalance can be expressed as:

$$Workload\ Imbalance_{max} = C_{max} - C_{min} \tag{18}$$

This formula is adapted from [18].

The workload imbalance from three scheduling strategies for identical parallel machines is shown in **Figure 8** for a total of 22 periods. The red dots in the graphic demonstrate that the largest workload imbalance is caused by the actual schedule. As the MILP model is used without the job-splitting property, this workload imbalance reduces. The least workload imbalance is provided by a different proposed model called MILP with job-splitting capabilities. This approach can also eliminate the workload imbalance in periods 4, 9, 17, 18, and 19.

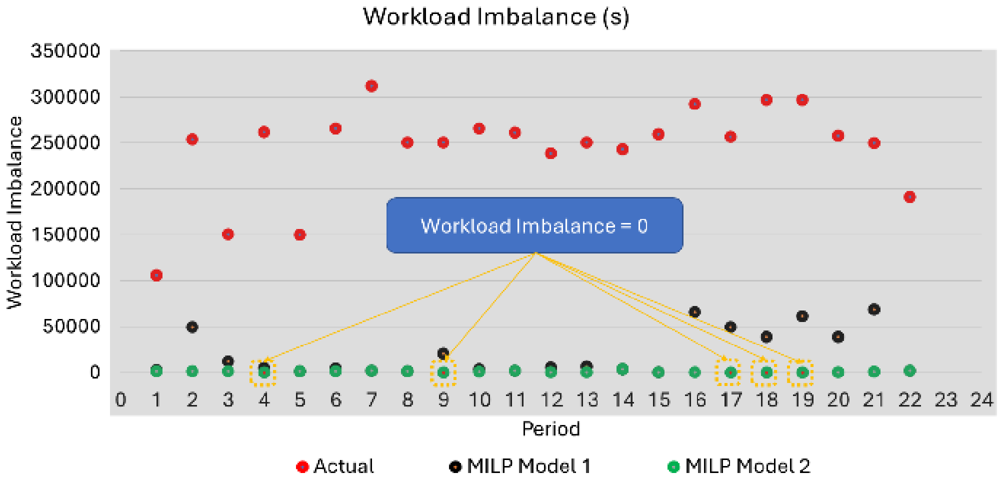

Figure 8. Workload Imbalance of three scheduling approaches with 4 identical parallel machines

3.4. Relative Percentage of Imbalance (RPI)

The effectiveness of work loading-based parallel machine scheduling solutions was examined by the authors. The findings of this study may be used to help define the goal of minimizing the greatest number of imbalances that currently exist in parallel machine scheduling. When the machines' workloads are balanced, the jobs can be distributed among them in the optimal way. The performance of the previously mentioned scheduling strategies—Actual Scheduling, MILP Model 1, and MILP Model 2 in parallel machine scheduling—was evaluated in this study using the performance metric of relative percentage of workload imbalance (RPI).

To compare the effectiveness of the scheduling strategies, the relative proportion of imbalances (RPI) in the workloads of all the machines is used. This shows the proportion by which machine workloads deviate from the maximum workload upper bound. It is adapted from the formula [19] written as:

$$Relative\ Percentage\ of\ Imbalance\ (RPI)_{max} = \frac{C_{max} - C_{min}}{C_{max}} \times 100\% \qquad (19)$$

Table 7. Comparison of RPI from three scheduling models with various numbers of machines

Number of Jobs	Maximum Relative Percentage of Imbalance (RPI)						
	3 Machines		4 Machines			5 Machines	
	MILP Model 1	MILP Model 2	Actual	MILP Model 1	MILP Model 2	MILP Model 1	MILP Model 2
11	3.40%	0.26%	90.15%	37.42%	0.36%	90.36%	2.03%
12	0.94%	0.94%	89.78%	21.08%	0.00%	67.48%	0.82%
13	1.44%	0.48%	86.89%	3.47%	0.11%	52.94%	0.60%
14	1.26%	0.06%	64.47%	0.92%	0.92%	29.17%	0.19%
15	0.90%	0.90%	70.79%	2.55%	0.00%	37.91%	0.19%
16	0.76%	0.00%	54.76%	2.00%	0.64%	34.31%	0.09%
17	0.32%	0.43%	53.40%	0.61%	0.61%	26.47%	0.18%
18	3.18%	0.31%	38.47%	1.22%	0.68%	15.85%	0.25%
Average	1.52%	0.42%	68.59%	8.66%	0.42%	44.31%	0.54%

Table 7 summarizes the maximum relative percentage of imbalance. It is clearly seen that the actual scheduling strategy with four machines gives high values of RPI. It is then followed by the MILP model 1 with five machines. In the previous

sub-section, it is stated that adding more machines will reduce the makespan. However, from the RPI point of view there exists a workload imbalance issue. The MILP Model 1 with three and four machines decrease the RPI dramatically, to the value under 1.52%. The RPI is even lower to the value under 0.54%, if the MILP model 2 is applied.

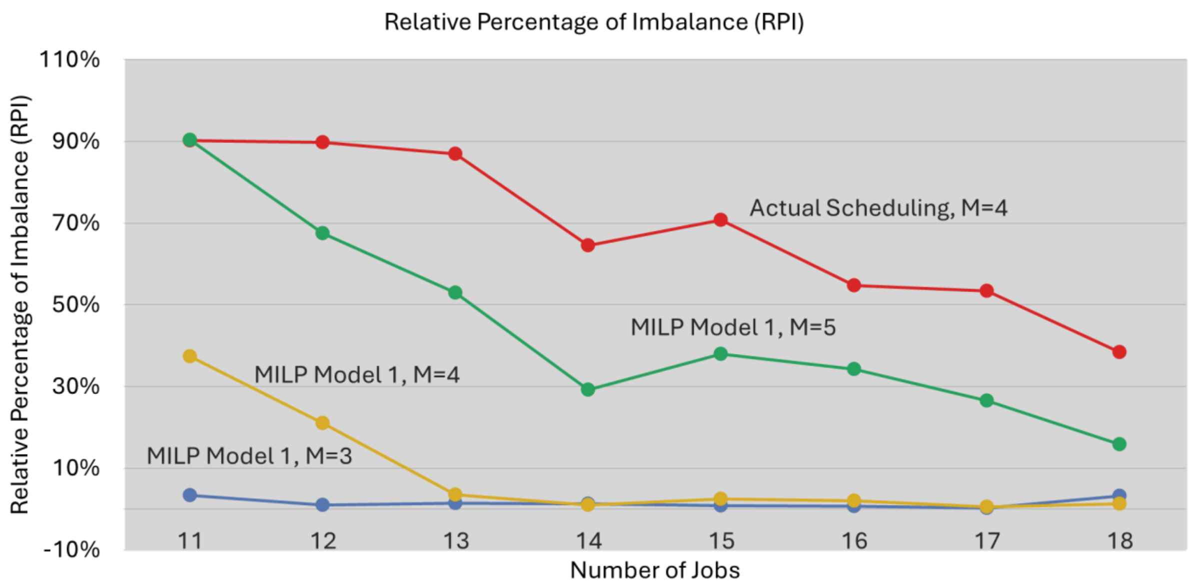

Figure 9. Improvement in RPI obtained from MILP Model 1 compared to the actual scheduling

The RPI values as illustrated in **Figure 9** shows that MILP model 1 successfully minimize the imbalance compared to the actual scheduling. With various numbers of jobs, MILP model 1 with three machines exhibits the lowest imbalance. MILP model 1 with four machines (the yellow line) shows a higher value of imbalance. When five machines are occupied the RPI of MILP model 1 approaches the imbalance value of actual scheduling strategy. From **Figure 9** also one can observe a trend that is when the number of jobs increases the RPI value tends to decrease. This trend is also determined by the previous study [18], [19]. When comparing the approaches, the MILP Model 1 with three machines yields lower imbalance percentages for all the N-jobs situations.

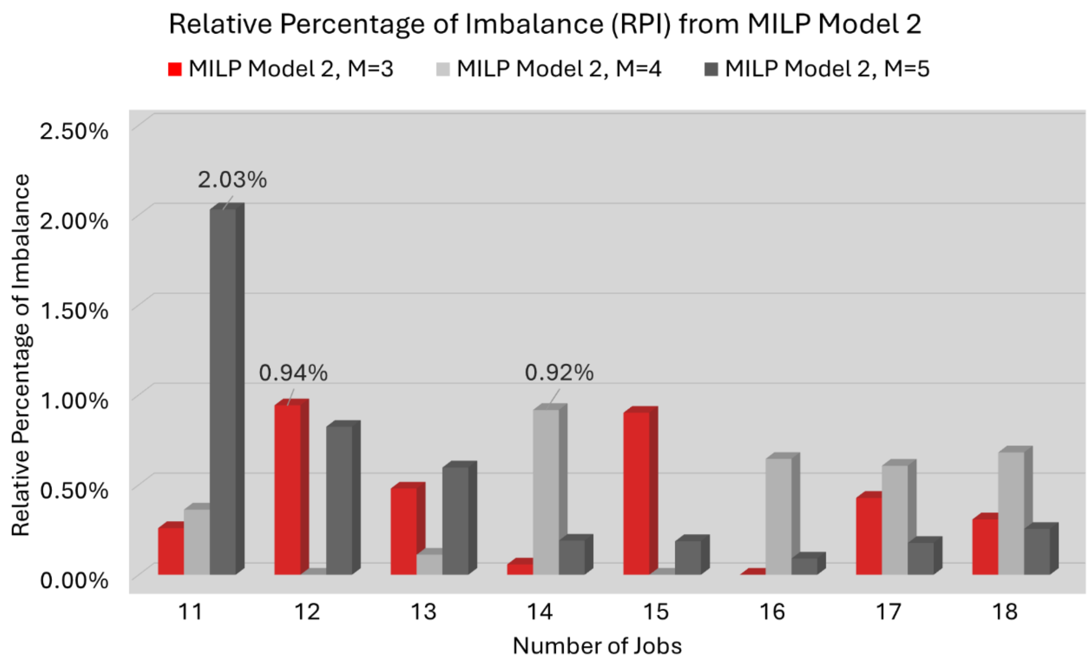

Figure 10. Improvement in RPI obtained from MILP Model 2

Since the RPI values of MILP Model 2 is lower than 2%, these will be illustrated in a separate graph. **Figure *10*** depicts that the highest RPI value for MILP model 2 with 3,4, and 5 machines are 0.94%, 0.92%, and 2.03% respectively. This graph still follows the similar trend as in **Figure *9***.

The result of this study being discussed above is limited to the condition of deterministic demand. The objective function is minimization of makespan with and without considering job-splitting property. The proposed models can be improved by considering other constraints such as limited number of operators, limited number of molds or maintenance time (resource constraint). The complexity of the MILP-based models increases as the demand becomes stochastic and other objective functions are applied. Furthermore, the models for identical parallel machine scheduling can be enhanced when the case is in multiple-stage scheduling environment. When it falls to this multiple-stage scheduling, the process sequence for each job plays an essential role. The setup dependent has to be also taken into consideration.

4. CONCLUSION

Without job-splitting property, MILP can handle the real problem of scheduling identical parallel machines on injection molding machines to minimize makespan, resulting in an average makespan reduction of 36%. With job-splitting property, MILP model can enhance the reduction of makespan by 2.40%. Numerous numerical computations for several sets of machines have been successfully conducted. Although adding one more machine can lower the makespan but the maximum relative percentage of imbalance increases. Therefore, it is not necessary to invest in an additional injection molding machine. The relative ranking of the various scheduling models studied based on makespan from the minimum is MILP with job splitting property, MILP without job splitting property and actual scheduling. For future work, the identical parallel machine scheduling can be enhanced its complexity such as considering stochastic demand, operators, and molds constraints.

Disclaimer

The authors whose names are written certify that they have no conflict of interest

REFERENCES

[1] T. C. E. Cheng and C. C. S. Sin, "A state-of-the-art review of parallel-machine scheduling research," *Eur. J. Oper. Res.*, vol. 47, no. 3, pp. 271–292, 1990, doi: 10.1016/0377-2217(90)90215-W.

[2] D. Biskup, J. Herrmann, and J. N. D. Gupta, "Scheduling identical parallel machines to minimize total tardiness," *Int. J. Prod. Econ.*, vol. 115, no. 1, pp. 134–142, 2008, doi: 10.1016/j.ijpe.2008.04.011.

[3] A. Kramer, M. Iori, and P. Lacomme, "Mathematical formulations for scheduling jobs on identical parallel machines with family setup times and total weighted completion time minimization," *Eur. J. Oper. Res.*, vol. 289, no. 3, pp. 825–840, 2021, doi: 10.1016/j.ejor.2019.07.006.

[4] A. Najat, C. Yuan, S. Gursel, and Y. Tao, "Minimizing the number of tardy jobs on identical parallel machines subject to periodic maintenance," *Procedia Manuf.*, vol. 38, no. 2019, pp. 1409–1416, 2019, doi: 10.1016/j.promfg.2020.01.147.

[5] N. Maisazahra, M. D. Astuti, and M. D. Prasetio, "Identical Parallel Machine Scheduling to Minimize Makespan Using Suggested Algorithm Method at XYZ Company," *Int. J. Innov. Enterp. Syst.*, vol. 6, no. 01, pp. 1–10, 2022, doi: 10.25124/ijies.v6i01.146.

[6] J. H. Lee and H. J. Kim, *A heuristic algorithm for identical parallel machine scheduling: splitting jobs, sequence-dependent setup times, and limited setup operators*, vol. 33, no. 4. Springer US, 2021.

[7] D. Nait Tahar, F. Yalaoui, C. Chu, and L. Amodeo, "A linear programming approach for identical parallel machine scheduling with job splitting and sequence-dependent setup times," *Int. J. Prod. Econ.*, vol. 99, no. 1–2, pp. 63–73, 2006, doi: 10.1016/j.ijpe.2004.12.007.

[8] D. Laha and J. N. D. Gupta, "An improved cuckoo search algorithm for scheduling jobs on identical parallel machines," *Comput. Ind. Eng.*, vol. 126, pp. 348–360, 2018, doi: 10.1016/j.cie.2018.09.016.

[9] J. H. Lee, H. Jang, and H. J. Kim, "Iterative job splitting algorithms for parallel machine scheduling with job splitting and setup resource constraints," *J. Oper. Res. Soc.*, vol. 72, no. 4, pp. 780–799, 2021, doi: 10.1080/01605682.2019.1700191.

[10] O. Ozturk, M. L. Espinouse, M. Di Mascolo, and A. Gouin, "Makespan minimisation on parallel batch processing machines with non-identical job sizes and release dates," *Int. J. Prod. Res.*, vol. 50, no. 20, pp. 6022–6035, 2012, doi: 10.1080/00207543.2011.641358.

[11] V. H. Nguyen, N. H. Tuong, V. H. Tran, and N. Thoai, "An MILP-based makespan minimization model for single-machine scheduling problem with splitable jobs and availability constraints," *2013 Int. Conf. Comput. Manag. Telecommun. ComManTel 2013*, pp. 397–400, 2013, doi: 10.1109/ComManTel.2013.6482427.

[12] F. Schalekamp, R. Sitters, S. van der Ster, L. Stougie, V. Verdugo, and A. van Zuylen, "Split scheduling with uniform setup times," *J. Sched.*, vol. 18, no. 2, pp. 119–129, 2015, doi: 10.1007/s10951-014-0370-4.

[13] P. Godhandaraman, V. Poongothai, and A. Yuvashree, "Parallel machine with learning, tool changes and job splitting to reduce the makespan," *AIP Conf. Proc.*, vol. 2277, no. November, 2020, doi: 10.1063/5.0025324.
[14] Y. D. Kim, S. O. Shim, S. B. Kim, Y. C. Choi, and H. M. Yoon, "Parallel machine scheduling considering a job-splitting property," *Int. J. Prod. Res.*, vol. 42, no. 21, pp. 4531–4546, 2004, doi: 10.1080/00207540410001720745.
[15] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. R. Kan, "Optimization and heuristic in deterministic sequencing and scheduling: a survey," *Ann. Discret. Math.*, vol. 5, pp. 287–326, 1979, [Online]. Available: https://ac.els-cdn.com/S016750600870356X/1-s2.0-S016750600870356X-main.pdf?_tid=cbf345a3-808d-42df-9560-bf8a52069d0e&acdnat=1550949881_e9837bfdf6316caefaf71ae2f3779b10.
[16] I. Kantor, J. L. Robineau, H. Bütün, and F. Maréchal, "A Mixed-Integer Linear Programming Formulation for Optimizing Multi-Scale Material and Energy Integration," *Front. Energy Res.*, vol. 8, no. April, 2020, doi: 10.3389/fenrg.2020.00049.
[17] M. Boccia, G. Bruno, and C. Sterle, "A MILP formulation for a batch scheduling problem on parallel machines in the aircraft industry," *2013 5th Int. Conf. Model. Simul. Appl. Optim. ICMSAO 2013*, 2013, doi: 10.1109/ICMSAO.2013.6552650.
[18] S. Rajakumar, V. P. Arunachalam, and V. Selladurai, "Workflow balancing strategies in parallel machine scheduling," *Int. J. Adv. Manuf. Technol.*, vol. 23, no. 5–6, pp. 366–374, 2004, doi: 10.1007/s00170-003-1603-4.
[19] Y. Ouazene, F. Yalaoui, H. Chehade, and A. Yalaoui, "Workload balancing in identical parallel machine scheduling using a mathematical programming method," *Int. J. Comput. Intell. Syst.*, vol. 7, no. SUPPL.1, pp. 58–67, 2014, doi: 10.1080/18756891.2013.853932.

*ayuditaoktafiani@telkomuniversity.ac.id