The Heuristic Methods for Production Scheduling with Unrelated Parallel Machine, and Job- and Machine-Sequence-Dependent Setup Time

Prasal Soeurn
School of Manufacturing Systems and Mechanical Engineering (MSME)
Sirindhorn International Institute of Technology
Thammasat University
Pathum Thani, Thailand
m6422040474@g.siit.tu.ac.th

Pham Duc Tai
School of Manufacturing Systems and Mechanical Engineering (MSME)
Sirindhorn International Institute of Technology
Thammasat University
Pathum Thani, Thailand
tai.pham@siit.tu.ac.th

Jirachai Buddhakulsomsiri
School of Manufacturing Systems and Mechanical Engineering (MSME)
Sirindhorn International Institute of Technology
Thammasat University
Pathum Thani, Thailand
jirachai@siit.tu.ac.th

Abstract

This study focuses on production scheduling problems in metal pipe manufacturing, where customer orders differ in terms of type, thickness, and quantity. Each order is considered a job with a processing time that varies based on the machines involved in the production process. To produce the next job, there is a setup time required that depends on the machine, which is assigned to, and the current job being processed on that machine which is referred to as a job- and machine-sequence-dependent setup time. Since it is not possible to obtain an optimal solution for a large scheduling problem by using the mathematical programming approach, a heuristic method is a preferable choice to handle this. Therefore, this paper aims to illustrate the application of heuristics that features dispatching rules to solve the aforementioned scheduling problem through a numerical example. These rules, which explicitly consider the job- and machine-dependent setup time include the shortest job completion time, shortest job completion time based on the longest processing time, and shortest job completion time based on the earliest due date, and shortest job completion based on the minimum slack. A comparison regarding the effectiveness of these rules is also conducted. The results show that the shortest job completion time based on the minimum slack yields the best performance concerning the makespan and the second-best performance in terms of total tardiness. In addition, the shortest job completion time based on the earliest due date produces the best performance for both total tardiness and the number of tardy jobs.

Keywords
Unrelated parallel machines, Job- and machine sequence-dependent setup time, Heuristic methods, and Dispatching rules.

1. Introduction

In today's manufacturing, effective scheduling of production processes has become increasingly critical to guarantee timely delivery and high-quality products. One of the most challenging scheduling problems that manufacturers face is the unrelated parallel machine scheduling problem, where each machine has a production rate and is capable of processing a subset of jobs. Unrelated parallel machines refer to a scheduling problem, where a set of independent jobs are assigned to a set of machines, each with its own processing time (Zhang et al, 2021; Bektur & Saraç, 2019). The machines are generally capable of processing jobs, but not all types of jobs, i.e. only a subset of jobs can be processed by each machine (Zheng et al. 2022; Cheng & Huang, 2017). In this problem, each job must be processed on a single machine, but the time required to finish the job is depended on both the job and the machine to which the job is assigned (Jaklinovi et al. 2021). This scheduling problem is further complicated by job- and machine-sequence-dependent setup times, which are necessary to prepare machines for processing a specific job. In other words, the setup times depend on the preceding job on the machine (Joo & Kim, 2015). A scheduling problem of this kind is common in steel pipe manufacturing. A manufacturer usually receives customer orders that are different in terms of type, thickness, and quantity. Each order has a specified due date and can be processed on a set of machines with different production rates. The manufacturer takes these customer orders as individual jobs and is supposed to schedule them efficiently to meet customer demand. Normally, the optimal solution for scheduling problems is found using a mixed-integer linear programming (MILP) model. However, MILP is only practical for solving small problem instances. Hence, this paper proposes heuristic methods that feature dispatching rules to handle larger problem instances. These rules include the shortest job completion time (SCT), the shortest job completion time based on the longest processing time (SC-LPT), the shortest job completion time based on the earliest due date (SC-EDD), and the shortest completion time based on the minimum slack (SC-MinSlack). The SCT rule selects the job to process on the machine that would take the shortest completion time, whereas the SC-LPT rule selects the job that has the longest processing time, the SC-EDD rule chooses the job that has the earliest due date, the SC-MinSlack rule assigns the job that provides the minimum slack. Finally, these three rules also allocate the selected job to the machine with the shortest completion time.

To evaluate the performance of the scheduling problem, three key measures, consisting of makespan, total tardiness, and the number of tardy jobs, are considered. The makespan is the time that it takes to complete all the jobs, the total tardiness is the sum of the lateness of each tardy job, and the number of tardy jobs is the total count of jobs that miss their deadline. Numerous research studies have investigated production scheduling problems on unrelated parallel machines. These problems can be categorized based on a range of criteria, such as unrelated parallel machines, job- and machine-dependent setup time, system performance measures, and solution techniques. Joo and Kim (2015) have studied production scheduling with the unrelated parallel machine to minimize the makespan; Zheng et al. (2022) focused on minimizing the completion time of all orders as well. Soleimani et al. (2020), and Cheng and Huang (2017) have studied an unrelated parallel machine for minimizing weighted tardiness. Fan et al. (2021) presented a mathematical model to minimize the mean weighted tardiness of the jobs. While Della Croce et al. (2021) studied parallel machines with the minimum number of tardy jobs. The minimization of the total weighted number of tardy jobs on single-machine scheduling has been studied by Chen et al. (2023). Lei and Yang (2022) have studied production scheduling to minimize both makespan and tardiness. Some studies have proposed approaches for production scheduling with sequence-dependent setup time. Zhao et al. (2022) have proposed the knowledge-incorporated construction heuristic for production scheduling. To find the best schedules utilizing metaheuristic methods, Kommadath et al. (2023) have proposed a revolutionary no-wait time heuristic mechanism. A recent study by Liu et al. (2022) proposed a new approach that incorporates dual resources and ready times in addition to sequence-dependent setup times. According to Ozsoydan and Sair (2021), the scheduling issue was handled by learning an iterated greedy search metaheuristic to reduce the makespan in a hybrid flexible flow shop problem with sequence-dependent setup delays that arise in a manufacturing facility. From the literature review, there is a lack of studying for scheduling with the unrelated parallel machine with the comparison of heuristics approaches.

2. Methods

To illustrate the heuristics in this study, the following notations are adopted.

Indices

i', j = indices of jobs, where N denotes the set of jobs $j \in N = \{1, 2, 3, \dots, n\}$; N' denote the set of remaining jobs $N' = \{1, 2, 3, \dots, n\}$, where N_0 denotes the set of jobs including a dummy job 0; $i' \in N_0 = \{0, 1, 2, \dots, n\}$

j^* = index of selected jobs, where $j^* \in N = \{1, 2, 3, \ldots, n\}$.
k = index of machines, where M represents the set of machines; $k \in M = \{1, 2, \ldots, m\}$. Each machine can process a different set of jobs; therefore, each given job can only be processed by a subset of M, i.e., M_j. That is, $M = M_1 \cup M_2 \ldots \cup M_n$.
i'_k = index of previous jobs i' on machine k, $i'_k \in N_0 = \{0, 1, 2, 3, \ldots, n\}$ and $k \in M = \{1, 2, 3, \ldots, m\}$

Parameters
$P_{j,k}$ = the processing time of job j on machine k (hour)
P_{j^*} = the minimum processing time of the selected job j^*(hour)
$S_{i',j,k}$ = the required setup time of machine k, when the machine is assigned to handle job j and has a previous job i'(hour)
$S_{i',j^*,k}$ = the required setup time of machine k, when the machine is set to process the selected job j^* and has a previous job i'(hour)
D_j = the due date of job j (hour)
D_{j^*} = the due date of the selected job j^*(hour)
$R_{j,k}$ = Restriction condition of processing job j on machine k, where its value is 1 meaning that the machine k can be used to produce the job j, 0 otherwise.
$Sl_{j,k}$ = the amount of slack of job j on machine k (hour)
Sl_{j^*} = the minimum slack of the selected job j^* (hour)
$C_{j,k}$ = the amount of completion time of job j on machine k (hour)
C_{max} = the completion time of all jobs j (hour)
e_j^+ = the amount of tardy time of job j (hour)
T = the total tardiness of all jobs j (hour)
N = the number of tardy jobs (job)

SCT (shortest job completion time): the rule selects the next job $j \in N'$ to be processed, and simultaneously selects machine $k \in M_j$ to process job j such that the time to complete j is minimized. In addition, in most cases, the selected machine k is processing the job i' (the preceding job of job j) which affects the setup time of job j on machine k, i.e., $S_{i',j,k}$. That is, the selected job j may not be the job that has the shortest processing time, but it is the job that has the shortest completion time from the setup time and processing time. In other words, we can write the description of the rule as:

$$\text{SCT: } \min_{j \in N',\, k \in M_j} (S_{i',j,k} + P_{j,k}).$$

SC-LPT (shortest job completion time based on the longest processing time): the rule selects the next job $j \in N'$ to be processed that has the longest processing time first, say job j^*, then the rule chooses machine $k \in M_j$ to process job j^*, while taking the setup time $S_{i',j^*,k}$ into account. Finally, the job and the machine selection result in the shortest completion time of the job j^*. In other words, we can write the description of the rule as:

$$\text{SC-LPT: } \min_{k \in M_{j^*}} (S_{i',\, j^*,k} + P_{j^*,k}) | [\, P_{j^*} = \max_{j \in N',\, k \in M_j} P_{j,k}].$$

SC-EDD (shortest job completion time based on the earliest due date): the rule picks the next job $j \in N'$ to be processed that has the earliest due date, say job j^*, then the rule chooses machine $k \in M_j$ to process job j^*, while taking the setup time $S_{i',j^*,k}$ into account. Ultimately, the job and the machine selection result in the shortest completion time of the job j^*. In other words, we can write the description of the rule as:

$$\text{SC-EDD: } \min_{k \in M_{j^*}} (S_{i',\, j^*,k} + P_{j^*,k}) | [\, D_{j^*} = \min_{j \in N',\, k \in M_j} D_j].$$

SC-MinSlack (shortest job completion based on the minimum slack): This rule selects the next job $j \in N'$ to process job j that has the shortest minimum slack time first, say j^*, then the rule chooses the machine $k \in M_j$. To complete the selected job j^*, it requires the setup time $S_{i',j^*,k}$ and processing time $P_{j^*,k}$. Finally, the machine selection results in the shortest completion time for the selected job j^*. In other words, we can write the description of the rule as:

$$\text{SC-MinSlack: } \min_{k \in M_{j^*}} \left(S_{i',j^*,k} + P_{j^*,k}\right) | [Sl_{j^*} = \min_{j \in N', k \in M_j} (D_j - P_{j,k} - S_{i',j,k})].$$

3. Numerical Example

In this numerical experiment, 10 jobs may be processed on three unrelated parallel machines. The due date, processing time on each machine, and job- and machine-sequence-dependent setup time for every job are presented below:

$$P_{j,k} = \begin{bmatrix} 66.90 & 11.80 & 26.80 & 6.45 & 30.74 & 11.23 & 4.51 & 5.09 & 9.70 & 3.18 \\ 100 & 100 & 33.49 & 8.07 & 38.43 & 14.03 & 100 & 100 & 12.13 & 3.97 \\ 66.90 & 11.80 & 26.80 & 6.45 & 30.74 & 11.23 & 4.51 & 5.09 & 100 & 100 \end{bmatrix}$$

$$D_j = [79 \quad 30 \quad 60 \quad 25 \quad 57 \quad 38 \quad 11 \quad 7 \quad 42 \quad 4]$$

$$R_{j,k} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

$$S_{i',j,1} = \begin{bmatrix} 0 & 3 & 5 & 5 & 5 & 5 & 4 & 4 & 5 & 5 \\ 3 & 0 & 5 & 5 & 5 & 5 & 4 & 4 & 5 & 5 \\ 5 & 5 & 0 & 0.5 & 0.5 & 3 & 5 & 5 & 5 & 5 \\ 5 & 5 & 0.5 & 0 & 0.5 & 3 & 5 & 5 & 5 & 5 \\ 5 & 5 & 0.5 & 0.5 & 0 & 3 & 5 & 5 & 5 & 5 \\ 5 & 5 & 3 & 3 & 3 & 0 & 5 & 5 & 5 & 5 \\ 4 & 4 & 5 & 5 & 5 & 5 & 0 & 0.5 & 5 & 5 \\ 4 & 4 & 5 & 5 & 5 & 5 & 0.5 & 0 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 0 & 0.5 \\ 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 0.5 & 0 \end{bmatrix}$$

$$S_{i',j,2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 & 3 & 0 & 0 & 5 & 5 \\ 0 & 0 & 0.5 & 0 & 0.5 & 3 & 0 & 0 & 5 & 5 \\ 0 & 0 & 0.5 & 0.5 & 0 & 3 & 0 & 0 & 5 & 5 \\ 0 & 0 & 3 & 3 & 3 & 0 & 0 & 0 & 5 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 5 & 5 & 5 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & 5 & 5 & 5 & 5 & 0 & 0 & 0.5 & 0 \end{bmatrix}$$

$$S_{i',j,3} = \begin{bmatrix} 0 & 3 & 6 & 6 & 6 & 6 & 4 & 4 & 0 & 0 \\ 3 & 0 & 6 & 6 & 6 & 6 & 4 & 4 & 0 & 0 \\ 6 & 6 & 0 & 4 & 4 & 3 & 6 & 6 & 0 & 0 \\ 6 & 6 & 4 & 0 & 0.5 & 4 & 6 & 6 & 0 & 0 \\ 6 & 6 & 4 & 0.5 & 0 & 4 & 6 & 6 & 0 & 0 \\ 6 & 6 & 3 & 4 & 4 & 0 & 6 & 6 & 0 & 0 \\ 4 & 4 & 6 & 6 & 6 & 6 & 0 & 0.5 & 0 & 0 \\ 4 & 4 & 6 & 6 & 6 & 6 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

4. Results and Discussion

4.1 Shortest Job Completion Time (SCT)

Table 1. The completion time of jobs and their tardiness from rule SCT

Job	Completion time	Due date	Tardiness
1	117.09	79	38.09
2	25.91	30	0
3	45.18	60	0
4	8.07	25	0
5	62.65	57	5.65
6	25.1	38	0
7	4.51	11	0
8	10.1	7	3.1
9	13.39	42	0
10	3.18	4	0

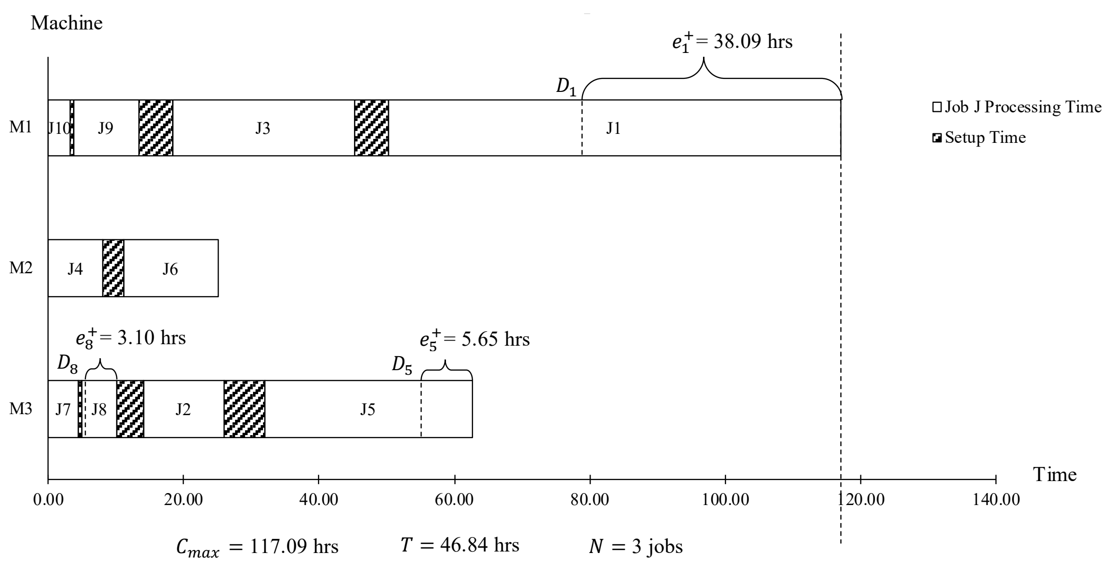

Figure 1. Jobs sequence on each machine from rule SCT

As shown in Table 1, the results of applying rule SCT display a makespan of 117.09 hours, with three jobs finished late and a total of 46.84 hours of tardiness. From Figure 1, Machine 1 processes jobs 10, 9, 3, and 1, while Machine 2 processes jobs 4 and 6, and Machine 3 processes jobs 7, 8, 2, and 5. Notably, jobs 1, 8, and 5 were completed beyond the due date.

4.2 Shortest Job Completion Time Based on the Longest Processing Time (SC-LPT)

Table 2. The completion time of jobs and their tardiness from rule SC-LPT

Job	Completion time	Due date	Tardiness
1	66.9	79	0
2	63.77	30	33.77
3	33.49	60	0
4	63.69	25	38.69
5	30.74	57	0
6	45.97	38	7.97
7	75.41	11	64.41
8	72.87	7	65.87
9	50.63	42	8.63
10	72.67	4	68.67

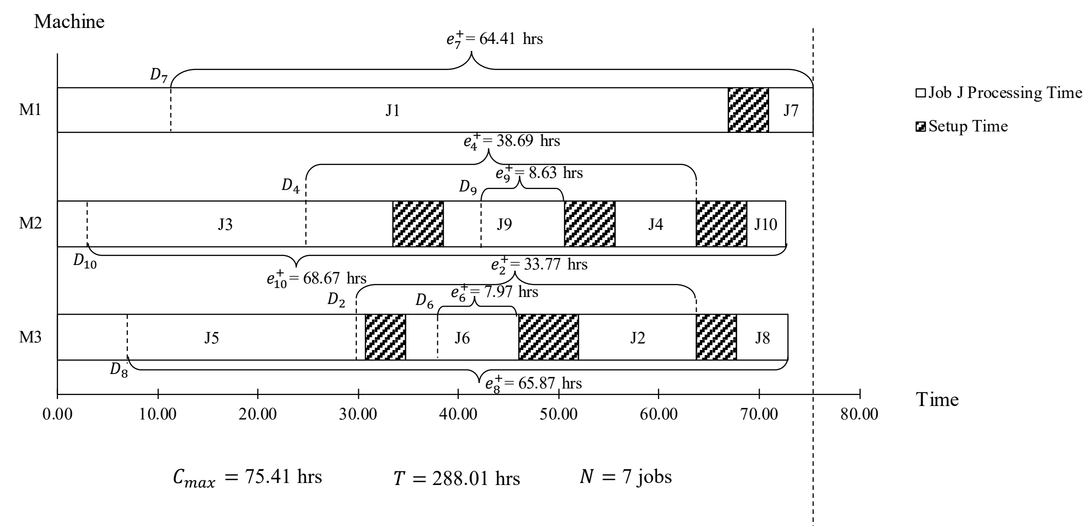

Figure 2. Jobs sequence on each machine from rule SC-LPT

Table 2 presents the results of applying rule SC-LPT, which indicate a makespan of 75.41 hours, with seven jobs being completed late, resulting in a total tardiness of 288.01 hours. Figure 2 displays the sequence of the job for each machine, indicating that Machine 1 is entrusted with processing jobs 1 and 7, Machine 2 handles jobs 3, 9, 4, and 10, and Machine 3 processes jobs 5, 6, 2, and 8. It is noteworthy that jobs 7, 4, 9, 10, 2, 6, and 8 exceeded their due dates.

4.3 Shortest Job Completion Time Based on the Earliest Due Date (SC-EDD)

Table 3. The completion time of jobs and their tardiness from rule SC-EDD

Job	Completion time	Due date	Tardiness
1	106.59	79	27.59
2	19.99	30	0
3	61.59	60	1.59
4	8.07	25	0
5	46.85	57	0
6	25.1	38	0
7	10.1	11	0
8	5.1	7	0
9	34.69	42	0
10	3.18	4	0

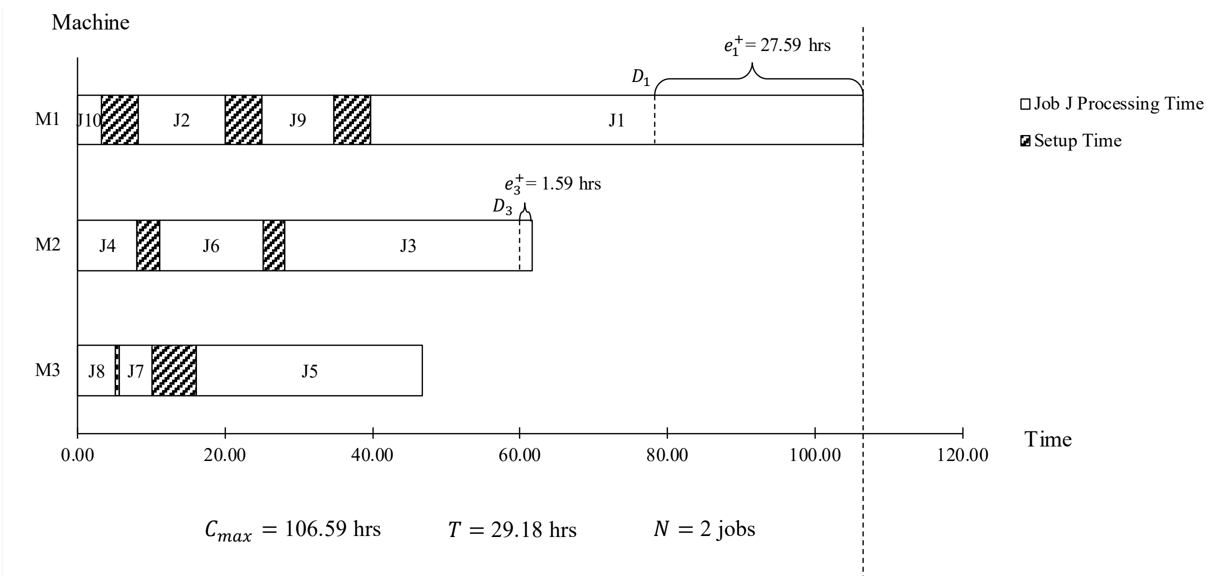

Figure 3. Jobs sequence on each machine from rule SC-EDD

Table 3 presents the results of applying rule SC-EDD, which indicate a makespan of 106.59 hours, with only two jobs being completed late, resulting in a total tardiness of 29.18 hours. Each machine's job sequence is shown in Figure 3, which shows that Machine 1 is in charge of processing jobs 10, 2, 9, and 1. Machine 2 handles jobs 4, 6, and 3. Jobs 8, 7, and 5 are processed by Machine 3. Notably, jobs 1 and 3 were completed after their due date.

4.4 Shortest Job Completion Time Based on the Minimum Slack (SC-MinSlack)

Table 4. The completion time of jobs and their tardiness from rule SC-MinSlack

Job	Completion time	Due date	Tardiness
1	75.08	79	0
2	25.91	30	0
3	72.93	60	12.93
4	8.07	25	0
5	46.99	57	0
6	43.13	38	5.13
7	10.1	11	0
8	5.1	7	0
9	64.13	42	22.13
10	3.18	4	0

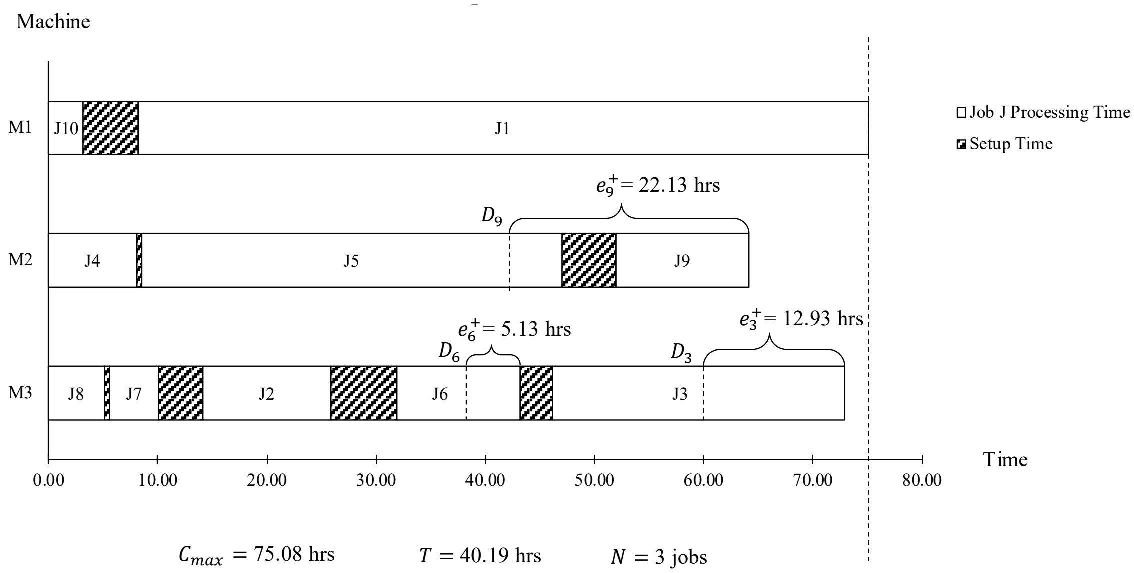

Figure 4. Jobs sequence on each machine from rule SC-MinSlack

Table 4 presents the results of applying rule SC-MinSlack, which indicates a makespan of 75.08 hours, with three jobs being completed late, resulting in a total tardiness of 40.19 hours. Each machine's job sequence is shown in Figure 4, which shows that Machine 1 is in charge of processing jobs 10, and 1. Machine 2 handles jobs 4, 5, and 9. Jobs 8, 7, 2, 6, and 3 are processed by Machine 3. Notably, jobs 9, 6, and 3 were completed after their due date.

4.5 Summary of Results for the Four Heuristic Methods

According to Table 5, among the four rules, SC-MinSlack provides the best makespan; SC-LPT offers the second-best makespan. While SC-MinSlack gives the second-best both total tardiness and the number of tardy jobs, SC-LPT provides the worst total tardiness and number of tardy jobs. In addition, SC-EDD produces the smallest total tardiness and the least number of tardy jobs, while producing a much larger makespan than those of the SC-LPT and SC-MinSlack. Unexpectedly, the SCT performs poorly in terms of the makespan, while producing relatively similar performance in terms of tardiness as SC-MinSlack. To summarize, the SC-MinSlack seems to perform well in all three measures of performance.

Table 5. Summary Results of the Four Heuristic Methods

N	Rules	Makespan	Total Tardiness	Number of Tardy Jobs
1	SCT	117.09	46.84	3
2	SC-LPT	75.41	288.01	7
3	SC-EDD	106.59	**29.18**	**2**
4	SC-MinSlack	**75.08**	40.19	3

5. Conclusion

In this paper, four heuristics are proposed to solve the unrelated parallel machine scheduling problem with the job- and machine-sequence-dependent setup time. Their effectiveness is evaluated according to three performance metrics, i.e., makespan, total tardiness, and the number of tardy jobs through a numerical example. The results indicate that SC-Minslack is the most effective heuristic method for minimizing makespan, while SC-EDD provides the best performance for both total tardiness and the number of tardy jobs. Although SC-LPT is the second-best heuristic method for minimizing makespan, it produces the worst results for both total tardiness and the number of tardy jobs. For future research, developing metaheuristic algorithms may be a promising avenue to obtain schedules that offer a better trade-off among the three measures of performance.

References

Bektur, G., & Saraç, T., 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, *Computers & Operations Research*, vol.103, pp. 46-63, 2019.

Cheng, C. Y., & Huang, L. W., Minimizing total earliness and tardiness through unrelated parallel machine scheduling using distributed release time control, *Journal of manufacturing systems*, vol.42, pp. 1-10, 2017.

Chen, K., Cheng, T. C. E., Huang, H., Ji, M., & Yao, D., Single-machine Scheduling with Autonomous and Induced Learning to Minimize Total Weighted Number of Tardy Jobs, *European Journal of Operational Research*, 2023.

Della Croce, F., T'kindt, V., & Ploton, O., Parallel machine scheduling with minimum number of tardy jobs: Approximation and exponential algorithms, *Applied Mathematics and Computation*, vol.397, 2021.

Fan, H., Xiong, H., & Goh, M., Genetic programming-based hyper-heuristic approach for solving dynamic job shop scheduling problem with extended technical precedence constraints. *Computers & Operations Research*, vol.134, 2021.

Jaklinović, K., Đurasević, M., & Jakobović, D. (2021). Designing dispatching rules with genetic programming for the unrelated machines environment with constraints. *Expert Systems with Applications*, 172, 114548.

Joo, C. M., & Kim, B. S., Hybrid genetic algorithms with dispatching rules for unrelated parallel machine scheduling with setup time and production availability, *Computers & Industrial Engineering*, vol.85, pp. 102-109, 2015.

Kommadath, R., Maharana, D., & Kotecha, P., Efficient scheduling of jobs on dissimilar parallel machines using heuristic assisted metaheuristic techniques, *Chemical Engineering Research and Design*, vol.188, 2022.

Lei, D., & Yang, H., Scheduling unrelated parallel machines with preventive maintenance and setup time: Multi-sub-colony artificial bee colony, *Applied Soft Computing*, vol.125, 2022.

Ozsoydan, F. B., & Sağir, M., Iterated greedy algorithms enhanced by hyper-heuristic based learning for hybrid flexible flowshop scheduling problem with sequence dependent setup times: a case study at a manufacturing plant, *Computers & Operations Research*, vol.125, 2021.

Munoz, L., Villalobos, J. R., & Fowler, J. W., Exact and heuristic algorithms for the parallel machine total completion time scheduling problem with dual resources, ready times, and sequence-dependent setup times, *Computers & Operations Research*, vol.143, 2022.

Soleimani, H., Ghaderi, H., Tsai, P. W., Zarbakhshnia, N., & Maleki, M., Scheduling of unrelated parallel machines considering sequence-related setup time, start time-dependent deterioration, position-dependent learning and power consumption minimization, *Journal of Cleaner Production*, vol.249, 2020.

Zhao, F., Bao, H., Wang, L., Xu, T., & Zhu, N., A heuristic and meta-heuristic based on problem-specific knowledge for distributed blocking flow-shop scheduling problem with sequence-dependent setup times, *Engineering Applications of Artificial Intelligence*, vol.116, 2022.

Zhang, L., Deng, Q., Lin, R., Gong, G., & Han, W., A combinatorial evolutionary algorithm for unrelated parallel machine scheduling problem with sequence and machine-dependent setup times, limited worker resources and learning effect, *Expert Systems with Applications*, vol.175, 2021.

Zheng, F., Jin, K., Xu, Y., & Liu, M., Unrelated parallel machine scheduling with processing cost, machine eligibility and order splitting, *Computers & Industrial Engineering*, vol.171, 2022.

Biographies

Prasal Soeurn is a master's degree student in Logistics and Supply Chain Systems Engineering in the international program at Sirindhorn International Institute of Technology, Thammasat University in Thailand. He earned an engineer's degree in Computer Science from the Institute of Technology of Cambodia in Phnom Penh, Cambodia.

Pham Duc Tai is a lecturer at Sirindhorn International Institute of Technology (SIIT), Thammasat University, Pathum Thani, Thailand. Formerly, he was a lecturer at the International Academy of Aviation Industry, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand. He received a bachelor's degree in Industrial Systems Engineering from Ho Chi Minh City University of Technology, Vietnam, a master's degree in Logistics and Supply Chain Systems Engineering, and a doctoral degree in Engineering and Technology from SIIT, Thammasat University, Thailand. His research interests include those related to supply chain management, such as supply chain network design (SCND), supply chain optimization, retail management, inventory management, supply contract, marketing-operations interface, bullwhip effect, and information sharing.

Jirachai Buddhakulsomsiri is an Associate Professor in the School of Manufacturing Systems and Mechanical Engineering, Sirindhorn International Institute of Technology (SIIT), Thammasat University, Thailand. He received his Ph.D. in Industrial Engineering from the Department of Industrial and Manufacturing Engineering, Oregon State University, USA, in 2003. His research interests include (1) logistics and supply chain systems in the problem areas of supply chain network design, vehicle routing problems, inventory optimization, (2) data analysis using machine learning, computer visions, statistical analysis, and (3) system modeling using discrete-event system simulation, Monte Carlo simulation, system dynamics.