The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM)
ISSN: 2602-3199

The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM), 2024

Volume 32, Pages 311-319

IConTES 2024: International Conference on Technology, Engineering and Science

Parallel Machine Scheduling with Re-entrant Jobs with Consideration of Set up Times

Betul Kayisoglu
Abdullah Gul University

Seyma Bekli
Abdullah Gul University

Ayse Sena Sahin
Abdullah Gul University

Gamze Gul Akyurek
Abdullah Gul University

Ruveyda Aydinli
Abdullah Gul University

Sevda Nur Copur
Abdullah Gul University

Tugba Ekinci
Abdullah Gul University

Abstract: We study the identical parallel machine problem with re-entrant jobs. Re-entrant jobs require to pass through the processing line multiple times. In many real-life manufacturing systems with parallel machine environments, one of the scheduling problems that needs to be addressed is the order of jobs on each machine with re-entrant jobs. In addition, manufacturing systems may require periodic maintenance, systematic manufacturing equipment cleaning, or predetermined upper limits on the overtime. Therefore, machine availability may vary during the scheduling horizon. We propose an integer programming model to find the optimal sequence of the re-entrant jobs at parallel machines with consideration of machine availability. The model aims to reduce setup times and maximize capacity utilization by scheduling tasks with similar set up requirements consecutively. We tested the proposed model at a panel line manufacturing company located in Turkey. The order of the panels is scheduled optimally by the proposed model for 3 different instances on the identical parallel machines for the coating process. We also provided relevant information on the user interface we developed to make the proposed scheduling model usable to by the company. The proposed model and interphase offer a systematic approach to panel line planning and can also be implemented in other industries.

Keywords: Production scheduling, Re-entrant scheduling, Parallel machine scheduling, Sustainable manufacturing, Integer programming model.

Introduction

In today's globalizing world, one of the key aspects for companies to survive and compete is efficiency. In manufacturing companies, production efficiency could be considered as the primary factor to success. Production efficiency can be defined as a metric of how well the companies are using their available resources to produce the goods/services they are providing. The production efficiency of a manufacturing company (or shop floor) can be found by comparing the actual production amount to the company's (shop floor's) theoretical maximum value. Therefore, understanding the maximum achievable value as the production capacity is important. Production efficiency depends heavily on effective scheduling, as optimized schedules allocate resources optimally, minimizing idle time and/or reducing the setup time between tasks.

The scheduling problem has been of interest to researchers since it was first introduced by (Johnson (1954). The literature on scheduling is quite extensive and covers process optimizations that have been developed over many years. Since the manufacturing process varies by product, different solution methods and algorithms have been created theoretically. Processes involving re-entrant jobs differ from standard machine scheduling operations and vary in this respect.

Re-entrant scheduling problems are characterized by the jobs entering a facility or a machine more than once. Many real-world manufacturing processes require products to be processed on the same machine more than once. Mirror manufacturing, dying process, printed circuit board production, tests done more than once in microbiological laboratories, and surgery schedules when patients need operations more than once could be given as re-entrant scheduling examples Chakhlevitch and Glass (2009), Choi and Kim (2008), Wang et al. (2021). Out of many re-entrant scheduling problems, motivated by a panel line manufacturing facility, we focus on an identical parallel machine scheduling problem with re-entrant jobs (IPMS_RJ). In real-world manufacturing environments, not all the machines are available throughout the given time horizon. Periodic maintenance or cleaning reduces the availability of the machines for a certain period.

Even though parallel machine literature is vast, scheduling re-entrant jobs in parallel machines has not been adequately addressed. To the best of our knowledge, no study has examined IPMS_RJ with the consideration of deterministic machine unavailability. To close this literature gap, we present a new integer programming model that optimizes the sequence of the jobs which may or may not require additional processing on parallel machines with deterministic unavailability.

The rest of this paper is organized as follows: first, we present a summary of related literature. Next, an integer programming model is introduced to optimally solve the IPMS_JS problem. We discuss the results in the Computational Experiments section using real-world data. We also give an interphase for solving the IPMS_JS problem. Finally, concluding remarks and future research directions are given.

Related Literature

In this section, we present a literature review on scheduling with a focus on re-entrant scheduling. For a detailed re-entrant scheduling literature review, readers are referred to Danping and Lee (2011). Re-entrant scheduling could be analyzed through the single-machine environment, flow-shop, and job-shop environments (Chen & Chao-Hsien Pan, 2006). Since we have a parallel machine environment in our problem, we focus on multi-machine environments (i.e. flow-shop and job-shop environments) in the related literature.

In the flow-shop environment, all the jobs require similar manufacturing techniques. This is why; the machine sequence of the jobs is similar. Minimizing makespan in a re-entrant flow-shop environment is proved to be at least binary NP-hard for even 2-machine case Wang et al. (1997) and Chen et al. (2008) investigate the use and performance of hybrid genetic algorithms while minimizing makespan for re-entrant flow-shop environments. The study shows that hybrid genetic algorithms work well in finding the near-optimal values. Choi and Kim (2007) also investigates similar problems of minimization makespan by using different types of heuristic algorithms such as lower bound-based, idle time-based, constructive heuristics, and simulated annealing algorithms. The computational experiments show that the suggested heuristics work well and provide good results in a short amount of time. Chamnanlor et al. (2017) considered minimizing makespan at a re-entrant flow-shop with time-window constraints. This study is representative of real-life manufacturing problems hard-disk drive sector where the product quality has to be monitored within certain time intervals. Genetic algorithms and ant colony algorithms are used together as a heuristic algorithm.

Since parallel machine environments can be considered a special type of flexible job-shop environment, we present literature on re-entrant scheduling in job-shop environments. In job-shop environments, each job follows its predetermined path. Low et al. (2005) investigated re-entrant scheduling in a job shop environment with the sequence-dependent set-up time consideration. Three optimization models are proposed for minimizing total job flow time, total tardiness, and machine idle. Since the optimization models related to job-shop scheduling with re-entrant jobs are computationally intractable, many of researchers have focused on heuristic algorithms to provide near-optimal solutions. Zoghby et al. (2005) addressed the re-entrant scheduling problem by incorporating sequence-dependent set-up times in a job-shop environment. They have investigated the feasibility conditions for constructing heuristic algorithms. Sun (2009) also studied a similar problem where set-up times are defined by the sequences of the machines. Two heuristic algorithms and a genetic algorithm have been developed as solution algorithms. The computational experiments show that the genetic algorithm is providing better quality solutions within a reasonable computational time. Mason et al. (2002) studied a re-entrant job-shop problem with the goal of minimizing total weighted tardiness. A modified shifting bottleneck-type heuristic is proposed as a solution algorithm. The proposed algorithm is tested using a semiconductor manufacturer's data. It is shown that the proposed heuristic algorithm works well in delivering the orders on time. Belkaid et al. (2016) studied the re-entrant parallel machine scheduling problem under the consideration of consumable resources. It is assumed that each job consumes a certain number of resources to be processed at the machines. An integer linear programming model and a genetic algorithm are proposed for solving the problem to minimize makespan. The focus of the research reported in this paper is on modelling the problem of optimal sequences of the jobs in parallel machines taking into account re-entrant jobs and changes in machine availability. This problem is intended to represent a real-world scheduling problem that is often encountered when using the identical parallel machines with jobs requiring more than one process at the machines such as coating of the panel lines. This study contributes to the current state of knowledge by introducing a new integer programming model that aims to find the optimal schedules of the jobs on parallel machines while considering the required time/shift between consecutive executions of the re-entrant jobs.

Problem Definition

We focus on scheduling of jobs on identical parallel machines with re-entrant jobs. By re-entrant, we mean that certain jobs are processed once, while others re-enter the system after a waiting period, such as for drying. We consider a scheduling horizon, which is a predetermined period, such as a day, week, or month, within which all jobs will be processed. This scheduling horizon is divided into shifts, represented by the set K. The jobs to be processed during this scheduling horizon are known in advance. These jobs are assigned to a group of identical parallel machines, meaning each job can be processed on any machine since they share the same capabilities. Let L represent the set of parallel machines available for processing. Each machine may be restricted to certain shifts; for example, some machines may be under maintenance and may not work on predetermined shifts. We define a parameter b_{kl} to specify the shifts that machine l will work within the scheduling horizon, b_{kl} gets the value 1 if machine l can work on shift k, 0 otherwise.

Let I represent the set of jobs with subsets I_1 for jobs processed once and I_2 for re-entrant jobs processed more than once. We define a parameter m_i to denote the number of times job $i \in I$ requires processing. Additionally, we create another subset I_3 for dummy jobs that are not actual jobs but are introduced for modeling purposes. Each parallel machine has a dummy starting job and a dummy ending job. Let n be the number of jobs. We assign job 0 as the starting dummy job and job n+1 as the ending dummy job for each parallel machine. Thus, $I = I_1 \cup I_2 \cup I_3$. Each job has processing time that we represent by the parameter p_i where $i \in I$. For dummy jobs in I_3 , we set $p_i = 0$. When a job enters the system, it requires a setup time before processing begins. If the preceding job is similar to the current job, the setup time may be reduced. Therefore, the setup time varies depending on the specific sequence of jobs. We define the parameter s_{ij} to represent the setup time required when job j is processed immediately after job i, where $i, j \in I$. The aim of IPMSP_RJ is to minimize total setup time.

Mathematical Model for IPMSP_RJ

In this study, a mathematical model was developed for IPMSP_RJ presented below.

Sets

I	$Set\ of\ jobs$
I_1	$Set\ of\ jobs\ processed\ once$
I_2	$Set\ of\ re-entrant\ jobs\ that\ are\ processed\ multiple\ times$

I_3	$Set\ of\ dummy\ jobs\ that\ are\ not\ actual\ jobs$
K	$Set\ of\ shifts$
L	$Set\ of\ identical\ machines$

Parameters

p_i	$Processing\ time\ of\ job\ i \in I$
s_{ij}	$Setup\ time\ required\ when\ job\ j\ is\ processed\ immediately\ after\ job\ i, where\ i, j \in I$
v_k	$Total\ working\ time\ of\ shift\ k \in K$
b_{kl}	$1\ if\ machine\ l \in L\ can\ work\ on\ shift\ k \in K, 0\ otherwise$
a	$Number\ of\ shifts\ re-entrant\ jobs\ must\ wait\ before\ reentering\ the\ system$
m_i	$Number\ of\ times\ job\ i\ \in I\ is\ processed$

Decision Variables

Y_{jlk}	1,	$if\ job\ j \in I\ is\ assigned\ to\ machine\ l\ \in L\ in\ shift\ k \in K;\ 0\ otherwise$
X_{kijl}	1,	$if\ job\ j \in I\ is\ assigned\ to\ machine\ l \in L$ just after job i $\in I\ in\ shift\ k \in K;$ $0, otherwise$
U_{ikl}		$Order\ of\ job\ i \in I\ on\ machine\ l \in L\ in\ shift\ k \in K$

With these definitions, the proposed model for IPMSP_RJ is given below:

$$Z = \text{Min} \sum_{i,j,l,k} s_{ij}\ X_{ijkl} \quad (1)$$

s.t.

$$\sum_{l,k} Y_{ilk} = 1 \qquad \forall i\ \in I_1 \quad (2)$$

$$\sum_{l,k} Y_{ilk} = m_i \qquad \forall i\ \in I_2 \quad (3)$$

$$Y_{ilk}\ \leq b_{lk} \qquad \forall i\ \in I, \forall k\ \in\ K, \forall l\ \in\ L \quad (4)$$

$$Y_{0,l,k} = 1 \qquad \forall k\ \in\ K, \forall l\ \in\ L \quad (5)$$

$$Y_{n+1,l,k} = 1 \qquad \forall k\ \in\ K, \forall l\ \in\ L \quad (6)$$

$$\sum_l \sum_{k+1}^{k+a} Y_{ilk}\ \leq 1 - \sum_l Y_{ilk} \qquad \forall i\ \in I_2,\ \forall k \in\ K \quad (7)$$

$$\sum_{i,j} s_{ij}\ X_{kijl} + \sum_j p_j\ Y_{jlk}\ \leq\ v_k \qquad \forall k\ \in\ K, \forall l\ \in\ L \quad (8)$$

$$\sum_j X_{kijl}\ =\ Y_{ilk} \qquad \forall k\ \in\ K,\ \forall l\ \in\ L,\ \forall i\ \in I_1 \cup I_2 \quad (9)$$

$$\sum_j X_{kjil}\ = Y_{ilk} \qquad \forall k\ \in\ K,\ \forall l\ \in\ L, \forall i\ \in I_1 \cup I_2 \quad (10)$$

$$\sum_j X_{k,j,0,l}\ = 0 \qquad \forall k\ \in\ K, \forall l\ \in\ L \quad (11)$$

$$\sum_j X_{k,n+1,j,l}\ = 0 \qquad \forall k\ \in\ K, \forall l\ \in\ L \quad (12)$$

$$\sum_j X_{k,0,j,l}\ = 1 \qquad \forall k\ \in\ K, \forall l\ \in\ L \quad (13)$$

$$\sum_j X_{k,j,n+1,l}\ = 1 \qquad \forall k\ \in\ K, \forall l\ \in\ L \quad (14)$$

$$U_{ikl} - U_{jkl} + (n+1)\ X_{kijl} \leq n \qquad \forall k\ \in\ K,\ \forall l\ \in\ L, \forall i,j\ \in I \quad (15)$$

$$X_{kijl}\ + X_{kjil}\ \leq 1 \qquad \forall k\ \in\ K,\ \forall l\ \in\ L, \forall i,j\ \in I \quad (16)$$

$$Y_{ilk} \in \{0,1\} \qquad \forall k\ \in\ K,\ \forall l\ \in\ L, \forall i\ \in I \quad (17)$$

$$X_{kijl} \in \{0,1\} \qquad \forall k\ \in\ K,\ \forall l\ \in\ L, \forall i,j\ \in I \quad (18)$$

$$U_{ikl} \geq 0 \qquad \forall k\ \in\ K,\ \forall l\ \in\ L, \forall i\ \in I \quad (19)$$

The model aims to minimize the total setup time required on all the machines during the scheduling horizon as defined in the objective function (1). Constraints (2) assign the jobs once that require only a single processing stage, while Constraints (3) assign each re-entrant jobs i, m_i times to the machines. Constraints (4) ensure that assignments to machine l in shift k is possible if machine l can work during shift k; otherwise, assignments are not allowed. Constraints (5) and (6) ensure that dummy starting and ending jobs will be assigned to each machine in every shift. When a re-entrant job is processed, Constraints (7) ensure that it waits at least a shifts before re-entering the system. Constraints (8) impose that the total setup and production time of a machine in each shift should not exceed the total available working time for that machine in that shift. Constraints (9) and

(10) ensure that if a non-dummy job is assigned to a machine in a shift, a job must be scheduled before and after that job, respectively. Constraints (11) prevent any job from being scheduled before the starting job, while Constraints (12) prevent any job from being scheduled after the ending job in each shift on each machine. Similarly, Constraints (13) impose a job to be scheduled after the starting job, and Constraints (14) impose a job to be scheduled before the ending dummy job. Constraints (15) determine the job sequence of each machine at each shift. Constraints (16) ensure that job i is scheduled either before or after job j. Constraints (17), (18) and (19) define the domains of the variables.

Case Study: IPMSP_RJ in Panel Line Production

This study examines the identical parallel machine scheduling problem in the panel production section of a company that specializes in providing semi-finished wood-based panels used in furniture and interior decoration. The company is located in Turkey. The process involves applying coatings to the panels, which include glue and coating materials (bobbins). Product characteristics dictate the need for glue and bobbin changes, depending on order specifics. When transitioning between different product types, a bobbin change is required. Moreover, since the type of glue can vary depending on the product, there may be glue setup process that is more complex and time-consuming than the bobbin change. The glue change process begins with washing and drying the tank, adding new glue, and resuming production. The company knows in advance the setup times required for both bobbin and glue when transitioning between different product types. There are two identical parallel machines with same capabilities operating day and night shift in a day. The company creates a weekly production plan based on incoming orders, scheduling all products to be produced in two shifts each day: a day shift and a night shift, over five days, for a total of 10 shifts. During the day shifts, both machines operate simultaneously. However, on night shifts, only one machine is operating, as the other undergoes a scheduled maintenance in a rotating sequence each night. Certain products require double-sided surface coating. After coating the front side, the products are left in a designated area for one day before the backside can be coated. The production planning department knows the time required to coat each product and calculates the total coating time for each order based on the order quantity.

The aim of the company is to schedule the orders (jobs) in a way that minimizes total setup time. While the setup times required between each job are known in advance, sequencing jobs with lower setup times may seem like a viable solution. However, re-entrant jobs that enter the system the next time at least two shifts later can encounter significantly higher setup times. This highlights the need for precise scheduling in the production line. The lack of planning in sequencing can lead to job delays, increased lead times, and consequently, missed deadlines, ultimately resulting in customer dissatisfaction.

Computational Experiments with the Mathematical Model for IPMSP_RJ

We conduct computational experiments to verify and validate the proposed model developed for IPMSP_RJ. Also with these computational experiments we tested the performance of the proposed model. The company creates a weekly production plan based on incoming orders. For our analysis, we got three distinct weekly production plans from the company, referred to as Case 1, Case 2, and Case 3. These cases contain 74, 88, and 100 jobs, respectively. We used CPLEX to solve each case and obtained the optimal solutions. Moreover, we conducted a sensitivity analysis on the number of the shifts. The company is operating across 10 shifts per week, but we examined the impact of reducing the number of shifts. When we rerun our model with fewer shifts, we found that the same objective function value could be achieved with 8 shifts in all three cases, effectively reducing the whole schedule by one working day. When we tested the model with 7 shifts, we were unable to achieve a feasible solution. Based on these results, we continued our analysis using an 8-shift schedule.

Figure 1 presents the Gantt chart for the schedule obtained by the proposed model in Case 1. The left side of the chart represents Line 1, that operates on both day and night shifts, while the right side shows Line 2, that operates only during day shifts. Each section of the chart shows the job schedule in a single shift. Dark blue segments denote setup times, whereas light blue segments represent production times. In this schedule, all jobs have been allocated within the production plan. Case 1 includes six re-entrant jobs, which require re-processing after at least 2-shift long waiting period. For instance, Job 39, a re-entrant job, is firstly processed on Day 1 in the night shift on Machine 1 and afterwards processed on Day 3 in the day shift on Machine 2. This schedule meets the requirement of a minimum waiting period of two shifts between re-entries. Re-entrant jobs can return to any machine, as both machines are identical in function.

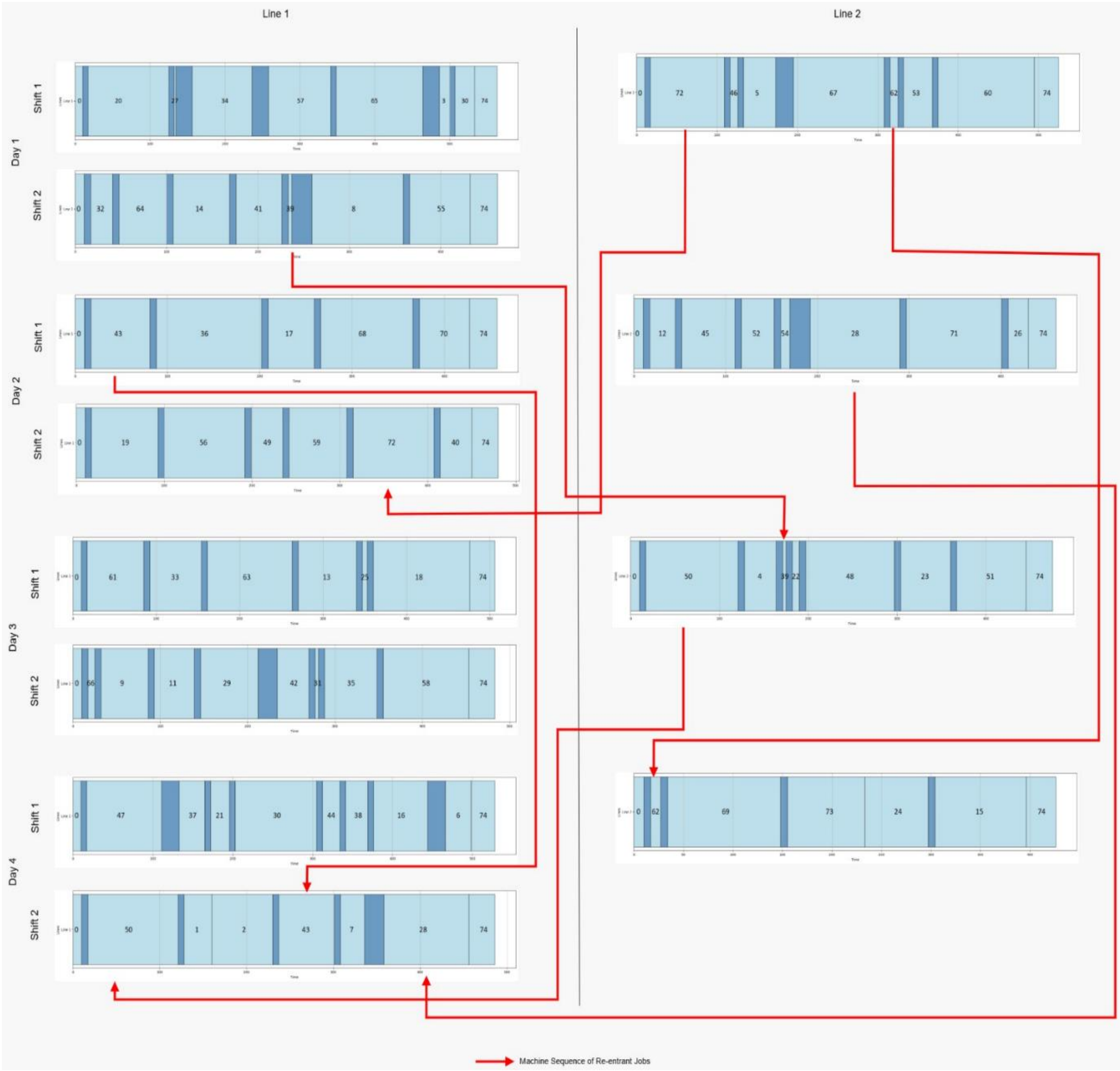

Figure 1. Gannt chart of the schedule obtained by the proposed model in Case 1

Figure 2 shows the percentage of total setup and production times for each case. In case 1, there is a higher proportion of similar jobs, resulting in a setup time percentage of 8%, while case 3 has jobs requiring longer setups, with the setup time percentage reaching 20%. The average setup time across all cases is 14%, resulting an average production time of 86%. This production percentage shows a worthy level of productivity, particularly when compared to the company's current productivity rates.

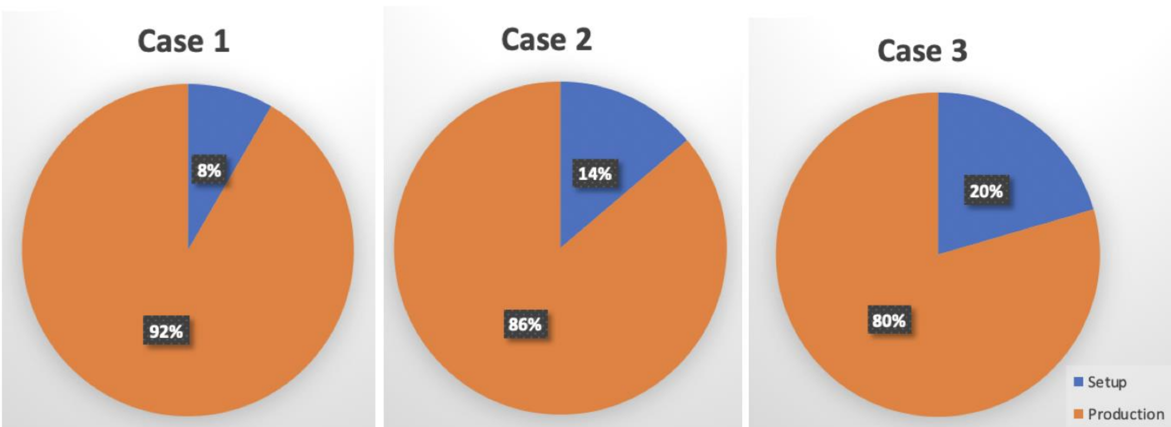

Figure 2. Percentage of setup and production times across three cases

Interface Developed for IPMSP_RJ.

In this study, we present a user-friendly Phyton-based interface developed for solving the proposed model for IPMSP_RJ in panel line production. This interface integrates the proposed model that operates in the background to schedule the jobs. The primary objective of the interface is to enhance accessibility for users, enabling them to input custom datasets and achieve scheduling of the jobs with minimum setup times. The interface design prioritizes usability, offering a step-by-step workflow for users from data input to solution visualization. After successful login, users upload a dataset formatted to system specifications. Once uploaded, the dataset is displayed for verification, enhancing data accuracy and transparency in the scheduling process.

Users define the scheduling duration by specifying the number of shifts in the “Shift Select” section. The “Make a Scheduling” function initiates the scheduling process, after which users can visualize the schedule through a Gantt chart. This chart displays the sequence of products assigned to each shift and production line, providing a clear representation of the production timeline and order distribution. Figure 3 shows an example for the Gantt charts for schedules obtained by the interface developed and Figure 4 gives the details of a Gantt chart. In summary, the interface efficiently guides users through dataset input, scheduling configuration, and solution visualization. This structured approach ensures a seamless experience while facilitating optimal scheduling in complex production environments.

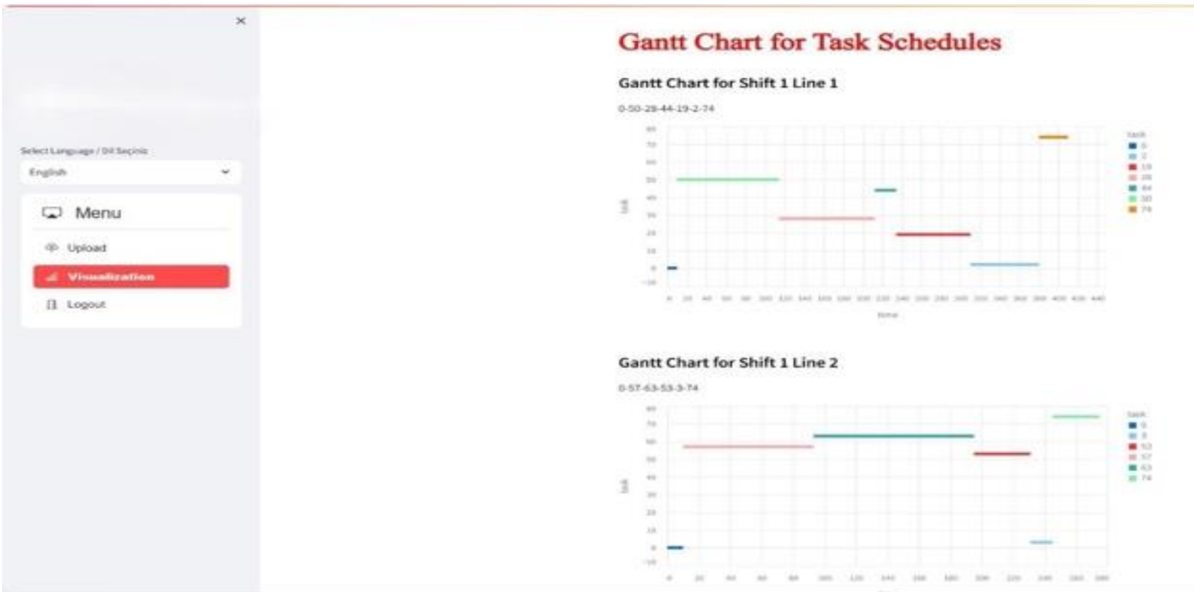

Figure 3. Gantt charts presented by the interface

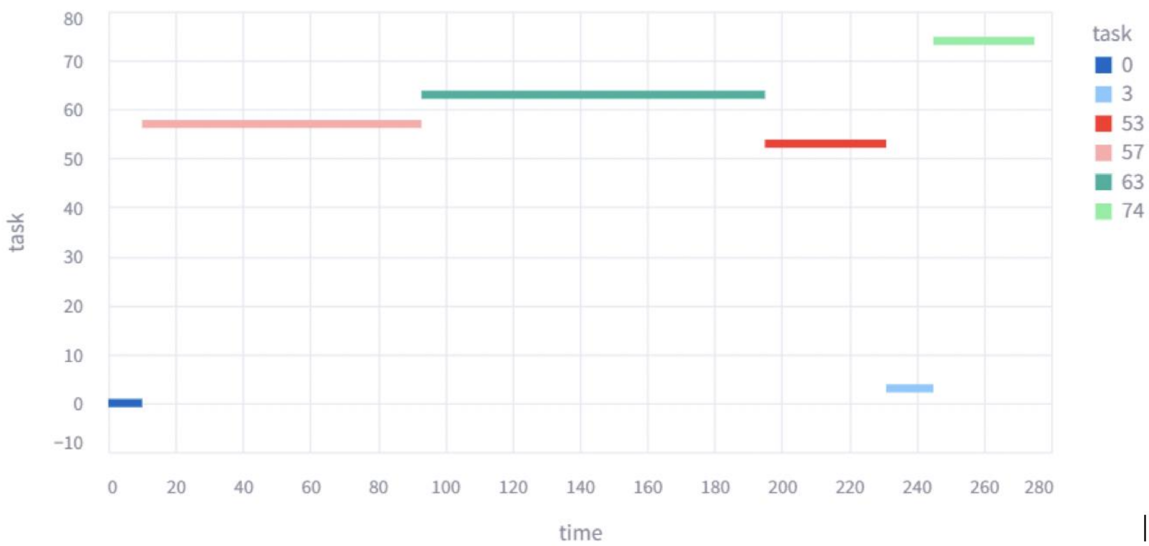

Figure 4. Details of a gantt chart presented by the interface

Concluding Remarks and Future Research Directions

This study addresses IPMSP_RJ, an identical parallel machine scheduling problem with re-entrant jobs that re-enter the system after a designated waiting period with the objective of minimizing the total setup time in the scheduling horizon. Minimizing setup time guarantees the production efficiency. In today's industrial environment, production efficiency is critical for companies to remain competitive.

An integer programming model is developed for IPMSP_RJ. This model takes into account re-entrant job requirements and deterministic machine unavailability, that is a realistic feature often seen in industrial settings. The model was tested on three distinct cases from the production facility, with the number of jobs 74, 88, and 100 respectively. We use CPLEX to obtain optimal solutions. When we investigate the solutions, we validated that the model represents the problem accurately, satisfying all specified requirements. To further understand the impact of number of shifts, a sensitivity analysis was conducted on it. While the company typically schedules over 10 shifts per week, we observe that same production outcomes can be achieved with only 8 shifts. This results in savings of one working day per week, improving resource utilization. Moreover, we developed a Python-based user interface to make the proposed scheduling model usable to production planners in the company. This interface allows users to input datasets, decide number of shifts, and visualize the optimal schedules via Gantt charts. The interface is designed to provide a valuable tool for the production planning personnel to generate production schedules without extensive technical knowledge of scheduling models.

In conclusion, by incorporating re-entrant job requirements and machine unavailability constraints into a parallel machine scheduling model, this study addresses an important gap in the literature. There is also a case study from real life showing the applicability of the proposed mathematical model. The analysis shows the potential for increased efficiency in complex production scheduling problems and provide a framework for further research on re-entrant scheduling problems. Future studies could extend this model by exploring stochastic machine availability, considering different objectives, e.g., minimizing makespan or/and number of shifts or integrating additional constraints based on more different industrial settings.

Scientific Ethics Declaration

The authors declare that the scientific ethical and legal responsibility of this article published in EPSTEM Journal belongs to the authors.

Acknowledgements or Notes

This article was presented as an poster presentation at the International Conference on Technology, Engineering and Science (www.icontes.net) held in Antalya/Turkey on November 14-17, 2024.

References

Belkaid, F., Yalaoui, F., & Sari, Z. (2016). An efficient approach for the reentrant parallel machines scheduling problem under consumable resources constraints. *International Journal of Information Systems and Supply Chain Management*, *9*(3), 1–25.

Chakhlevitch, K., & Glass, C. A. (2009). Scheduling reentrant jobs on parallel machines with a remote server. *Computers & Operations Research*, *36*(9), 2580–2589.

Chamnanlor, C., Sethanan, K., Gen, M., & Chien, C.-F. (2017). Embedding ant system in genetic algorithm for re-entrant hybrid flow shop scheduling problems with time window constraints. *Journal of Intelligent Manufacturing*, *28*(8), 1915–1931.

Chen, J. S., & Pan, C. H. J. (2006). Integer programming models for the re-entrant shop scheduling problems. *Engineering Optimization*, *38*(5), 577–592.

Chen, J. S., Pan, J. C.H., & Lin, C. M. (2008). A hybrid genetic algorithm for there-entrant flow-shop scheduling problem. *Expert Systems with Applications*, *34*(1), 570–577.

Choi, S.-W., & Kim, Y.-D. (2007). Minimizing makespan on a two-machine re-entrant flowshop. *Journal of the Operational Research Society*, *58*(7), 972–981.

Choi, S.-W., & Kim, Y.-D. (2008). Minimizing makespan on an -machine re-entrant flowshop. *Computers & Operations Research*, *35*(5), 1684–1696.

Danping, L., & Lee, C. K. M. (2011). A review of the research methodology for the re-entrant scheduling problem. *International Journal of Production Research*, *49*(8), 2221–2242.

Johnson, S. M. (1954). Optimal two- and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly*, *1*(1), 61–68.

Low, C., Wu, T.-H., & Hsu, C.M. (2005). Mathematical modelling of multi-objective job shop scheduling with dependent setups and re-entrant operations. *The International Journal of Advanced Manufacturing Technology*, *27*(1–2), 181–189.

Mason, S. J., Fowler, J. W., & Matthew Carlyle, W. (2002). A modified shifting bottleneck heuristic for minimizing total weighted tardiness in complex job shops. *Journal of Scheduling*, *5*(3), 247–262.

Sun, J. U. (2009). A genetic algorithm for a re-entrant job-shop scheduling problem with sequence-dependent setup times. *Engineering Optimization*, *41*(6), 505–520.

Wang, K., Qin, H., Huang, Y., Luo, M., & Zhou, L. (2021). Surgery scheduling in outpatient procedure centre with re-entrant patient flow and fuzzy service times. *Omega*, *102*, 102350.

Wang, M. Y., Sethi, S. P., & van de Velde, S. L. (1997). Minimizing makespan in a class of reentrant shops. *Operations Research*, *45*(5), 702–712.

Zoghby, J., Wesley Barnes, J., & Hasenbein, J. J. (2005). Modeling the reentrant job shop scheduling problem with setups for metaheuristic searches. *European Journal of Operational Research*, *167*(2), 336–348.

Author Information

Betul Kayısoglu
Abdullah Gul University, Kayseri, Türkiye
Email: betul.kayisoglu@agu.edu.tr

Seyma Bekli
Abdullah Gul University, Kayseri, Türkiye

Ayse Sena Sahin
Abdullah Gul University, Kayseri, Türkiye

Gamze Gul Akyurek
Abdullah Gul University, Kayseri, Türkiye

Ruveyda Aydınlı
Abdullah Gul University, Kayseri, Türkiye

Sevda Nur Copur
Abdullah Gul University, Kayseri, Türkiye

Tugba Ekinci
Abdullah Gul University, Kayseri, Türkiye

To cite this article:

Kayısoglu, B., Bekli, S., Sahin, A. S., Akyurek, G. G., Aydınlı, R., Copur, S. N., & Ekinci, T. (2024). Parallel machine scheduling with re-entrant jobs with consideration of set up times. *The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM), 32,* 311-319.