

Evaluation of Integer Programming Solvers to Improve the Efficiency of Individual Work Planning

Keiichi Takahashi

Faculty of Humanity-Oriented Science and Engineering, Kindai University, Fukuoka, Japan

Email: ktakahas@fuk.kindai.ac.jp

How to cite this paper: Takahashi, K. (2025) Evaluation of Integer Programming Solvers to Improve the Efficiency of Individual Work Planning. *Open Journal of Modelling and Simulation*, 13, 51-64.
<https://doi.org/10.4236/ojmsi.2025.131003>

Received: December 3, 2024

Accepted: January 11, 2025

Published: January 14, 2025

Copyright © 2025 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

This study proposes a novel approach to optimizing individual work schedules for book digitization using mixed-integer programming (MIP). By leveraging the power of MIP solvers, we aimed to minimize the overall digitization time while considering various constraints and process dependencies. The book digitization process involves three key steps: cutting, scanning, and binding. Each step has specific requirements and limitations such as the number of pages that can be processed simultaneously and potential bottlenecks. To address these complexities, we formulate the problem as a one-machine job shop scheduling problem with additional constraints to capture the unique characteristics of book digitization. We conducted a series of experiments to evaluate the performance of our proposed approach. By comparing the optimized schedules with the baseline approach, we demonstrated significant reductions in the overall processing time. In addition, we analyzed the impact of different weighting schemes on the optimization results, highlighting the importance of identifying and prioritizing critical processes. Our findings suggest that MIP-based optimization can be a valuable tool for improving the efficiency of individual work schedules, even in seemingly simple tasks, such as book digitization. By carefully considering specific constraints and objectives, we can save time and leverage resources by carefully considering specific constraints and objectives.

Keywords

Mixed Integer Programming, Scheduling Optimization, Work Planning, Book Scanning, Gantt Chart

1. Introduction

Mixed Integer Programming (MIP) solvers are advanced software tools designed

to solve complex optimization problems involving integer and continuous variables [1]. These solvers have been applied to a variety of real-world optimization problems, such as operational planning and scheduling problems, and have become indispensable tools in areas such as operational research and supply chain management [2]. Research on MIP solvers dates back to the 1960s, but in recent years, machine-learning techniques and their integration with machine learning have attracted much attention [3]. For example, studies using neural networks to improve the key subtasks of MIP solvers have shown very high performance compared to conventional MIP solvers [4]. The application of MIP solvers to real-world problems has resulted in various benefits. For example, they have been used in the automotive industry to optimize large-scale vehicle production sequences [5]. In the energy industry, MIP solvers are used to optimize power plant schedules, helping reduce costs and environmental impacts [6]. Current MIP solver research addresses a wide range of issues, including increased scalability for larger and more complex problems, further integration with machine learning, and improved parallel and distributed computing capabilities [7]-[9]. Research is also being conducted on the use and integration of large language models (LLMs), which have received considerable attention in recent years [10] [11]. There are many research cases in which the MIP solver has been used to optimize work planning, including in the medical field, manufacturing, logistics, and distribution operations. In the medical field, MIP solver is used to optimize shift planning and task allocation for doctors and nurses [12]-[14]. In the manufacturing industry, it optimizes the work schedules of workers and equipment operations [15]-[17]. In the logistics field, it plans for efficient execution of tasks by specific delivery personnel [18]-[20]. To the best of our knowledge, no case studies have been reported in recent years on the application of an MIP Solver to individual work planning.

To expand the potential of MIP solvers, it is important for the development of the field to apply and evaluate MIP solvers, even on a small scale, in areas where MIP solvers have not been tried before. Therefore, in this study, we will optimize individual work schedules using MIP solvers.

In recent years, small and low-cost tablet devices have been offered by many companies, and the number of users who enjoy gathering information and reading books on such devices has increased [21] [22]. Some tablet users not only use commercially available e-books but also digitize their own paper books and read them on their tablets [23]. Under these circumstances, image scanners specialized for digitizing paper documents and books have been offered at low prices for several years, and their sales are increasing [24] [25]. Although image scanners that can be used by individuals have been around for a long time, the models offered in recent years have been improved with an auto-feed function that automatically reads multiple sheets of paper as a standard feature, making it easier to digitize books and other documents with many pages. In this article, these image scanners are referred to as book-scanners. There are several methods for digitizing books, and one of them, the destructive method, is used here [23]. This method consists

of three steps: cutting the book, scanning the pages, and binding the cut pages. None of these processes is difficult, but if many books are to be digitized, the time required for the work will be longer, and several processes will be performed in parallel. Because the work of digitizing books is an individual task, there is no specific deadline. However, because all processes are monotonous tasks, the work should be completed in as short a time as possible.

In this paper, we propose a method of preplanning using a MIP solver to solve these problems [26]. We evaluated the efficiency of digitizing books by following the work plan through experiments using numerical simulations. To develop the work plan, we used Gurobi Optimizer, a MIP solver provided by Gurobi Optimization Inc [27]. The remainder of this paper is organized as follows. Section 2 defines the book digitization work planning problem, and Section 3 describes the preliminary experiments conducted to obtain the lead-time for each step of the book digitization process. Section 4 describes the simulation experiments using a MIP solver based on the results of the preliminary experiments. Section 5 discusses the differences in the results of the simulation experiments depending on the optimization strategy. Section 6 describes the limitations of this study. Finally, Section 7 concludes the study.

2. Book Digitization Work Planning Problem (BDWPP)

2.1. Process Description

The book digitization work to be optimized in this study is described. The work process is presented in **Table 1**. There are three processes involved: cutting, scanning, and binding. The following is a description of each process.

Table 1. Book digitization workflow.

Processes	Equipment	Work descriptions
Cutting	Cutting machine	<ul style="list-style-type: none"> The paper was cut to a thickness that could be cut using a cutting machine. Cut the glued spine on the cutting machine. Separate the book into BPs so that they can be set in the book scanner.
Scanning	Book scanner	Place the BPs on the book scanner and press the scan start button.
Binding	Punch machine	<ul style="list-style-type: none"> Punch the BP with a punch machine. Bind the BPs together with binding bands.

Cutting: This is the process of cutting the spine of a book, where each page is glued to the spine using a cutting machine. In this paper, a piece of paper after cutting is called a book piece (BP). Because there is a limit to the thickness of a piece of paper that can be cut simultaneously, a book with many pages must be cut into pieces using a box cutter. The book scanner also has an upper limit on the

number of sheets of paper that can be set at one time, so the BPs divided for cutting need to be further divided. This process also includes removing the front and back covers from the book and finding and separating pages that still have glue on them from the BPs so that they will not interfere with the scanning process.

Scanning: This is the process in which BPs cut in Cutting are placed on the book scanner and scanned in page order. Paper jams may occur during scanning, but they occur stochastically, depending on the nature of the book paper and its conditions; this handling process is ignored.

Binding: This is the process of binding the BPs to restore them to readability as a paper book after all the BPs have been scanned. There are various methods for book binding, but here, we use a binding band as one of the simplest methods. Each BP was punched (two holes) with a punching machine, and the BPs were bound together with plastic binding bands.

2.2. Mathematical Model

This section defines a mathematical model for planning book digitization work. In this paper, we refer to this problem as the book digitization work planning problem (BDWPP). The smallest unit of work to process a book is called a Job. A Job has a start time and a work duration. Although there are three processes in book digitization work, it can be called a flow-shop type scheduling problem because the order of the processes to process books is the same for all books [28]. The objective of this study was to digitize books within the shortest possible time. This can be achieved by changing the order of the work within each process. Therefore, it is necessary to allow Jobs to overtake each other in each process. In addition, it is difficult to apply the formulation of the permutation flow storage problem to the scanning process because a Job must be processed multiple times [29]. Therefore, we focus on the fact that not all processes can be started without a worker and formulate the problem as a one-machine job shop scheduling problem, where the worker is considered a machine. Only one worker was included in this study. In addition, it is necessary to consider that the worker can perform other tasks because the process is automatic after the BPs are placed in the book scanner. Based on these constraints, we formulated the system as follows [30]:

Variables:

s_j Start time of Job j .

p_j Work duration of Job j .

q_j Work-inhibit duration after Job j is completed.

x_{jk} A variable that is set to 1 when Job j precedes another Job k and 0 otherwise.

a_j Process name of Job j .

Objective function:

$$\sum_{j=1}^n w_j s_j + \sum_{j=1}^n w_j p_j \quad (1)$$

Constraint conditions:

$$s_j + p_j + q_j - M(1 - x_{jk}) \leq s_k \quad (\forall j \neq k, a_j = a_k), \quad (2)$$

$$s_j + p_j - M(1 - x_{jk}) \leq s_k \quad (\forall j \neq k, a_j \neq a_k), \quad (3)$$

$$s_j - s_{j-1} \geq p_{j-1} + q_{j-1} \quad (\forall j, a_j \neq a_1), \quad (4)$$

$$x_{jk} + x_{kj} = 1 \quad (\forall j < k), \quad (5)$$

$$x_{jk} \in \{0, 1\} \quad (\forall j \neq k). \quad (6)$$

The objective function is the sum of the completion times of all Jobs. Equations (2) and (3) are the disjunctive constraints. Equation (2) is a constraint if the process names of Jobs j and k are the same, and Equation (3) is a constraint if the process names are different. This constraint formula considers the fact that there are two operations in the scanning process: placement of the BPs and automatic scanning by the book scanner. If Job j precedes Job k and they are in the same process, Job k can start after $p_j + q_j$ has elapsed since Job j starts. If the processes for Jobs j and k are different, only p_j is considered. Equation (4) represents the process-order constraint. This order constraint is applied when the process name of Job j is after the second process, that is, after the scanning process.

3. Preliminary Experiment

3.1. Method

To evaluate the work plan resulting from Gurobi Optimizer calculation, we first recorded the real Job start time and work duration when the subject processed freely selected books without consideration. The books were processed in the order in which they were cut during Cutting and in subsequent processes. **Table 2** and **Table 3** list the specifications of the books and equipment used in this experiment. The number of pages in both books differed from the original because Japanese translations were used. Note that the subject of this experiment was the author himself, and the number of trials was one.

Table 2. Specifications of books used in the experiment.

Title of books	Pages
Decompiling Java	156
Textbook for agile development	160
Five core metrics: the intelligence behind	182

Table 3. Specifications of machines used in the experiment.

Machines	Specifications
Cutter	Cutting capacity: 180 sheets of PPC paper
Book scanner	Scanning speed: 20 sheets/minute Number of sheets loaded: 50
Paper punch	Drilling capacity: 180 sheets of PPC paper

3.2. Result

The Gantt chart shows the start and elapsed times for each Job recorded during the experiment (Figure 1). In Section 2.1, there were three steps in the book digitization process. Let Scanning 1 be the manual process of setting BPs and Scanning 2 be the process of automatically scanning the BPs with a book scanner. Compared to the other processes, Scanning 2 took more time. In other words, Scanning 2 is the bottleneck in the entire process, and it can be predicted that increasing the utilization rate of Scanning 2 will contribute to improving the efficiency of the work plan.

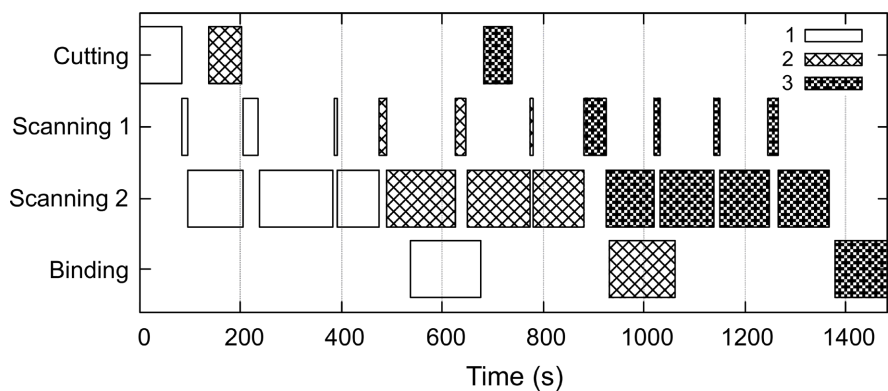


Figure 1. Work start time and work duration when BPs are processed in the order in which they are cut.

4. Evaluation for Optimizing Preliminary Experiment

4.1. Method

Table 4. Weight coefficient w for each experimental condition.

Experimental conditions	Weight coefficient w
1A	Set w of all processes to 1
1C	Set w of Cutting to 2 and the others to 1
1S	Set w of Scanning to 2 and the others to 1

The experiment aimed to verify whether work scheduling could be made more efficient. The work duration of each Job obtained from preliminary experiments was applied to the MIP solver. Gurobi Optimizer provided by Gurobi Optimization Inc. was used as the MIP solver under an academic license. Figure 1 shows that there are a total of 26 tasks in the preliminary experiment. These work durations are given to Gurobi Optimizer, which adjusts the work start time for each task to optimize the objective function. Simulation experiments were conducted under the three conditions listed in Table 4. w represents the weight coefficient in Equation (1). Because a larger value of w contributes to a larger value of the start time s_j of a Job in this process to increase Equation (1), Gurobi Optimizer

calculates the work duration p_j of a Job in this process to be smaller. The PC used for the computation was Intel Core i7 (1.7 GHz) with 8 GB of memory. This PC was used in subsequent experiments.

4.2. Results

The optimal solutions obtained by the simulation calculations under the three conditions are shown in **Figure 2** and **Figure 3**. The same optimal solutions were obtained under conditions 1A and 1S. The result of condition 1C was similar to the other conditions, except that the order of the Job started in Cutting. In both conditions, the order of starting the Job was different from that in the preliminary experiment.

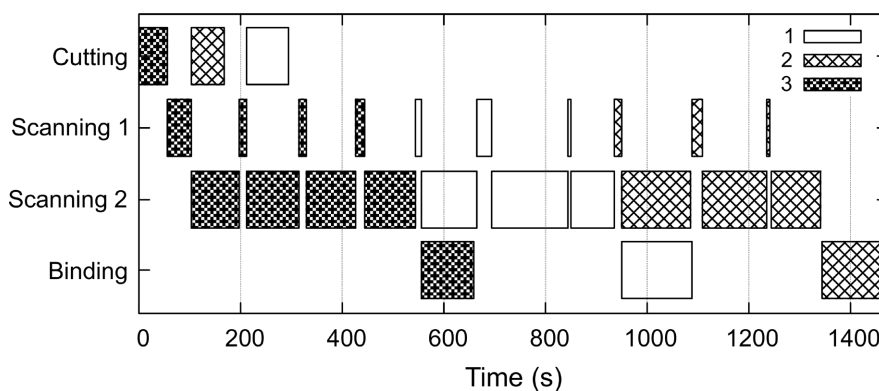


Figure 2. Optimal solution for conditions 1A and 1S.

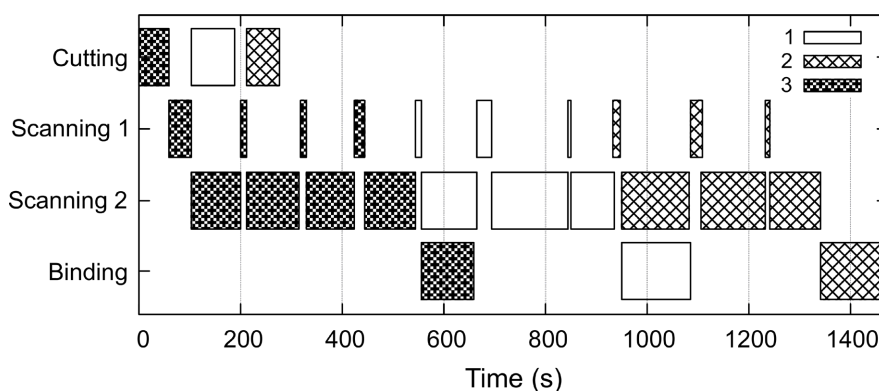


Figure 3. Optimal solution for condition 1C.

4.3. Data Analysis

The objective function and maximum completion time (makespan) of the optimal solution for each condition are listed in **Table 5**. The makespan is the end time at which the work is completed last among all tasks and is one of the typical schedule evaluation indices in scheduling problems. **Table 5** shows that the makespan of the optimal solution using Gurobi Optimizer was 13 s shorter than that in the preliminary experiment under all conditions. Although 13 s is only 1% of the makespan, it confirms that Gurobi Optimizer can improve the work plan. The

objective function value for condition 1A was 1046 s lower than that in the preliminary experiment. The results indicate that work on many Jobs completed ahead of the preliminary experiments. Note that the objective function values for conditions 1C and 1S cannot be compared because the weight coefficients are different from those in the preliminary experiment.

Table 5. Objective function and makespan of the optimal solution for each condition.

Experimental conditions	Objective function (s)	Makespan (s)
Preliminary experiment	11,261	1486
Condition 1A	10,215	1473
Condition 1C	10,734	1473
Condition 1S	16,694	1473

5. Analysis of BDWPP

5.1. Method

The results in Section 4 indicate that work planning using Gurobi Optimizer can improve the efficiency of the book digitization process. To further analyze BDWPP, we conducted simulation experiments by varying the number of books and BPs from 1 to 5. As in the experiments described in Section 4, the calculations were performed under the three conditions listed in **Table 6**. The work duration for each process is given by the average value obtained in the preliminary experiment. This implies that the work duration for all Jobs in the same process is the same. The upper limit of the computation time was set to 600 s because the computation time increased as the number of Jobs increased.

Table 6. Weight coefficient w for each experimental condition.

Experimental conditions	Weight coefficient w
2A	Set w of all processes to 1
2C	Set w of Cutting to 2 and the others to 1
2S	Set w of Scanning to 2 and the others to 1

5.2. Results

Simulation calculations were performed using Gurobi Optimizer under three conditions based on the number of books and BPs. The makespan for the optimal solutions obtained for each condition is listed in **Table 7**. The small brackets in the table indicate the solutions for which the search was aborted because the computation time of Gurobi Optimizer reached its upper limit. The red and underlined numbers indicate that different results were obtained under the three conditions. In the next section, we analyze the factors that led to the different results for optimal solutions (A) to (C) in **Table 7**.

Table 7. Makespans used in the experiment.

Book Count	BP Count	Makespan (s)		
		2A	2C	2S
1	1	326	326	326
1	2	458	458	458
1	3	590	590	590
1	4	722	722	722
1	5	854	854	854
2	1	474	474	474
2	2	738	738	738
2	3	1002	1002	1002
2	4	1266	1266	1266
2	5	1530	1530	1530
3	1	651	651	651
3	2	1018	1018	1018
3	3	1414	1414	1414
3	4	1810	1810	1810
3	5	2206	2206	2206
4	1	868	868	908
4	2	1298	1327	1298
4	3	1826	1826	1826
4	4	(2354)	(2354)	2354
4	5	(2882)	(2882)	2882
5	1	1085	1085	1085
5	2	(1607)	(1607)	1578
5	3	(2238)	(2267)	2238
5	4	(2898)	(2898)	2898
5	5	(3558)	(3558)	3558

5.3. Schedule Analysis

Case (A): In the condition with five books and two BPs in **Table 7**, the Job completion time of the optimal solution for condition 2S was earlier than that of the other conditions. Gantt charts for conditions 2A and 2S are shown in **Figure**

4 and Figure 5, respectively. Note that all work durations within the same process were the same; therefore, the difference in Job IDs had no meaning. Comparing the results from this perspective, there was no significant difference in work plan. Focusing on Scanning 2, the bottleneck process, we see that downtime (red arrow in Figure 4) is likely caused by the delayed start of Job 1, as condition 2A optimized the start time of the entire cut Job. On the other hand, Scanning 2 was prioritized under condition 2S, resulting in Job 1 being prioritized and downtime being reduced. This difference influenced the subsequent process, reducing the makespan for condition 2S. From this result, it is expected that a better work plan can be obtained by prioritizing Scanning.

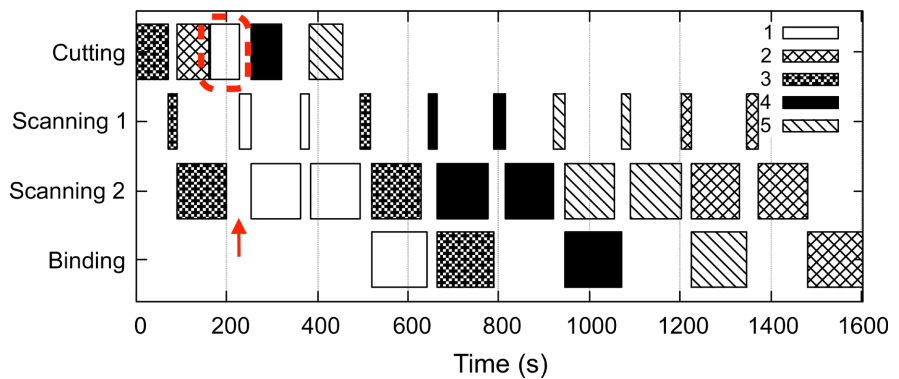


Figure 4. Optimal solution for condition 2A (number of books 5, number of BPs 2).

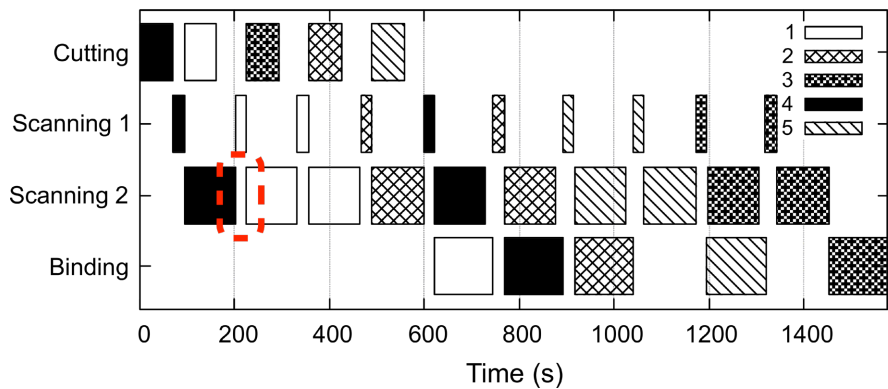


Figure 5. Optimal solution for condition 2S (number of books 5, number of BPs 2).

Case (B): In the condition with four books and one BP in Table 7, the makespan for condition 2S was longer than those for the other conditions, the opposite of case (A). Gantt charts for conditions 2A and 2S are shown in Figure 6 and Figure 7, respectively. In condition 2A, the book scanner was paused because Job 1 of Scanning 2 could not be started (red arrow in Figure 6) because Job 2 of Binding had priority (boxed area in Figure 6). On the other hand, in condition 2S, Job 4 was processed in Scanning 2 during this period (boxed area in Figure 7). This may have caused Jobs 4, 2, and 3 in the Binding to be unprocessed, thus increasing the maximum completion time. While pausing the bottleneck process (Scanning) in

condition 2A might seem inefficient, it can actually minimize the overall completion time, depending on the specific processing times.

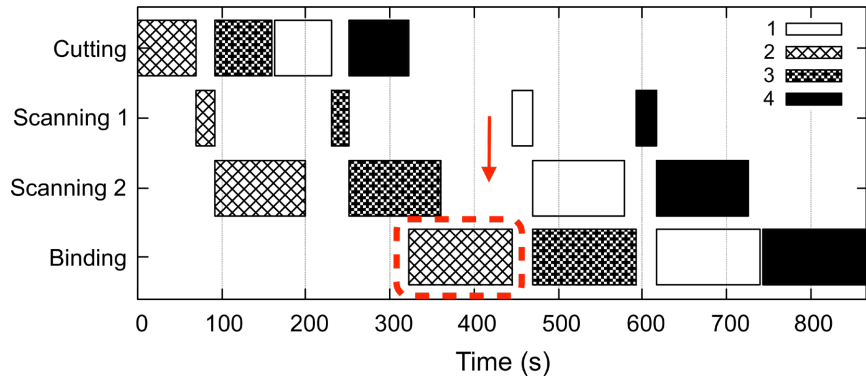


Figure 6. Optimal solution for condition 2A (number of books 4, number of BPs 1).

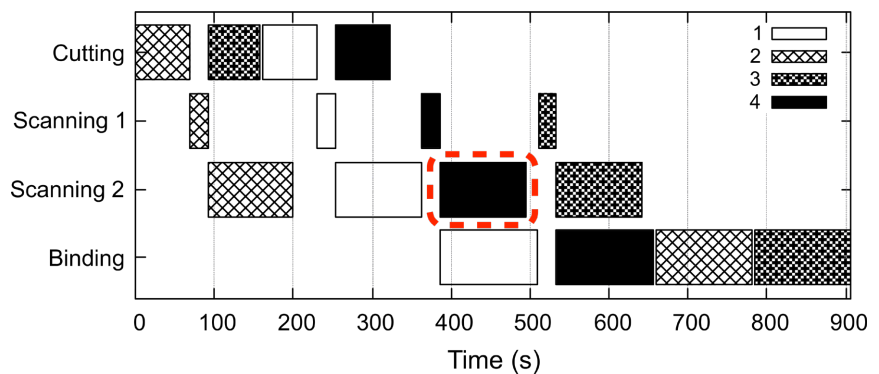


Figure 7. Optimal solution for condition 2S (number of books 4, number of book fragments 1).

Case (C): In the condition with four books and two BPs in Table 7, the makespan for condition 2C was longer than that for the other conditions. Gantt charts for conditions 2A and 2C are shown in Figure 8 and Figure 9, respectively. Because condition 2C prioritizes Cutting, Job 3 (boxed area in Figure 9) of Cutting was prioritized, and there was a pause in Scanning 2 (red arrow in Figure 9), which increased the makespan for Condition 2C.

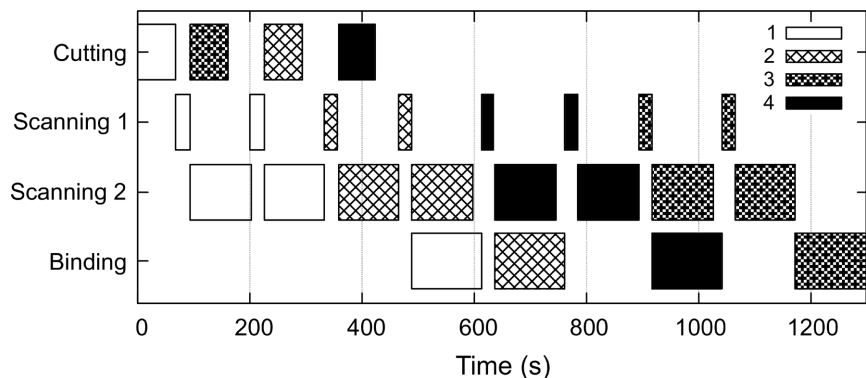


Figure 8. Optimal solution for condition 2A (number of books 4, number of BPs 1).

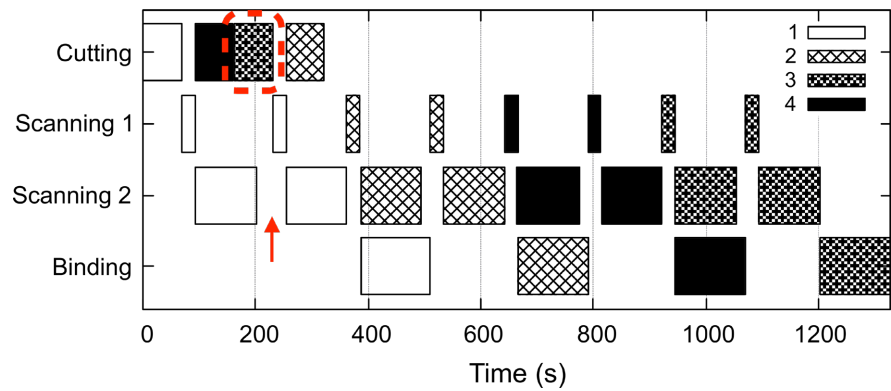


Figure 9. Optimal solution for condition 2C (number of books 4, number of BPs 1).

6. Limitations

The proposed model does not consider paper jams, variations in book conditions, or operator fatigue. Because the number of trials in the preliminary experiment is one, different results may be obtained if the number of trials is increased. In addition, the simulation experiments assume the same lead time for all processes; therefore, variations in the size of the BPs may change the calculation results. Despite these limitations of the results in this paper, we believe that the cost of model construction and maintenance should be more important than the proof of statistical robustness in general MIP studies, since the modeling targets are individual work plans. The purpose of this study is to disseminate the methods and possibilities of applying MIP to individual work.

7. Conclusions

In this paper, we investigate through a simulation experiment whether the work plan can be improved by planning the work for book digitization using a MIP solver, an integer programming Gurobi optimizer. The experiment demonstrated that the makespan could be reduced by 13 s by planning the work in advance. In addition, the results of the experiment with different numbers of books and BPs showed that prioritizing certain processes can reduce work time, but it can also increase work time.

In the future, we plan to develop a book digitization work plan scheduler based on the results of this study and to verify its effectiveness. In addition, we plan to revise the model to reduce the computation time, study the effects of paper jams and other problems, and learn how to deal with them.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Zhang, J., Liu, C., Li, X., Zhen, H., Yuan, M., Li, Y., *et al.* (2023) A Survey for Solving Mixed Integer Programming via Machine Learning. *Neurocomputing*, **519**, 205-217.

- <https://doi.org/10.1016/j.neucom.2022.11.024>
- [2] Bixby, R.E. (2012) A Brief History of Linear and Mixed-Integer Programming Computation. In: *Documenta Mathematica Series*, EMS Press, 107-121. <https://doi.org/10.4171/dms/6/16>
- [3] Bengio, Y., Lodi, A. and Prouvost, A. (2021) Machine Learning for Combinatorial Optimization: A Methodological Tour D'horizon. *European Journal of Operational Research*, 290, 405-421. <https://doi.org/10.1016/j.ejor.2020.07.063>
- [4] Nair, V., Bartunov, S., Gimeno, F., Von Glehn, I., Lichocki, P., Lobov, I. and Zwols, Y. (2020) Solving Mixed Integer Programs Using Neural Networks. <https://doi.org/10.48550/arXiv.2012.13349>
- [5] Solnon, C., Cung, V.D., Nguyen, A. and Artigues, C. (2008) The Car Sequencing Problem: Overview of State-of-the-Art Methods and Industrial Case-Study of the ROADEF'2005 Challenge Problem. *European Journal of Operational Research*, 191, 912-927. <https://doi.org/10.1016/j.ejor.2007.04.033>
- [6] Morales-Espana, G., Latorre, J.M. and Ramos, A. (2013) Tight and Compact MILP Formulation for the Thermal Unit Commitment Problem. *IEEE Transactions on Power Systems*, 28, 4897-4908. <https://doi.org/10.1109/tpwrs.2013.2251373>
- [7] Koch, T., Berthold, T., Pedersen, J. and Vanaret, C. (2022) Progress in Mathematical Programming Solvers from 2001 to 2020. *EURO Journal on Computational Optimization*, 10, Article 100031. <https://doi.org/10.1016/j.ejco.2022.100031>
- [8] González, M., López-Espín, J.J. and Aparicio, J. (2020) A Parallel Algorithm for Matheuristics: A Comparison of Optimization Solvers. *Electronics*, 9, Article 1541. <https://doi.org/10.3390/electronics9091541>
- [9] Turner, M., Chmiela, A., Koch, T. and Winkler, M. (2023) PySCIPOpt-ML: Embedding Trained Machine Learning Models into Mixed-Integer Programs. <https://doi.org/10.48550/arXiv.2312.08074>
- [10] Ahmadi Teshnizi, A., Gao, W. and Udell, M. (2023) Optimus: Optimization Modeling Using MIP Solvers and Large Language Models.
- [11] Wang, T., Yu, W.Y., She, R., Yang, W., Chen, T. and Zhang, J. (2024) Leveraging Large Language Models for Solving Rare MIP Challenges. <https://doi.org/10.48550/arXiv.2409.04464>
- [12] Brech, C., Ernst, A. and Kolisch, R. (2019) Scheduling Medical Residents' Training at University Hospitals. *European Journal of Operational Research*, 274, 253-266. <https://doi.org/10.1016/j.ejor.2018.04.003>
- [13] Vermuyten, H., Namorado Rosa, J., Marques, I., Beliën, J. and Barbosa-Póvoa, A. (2018) Integrated Staff Scheduling at a Medical Emergency Service: An Optimisation Approach. *Expert Systems with Applications*, 112, 62-76. <https://doi.org/10.1016/j.eswa.2018.06.017>
- [14] Luo, L., Liu, X., Cui, X., Cheng, Y., Yu, X., Li, Y., *et al.* (2020) Applying Queuing Theory and Mixed Integer Programming to Blood Center Nursing Schedules of a Large Hospital in China. *Computational and Mathematical Methods in Medicine*, 2020, 1-7. <https://doi.org/10.1155/2020/9373942>
- [15] Hewitt, M., Chacosky, A., Grasman, S.E. and Thomas, B.W. (2015) Integer Programming Techniques for Solving Non-Linear Workforce Planning Models with Learning. *European Journal of Operational Research*, 242, 942-950. <https://doi.org/10.1016/j.ejor.2014.10.060>
- [16] Wang, H., Alidaee, B., Ortiz, J. and Wang, W. (2020) The Multi-Skilled Multi-Period Workforce Assignment Problem. *International Journal of Production Research*, 59,

- 5477-5494. <https://doi.org/10.1080/00207543.2020.1783009>
- [17] Vahedi-Nouri, B., Tavakkoli-Moghaddam, R., Hanzálek, Z. and Dolgui, A. (2022) Workforce Planning and Production Scheduling in a Reconfigurable Manufacturing System Facing the COVID-19 Pandemic. *Journal of Manufacturing Systems*, **63**, 563-574. <https://doi.org/10.1016/j.jmsy.2022.04.018>
- [18] Köseli, İ., Soysal, M., Çimen, M. and Sel, Ç. (2023) Optimizing Food Logistics through a Stochastic Inventory Routing Problem under Energy, Waste and Workforce Concerns. *Journal of Cleaner Production*, **389**, Article 136094. <https://doi.org/10.1016/j.jclepro.2023.136094>
- [19] Laesanklang, W., Landa-Silva, D. and Arturo Castillo Salazar, J. (2015) Mixed Integer Programming with Decomposition to Solve a Workforce Scheduling and Routing Problem. *Proceedings of the International Conference on Operations Research and Enterprise Systems*, Lisbon, 10-12 January 2015, 283-293. <https://doi.org/10.5220/0005223602830293>
- [20] Moussavi, S.E., Mahdjoub, M. and Grunder, O. (2019) A Matheuristic Approach to the Integration of Worker Assignment and Vehicle Routing Problems: Application to Home Healthcare Scheduling. *Expert Systems with Applications*, **125**, 317-332. <https://doi.org/10.1016/j.eswa.2019.02.009>
- [21] IDC (2024) Tablet Shipments Show Signs of Recovery in Q1 2024. <https://www.idc.com/getdoc.jsp?containerId=prUS52105224>
- [22] Nagaraj, A. and Reimers, I. (2023) Digitization and the Market for Physical Works: Evidence from the Google Books Project. *American Economic Journal: Economic Policy*, **15**, 428-458. <https://doi.org/10.1257/pol.20210702>
- [23] Library Vision (2022) How to Scan Old Books: Digitizing Used Books into a PDF. <https://www.libraryvision.org/digitization/how-to-scan-old-books/>
- [24] Fortune Business Insights (2024) Document Scanner Market Size, Share Industry Analysis 2032. <https://www.fortunebusinessinsights.com/document-scanner-market-106606>
- [25] The Business Research Company (2024) Scanner Market Report 2024—Scanner Market Analysis and Forecast 2033. <https://www.thebusinessresearchcompany.com/report/scanner-global-market-report>
- [26] Takahashi, K. (2014) An Analysis on Work Efficiency of Personal Book Scanning Process with Integer Programming Solver. *Scheduling Symposium 2014*, Toyama, 29-30 September 2014, 171-176.
- [27] Gurobi Optimization (2024) Gurobi Optimizer: The World's Fastest Solver. <https://www.gurobi.com/solutions/gurobi-optimizer/>
- [28] Johnson, S.M. (1954) Optimal Two and Three-Stage Production Schedules with Setup Times Included. *Naval Research Logistics Quarterly*, **1**, 61-68. <https://doi.org/10.1002/nav.3800010110>
- [29] Potts, C.N. and Van Wassenhove, L.N. (1982) A Decomposition Algorithm for the Single Machine Total Tardiness Problem. *Operations Research Letters*, **1**, 177-181. [https://doi.org/10.1016/0167-6377\(82\)90035-9](https://doi.org/10.1016/0167-6377(82)90035-9)
- [30] Kubo, M., Pedro, J., Muramatsu, M. and Rais, A. (2012) Mathematical Optimization: Solving Problems Using Gurobi and Python. Kindai Kagaku Sha.