Optimizing Café Operations: Applying Mixed Integer Optimization Techniques to An Extended Job-Shop Scheduling Problem

Dongming Shen (dshen999@mit.edu)
Yongchan (Yolanda) Wang (ycwang17@mit.edu)

15.093/6.7201 Optimization

Prof. Alexandre (Alex) Jacquillat
Prof. Dimitris Bertsimas

Fall 2023

1 Problem Description

The Job-Shop Scheduling Problem (JSSP) involves assigning a set of jobs to a set of machines, where each job comprises a series of operations that need to be processed by different machines in a specific order. This problem is prevalent in manufacturing and logistics, showcasing its crucial role in optimizing resource allocation and minimizing completion times.

JSSP contains inherent complexity, making it an NP-hard problem, indicating that the time required to solve instances of the problem increases exponentially with the size of the input. This complexity stems from the combinatorial nature of the scheduling decisions and the constraints on operation order and machine assignments.

In the context of a coffee shop, it requires organizing a variety of drink and food orders, each with unique preparation steps, across several machines, equipment, workstations, and human workers. Traditional processing methods, typically First-in-First-out (FIFO), can be straightforward but suboptimal under high demand during rush hours. To enhance resource efficiency and service speed, we propose utilizing a Mixed Integer Optimization (MIO) framework to optimize coffee shop operational strategies.

2 Data

We generate synthesized data to simulate a coffee shop's operations. This includes a series of hypothetical customer orders and a set of machines (equipment, stations, and human workers) typically found in a coffee shop environment. The data includes common orders, such as various coffee drinks and food items, each with its required sequence of operations, required machines, and estimated preparation times.

We create the data in CSV format for extensibility. The dataset contains information about the jobs, operations, machine assignments, processing times, and any special constraints that need to be addressed. This format ensures that the dataset can be easily parsed and therefore can be readily extended or modified to test various aspects of the model, providing a robust means of validation.

In our study, we have developed two distinct datasets, Dataset1 and Dataset2, each was created based on different assumptions about machine assignments in a coffee shop setting.

2.1 Dataset1 (Single Machine Assignment Assumption)

Dataset1 adheres to a traditional assumption often found in standard JSSP models. It assumes that each job (e.g., an order for a coffee or a food item) requires the use of exactly one machine, and each machine is used no more than once for any given job. This dataset offers a simplified view of café operations, which is a good starting point, but may not fully capture the complexity of real-world café tasks. Below is an example table of Dataset1 with 4 jobs.

Job ID	Job Meaning	Group ID	Op ID	Op Meaning	Machine ID	Machine Meaning	Time (s)
1	Latte	1	1	Grinding	1	Grinder	30
		2	2	Espresso	2	Coffee Machine	30
		2	3	Hot Milk	4	Heater	60
		3	4	Mixing	6 / 7	Any Staff	20 / 10
2	Cappuccino	1	1	Grinding	1	Grinder	30
		2	2	Espresso	2	Coffee Machine	30
		2	3	Hot Milk	4	Heater	60
		3	4	Milk Frothing	5	Frother	20
		4	5	Mixing Latte Art	6 / 7	Any Staff	40 / 20
3	Americano	1	1	Grinding	1	Grinder	30
		2	2	Espresso	2	Coffee Machine	30
		2	3	Hot Water	4	Heater	60
		3	4	Mixing	6 or 7	Any Staff	20 / 10
4	Mocha	1	1	Grinding	1	Grinder	30
		2	2	Mocha Pot	3	Mocha Pot	90
		3	3	Submitting	6 / 7	Any Staff	10 / 5

Table 1: Example Dataset1 with 4 Jobs

2.2 Dataset2 (Multiple Machine Assignment Flexibility)

Recognizing the limitations of Dataset1's assumptions, we create Dataset2 which relaxes this constraint. In this dataset, we allow for a more flexible job-machine relationship, where a single job can require multiple machines or the same machine can be used multiple times for different steps of a job. This dataset is designed to more accurately reflect the diverse and often complex nature of tasks in a café, such as preparing multi-step beverages or food items that require several types of equipment multiple times. Below is an example table of Dataset2 with 4 jobs. Notice that the main difference with the Dataset1 example above is the ability to reuse machines, which also allows a machine to be able to perform multiple tasks - a more realistic assumption in our use case.

Job ID	Job Meaning	Group ID	Op ID	Op Meaning	Machine ID	Machine Meaning	Time (s)
1	Latte	1	1	Grinding	1	Grinder	30
		2	2	Espresso	2	Coffee Machine - Espresso	30
		2	3	Hot Milk	3	Heater	60
		3	4	Mixing	4 / 5	Any Staff	20 / 10
		4	5	Submitting	4 / 5	Any Staff	5
2	Cappuccino	1	1	Grinding	1	Grinder	30
		2	2	Espresso	2	Coffee Machine - Espresso	30
		2	3	Hot Milk	3	Heater	60
		3	4	Milk Frothing	2	Coffee Machine - Frothing	20
		4	5	Mixing Latte Art	4 / 5	Any Staff	40 / 20
		5	6	Submitting	4 / 5	Any Staff	5
3	Americano	1	1	Grinding	1	Grinder	30
		2	2	Espresso	2	Coffee Machine - Espresso	30
		2	3	Hot Water	3	Heater	60
		3	4	Mixing	4 / 5	Any Staff	20 / 10
		4	5	Submitting	4 / 5	Any Staff	5
4	Hand Brew	1	1	Grinding	1	Grinder	30
		1	2	Preparing Hand Brew Pot	5	Advanced Staff	30
		2	3	Hand Brewing	5	Advanced Staff	100
		3	4	Submitting	4 / 5	Any Staff	5

Table 2: Example Dataset2 with 4 Jobs

By contrasting these two datasets, our analysis aims to explore the impact of different machine assignment assumptions on the optimization of café operations, providing insights into the practical applicability of our MIO model in various real-world scenarios.

3 Methods

MIO is a suitable technique to formulate JSSP as it is capable of capturing both the discrete elements (like the assignment of specific operations to various machines) and the continuous elements (such as the precise timing of each operation). The flexibility of MIO positions it as a powerful method for deriving solutions that are either optimal or very close to optimal. Our approach employs an MIO formulation of JSSP, drawing inspiration from Harvey M. Wagner's Rank-based approach. This comprehensive formulation is tailored to effectively simulate and optimize the unique scheduling challenges present in dynamic coffee shop environments.

3.1 Formulation1: MIO Formulation for JSSP-Standard

Our initial model, Standard Formulation, is designed to address Dataset1, following the basic principles of job-machine assignment. This formulation serves as our foundational method, establishing a starting point for our optimization problem.

3.1.1 Terms, Variables, and Parameters

- Let J_i denote job i.
- Let M_k denote machine k.
- Let MG_g^i be the set of machine indices that G_g^i might use.

- Let S_i be the set of machine indices that J_i might use.
- X_{ij}^k: Binary variable, 1 if job J_i is on the order position j on machine M_k.
- h_j^k: Start time of order position j on machine M_k.
- s_j^k: Time elapsed between order positions j and $j+1$ on machine M_k.
- s_0^k: Idle time from start to the first order position on machine M_k.
- $n(k)$: Maximum number of order positions machine M_k can process.
- $N(k)$: Set of jobs J_i that can use machine M_k.
- TX_j^k: Process time of the j-th order position task on machine M_k.
- t_i^k: Processing time required for job J_i on machine M_k.
- $c_i^{S_i}$: Total number of times job J_i will use any machine(s) in set S_i.

3.1.2 Objective Function

Minimize the maximum completion time (makespan):

$$\min \quad h^*$$

3.1.3 Constraints

Each job is completed by its required machine(s) the required number of times:

$$\sum_{k \in S_i} \sum_{j=1}^{n(k)} X_{ij}^k = c_i^{S_i}, \quad \forall i$$

At most one operation at each order position on each machine:

$$\sum_{i \in N(k)} X_{ij}^k \leq 1, \quad \forall j \in [n(k)], \forall k$$

Define the processing time for each order position on each machine:

$$TX_j^k = \sum_{i \in N(k)} t_i^k X_{ij}^k, \quad \forall j \in [n(k)], \forall k$$

Ensure precedence constraints for each job (each group must be complete in the given group order):

$$h_{j'}^a + t_i^a X_{ij'}^a \leq h_{j''}^b + M(1 - X_{ij'}^a) + M(1 - X_{ij''}^b),$$
$$\forall a \in MG_g^i, \forall b \in MG_{g+1}^i, \forall j' \in [n(a)], \forall j'' \in [n(b)], \forall i$$

Define the makespan (maximum completion time):

$$h^* \geq h_{n(k)}^k + TX_{n(k)}^k, \quad \forall k$$

We estimated the worst-case number of binary variables and constraints for Formulation 1. Given n jobs, each with at most k operations, and m machines, there will be at most $O(mkn^2)$ variables and $O(m^2n^3k^2)$ constraints. This bound will be very loose in reality.

3.2 Formulation2: MIO Formulation for JSSP-Extended

Recognizing the need to accommodate the more complex and realistic scenarios depicted in Dataset2, we upgraded Formulation1 into Formulation2.

3.2.1 Terms, Variables, and Parameters

- Let J_i denote job i.
- Let P_l^i denote operation l of job i.
- Let G_g^i denote operation indices in group g of job i.
- Let M_k denote machine k.
- Let MG_g^i be the set of machine indices that G_g^i might use.
- Let S_l^i be the set of machine indices that P_l^i might use.
- X_{ilj}^k: Binary variable, 1 if operation P_l^i is on the order position j on machine M_k.
- h_j^k: Start time of order position j on machine M_k.
- s_j^k: Time elapsed between order positions j and $j+1$ on machine M_k.
- s_0^k: Idle time from start to the first order position on machine M_k.
- $n(k)$: Maximum number of order positions machine M_k can process.
- $N(k)$: Set of job-operation index pairs (i, l) that can use machine M_k.
- TX_j^k: Process time of the j-th order position task on machine M_k.
- t_{il}^k: Processing time required for operation P_l^i on machine M_k.
- $c_{il}^{S_l^i}$: Total number of times operation P_l^i will use any machine(s) in set S_l^i.

3.2.2 Objective Function

Minimize the maximum completion time (makespan):

$$\min \quad h^*$$

3.2.3 Constraints

Each job-operation is completed by its required machine(s) the required number of times:

$$\sum_{k \in S_l^i} \sum_{j=1}^{n(k)} X_{ilj}^k = c_{il}^{S_l^i}, \quad \forall (i,l)$$

At most one operation at each order position on each machine:

$$\sum_{(i,l) \in N(k)} X_{ilj}^k \leq 1, \quad \forall j \in [n(k)], \forall k$$

Define processing time for each order position on each machine:

$$TX_j^k = \sum_{(i,l) \in N(k)} t_{il}^k X_{ilj}^k, \quad \forall j \in [n(k)], \forall k$$

Ensure precedence constraints for each job (each group must be complete in the given group order):

$$h_{j'}^a + t_{il'}^a X_{il'j'}^a \leq h_{j''}^b + M(1 - X_{il'j'}^a) + M(1 - X_{il''j''}^b),$$
$$\forall a \in MG_g^i, \forall b \in MG_{g+1}^i, \forall P_{l'}^i \in G_g^i, \forall P_{l''}^i \in G_{g+1}^i, \forall j' \in [n(a)], \forall j'' \in [n(b)], \forall i$$

Define makespan (maximum completion time):

$$h^* \geq h^k_{n(k)} + TX^k_{n(k)}, \quad \forall k$$

[Distributed Task]

- Let DM^i_l denote the set of machine pairs that must be used together by operation l of job i.

Ensure each job-operation is completed by its required group of machines (if applicable):

$$\begin{aligned}
&X^a_{ilj'} \geq Z^{ab}_{ilj'j''}, \quad X^b_{ilj''} \geq Z^{ab}_{ilj'j''}, \quad Z^{ab}_{ilj'j''} \geq (X^a_{ilj'} + X^b_{ilj''} - 1), \\
&-M(Z^{ab}_{ilj'j''} - 1) \leq (h^a_{j'} - h^b_{j''}) \leq M(Z^{ab}_{ilj'j''} - 1), \\
&-M(Z^{ab}_{ilj'j''} - 1) \leq (h^a_{j'} + t^a_{il}X^a_{ilj'} - h^b_{j''} - t^a_{il}X^a_{ilj''}) \leq M(Z^{ab}_{ilj'j''} - 1), \\
&\forall (i,l), \forall (a,b) \in DM^i_l, \forall j' \in [n(a)], \forall j'' \in [n(b)]
\end{aligned}$$

This enhanced formulation introduces an additional dimension to our decision variables, extending beyond the basic job-machine assignment to encompass job-operation-machine assignments. The expansion of these variables is necessary to capture the complicated relationships and multiple interactions typical in a coffee shop setting. As a result, Formulation2 is significantly more powerful than Formulation1, despite the fact that it introduces an increased number of variables. This complexity, however, is a compromise to its enhanced flexibility and generalizability, but Formulation2 enables our optimization solution to reflect real-world café operations more effectively and accurately. The results section will further discuss the strengths and practical implications of this advanced formulation.

We estimated the worst-case number of binary variables and constraints for Formulation 2. Given n jobs, each with at most k operations, and m machines, there will be at most $O(m(nk)^2)$ variables and $O(m^2n^3k^4)$ constraints. As before, in reality, this bound will be very loose, but this gives an idea that the scalability of the problem is not great, and is indeed more complex than Formulation 1.

3.3 Models Implementation and Testing

We implemented both formulations in Julia, and we will test the formulations on Dataset1 and Dataset2, respectively using JuMP and Gurobi. Specifically, for Dataset1 we randomly generated task combinations from 5 available tasks to create 11 datasets ranging from 4 jobs to 24 jobs (step size = 2). For Dataset2 we randomly generated task combinations from 5 available tasks to create 9 datasets ranging from 4 jobs to 12 jobs (step size = 1). Considering that in reality, normal coffee shop scheduling should be nearly instant, we set the termination time for all problems to 10 seconds.

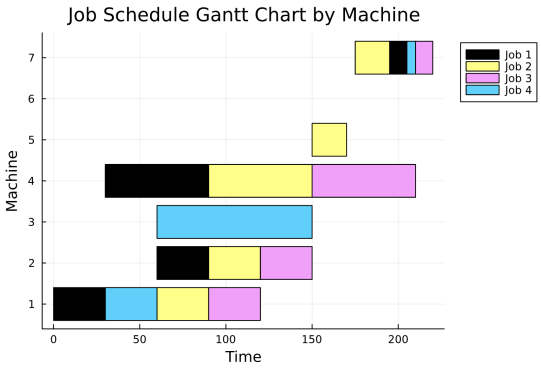
We expect our method applying the MIO formulation to outperform the traditional FIFO and algorithmic-based methods, especially during peak hours. Our results will be evaluated against these conventional approaches, with a focus on minimizing the makespan. We aim to provide visualizations of the job schedules for each machine to illustrate the feasibility and efficiency of the optimized schedules.

4 Results & Key Findings

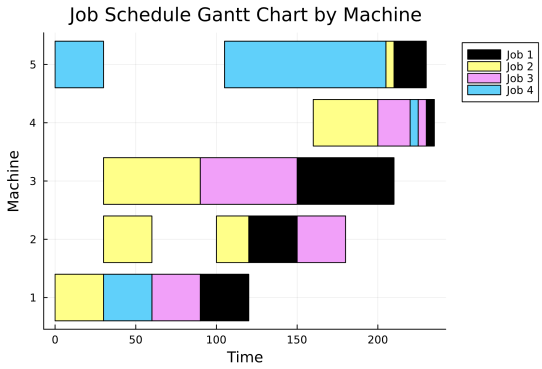
Our model surpasses the baseline FIFO approach, demonstrating superior performance even with a time constraint as short as 10 seconds. This highlights the efficiency of our MIO approach in managing complex scheduling tasks.

Figure 1 illustrates the job schedules optimized using both formulations. The Gantt Chart representation allows for an intuitive understanding of the scheduling dynamics. On the x-axis, each task's duration is depicted by the length of its corresponding bar, while the y-axis represents the machines utilized for each job.

Formulation1: A critical observation from the chart is the feasibility and optimality of the problem's solution. For instance, in Formulation1 as shown in Figure 1(a), Job 2 commences with Machine 1, followed by two subsequent operations on different machines that initiate concurrently. This simultaneity is achievable because the operations are categorized within the same group, highlighting the formulation's adherence to group constraints. In scenarios where operations are not in the same group, the chart distinctly shows no overlapping in operation times. This enforces the stipulation that simultaneous



(a) Job Schedule Gantt Chart under Formulation1



(b) Job Schedule Gantt Chart under Formulation2

Figure 1: Job Schedule Gantt Charts for 4 jobs under 2 Different Formulations

operations within a job must comply with predefined group constraints. Operations within the same group possess flexibility in execution order. Operations belonging to different groups must adhere strictly to the prescribed group order.

Formulation2: In contrast, Formulation2, as depicted in Figure 1(b), introduces greater flexibility while ensuring similar constraints at the same time, particularly in machine reuse. A notable instance is Job 2 utilizing Machine 2 for two distinct operations, one in the second group and another in the third group. This enhanced flexibility under Formulation2 is evident in the job schedules, demonstrating the model's capacity to accommodate more complex machine allocation strategies.

Each job schedule, upon verification of constraint satisfaction, demonstrates the optimized utilization of resources, highlighting interdependencies between machines for different jobs and enabling effective tracking of job progress over time.

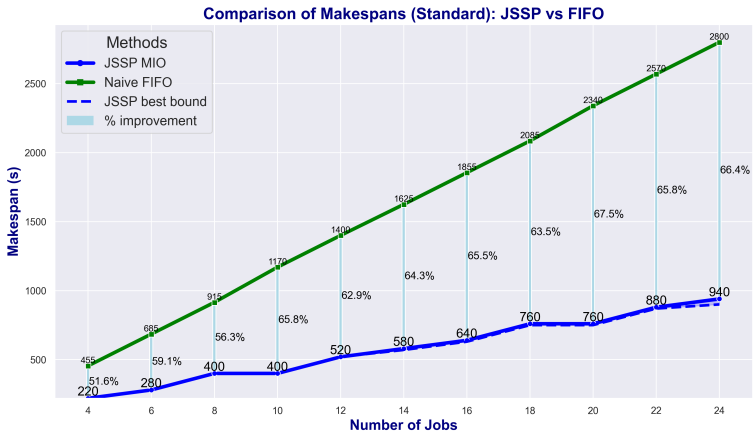


Figure 2: Makespans Comparison: JSSP-Standard v.s. FIFO

In addition, Figure 2 and Figure 3 demonstrate the effectiveness of MIO-formulated JSSP over the traditional FIFO approach. The MIO formulations, both Standard (Formulation1) and Extended (Formulation2), consistently show a significant reduction in the makespan when compared to the FIFO baseline, underscoring the benefits of advanced scheduling techniques in operational environments.

Reduction in Makespan: The MIO formulations are observed to reduce the FIFO baseline makespan by over 50% across varying numbers of jobs. This substantial decrease in makespan illustrates the efficiency and capability of MIO in optimizing job scheduling, in contrast to FIFO's sequential and less flexible approach. The ability of MIO to evaluate multiple scheduling options simultaneously allows for a more effective allocation of resources, thereby reducing idle times and increasing overall throughput.

Benefits Amplified with Increased Job Numbers: As the number of jobs increases, the advantages of MIO over FIFO become increasingly apparent. With a higher volume of jobs, FIFO's simplistic and

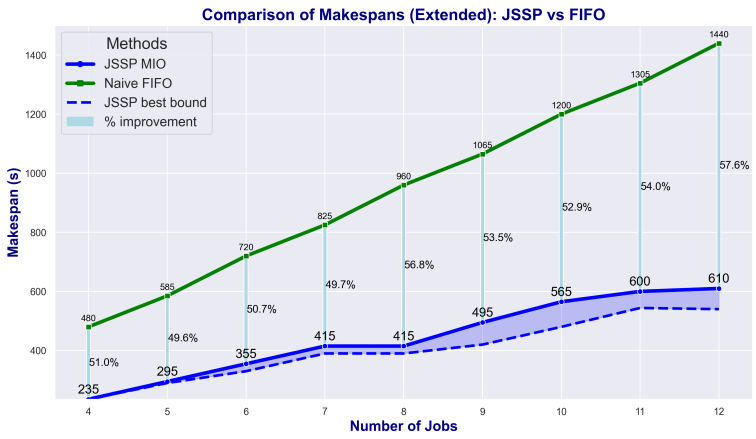


Figure 3: Makespans Comparison: JSSP-Extended v.s. FIFO

linear scheduling becomes less effective, leading to suboptimal machine usage and longer waiting times. In contrast, MIO's dynamic job allocation, which takes into account various constraints and factors, leads to a more efficient utilization of machines. This trend is especially notable in complex scheduling scenarios where the decision space is significantly larger. In reality, when there are more constraints such as setup time and clean-up time, this gap is expected to increase even more with the increasing number of jobs.

Optimality Gap in Complex Scenarios: In the Extended MIO Formulation (Formulation2), an increase in the optimality gap is observed. This is due to the limited solution time of 10 seconds, a constraint that becomes more pronounced as the complexity of the scheduling task increases. The optimality gap, indicating the difference between the best-found feasible solution and the lower bound of the objective, highlights a critical trade-off between solution quality and computational time, particularly in complex scheduling environments. The results indicate that while the MIO approach significantly improves scheduling efficiency, it is also subject to computational limitations, especially under strict time constraints. This observation is crucial for industries where scheduling efficiency directly impacts operational success, suggesting that investments in optimization techniques like MIO can lead to significant improvements, but they need to consider computational resources and more advanced solving techniques.

By conducting a detailed comparison of our JSSP model against the FIFO baseline in various task scenarios, we examine the robustness and adaptability of our approach. Our method also shows a high potential for generalizability, simulating real-world complexities more accurately through the incorporation of additional constraints. In the next section, we include some potential extensions that can be added to our formulation and their corresponding constraints.

5 Extensions on Flexible and Realistic Constraints

Machine Flexibility & Order Flexibility: Our model already extends the traditional JSSP by allowing for greater flexibility in machine assignments and order processing, addressing the dynamic nature of café operations. Specifically, a job can take a machine any number of times, and we can enforce strict sequencing between groups while relaxing sequencing and allowing parallelism within a group.

Distributed Tasks: We introduce scenarios where operations must be processed simultaneously by multiple machines. For example, [operation1, operation2] of Job1 needs to be completed by machine1 and machine2 at the same time. This extension reflects the collaborative aspects of certain tasks in a real café environment. We included the constraints in Formulation2 to indicate the formulation generalizability. Formulation1 can be generalized similarly.

Human Global Coordinator: Formulation2 can be easily modified to incorporate a "human global coordinator" function, which functions as an additional machine that intervenes either in between or alongside activities. This element represents the human oversight necessary in a café setting. This

extension cannot be easily applied to Formulation1.

Fairness in Task Execution: Our model allows easy generalization to include fairness considerations, such as penalizing out-of-order delivery or introducing additional constraints to maintain a fair and efficient service flow. For example, we can make sure that order 1's completion time does not exceed order 2's completion time, and that the difference between the two completion times remains near constant.

Special Requests Handling: To accommodate special requests, such as prioritizing completing order1, our model is designed to easily integrate these as additional constraints, demonstrating adaptability to customer needs.

Managing Setup and Idle Times: Considering the importance of maintenance and machine readiness, Formulation2 can be easily generalized to consider setup and idle times, such as requiring cleaning after every three iterations, ensuring continual operational efficiency.

6 Impacts of the Study

Time Savings During Rush Hours: One of the most significant impacts of applying our MIO formulations to coffee shop operations is the considerable time savings during peak or rush hours. By optimizing job-operation-machine assignments, our model efficiently manages the influx of orders, reducing wait times and enhancing customer satisfaction.

Sensitivity Analysis for Equipment Investment: Another key impact of our study is its utility in guiding investment decisions, particularly in determining which additional machines or equipment would bring the most value to a café. By running the model multiple times, each time with a new machine, our model can predict the effect of adding or upgrading different machines on overall operational efficiency. This analysis is helpful for café owners who are considering investments in new equipment, as it helps prioritize purchases that will have the greatest impact on reducing bottlenecks and improving service speed.

The application of MIO in a coffee shop setting can offer valuable insights into the balance between operational efficiency and customer experience. We can assess different trade-offs, such as the tension between minimizing makespan and maintaining fairness. This project could serve as an investigation into the viability of MIO for small-scale business improvements, with the potential of extending to a broader range of service-oriented industries. With real-time scheduling optimization, businesses can adapt to sudden changes in demand, manage staffing more effectively, and ultimately increase profitability.

7 Future Directions

Implementing Cutting Plane Methods for Speedup: A promising area for future research is the application of cutting plane methods to our MIO formulations. Currently, the complexity of our models, especially Formulation2 with the expanded, 4-dimensional, decision variables, leads to a huge increase in constraints, which can slow down runtime significantly. Cutting plane methods offer a potential solution to this issue by iteratively adding only the most relevant constraints during the optimization process. This approach can drastically reduce the number of constraints needed, thereby speeding up the solving process without compromising the accuracy or effectiveness of the results.

Exploring Additional Optimization Techniques: Beyond cutting planes, further exploration into other advanced optimization techniques could yield even more efficient solutions. Techniques such as branch-and-cut algorithms, heuristic methods, or even the integration of machine learning algorithms could offer new ways to handle the complexity of the scheduling problems in café settings. These methods might balance computational efficiency and solution accuracy, especially for highly dynamic and complex scenarios.

8 Conclusion

In conclusion, we explored optimizing café operations using Mixed Integer Optimization (MIO) applied to an extended Job-Shop Scheduling Problem (JSSP) formulation. Our study introduced two datasets

- Dataset1 with strong assumptions for simplification and Dataset2 allowing for more complex, realistic scenarios. We developed two formulations: Formulation1 and Formulation2 for the above datasets. Both formulations aimed to minimize the maximum completion time (makespan) for tasks in a café setting, under different assumptions.

Our findings showed that these MIO formulations significantly improved efficiency over traditional First-in-First-out (FIFO) methods, especially noticeable as the number of jobs increased. However, the complexity led to a trade-off between solution quality and computational time under tight constraints. The project also highlighted potential model extensions to better mimic real-world café operations and suggested future directions, including cutting plane methods and exploring new optimization techniques. Overall, this project illustrates the effectiveness of MIO in enhancing operational efficiency in cafés and potentially benefiting a wider range of service industries.