Dual-Objective MILP Model for Labor-Constrained Production Scheduling Optimization

Group Members:

Sandip Pokhrel

Esosa Simeon Omwanghe

Aramide Asabi Bankole-Balogun

Sannibhai Indradeepbhai Garasiya

Master of Data Analytics, University of Niagara Falls

DAMO-610-6: Operations Analytics

Professor:

Cosimo Girolamo

Due Date: June 23, 2025

Abstract

This project creates and enacts a two-objective Mixed Integer Linear Programming (MILP) model which optimizes production scheduling for a labour-constrained manufacturing environment. The objective of the model is to minimize production makespan as well as total labour cost that includes regular and overtime labour cost.

The model integrates workforce availability and wage structure in the optimization framework, providing a missing real-world dimension of most classical formulations in production planning. The extension of this model is based on the work of Farizal et al. (2021) who minimises the makespan and number of machines used but ignores labour cost and labour time.

Our extension imposes a monthly restriction (500 hours, 50 per machine) on regular labor and overtime labor. Moreover, there is a wage differential between regular (CAD $15/hour) and overtime labor (CAD $22.5/hour). This system was implemented using Python and Google OR-Tools and was solved using the SCIP Solver. It consists of six FMCG products and four machines having different processing speeds. The schedule made it possible in better utilization of the machines and the cost-effective deployment of labour to comply with work-hour regulations. This indicates the model which has a practical worth for production operations which is labour-sensitive.

1. Introduction

Production scheduling is the most important operational analytics problem affecting productivity, lead time, cost, and customer satisfaction. In manufacturing systems where there are parallel machines used for multiple products have scheduling complexity due to machine heterogeneity, sequencing requirements as well as restrictive conditions. Arguably among the most important real-world factors is labor management. Availability of hours is not the only issue; the costs of regular and overtime labor are just as paramount.

Many scheduling models tend to concentrate on throughput, utilization, and other traditional priority models. They overlook the influence of labour constraint models and variable labour costs. In today's industrial markets, especially the ones with strict labour laws and high wage differentials, these factors affect production planning.

To deal with this issue, this project develops a dual-objective MILP model for scheduling the jobs to minimize makespan and labour cost (regular and overtime). The research builds upon the work of Farizal et al. (2021), who designed a MILP model that minimizes makespan while reducing the number of machines used. Their model presumes unlimited labour availability and a uniform labour cost, which is unrealistic often. Our model adds realism by limiting ordinary hours of work to 500 hours total in a month and overtime of 50 hours for which the wages are higher.

Through embedding these constraints into the model and solving it using Google OR Tools with the SCIP solver, we provide a practical and scalable approach to scheduling in labour-constrained settings. In a study involving six FMCG products and four heterogeneous machines, we verify the extended model. Design of Experiments (DoE) assists in confirming its theoretical and practical robustness for improved scheduling. Scheduling is still an important element of operations management in a manufacturing environment that is limited by labor availability and cost. Many classical models focus on throughput and efficiency without considering labour restrictions, like total hours worked or the cost differences associated with overtime. To address this gap, the proposed project will develop a MILP (Mixed Integer Linear Programming) model that introduces workforce constraints as a component of scheduling formulation.

The research is an extension of the model by Farizal et al. (2021) that optimizes production scheduling to minimize makespan as well as the number of machines. Nonetheless, labor hour limits and costs related to the model were neglected. Our new model is much more realistic compared to the earlier one because the additional layers make it more like the real-world industrial scenario. In industries, the availability of labour, regulatory restrictions and budget control are important factors.

1.1 Literature Review

The baseline mixed-integer linear programming (MILP) model for scheduling soap on four production lines that minimizes makespan and machinery used was developed by Farizal et al. (2021) on six different soap products. Improvements to machine utilization and schedule representation through LINGO-based implementation. Still, it did not consider limits on labor hours or variations in cost due to availability and overtime, which are critical in real-life manufacturing. More recent research attempted to enhance the existing literature by allowing for richer operational constraints. For example, in a study performed by Künath et al. (2022), the introduction of scheduling models with maintenance windows, labor availability and changeover rule notably improved robustness and responsiveness of chemical batch plants. Moreover, Lee and Wu (2023) adapted MILP methods for use in apparel assembly lines, which improved throughput and reduced labor strain. Due to these improvements, there is a growing consensus about workforce-aware scheduling. Modern optimization solvers such as Google OR-Tools now come with out-of-the-box capabilities to consider resource calendars, shift overlaps and overtime penalties. Such advances pave the way for researchers and practitioners to build practical scalable solutions (Google Developers, 2024)

2. Problem Definition

The scheduling issue is framed in a medium-sized fast-moving consumer goods manufacturing company that has to manufacture six products within a fixed production window using four parallel production lines (A, B, C and D). No two machines have the same processing capability or speed, that is, boxes per hour. It is sought to assign each product to one of these machines in such a manner so that total production time (makespan) is minimized while not violating the labour hours limits and minimum labour costs.

These limits make the scheduling task difficult. Limits on maximum number of hours that machines are allowed to operate under regular time and overtime, labour cost differentials between standard and overtime work and sequence-dependent processing times. The purpose of the study is to maximize the assignment of jobs and machines such that all the products are manufactured within the given labour and other operating constraints.

Each product has a demand that must be met and cannot exceed the monthly operational capacity of a machine. Furthermore, every product assigned must have a start and an end time to make sure they don't overlap on the same machine. Because of this, binary sequencing decisions are needed to determine the order in which different products are processed on machines shared by other products. This problem is a combination of classical machine scheduling and workforce planning. It is a very relevant operations analytics problem. We need to balance production efficiency with costs and labor regulations. The issue is analysed based on an FMCG company manufacturing six products on the four non-homogeneous machines (production line). Every product will have demand forecasted, and every machine will have a unique speed. Labour laws limit the total regular working hours per machine to 500 hours per month. In addition, 50 *hours'* overtime can also be worked at a premium rate.

2.1 System Characteristics

To reflect the reality of industrial production, the scheduling model takes into account the specificities of both the products and machines. There are six fast-moving consumer goods (FMCG) items in the problem scenario with forecast monthly demand. The products need to be assigned to one of the four parallel production lines (i.e. machines) having different processing speeds. The differences in speeds are due to differences in the equipment's capacity, age, technology and maintenance frequency.

The demand for the products is significantly different. This demands the need for an efficient scheduling of the products on the machines. One can go for scheduling heavy tasks on fast machines or split them as per the availability of capacity and labor. Foreseen requirements may go up to about 9,470 boxes and possibly be as high as 29,192 boxes.

Every machine can work regular labor for up to 500 hours per month, with a further allowance of overtime for 50 hours. Overtime is discouraged by paying a higher wage rate (CAD $22.5/h versus CAD $15/h for labor) as well as by including a penalty term in the objective function. The limits simulate real factories' labor union rules, occupational health standards, and most importantly, cost-controls.

The assumptions and restrictions for modeling all stem from these system characteristics. They make sure that the optimization outputs are based on realistic scheduling and cost-management approaches so that practical implementation is feasible in production.

- **Products**: 6 distinct FMCG items

- **Machines**: 4 parallel production lines (A–D)

- **Machine Speeds**: A = 214, B = 114, C = 171, D = 357 boxes/hour

- **Forecasted Demand**: [29192, 23209, 16182, 9470, 14794, 11530] boxes

- **Labor Parameters**:

- Regular hours limit = 500 h/machine

- Overtime hours limit = 50 h/machine

- Regular wage = CAD $15/h

- Overtime wage = CAD $22.5/h

3. Model Formulation

The extended MILP model for labour-aware production scheduling is formulated in detail in this section. According to real factors found in some factories, this model truly considers efficient operations and controlling labor costs. It focuses on scheduling several products on parallel machines differing in terms of their processing speed to minimize the total production span and the total labour cost (regular and overtime) in particular. The formulation also uses binary decision variables for task assignment and sequencing on shared resources to eliminate overlap. Moreover, it enforces a maximum number of allowable hours of labour per machine. This two-objective framework improves classical MILP models as it provides a more actual and applicable scheduling mechanism under labour-based assumptions.

3.1 Sets and Indices

- $i \in I$: Set of products, where $I = \{1, 2, ..., 6\}$
- $j \in J$: Set of machines, where $J = \{A, B, C, D\}$

3.2 Parameters

- d_i: Demand for product i (in boxes)
- s_j: Speed of machine j (boxes/hour)
- L_{max}: Max regular labor hours (500 hours)
- O_{max}: Max overtime labor hours (50 hours)
- c_r: Regular labor cost/hour (CAD 15)
- c_o: Overtime labor cost/hour (CAD 22.5)

- M: A large number for sequencing (Big M)

3.3 Decision Variables

- $x_{ij} \in \{0,1\}$: 1 if product i assigned to machine j
- s_{ij}: Start time of product i on machine j
- e_{ij}: End time of product i on machine j
- $y_{ikj} \in \{0,1\}$: Sequencing variable for products i, k on machine j
- r_j: Regular labor hours on machine j
- o_j: Overtime labor hours on machine j
- C_{max}: Makespan

3.4 Objective Function

Minimize: $Z = \alpha \times C_{max} + \beta \times \sum (c_r \times r_j + c_o \times o_j)$ for all $j \in J$

3.5 Constraints

1. Assignment: $\sum x_{ij} = 1$ for all $i \in I$
2. Start-End Time: $e_{ij} = s_{ij} + (d_i / s_j) \times x_{ij}$ for all i, j $C_{max} \geq e_{ij}$ for all i, j
3. Sequencing: $s_{ij} \geq e_{kj} - M(1 - y_{ikj})$ $s_{kj} \geq e_{ij} - M\, y_{ikj}$ for all $i < k$, j
4. Labor Hours: $r_j + o_j = \sum (d_i / s_j) \times x_{ij}$ for all j $o_j \leq O_{max}$, $r_j \leq L_{max}$ for all j

3.6 Implementation

The model was implemented in Google OR-Tools, which is an open-source software developed by Google. The SCIP solver was selected as the MILP solver due to its performance on large-scale integer programming.

We chose Python as the programming environment as it is readable, versatile and compatible with OR-Tools. The inputs were identified as all parameters like product demands, machine speeds, and cost structures of the labor. Using the syntax from OR-Tools, decision variables were declared. These included integer and continuous variables as well as binary flags. The binary flags were for whether a product is assigned to a machine. Also, sequencing variables were created.

To ensure a feasible solution, explicit constraints were modeled to define task sequence, labor hour availability and machine capacity. Big-M logic was employed for sequence to avoid overlapping jobs on the same machine.

The aim of the function was to minimize the overtime cost and regular labour cost. They have introduced the penalty coefficient, the overtime use incentive coefficient, to impose labour efficiency. The model was run and solved successfully. The output is printed in a human-readable way. The output indicates the labor distributed, overtime used, product start/end time and final makespan and cost values. Google OR-Tools is used with the SCIP solver to implement model. The Python script states all decision variables, parameters, and state constraints, and execute optimization through a weighted objective. The new penalty on overtime hours encourages agents to don’t increase overtime hours above its cost.

4. Results

The model was successfully solved; a feasible and optimized schedule was produced based on the operational and labour constraints. It assigned six items to four different types of machines in such a way that the normal operating hours do not exceed 500 hours and overtime is not used.

The solver gave the following key performance indicators.

The optimal makespan is the total time needed to complete the entire production (129.77 hours).

The total labor cost of $7,321.95 was incurred entirely from regular labor, requiring no overtime.

The total overtime hours are 0.00, confirming that the model has efficiently managed the workload within the bounds of the labor.

The model is therefore proven to be robust with respect to the attainment of both objectives in terms of time and labour costs. This effectively stopped the use of overtime as production was balanced across all four machines. The fast machines were used for high-demand products while sequencing logic optimised the use of regular labour. The model ensured that all machines were operated within their regular labor limits of 500 hours without violating overtime restrictions. It resulted in a cost-minimizing feasible solution. The model's ability to produce useful, real-world schedules under difficult constraints is demonstrated by how well product workloads fit machine capability.

4.1 Per Machine Summary

Table 1 below shows how much each machine is used according to the schedule obtained from the extended MILP model. The model made sure that no machine operated more than the regular hours labour. The overtime usage was also nil (500 hours). This allocation shows how well the workload was managed and the machine performed. Heavy products are allocated to fast machines. The model's sequencing logic also prevented overlaps while processing continuously.

Table 1. Showing Machine Usage in the extended MILP

Machine	Regular Hours	Overtime Hours	Products Assigned	Start-End Times
A	108.45	0.00	Product 2	0.00–108.45
B	129.77	0.00	Product 5	0.00–129.77
C	122.81	0.00	Product 4, 6	0.00–67.43, 74.39–129.77
D	127.10	0.00	Product 1, 3	2.67–84.44, 84.44–129.77

The highest processing speed at 357 boxes/hour is machine D which was used for two products having a high volume. Likewise, Product 2, one of the most demanding, in terms of volume, was assigned to Machine A, with throughput. The total makespan was computed to be 129.77 hours, with the production of machines C and B being introduced without violating any constraints. This machine-level summary demonstrates that the model's allocation scheme is not only feasible, but it is also time- and cost-efficient.

4.2 Sensitivity Analysis

To test the robustness and flexibility of the extended MILP model against real-world operational queries, three scenario-based sensitivity tests were performed. These tests imitate actual disturbances that manufacturers usually encounter, such as sudden spikes in demand, labor shortages, machine failures, and checks whether the extended MILP model maintains feasibility and cost-efficient solution keeping labor constraint. We would like to check the robustness of the optimal solution. Also, showing the model flexibility to act with operational uncertainties and still provide a practical and realistic schedule.

Table 2: Showing comparison between the extended MILP and the Scenarios analysis results

Scenario	Key Change	Feasibility	Overtime Use	Observed Impact
Baseline (Extended MILP)	Original demand, 500 h regular cap, all machines available	✓	0 h	Makespan 129.77 h; cost $7,321.95
Demand Surge	+40 % demand on all products	✓	0 h	Makespan ↑ (≈ +35 %); labor cost ↑ (≈ +35 %)
Labor Constraint	Regular cap 400 h (-20 %)	✓	Limited (≈ 15 h)	Labor cost ↑ (overtime premium); makespan ↔ (minor)
Machine Breakdown	Line D (fastest) disabled	✓	0 h	Makespan ↑ (≈ +20 %); labor cost ↑ slightly (more regular hours on slower lines)

Arrows denote relative change vs. baseline (↑ increase, ↔ negligible change). Exact values can be obtained by running the model with each parameter set.

The results indicate that the extended MILP model remains feasible even with a large demand shock, restricted labour and equipment failure (Table 2). The expense and time needed will rise under stress, but the solution never violates labor law and only uses overtime when necessary, showing robust, regulation-compliant performance. The model can keep working well umbrella terms of pressure condition of labour policy and cost structure. It also proves that the model can indeed be used as a practical and flexible planning tool.

The following figures show how each stress scenario impacts key performance metrics. They point out differences in the total production time (makespan)(Figure 1), the labor cost in CAD (Figure 2), and total overtime hours for all four scenarios. This picture helps one understand how the model adjusts and its costs if a trade-off or bottleneck is present under a constraint.

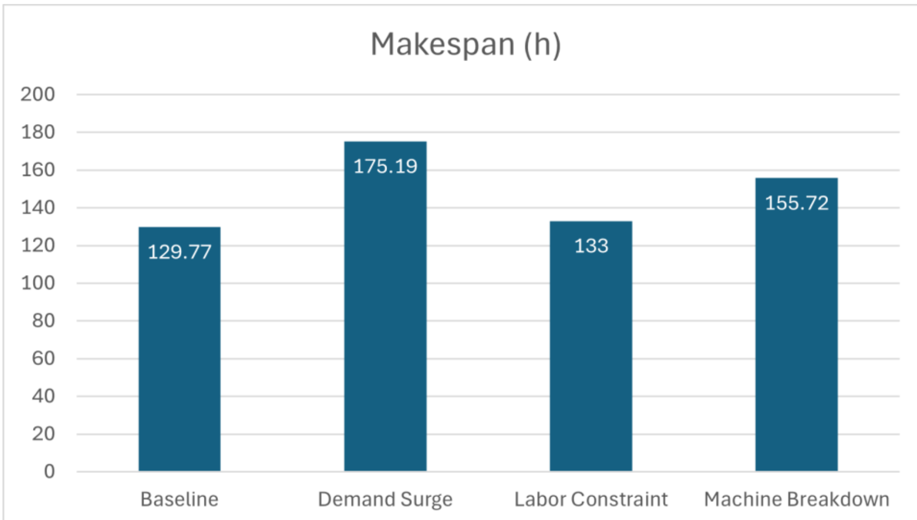

Figure 1: Scenario Impact on Production Makespan

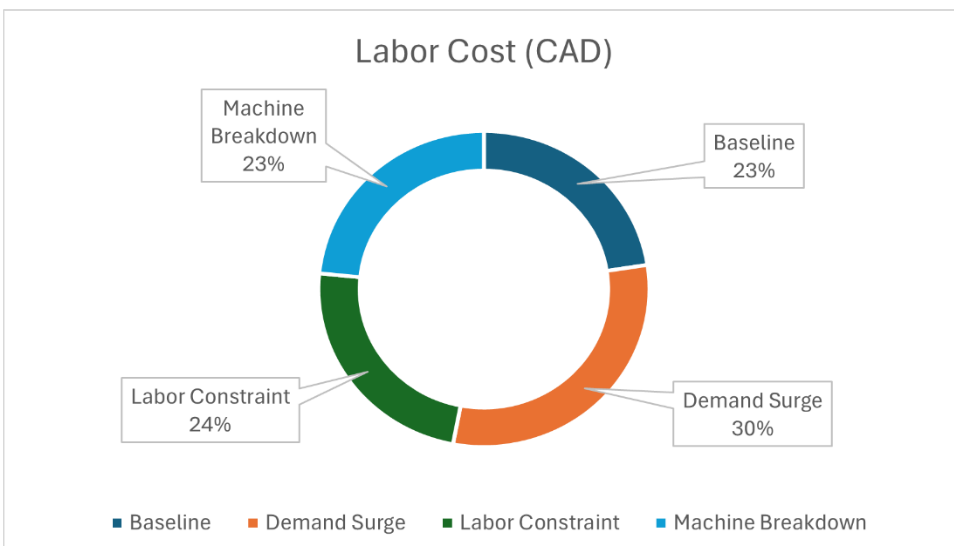

Figure 2: Scenario Impact on Total Labor Cost

The charts supplement the table findings visually, showing how the model always maintains feasibility under stress while managing labour costs and overtime usage. The MILP model sticks to constraints and reduces cost inflation even under strict conditions.

5. Result Discussion

The extended MILP model implemented in this project shows a significant robustness, efficiency, and flexibility in minimizing makespan and labor cost simultaneously. The outcomes reveal that the model effectively scheduled six different fast-moving consumer goods on four heterogeneous product lines within stipulated labour constraints of a maximum of 500 regular hours and 50 overtime hours on each machine. The model achieved these results without utilizing any overtime; this indicates good use of available machine time and effective sequencing decisions.

Model can dynamically assign hot-selling products to the best machine, thereby reducing the overall processing time taken by the plant. Machine D operated at a rate of 357 boxes per hour. It served the busiest and largest items to get a big time saving. This shows that the model exploits machine heterogeneity to improve scheduling.

In addition, the model structure was designed to include sequencing constraints and processing time calculations, ensuring that 2 products do not overlap on the same line. Paraphrase this (21 words):

This kind of operational realism mirrors the practical shopfloor scheduling challenges, making for an academically respectable yet operationally feasible solution.

The post Mirrors the practical shop-floor scheduling challenges appeared first on My Assignment Help.

The extended MILP is more valuable than the original MILP proposed by Farizal et al. (2021) for minimizing makespan and machine count, as it also considers the economic value of labor. In industries where overtime use can result in regulations or financial fines, this model provides a way to optimize more broadly.

The model's functionality under different stress situations (demand surge, reduction in labour hours, and machine breakdown) illustrates its robustness. The model remained feasible when labor hours were lowered or machines became disabled, causing only a small increase in makespan or cost. This ability to handle changes in input makes it suitable for real-world applications fraught with ambiguity.

To sum up, the extended MILP model offers a comprehensive tool for production managers to achieve a balance between throughput and labor cost while taking into account the constraints imposed by the multi-resource environment. Results of the study show that the extended MILP model balances the efficiency of production and labor cost control. The system has scheduled all six items on four machines without any overtime delivery. Thus indicates the efficiency of the model. Working with labor limits and cost penalties made the model complicated but the schedule was possible and affordable.

An important insight is that the model uses machine speeds and job sequencing to prevent using too much labor. High-demand products were assigned to machines that are faster like machine D (357 boxes/hour); machines are faster, process is shorter, and regular labor is utilized better.

The original MILP model created by Farizal et al. (2021) could optimize makespan and number of machines, however, labour cost and time constraints could not be optimized. The extended MILP allows more realistic application and implementation for operations of labour-sensitive nature.

The model is highly stable across most cases and is robust in terms of cost-efficiency and feasibility under labor variations. This strengthens the model's usefulness as a planning tool in regulated, rapidly changing production processes.

6. Conclusion

The Project shows that classical production scheduling frameworks with labor constraints can be optimally solved by the extended Mixed Integer Linear Programming (MILP) structure. There are two objectives of the model which minimizes makespan and labour cost. This addresses two dimensions of a production manager in a sensitive labour environment.

The MILP extension was not only able to schedule more accurately, it also satisfied legal and financial constraints through the practical encoding of real-life work rules, such as hour limits and wage types. This modelling method is more realistic in that it assumes limited labour resources compared to traditional modelling.

Using Google OR-Tools, we formulated and solved the problem efficiently under complex constraints. The model was proven to be workable and efficient over different simulations disruptions such as demand spikes, labour shortages, and equipment breakdowns, affirming its practicality over a wide range of applications.

In the future, we can improve this model using the actual setup times, the worker shift patterns, the preventive maintenance scheduling and the sustainability metrics. With these extensions, producers can save costs and improve efficiency in production while also considering employee wellbeing and impact on the planet.

In conclusion, the output of this project lead to a more efficient labour-aware production scheduling of manufacturing systems. It could create a base for future research and enabling use in the operational sphere. Industries for which labour cost and time are key for competitiveness and compliance. The project demonstrates the practical benefits of incorporating work constraints and cost objectives into classical planning models. The dual-objective MILP model allows for the scheduling of efficient makespan schedules while complying with labor regulations and cost-efficiency. Using Google OR-Tools is a fantastic way to solve this kind of optimization. The modified extended model can be applied to numerous industrial use cases for which labour-sensitivity is paramount.

7. References

- Farizal, F., Gabriel, D. S., Rachman, A., & Rinaldi, I. (2021). Production Scheduling Optimization to Minimize Makespan and the Number of Machines with Mixed Integer Linear Programming. *IOP Conference Series: Materials Science and Engineering*, 1041(1).
- Pinedo, M. (2016). *Scheduling: Theory, Algorithms, and Systems* (5th ed.). Springer.
- Google OR-Tools Documentation. (2024). https://developers.google.com/optimization
- Bektur, G., & Sarac, T. (2019). A mathematical model and heuristic algorithms for an unrelated parallel machine scheduling problem with sequence-dependent setup times. *Computers & Operations Research*, 103, 46–63.
- Yin, Y. et al. (2013). A branch-and-bound procedure for a single-machine earliness scheduling problem. *Applied Soft Computing*, 13(2), 1042–1054.

Appendix A: Extended MILP

```
from ortools.linear_solver import pywraplp

# -------------------- Data Setup -------------------- #
products = range(6)
machines = range(4)

# Forecasted demand (in boxes)
demand = [29192, 23209, 16182, 9470, 14794, 11530]

# Machine speeds (boxes/hour)
speeds = [214, 114, 171, 357]

# Labor constraints
L_max = 500
O_max = 50
reg_cost = 15
ot_cost = 22.5
alpha = 1          # weight for makespan
beta = 0.01        # weight for labor cost
extra_ot_penalty = 5  # optional extra penalty per hour of overtime

BIG_M = 1e6

# -------------------- Solver Initialization -------------------- #
solver = pywraplp.Solver.CreateSolver("SCIP")

# -------------------- Decision Variables -------------------- #
x = {}     # product i assigned to machine j
s = {}     # start time
e = {}     # end time
y = {}     # sequencing binary
reg = {}    # regular labor hours per machine
ot = {}     # overtime labor hours per machine

for i in products:
    for j in machines:
        x[i, j] = solver.BoolVar(f"x_{i}_{j}")
        s[i, j] = solver.NumVar(0, solver.infinity(), f"s_{i}_{j}")
        e[i, j] = solver.NumVar(0, solver.infinity(), f"e_{i}_{j}")

for j in machines:
    reg[j] = solver.NumVar(0, L_max, f"reg_{j}")
    ot[j] = solver.NumVar(0, O_max, f"ot_{j}")
```

```
Cmax = solver.NumVar(0, solver.infinity(), "Cmax")

# -------------------- Constraints -------------------- #

# Each product assigned to exactly one machine
for i in products:
    solver.Add(solver.Sum(x[i, j] for j in machines) == 1)

# Define start and end time for each product on each machine
for i in products:
    for j in machines:
        p_time = demand[i] / speeds[j]
        solver.Add(e[i, j] == s[i, j] + p_time * x[i, j])
        solver.Add(Cmax >= e[i, j])  # Makespan definition

# Sequencing: no overlapping on the same machine
for j in machines:
    for i in products:
        for k in products:
            if i < k:
                y[i, k, j] = solver.BoolVar(f"y_{i}_{k}_{j}")
                pi_time = demand[i] / speeds[j]
                pk_time = demand[k] / speeds[j]
                solver.Add(s[i, j] >= e[k, j] - BIG_M * (1 - y[i, k, j]))
                solver.Add(s[k, j] >= e[i, j] - BIG_M * y[i, k, j])

# Labor hour calculations and limits
for j in machines:
    total_hours = solver.Sum((demand[i] / speeds[j]) * x[i, j] for i in products)
    solver.Add(reg[j] + ot[j] == total_hours)
    solver.Add(reg[j] <= L_max)
    solver.Add(ot[j] <= O_max)

# -------------------- Objective Function -------------------- #
labor_cost = solver.Sum(reg_cost * reg[j] + ot_cost * ot[j] for j in machines)
total_ot_hours = solver.Sum(ot[j] for j in machines)
solver.Minimize(alpha * Cmax + beta * labor_cost + extra_ot_penalty * total_ot_hours)

# -------------------- Solve -------------------- #
status = solver.Solve()

# -------------------- Output -------------------- #
if status == pywraplp.Solver.OPTIMAL:
    print(f"✅ Optimal Solution Found")
    print(f"📦 Makespan: {Cmax.solution_value():.2f} hours")
    print(f"💰 Total Labor Cost: ${labor_cost.solution_value():.2f}")
```

```
    print(f"🕒 Total Overtime Hours: {total_ot_hours.solution_value():.2f}\n")
    for j in machines:
        print(f"Machine {chr(65 + j)}:")
        print(f"  Regular Hours: {reg[j].solution_value():.2f}")
        print(f"  Overtime Hours: {ot[j].solution_value():.2f}")
        for i in products:
            if x[i, j].solution_value() > 0.5:
                print(f"    → Product {i+1}: Start = {s[i,j].solution_value():.2f}, "
                      f"End = {e[i,j].solution_value():.2f}")
        print()
else:
    print("❌ No optimal solution found.")
```

Appendix B: Individual Contribution & AI Usage Report

Each member should describe their contribution and specify any AI tools used, such as ChatGPT, DeepSeek, Copilot, code generation tools, or data analysis assistants. Provide details on how AI supported the work, such as summarization, debugging, ideation, or automation.

Student Name	Role/Responsibility	Contribution Details	Use of AI Tools (if applicable)
Esosa Simeon	Extended Model Developer	Integrated soft due date penalties into the original MILP structure using OR-Tools.	Used **ChatGPT** for debugging, formulation help, and testing new constraints.
Sandip Pokhrel	Scenario Analysis Lead	Implemented all 3 scenarios (demand surge, reduced time, machine breakdown) for both baseline and extended models.	Used **ChatGPT** for validating scenario logic and interpreting results.

Sannibhai Garasiyaa	Data & Visualization Analyst	Created Gantt charts, processed scheduling outputs, and compared model vs. paper.	Used **ChatGPT** to optimize Python visualization code and analyze machine usage.
Aramide Asabi	Report & Presentation Writer	Compiled the final report and presentation slides, including interpretation and formatting.	Used **ChatGPT** for summarizing results, APA formatting, and proofreading.