Contents

List of Tables

Nomenclature

Sets and Variables

- **Sets:**
  - P: Set of different products.
  - M: Set of different types of machines.
  - S: Set of different types of scenarios.
- **Parameters:**
  - $T_{p,m}$: Minutes required to produce one unit of product p on machine m.
  - C_p: Profit for each unit of product p.
  - F_m: Financial cost for each machine m per day.
  - MAX_MINUTES: Maximum minutes available for production each day on each machine.
  - MAX_MACHINES: Maximum total number of machines that can be purchased, limited by the size of the factory.

1 MIP model for Benders Production Company

Variables

- x_m: Number of machines of type m to use, $\forall m \in \{1, \ldots, M\}$
- y_p: Number of units of product p produced, $\forall p \in \{1, \ldots, P\}$

Objective Function

$$\text{Maximize} \quad Z = \sum_{p=1}^{P} \mathrm{C}_p \cdot y_p - \sum_{m=1}^{M} \mathrm{F}_m \cdot x_m \tag{1}$$

Subject to

Machine Utilization

The total production time per machine does not exceed available machine minutes:

$$\sum_{p=1}^{P} \mathrm{Time}_{p,m} \cdot y_p \leq \mathrm{MAX\_MINUTES} \cdot x_m, \quad \forall m \in \{1, \ldots, M\} \tag{2}$$

Machine Availability

The total number of all machines does not exceed the factory limit:

$$\sum_{m=1}^{M} x_m \leq \mathrm{MAX\_MACHINES} \tag{3}$$

Variable Constraints

$$x_m \in Z^+, \quad \forall m \in \{1, \ldots, M\} \tag{4}$$

$$y_p \in R^+, \quad \forall p \in \{1, \ldots, P\} \tag{5}$$

Benders Production will achieve a daily profit of $10,430 by strategically acquiring machinery, including **two units** each of Machine Types **1, 5, and 6**, and **three units** each of Machine Types **2, 3, and 4.** The optimization model has determined that to maximize profitability, **only Product 7** should be produced, with a daily output set at **160 units**. This targeted production strategy optimizes resource allocation and maximizes returns efficiently. The code can be found in the file **Q1.jl**

2 Benders Subproblem

In this implementation of Benders Decomposition, we separate the decision variables by their nature: **continuous variables**, representing production quantities (y_p), are handled in the **subproblem**, while integer decisions about machine counts (x_m) remain in the master problem.

2.1 Benders Subproblem (BSP)

The Benders subproblem optimizes production levels assuming a given set of machines ($\bar{x}_m$). Here, C_p represents the profit per unit of product p and $T_{p,m}$ the number of minutes of production time required for product p on machine m. Thus the model is the following:

$$\begin{aligned} \text{Maximize} \quad & \sum_{p=1}^{P} C_p y_p \\ \text{Subject to} \quad & \sum_{p=1}^{P} T_{p,m} y_p \leq \text{MAX\_MINUTES} \cdot \bar{x}_m, \quad \forall m \in \{1, \dots, M\} \\ & y_p \geq 0, \quad \forall p \in \{1, \dots, P\} \end{aligned} \tag{6}$$

2.2 Dual of the Benders Subproblem

By taking the dual of the BSP, we can remove $\bar{x}_m$ from the constraint, allowing us to study the effect of this variable on the objective values. The dual variable (α_m) correspond to the constraints on production time, providing a mechanism to incorporate the time constraints impact on the objective function.

$$\begin{aligned} \text{Minimize} \quad & \sum_{m=1}^{M} (\text{MAX\_MINUTES} \cdot \bar{x}_m) \cdot \alpha_m \\ \text{Subject to} \quad & \sum_{m=1}^{M} T_{pm} \alpha_m \geq C_p, \quad \forall p \in \{1, \dots, P\} \\ & \alpha_m \geq 0, \quad \forall m \in \{1, \dots, M\} \end{aligned}$$

3 Benders Master Problem

The Master Problem (MP) is formulated using dual variables from the Benders Subproblem (SB). The $\geq$ inequality in the first constraint ensures that the MP solutions adequately cover the costs identified in the SB, thereby anchoring q to the subproblem's optimal value. Finally, in this case **q** can be a **free variable**.

$$
\begin{aligned}
\text{Maximize} \quad & -\sum_{m=1}^{M} \mathrm{F}_m \cdot x_m + q \\
\text{Subject to} \quad & \sum_{m=1}^{M} (\text{MAX\_MINUTES} \cdot \bar{a}_m) \cdot x_m \geq q \\
& \sum_{m=1}^{M} x_m \leq \text{MAX\_MACHINES} \\
& x_m \in Z^+, \quad \forall m \in \{1, \dots, M\}, q \in R
\end{aligned}
$$

4 Benders Algorithm

Algorithm 1 presents a general outline of the Benders Decomposition. It is **initially assumed that** one unit of each machine is purchased. We then iterate until the difference between the Upper Bound and the Lower Bound is below a delta value. We call the subproblem function and we get back the objective and the dual. We update the Lower Bound (LB) since our problem is maximization, we calculate the master problem objective, updating the upper bound (UB) and repeat until convergence.

Algorithm 1 Iterative Benders Decomposition

$UB \leftarrow \infty$
$LB \leftarrow -\infty$
$\Delta \leftarrow 0$
$x\_bar \leftarrow \mathrm{ones}(\mathrm{Int}, M)$ ▷ Assume 1 of each machine initially
$it \leftarrow 1$
while $UB - LB > \Delta$ **do**
    $(\mathrm{sub\_obj}, \alpha_{bar}) \leftarrow \mathrm{solve\_sub}(x\_bar)$
    $LB \leftarrow \max(LB, \mathrm{sub\_obj} + \mathrm{mas\_obj})$
    $mas\_obj \leftarrow \mathrm{solve\_master}(\alpha_{bar})$
    $x\_bar \leftarrow \mathrm{value}(x)$
    $UB \leftarrow mas\_obj$
    Print(progress)
    $it \leftarrow it + 1$
end while

Iteration	Upper Bound (UB)	Lower Bound (LB)	Subproblem Objective (Sub)
1	27745.98	4012.58	5616.58
2	20728.61	4012.58	0.0
3	19496.99	4012.58	0.0
4	14688.88	4012.58	0.0
5	12350.99	4012.58	0.0
6	11820.02	4012.58	0.0
7	11812.99	8106.71	12205.71
8	10976.34	8437.12	12368.12
9	10430.00	10430.00	14240.0
10	10430.00	10430.00	14240.0

Table 1: Iteration results for Benders Decomposition

Analyzing the iteration results from the Benders Decomposition, several observations can be made about the progression and outcomes of the solution process:

1. **Convergence of Bounds:** The Upper Bound (UB) and Lower Bound (LB) progressively converge over the iterations. Initially, there is a significant gap between UB and LB, suggesting substantial room for improvement in the early stages of the solution process. By the 10th iteration, the UB and LB converge to the same value (10430.00), indicating that an optimal solution has likely been reached.

2. **Subproblem Objectives:** The Subproblem Objective (Sub) starts high in the first iteration, indicating a potentially profitable but unoptimized scenario. It then drops to zero for several iterations, which may suggest that during these iterations, the master problem adjustments failed to yield better or feasible solutions or possibly that the subproblems were infeasible. The sudden spikes in iteration 7 and onwards to higher objective values coincide with adjustments in the master problem that improved feasibility and profitability.

At the end of the run, the algorithm converges to the same solution obtained with the MIP model in Q1. Specifically, it chooses to purchase 2 units of machine 1, 5, 6 and 3 units of machine 2, 3 and 4 respectively. It should be noted that in this problem the dual of the **quantity of product** p is studied. Thus, in order to calculate these **quantities** we ensure to store the dual variables in the subproblem using the **dual.** command. For more information the reader can see the outputs of the **file Q4.jl.**

5 Ray Generation Subproblem

The original problem is adjusted to **include a minimum production requirement**, denoted by $min - prod_p$. If too few machines are selected in the master problem, the dual of the subproblem may become unbounded, indicating that the primal (our original subproblem) is impossible (infeasible) to meet the demand with the available machine capacity. The inclusion of a ray-generation subproblem helps identify such issues.

5.1 Updated MIP Model with Demand Constraint

$$
\begin{aligned}
\text{Maximize} \quad & \sum_{p=1}^{P} C_p y_p - \sum_{m=1}^{M} \mathrm{F}_m x_m \\
\text{subject to} \quad & \sum_{p=1}^{P} T_{p,m} y_p \leq \text{MAX\_MINUTES} \cdot x_m, \quad \forall m \in \{1, \ldots, M\} \\
& \sum_{m=1}^{M} x_m \leq \text{MAX\_MACHINES} \quad \forall m \in \{1, \ldots, M\} \\
& y_p \geq \text{min-prod}_p, \quad \forall p \in \{1, \ldots, P\} \\
& x_m \in Z^{+} \quad \forall m \in \{1, \ldots, M\} \\
& y_p \geq 0, \quad \forall p \in \{1, \ldots, P\}
\end{aligned}
$$

5.2 Benders Subproblem (BSP) with Rays

Given fixed machine availability $\bar{x}_m$, the subproblem maximizes production based on current machine allocations, ensuring production is at least the daily demand:

$$
\begin{aligned}
\text{Maximize} \quad & \sum_{p=1}^{P} C_p y_p \\
\text{subject to} \quad & \sum_{p=1}^{P} T_{p,m} y_p \leq \text{MAX\_MINUTES} \cdot \bar{x}_m, \quad \forall m \in \{1, \ldots, M\} \\
& - y_p \leq -\text{min-prod}_p, \quad \forall p \in \{1, \ldots, P\} \\
& y_p \geq 0, \quad \forall p \in \{1, \ldots, P\}
\end{aligned}
$$

5.3 Dual of the BSP

$$
\begin{aligned}
\text{Minimize} \quad & \sum_{m=1}^{M} \text{MAX\_MINUTES} \cdot \bar{x}_m \cdot \alpha_m - \sum_{p=1}^{P} \text{min-prod}_p \cdot \beta_p \\
\text{subject to} \quad & \sum_{m=1}^{M} T_{p,m}\alpha_m - \beta_p \geq C_p, \quad \forall p \in \{1, \ldots, P\} \\
& \alpha_m \geq 0, \quad \forall m \in \{1, \ldots, M\} \\
& \beta_p \geq 0, \quad \forall p \in \{1, \ldots, P\}
\end{aligned}
$$

5.4 Ray Generation Subproblem

To identify infeasibility in meeting production requirements:

$$
\begin{aligned}
\text{Minimize} \quad & -1 \\
\text{subject to} \quad & \sum_{m=1}^{M} \text{MAX\_MINUTES} \cdot \bar{x}_m \cdot \alpha_m - \sum_{p=1}^{P} \text{min-prod}_p \cdot \beta_p = -1 \\
& \sum_{m=1}^{M} T_{p,m}\alpha_m - \beta_p \geq 0, \quad \forall p \in \{1, \ldots, P\} \\
& \alpha_m \geq 0, \quad \forall m \in \{1, \ldots, M\} \\
& \beta_p \geq 0, \quad \forall p \in \{1, \ldots, P\}
\end{aligned}
$$

5.5 Benders Master Problem (BMP)

Incorporating the dual variables from the BSP:

$$
\begin{aligned}
\text{Maximize} \quad & -\sum_{m=1}^{M} \text{Cost}_m \cdot x_m + q \\
\text{subject to} \quad & \sum_{m=1}^{M} \text{MAX\_MINUTES} \cdot \bar{\alpha}_m^p \cdot x_m - \sum_{p=1}^{P} \text{min-prod}_p \cdot \bar{\beta}_p^p \geq q \\
& \sum_{m=1}^{M} \text{MAX\_MINUTES} \cdot \alpha_m^r \cdot x_m - \sum_{p=1}^{P} \text{min-prod}_p \cdot \beta_p^r \geq 0 \\
& \sum_{m=1}^{M} x_m \leq \text{MAX\_MACHINES} \\
& x_m \in Z^+ \quad \forall m \in \{1, \ldots, M\}, \quad 100000 \geq q \geq -100000
\end{aligned}
$$

6 Benders Algorithm w/ Rays

Iteration	Upper Bound (UB)	Lower Bound (LB)	Sub Objective	Num of Machines
0	Inf	-Inf	-	[1, 1, 1, 1, 1, 1]
1	99438.0	-Inf	-1	[0, 2, 0, 0, 0, 0]
2	99109.0	-Inf	-1	[1, 2, 0, 0, 0, 0]
3	98888.0	-Inf	-1	[1, 2, 1, 0, 0, 0]
4	98788.0	-Inf	-1	[1, 2, 1, 1, 0, 0]
5	98461.0	-Inf	-1	[1, 2, 1, 1, 1, 0]
6	98115.0	-Inf	-1	[1, 2, 1, 1, 1, 1]
7	97788.0	-Inf	-1	[1, 2, 1, 1, 2, 1]
8	97688.0	-Inf	-1	[1, 2, 1, 2, 2, 1]
9	97342.0	-Inf	-1	[1, 2, 1, 2, 2, 2]
10	97121.0	-Inf	-1	[1, 2, 2, 2, 2, 2]
11	96792.0	-Inf	-1	[2, 2, 2, 2, 2, 2]
12	15670.1	6464.5	9672.5	[2, 5, 2, 2, 2, 2]
13	12873.5	6489.2	10540.2	[2, 2, 2, 2, 5, 2]
14	9746.8	6489.2	9728.2	[2, 4, 3, 2, 2, 2]
15	9269.4	6860.8	10851.8	[2, 3, 3, 2, 3, 2]
16	9212.0	7150.1	11187.1	[2, 3, 3, 3, 2, 2]
17	8798.3	7806.6	11616.6	[2, 3, 2, 3, 3, 2]
18	8552.3	7806.6	11461.1	[2, 3, 2, 2, 3, 3]
19	7850.5	7806.6	11500.5	[2, 2, 2, 3, 3, 3]
20	7806.6	7806.6	10116.3	[2, 3, 3, 3, 2, 2]

Table 2: Iteration Summary for Benders Decomposition with Ray Generation

We can clearly see that up until iteration 11, feasible solution could not be obtained. Thus the ray objective was returning the -1 value. From iteration 12 onwards both UB and LB stabilize and gradually converge. The subproblem objectives suggest feasible solutions have been found that align closer to optimal. These values reflect the iterative refining process of the master problem as it incorporates feedback (Benders cuts) from both the subproblems and ray generation problems, making sure that all production requirements are feasibly met under the constraints imposed by the current machine allocations. The **quantities of produced products** can be found in the **file Q6.jl**. Finally, one can observe that the same **number of machines** are purchased (iteration 20) as in the previous questions of the assignment.

7 Stochastic Program for Benders Production

The original model is adapted to include stochastic demands for each product under various scenarios:

Variables

- x_m: Number of machines of type m, $\forall m \in \{1, \ldots, M\}$ (1st-Stage Decision Variable)
- $y_{p,s}$: Production of product p in scenario s, $\forall p \in \{1, \ldots, P\}$, $\forall s \in \{1, \ldots, S\}$ (2nd-Stage Decision Variable)

Objective Function

Maximize the expected profit:

$$\text{Maximize} \quad Z = -\sum_{m=1}^{M} \mathrm{F}_m \cdot x_m + \sum_{s=1}^{\mathrm{S}} \pi_s \left(\sum_{p=1}^{P} \mathrm{C}_p \cdot y_{p,s} \right)$$

where π_s is the probability of scenario s.

Constraints

Total Machines Constraint

$$\sum_{m=1}^{M} x_m \leq \mathrm{MAX\_MACHINES}, \quad m \in \{1, \ldots, M\}$$

Production Time Constraint

$$\sum_{p=1}^{P} T_{pm} \cdot y_{p,s} \leq \mathrm{MAX\_MINUTES} \cdot x_m, \quad \forall m \in \{1, \ldots, M\}, \forall s \in \{1, \ldots, \mathrm{S}\}$$

Minimum Production Requirement

$$y_{p,s} \geq \mathrm{MIN\_PROD}_p, \quad \forall p \in \{1, \ldots, P\}, \quad \forall s \in \{1, \ldots, \mathrm{S}\}$$

Maximum Demand Constraint

It should be noted that the **Demand matrix has been transposed** to better fit the indices of the decision variables

$$y_{p,s} \leq \text{MAX\_DEMAND}_{p,s}, \quad \forall p \in \{1, \ldots, P\}, \quad \forall s \in \{1, \ldots, \text{S}\}$$

Variables

$$x_m \in Z^+, \quad \forall m \in \{1, \ldots, M\}$$
$$y_{p,s} \in R^+, \quad \forall p \in \{1, \ldots, P\}, \quad \forall s \in \{1, \ldots, S\}$$

The results of the model show that the optimal number of each machine type to be bought are again 2 units of machines 1, 5 and 6 and 3 units of machines 2, 3 and 4 at the Profit of $7120.19. That is, **given the uncertainty introduced from the demand scenarios**, Benders Production should still aim to purchase the same number of machine types as before. More information about the **quantities produced** per product and scenario can be found in the **file "Q7.jl"**.

8 Benders Decomposition on Stochastic Program

8.1 Benders Subproblem (BSP)

Given fixed machine allocations, the subproblem aims to maximize production while satisfying daily stochastic demands. The objective is specifically to maximize the expected profit across scenarios:

$$\text{Maximize} \quad z = \sum_{s=1}^{\text{S}} \pi_s \left(\sum_{p=1}^{P} \text{C}_p \cdot y_{p,s} \right)$$

$$\begin{aligned}
\text{subject to} \quad & \sum_{s=1}^{S} \sum_{p=1}^{P} T_{p,m} \cdot y_{p,s} \leq \text{MAX_MINUTES} \cdot \bar{x}_m, \quad \forall m \in \{1, \ldots, M\} \\
& y_{p,s} \geq \text{MIN_PROD}_p, \quad \forall p \in \{1, \ldots, P\}, \quad \forall s \in \{1, \ldots, S\} \\
& y_{p,s} \leq \text{MAX_DEMAND}_{p,s}, \quad \forall p \in \{1, \ldots, P\}, \quad \forall s \in \{1, \ldots, S\} \\
& y_{p,s} \geq 0, \quad \forall p \in \{1, \ldots, P\}, \quad \forall s \in \{1, \ldots, S\}
\end{aligned}$$

8.2 Dual of the BSP with Scenarios

$$
\begin{aligned}
\text{Minimize} \quad & \sum_{s=1}^{S} \left(\sum_{m=1}^{M} \text{MAX\_MINUTES} \cdot \bar{x}_m \cdot \alpha_{m,s} - \sum_{p=1}^{P} \beta_{p,s} \cdot \text{min-prod}_p + \sum_{p=1}^{P} \gamma_{p,s} \cdot \text{DEMAND}_{p,s} \right) \\
\text{subject to} \quad & \sum_{m=1}^{M} T_{p,m} \alpha_{m,s} + \gamma_{p,s} - \beta_{p,s} \geq \pi_s C_{\text{p}}, \quad \forall p \in \{1, \dots, P\}, \quad \forall s \in \{1, \dots, S\} \\
& \alpha_{m,s} \geq 0, \quad \forall m \in \{1, \dots, M\}, \quad \forall s \in \{1, \dots, S\} \\
& \beta_{p,s} \geq 0, \quad \forall p \in \{1, \dots, P\}, \quad \forall s \in \{1, \dots, S\} \\
& \gamma_{p,s} \geq 0, \quad \forall p \in \{1, \dots, P\}, \quad \forall s \in \{1, \dots, S\}
\end{aligned}
$$

8.3 Ray Generation Subproblem

$$
\begin{aligned}
\text{Minimize} \quad & -1 \\
\text{subject to} \quad & \sum_{s=1}^{S} \left(\sum_{m=1}^{M} \text{MAX\_MINUTES} \cdot \bar{x}_m \cdot \alpha_{m,s} - \right. \\
& \left. \quad \sum_{p=1}^{P} \beta_{p,s} \cdot \text{MIN\_PROD}_p + \sum_{p=1}^{P} \gamma_{p,s} \cdot \text{MAX\_DEMAND}_{p,s} \right) = -1 \\
& \sum_{m=1}^{M} T_{p,m} \alpha_{m,s} + \gamma_{p,s} - \beta_{p,s} \geq 0, \quad \forall p \in \{1, \dots, P\}
\end{aligned}
$$

8.4 Benders Master Problem (BMP)

Incorporating the dual variables from the Benders Subproblems:

$$
\begin{aligned}
\text{Maximize} \quad & -\sum_{m=1}^{M} \mathrm{F}_m \cdot x_m + q \\
\text{subject to} \quad & \sum_{s=1}^{S} \left(\sum_{m=1}^{M} \text{MAX\_MINUTES} \cdot x_m \cdot \bar{\alpha}_{m,s} - \sum_{p=1}^{P} \text{min-prod}_p \cdot \bar{\beta}_{p,s} \right. \\
& \quad \left. + \sum_{p=1}^{P} \text{DEMAND}_{p,s} \cdot \bar{\gamma}_{p,s} \right) \geq q, \\
& \sum_{s=1}^{S} \left(\sum_{m=1}^{M} \text{MAX\_MINUTES} \cdot x_m \cdot \bar{\alpha}_{m,s} - \sum_{p=1}^{P} \text{min-prod}_p \cdot \bar{\beta}_{p,s} \right. \\
& \quad \left. + \sum_{p=1}^{P} \text{DEMAND}_{p,s} \cdot \bar{\gamma}_{p,s} \right) \geq 0, \\
& \sum_{m=1}^{M} x_m \leq \text{MAX\_MACHINES}, \\
& x_m \in Z^{+} \quad \forall m \in \{1, \ldots, M\}, \quad 100000 \geq q \geq -100000
\end{aligned}
$$

9 Benders Algorithm w/ Rays for Stochastic Program

Iteration	Upper Bound (UB)	Lower Bound (LB)	Sub Objective	Num of Machines
0	Inf	-Inf	-	[1, 1, 1, 1, 1, 1]
1	99438.0	-Inf	-1.0	[0, 2, 0, 0, 0, 0]
2	99109.0	-Inf	-1.0	[1, 2, 0, 0, 0, 0]
3	98888.0	-Inf	-1.0	[1, 2, 1, 0, 0, 0]
4	98788.0	-Inf	-1.0	[1, 2, 1, 1, 0, 0]
5	98461.0	-Inf	-1.0	[1, 2, 1, 1, 1, 0]
6	98115.0	-Inf	-1.0	[1, 2, 1, 1, 1, 1]
7	97788.0	-Inf	-1.0	[1, 2, 1, 1, 2, 1]
8	97688.0	-Inf	-1.0	[1, 2, 1, 2, 2, 1]
9	97342.0	-Inf	-1.0	[1, 2, 1, 2, 2, 2]
10	97121.0	-Inf	-1.0	[1, 2, 2, 2, 2, 2]
11	96792.0	-Inf	-1.0	[2, 2, 2, 2, 2, 2]
12	17844.4	6261.0	9468.7	[2, 5, 2, 2, 2, 2]
13	11644.0	6261.0	10305.0	[2, 3, 2, 2, 4, 2]
14	9633.0	6716.0	10859.4	[2, 3, 2, 3, 3, 2]
15	8555.3	7055.6	10971.6	[2, 3, 3, 3, 2, 2]
16	7613.7	7120.2	10930.2	[2, 3, 3, 2, 3, 2]
17	7551.4	7120.2	10896.1	[3, 3, 2, 2, 3, 2]
18	7379.2	7120.2	10873.2	[2, 3, 3, 2, 3, 3]
19	7186.1	7120.2	10897.2	[2, 3, 3, 2, 2, 2]
20	7120.2	7120.2	10464.2	[2, 3, 3, 3, 2, 2]

Table 3: Iteration Summary for Benders Decomposition

Initially, it is evident that the algorithm converges after 20 iterations (Benders cuts), consistent with the behavior observed in Q6. The introduction of additional constraints reduced the objective value and the final outcome of the algorithm aligns well with the MIP formulation detailed in Q7. For more information, see **Q9.jl**

10 Parallel Benders Decomposition and CPLEX Automatic Decomposition (9*)

Implementation of Parallel Benders Decomposition Using JuMP

In the updated approach for solving the Benders decomposition, each subproblem is treated distinctly for different scenarios. Unlike the previous version, where all scenarios were combined into a single optimization problem (Q8), Q10 (Q9*) introduces a scenario-specific parameter s to each function. This facilitates the separate resolution of each subproblem according to the unique conditions of its corresponding scenario. The code for this part is in **Q9 Star Parallel Benders.jl**

Advantages

- **Parallel Processing:** Each subproblem can be processed independently and concurrently. This is particularly advantageous when scenarios do not interact directly, allowing for the use of parallel computing resources to enhance computational efficiency.
- **Memory and Performance:** Solving smaller, separate problems can sometimes be more memory-efficient and faster, depending on the nature of the individual subproblems and the computational resources available.

Disadvantages

- **Optimality Concerns:** When subproblems are solved independently without considering their interdependencies, there is a risk that the solutions may not converge to a global optimum.

For the specific example discussed, the execution time for both Q8 and Q10 (Q9*) remains comparable, suggesting that in this instance, the choice between these strategies may be driven by factors other than performance, such as clarity of implementation or the availability of parallel processing capabilities

Iteration	UB	LB	SP Obj	Num of Machines
0	∞	$-\infty$	-	[1, 1, 1, 1, 1, 1]
1	99438.0	$-\infty$	[-1, -1, -1, -1, -1]	[0, 2, 0, 0, 0, 0]
2	99109.0	$-\infty$	[-1, -1, -1, -1, -1]	[1, 2, 0, 0, 0, 0]
3	98888.0	$-\infty$	[-1, -1, -1, -1, -1]	[1, 2, 1, 0, 0, 0]
4	98788.0	$-\infty$	[-1, -1, -1, -1, -1]	[1, 2, 1, 1, 0, 0]
5	98461.0	$-\infty$	[-1, -1, -1, -1, -1]	[1, 2, 1, 1, 1, 0]
6	98115.0	$-\infty$	[-1, -1, -1, -1, -1]	[1, 2, 1, 1, 1, 1]
7	97788.0	$-\infty$	[-1, -1, -1, -1, -1]	[1, 2, 1, 1, 2, 1]
8	97688.0	$-\infty$	[-1, -1, -1, -1, -1]	[1, 2, 1, 2, 2, 1]
9	97342.0	$-\infty$	[-1, -1, -1, -1, -1]	[1, 2, 1, 2, 2, 2]
10	97121.0	$-\infty$	[-1, -1, -1, -1, -1]	[1, 2, 2, 2, 2, 2]
11	96792.0	$-\infty$	[-1, -1, -1, -1, -1]	[2, 2, 2, 2, 2, 2]
12	17844.4	6260.8	[1912, 1834, 1868, 1928, 1927]	[2, 5, 2, 2, 2, 2]
13	11644.0	6260.8	[2102, 1990, 2095, 2018, 2100]	[3, 3, 2, 2, 4, 2]
14	9633.0	6716.4	[2152, 2132, 2204, 2137, 2235]	[3, 3, 2, 3, 3, 2]
15	8555.3	7055.7	[2152, 2198, 2225, 2142, 2255]	[3, 3, 3, 3, 2, 2]
16	7613.7	7120.2	[2323, 1990, 2276, 2018, 2323]	[3, 3, 3, 2, 3, 2]
17	7551.4	7120.2	[2188, 2132, 2204, 2137, 2235]	[3, 3, 2, 2, 3, 2]
18	7379.2	7120.2	[2154, 2132, 2204, 2145, 2239]	[3, 3, 2, 2, 3, 3]
19	7186.1	7120.2	[2158, 2137, 2204, 2162, 2235]	[3, 3, 3, 2, 2, 2]
20	7120.2	7120.2	[2168, 1990, 2132, 2018, 2154]	[2, 3, 3, 3, 2, 2]

Table 4: Benders Decomposition Iteration Details

Automatic Benders Decomposition Using CPLEX

In order to test the results of my algorithm I used the Automatic Benders Decomposition capabilities of the CPLEX solver, (see file **Q9 star Automatic Bender.jl**). Specifically, the following commands were used:

```
# Optimize the Model from Q7 using CPLEX solver
set_optimizer_attribute(model, "CPXPARAM_Preprocessing_Presolve", 0)
set_optimizer_attribute(model, "CPXPARAM_Benders_Strategy", 3)
set_optimizer_attribute(model, "CPX_PARAM_CUTPASS", -1)
```

To record the time for the CPLEX solver to run, the @time macro was used:

```
@time begin
    optimize!(model)
end
```

CPLEX Solver Performance and Configuration

A few aspects of the CPLEX commands are noteworthy. Setting the **CPXPARAM_Preprocessing_Presolve** command to 0 disables the presolve processes. Setting the **CPXPARAM_Benders_Strategy** to 3 configures CPLEX to use an aggressive Benders approach, automatically generating Benders cuts at every node in the search tree, rather than solely at the root node. Setting the **CPX_PARAM_CUTPASS** to -1 means that the cut process is limited.

Solver Performance and Comparison

The CPLEX solver efficiently applies Benders decomposition, leveraging its parallel processing capabilities. Key performance metrics and comparisons are highlighted below:

- CPLEX consistently outperformed our algorithm in terms of speed. The total elapsed solution time averaged 0.22 seconds for CPLEX, compared to 1.8 seconds for my algorithm. This difference likely reflects CPLEX optimized internal mechanisms and superior handling of parallel computations.
- With **CPXPARAM_Preprocessing_Presolve** set to 0, CPLEX executed 55 Benders cuts, reducing to 39 when set to 1. This variation highlights how presolve settings can influence the efficiency of the decomposition process, affecting both the number and effectiveness of the cuts.
- Setting **CPXPARAM_Benders_Strategy** to 0 disabled the use of Benders cuts entirely. Adjusting it to 2 resulted in CPLEX Error 2000, indicating "No Benders decomposition available."
- Setting **CPX_PARAM_CUTPASS** to 1 significantly reduced the algorithm's runtime to less than 0.03 seconds to reach optimality, again without utilizing Benders cuts, indicating that while Benders cuts are powerful, they may not always be necessary for achieving optimal solutions quickly.
- My algorithm applied 20 Bender cuts, and while it did take longer time, the final solution values were identical to those achieved by the CPLEX solver. This outcome supports the robustness of the approach, though potential improvement to enhance speed could be implemented.