**Programming Assignment: Finite Capacity Scheduling for a Factory Using Pyomo**

**Context:**

You are working for a smart manufacturing company that produces custom mechanical parts for industrial machinery. The company operates in a job shop environment where customer orders (referred to as “jobs”) consist of multiple sequential tasks. Each task must be processed on a specific machine for a fixed amount of time and may depend on the completion of a preceding task.

The factory has finite machine availability, works on a defined calendar (shift-based working hours), and faces typical operational challenges like missed delivery deadlines, machine underutilization, and high work-in-progress (WIP). To tackle these issues, the factory is transitioning to Finite Capacity Scheduling (FCS).

Your job is to build a Pyomo-based optimization model that creates realistic, conflict-free production schedules that take into account machine capacity, shift calendars, and task dependencies.

**Problem Statement:**

You are given the following:

* A list of machines and their respective working hours (calendar)
* A list of jobs, where each job consists of dependent tasks
* Each task has:
  + A required machine
  + A fixed processing time (in hours)
  + A dependency (must be scheduled after a predecessor task)

**Your objective:** Create a Pyomo model to generate an optimal schedule that:

* Respects machine working hours (calendar)
* Ensures no two tasks overlap on the same machine
* Respects the order of tasks within a job (task dependencies)
* Minimizes the makespan (time at which the last task finishes)

**Factory Input Data (Example):**

# Machines in the factory and their working hours  
machines = ['LaserCutter', 'CNC\_Mill', 'PaintStation']  
  
# Daily working hours (in 24-hour format)  
calendar = {  
 'LaserCutter': list(range(8, 17)), # 8 AM to 5 PM  
 'CNC\_Mill': list(range(8, 16)), # 8 AM to 4 PM  
 'PaintStation': list(range(9, 18)) # 9 AM to 6 PM  
}

Consider the time in minutes. For example 9AM to 6PM should be like [540, 1080]. For next day 9AM to 6PM it should be like [1980, 2520]

# Jobs with tasks (job\_id, task\_id, machine, duration\_in\_hours, predecessor\_task\_id)  
tasks = [  
 ('Job1', 'Cutting', 'LaserCutter', 3, None),  
 ('Job1', 'Milling', 'CNC\_Mill', 2, 'Cutting'),  
 ('Job1', 'Painting', 'PaintStation', 1, 'Milling'),  
  
 ('Job2', 'Cutting', 'LaserCutter', 2, None),  
 ('Job2', 'Painting', 'PaintStation', 1, 'Cutting'),  
  
 ('Job3', 'Milling', 'CNC\_Mill', 4, None),  
 ('Job3', 'Painting', 'PaintStation', 2, 'Milling')  
]

**Tasks to Complete:**

1. Model the scheduling problem using Pyomo:
   * Define decision variables for task start and end times
   * Add constraints to:
     + Ensure task precedence is respected
     + Prevent overlap on the same machine
     + Ensure tasks are scheduled within working calendar hours
   * Define an objective function to minimize the makespan
2. Display the final output:
   * Start and end time of each task
   * Machine used
   * Total makespan
3. (Optional Bonus):
   * Plot a Gantt chart using matplotlib to visualize the schedule
   * Calculate machine utilization

**Expected Outcome:**

A Python script that:

* Reads the task and machine data
* Builds and solves an optimization model using Pyomo
* Outputs a feasible and optimal job schedule minimizing the makespan

This assignment is designed to test your understanding of optimization modeling in manufacturing environments, practical use of Pyomo, and your ability to convert real-world production data into mathematical constraints.