**Prompt 1**

I'd like to ask you to help me solve a resource-constrained project scheduling problem. I will provide you with a piece of code and project information. Please read the code and project information for you first.

**Prompt 2**

Code:

from gurobipy import Model, GRB

# Define the project data

tasks = {

0: {'duration': 0, 'resource': 0, 'predecessors': []},

1: {'duration': 5, 'resource': 1, 'predecessors': [0]},

2: {'duration': 3, 'resource': 7, 'predecessors': [0]},

3: {'duration': 8, 'resource': 3, 'predecessors': [1, 2]},

4: {'duration': 5, 'resource': 7, 'predecessors': [0]},

5: {'duration': 9, 'resource': 6, 'predecessors': [3, 4]},

6: {'duration': 4, 'resource': 5, 'predecessors': [3]},

7: {'duration': 10, 'resource': 4, 'predecessors': [2, 4]},

8: {'duration': 2, 'resource': 5, 'predecessors': [5, 6, 7]},

9: {'duration': 2, 'resource': 6, 'predecessors': [8]},

10: {'duration': 9, 'resource': 6, 'predecessors': [1]},

11: {'duration': 0, 'resource': 0, 'predecessors': [9, 10]}

}

max\_resource\_availability = 10

T = sum(task['duration'] for task in tasks.values()) # planning horizon

# Initialize the model

model = Model('ProjectScheduling')

# Define the decision variables

x = {}

for i in tasks.keys():

for t in range(T):

x[i, t] = model.addVar(vtype=GRB.BINARY, name=f"x\_{i}\_{t}")

# Define the objective function

model.setObjective(

sum(t \* x[11, t] for t in range(T)),

GRB.MINIMIZE

)

# Add precedence constraints

for j in tasks.keys():

for i in tasks[j]['predecessors']:

for t in range(T):

model.addConstr(

sum(x[i, tau] for tau in range(t - tasks[i]['duration'] + 1)) -

sum(x[j, tau] for tau in range(t + 1)) >= 0,

name=f"precedence\_{i}\_{j}\_{t}"

)

# Add resource constraints

for t in range(T):

model.addConstr(

sum(tasks[i]['resource'] \* x[i, tau] for i in tasks.keys() for tau in range(max(0, t - tasks[i]['duration'] + 1), t + 1)) <= max\_resource\_availability,

name=f"resource\_{t}"

)

# Add task completion constraint

for i in tasks.keys():

model.addConstr(

sum(x[i, t] for t in range(T)) == 1,

name=f"completion\_{i}"

)

# Optimize the model

model.optimize()

# Extract the solution

schedule = {i: None for i in tasks.keys()}

for i in tasks.keys():

for t in range(T):

if x[i, t].X > 0.5: # If the variable is set to 1 in the solution

schedule[i] = t

break

# Print the schedule

for i in sorted(schedule.keys()):

print(f"Task {i} starts at time {schedule[i]}")

**Prompt 3**

Project information:

Here is information about the tasks for the production of MiC modules in the MiC project. This includes the task number, task name, task duration, task resource requirements, and each task's predecessor tasks. Specific details are as follows. Please convert the information in the table into a format that can be calculated by the code. Examples are as follows:

# Define the project data

tasks = {

0: {'duration': 0, 'resource': 0, 'predecessors': []},

1: {'duration': 5, 'resource': 1, 'predecessors': [0]},

2: {'duration': 3, 'resource': 7, 'predecessors': [0]},

3: {'duration': 8, 'resource': 3, 'predecessors': [1, 2]},

4: {'duration': 5, 'resource': 7, 'predecessors': [0]},

5: {'duration': 9, 'resource': 6, 'predecessors': [3, 4]},

6: {'duration': 4, 'resource': 5, 'predecessors': [3]},

7: {'duration': 10, 'resource': 4, 'predecessors': [2, 4]},

8: {'duration': 2, 'resource': 5, 'predecessors': [5, 6, 7]},

9: {'duration': 2, 'resource': 6, 'predecessors': [8]},

10: {'duration': 9, 'resource': 6, 'predecessors': [1]},

11: {'duration': 0, 'resource': 0, 'predecessors': [9, 10]}

}

max\_resource\_availability = 10

Resource available: 100

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Task | Durations (Day) | Resource demand | Predecessor |
| 1 | IPA30 BD submission & approval | 566 | 20 | N/A |
| 2 | MiC Drawing BD Submission | 211 | 30 | 1 |
| 3 | MiC Shop Drawing Submission | 179 | 30 | 1 |
| 4 | Material submission | 179 | 50 | 1 |
| 5 | Drawing submission | 59 | 30 | 2,3,4 |
| 6 | Fabrication of Mock Up | 89 | 60 | 5 |
| 7 | Off-site mock up set up and installation | 67 | 70 | 6 |
| 8 | Concrete Slab | 95 | 20 | 7 |
| 9 | Steel Structural Frame | 103 | 30 | 8 |
| 10 | FFR Material, Walls, installation | 50 | 60 | 9 |
| 11 | Window & Door, Fit out and E&M installation, | 53 | 40 | 9 |
| 12 | Module Ready for Delivery to Hong Kong | 1 | 20 | 10,11 |

**Prompt 4:**

The result obtained by the code is as follows:

[Please replace with program code results after ]

Now I would like to ask you to answer the following questions based on the calculation results. Please give the results directly without any explanation.

1. What is the critical path length?
2. How many tasks are there on the critical path?
3. What are the tasks on the critical path?
4. When is task 7 completed?
5. What is the minimum value of resource availability during the entire scheduling process?

**Answers:**

1.

2.

3.

4.

5.