Project-2

Advanced AI

1 Optimization of Energy Constrained Multi - Robot Warehouse System

We consider a warehouse represented as a discrete $m \times n$ grid, which can be modeled as a undirected graph $G = (V, E)$, where V denotes the set of nodes and E denotes the set of undirected edges. The edge set E is defined as:

$$E \subseteq \big\{\{u, v\} \mid u, v \in V,\ u \text{ and } v \text{ are adjacent grid nodes (horizontal and vertical)}\big\},$$

and the node set V is defined as: $V = \{S\} \cup D \cup C \cup N$, where, S denotes the central storage node, D denotes the set of destination nodes (one for each item type), C denotes the set of charging nodes, and N denotes the set of internal transit nodes. All items are initially located at the central storage node S.

Let us assume that the warehouse handles three types of items: fruits, medicines, and groceries for elaboration of the problem. You need to consider k different types of items. The set of destination nodes (D) is defined as:

$$D = \{D_{\text{fruits}}, D_{\text{medicines}}, D_{\text{groceries}}\},$$

where each destination node corresponds to its respective item type. Let, I denote the set of items, and each item $i \in I$ is associated with a type $\tau(i)$, and

$$\tau(i) \in \{\text{fruits}, \text{medicines}, \text{groceries}\}.$$

An item i must be delivered to the corresponding destination node $D_{\tau(i)}$.

Let, $R = \{1, 2, \ldots, m\}$ denote the set of homogeneous robots (i.e., robots having identical carrying capacity, battery capacity, weight, and energy consumption characteristics). Time is discretized as $t = 0, 1, 2, \ldots, T$. The position of robot $r \in R$ at time t is denoted by $x_r(t) \in V$, with initial condition $x_r(0) = S \quad \forall r \in R$. At any time step t, robots may either remain at their current node or move to an adjacent node:

$$x_r(t + 1) = x_r(t) \quad \text{or} \quad \{x_r(t), x_r(t + 1)\} \in E.$$

No two robots can traverse the same edge in opposite directions simultaneously. Each robot has a maximum carrying capacity Q_r, which is measured in weight units. Let w_i denote the weight of item $i \in I$, $L_r(t)$ denote the total weight of items carried by robot r at time t, and defined as:

$$L_r(t) = \sum_{i \in I_r(t)} w_i, \text{ where } I_r(t) \text{ is the set of items carried by robot } r \text{ at time } t.$$

Then the capacity constraint of robot r at any time t is given by:

$$0 \leq L_r(t) \leq Q_r \qquad \forall r, t.$$

Pickup is allowed only at the central storage node S. For each item $i \in I$, there exists exactly one robot $r \in R$ that carries item i. Delivery of item i is completed only when the robot r carrying item i reaches the corresponding destination node $D_{\tau(i)}$. A robot may carry items belonging to different types (fruits, medicines, and groceries) simultaneously, and in such a cases, the robot must visit each corresponding destination node $D_{\tau(i)}$ for which it is carrying items, within the same delivery cycle before returning to S.

Each robot r has a maximum battery capacity B_r, and its own weight W_r. Let $b_r(t)$ denote its battery level at time t. The energy consumption function $\gamma_r(\cdot)$ depends on the travel distance, the current load carried by the robot, and the robot's own weight. So, the battery dynamics are given by:

$$b_r(t + 1) = \begin{cases} B_r & \text{if } x_r(t) = x_r(t + 1) \in C \\ b_r(t) - \gamma_r\big(d(x_r(t), x_r(t + 1)), L_r(t), W_r\big) & \text{otherwise} \end{cases} \qquad \forall r \in R, \forall t.$$

where:

- $d(x_r(t), x_r(t+1))$ denotes the distance traveled by robot r between positions $x_r(t)$ and $x_r(t+1)$
- $L_r(t)$ denotes the load carried by robot r at time t
- W_r denotes the weight of the robot r
- The energy consumption function is defined as

$$\gamma_r\big(d(x_r(t), x_r(t+1)), L_r(t), W_r\big) = \alpha\,(W_r + L_r(t))\, d(x_r(t), x_r(t+1)),$$

  where $\alpha > 0$ is a predefined constant, and is representing the energy cost per unit weight per unit distance.

The battery levels at any time step t must satisfy:

$$0 \le b_r(t) \le B_r \qquad \forall r, t.$$

A safety threshold is imposed on the battery level at each time step to prevent deep discharge.

$$b_r(t) \ge B_r^{\text{threshold}}, \quad \forall r \in R, \forall t.$$

The threshold is defined as

$$B_r^{\text{threshold}} = \beta \cdot B_r,$$

where, $\beta \in (0,1)$ is a predefined percentage representing the minimum allowable fraction of the maximum battery capacity. At most one robot can occupy a non-charging node at time t:

$$x_r(t) \ne x_{r'}(t), \quad \forall r \ne r', \forall t, \quad \text{if } x_r(t) \in V \setminus C.$$

Multiple robots are allowed to occupy a charging node simultaneously at time t:

$$x_r(t) = x_{r'}(t), \quad \forall r \ne r', \forall t, \quad \text{if } x_r(t) \in C, .$$

A delivery cycle of robot r consists of:

1. departing from S,
2. transporting assigned items to their respective destination nodes,
3. optionally visiting charging nodes,
4. returning empty to S.

Since, the number of robots is limited, each robot may perform multiple delivery cycles. Let K_r denote the number of delivery cycles executed by robot r.

Objective 1: Minimize the total delivery cycle completion time of all robots. Let, T_f denote the completion time of all delivery cycles.

$$\min T_f$$

Objective 2: Minimize the total number of delivery cycles of all robots.

$$\min \sum_{r \in R} K_r$$

Objective 3: Minimize the total energy consumption of all robots.

$$\min \sum_{r \in R} \sum_{t=0}^{T-1} \gamma_r\big(d(x_r(t), x_r(t+1)), L_r(t), W_r\big)$$

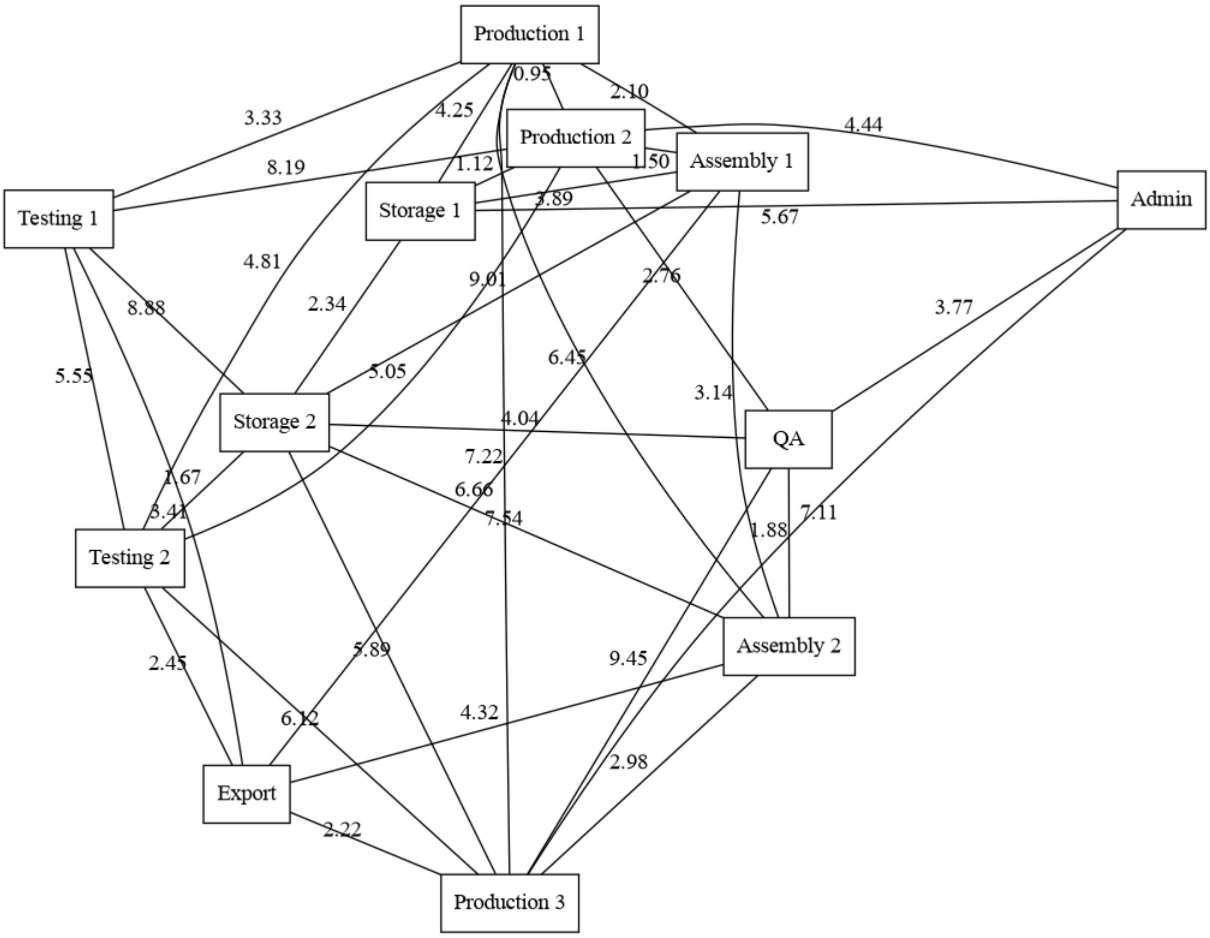
Contact person: Sanjib

2 The Automated Factory

Consider a fully automated factory populated exclusively by robots. Each robot is assigned a specific sequence of tasks to perform independently. The environment is deterministic, with no unaccounted delays.

Input:

You are given the floor plan as an undirected graph G=(V,E),(see Figure) where nodes represent locations and edges represent paths. Edge weights denote travel time (using floating-point precision). You are also provided with a Task List for each robot, detailing the sequence of locations it must visit.



Objective:

Automate the factory to maximize production efficiency by minimizing the Makespan (the time at which the last robot completes its task).

Constraints:

1. Collision Avoidance: Robots cannot traverse the same edge in opposite directions if they are meeting in the middle. Two robots will be able to occupy the same location or go on an edge in the same direction if they are separated by a specific time tolerance τ.
2. Waiting: A robot may wait at a node if doing so resolves a conflict or improves the global schedule.
3. Task: Generate a collision-free Time Sheet (Schedule) for every robot that satisfies all constraints and minimizes the total cycle time.

Contact person: Jenil

3 Garage Scheduling

As a garage owner with a background in Computer Science, you aim to optimize the daily schedule live to ensure both efficiency and mechanic well-being. Task Dynamics: Each task requires exactly 1 time unit. However, tasks possess a probabilistic element: completing a task may immediately trigger a new, mandatory sub-task (e.g., while servicing brakes, a mechanic discovers a defect in the axle).

Input:

- N types of cars, each represented by a Task Dependency Graph (DAG) where edge weights represent the probability of a task spawning a follow-up task.
- A workforce of M mechanics.

Constraints:

- Fatigue Management: To prevent burnout, a mechanic must take a mandatory break of 1 time unit after performing k consecutive tasks.
- Task Generation: After completing a task which has a probabilistic element, generate a random number between 0 and 1, if the number is higher than probability of task generation then the task is generated and add it to tasks list.

Objective:

Generate an initial optimal schedule disregarding all the probabilistic tasks and whenever a new task is added update the schedule to incorporate the new task.

Contact person: Jenil

Submission guidelines

- You need to upload a video (duration 15-20 minutes) of your project.
- You need to submit source code along with readme file.
- Keep the same group as in Project 1.
- **Deadline of submission: 10th April (Hard deadline)**