**Day 12 Decide how dynamic pricing will be calculated.**

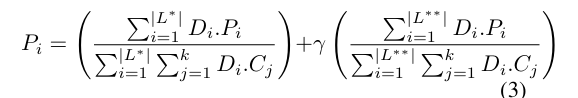
**✅ Step 1: Define Inputs**

Dynamic pricing depends on **what information you use**. From your project, we have:

1. **User’s request (θi)** → CPU, RAM, storage, etc.
2. **Deadline (t\*\_i)** → how fast they want it.
3. **Past requests (L\*, L\*\*)** → similar vs dissimilar demands.
4. **Weights (δ)** → how much importance we give to similar vs different past requests.

**✅ Step 2: Choose Pricing Formula**

We can base it on your GTRADPMFC method (weighted average):



* **L\*** → similar past requests.
* **L\*\*** → dissimilar past requests.
* **δ** → weight factor (0 → ignore dissimilar, 0.5 → give equal importance).

**✅ Step 3: Decide Algorithm**

You can decide:

* **Option 1:** Weighted Average (like GTRADPMFC).
* **Option 2:** Minimum-based (First Minimum / Second Minimum benchmarks).
* **Option 3:** Hybrid (Urgency + Weighted past demand).

Using **Weighted Dynamic Pricing (GTRADPMFC)** Option 1 because:

* It balances **history (fairness)** + **demand (profitability)**.
* You can easily adjust **δ** to test different behaviors.
* It already fits your project’s simulation setup.

**Day 13 & 14 Refer Problem statement document**

**Day 15** Implement user request generation (random CPU, memory, latency needs).

**Day 16** Implement fog nodes with limited resources.

**Day 17** Assign users randomly to fog nodes (baseline).

**Day 18** Add simple **fixed pricing** system.

Refer p1.py in code files for day 15,16,17 and p2.py for day 18