ONLINE LEARNING APPLICATIONS

Project: Dynamic Pricing with Production Constraints

PRESENTED BY:

Alperen Dagi
Nikola Dimic
Simar Ahmet Kahya
Veljko Tatalovic
Name 5

Requirements

01	Single product and stochastic environment
02	Multiple products and stochastic environment
03	Best-of-both-worlds algorithms with a single product
04	Best-of-both-worlds with multiple products
05	Slightly non-stationary environments with multiple products

Requirement 1 :
Single product and stochastic environment

Environment:

- Valuation distribution sampled from different Beta distributions.

Algorithms implemented:

- UCB1 (ignoring inventory)
- UCB1 with Inventory Constraint

Evaluation:

- Cumulative Reward
- Cumulative Regret vs $\sqrt{T}$ and T bounds

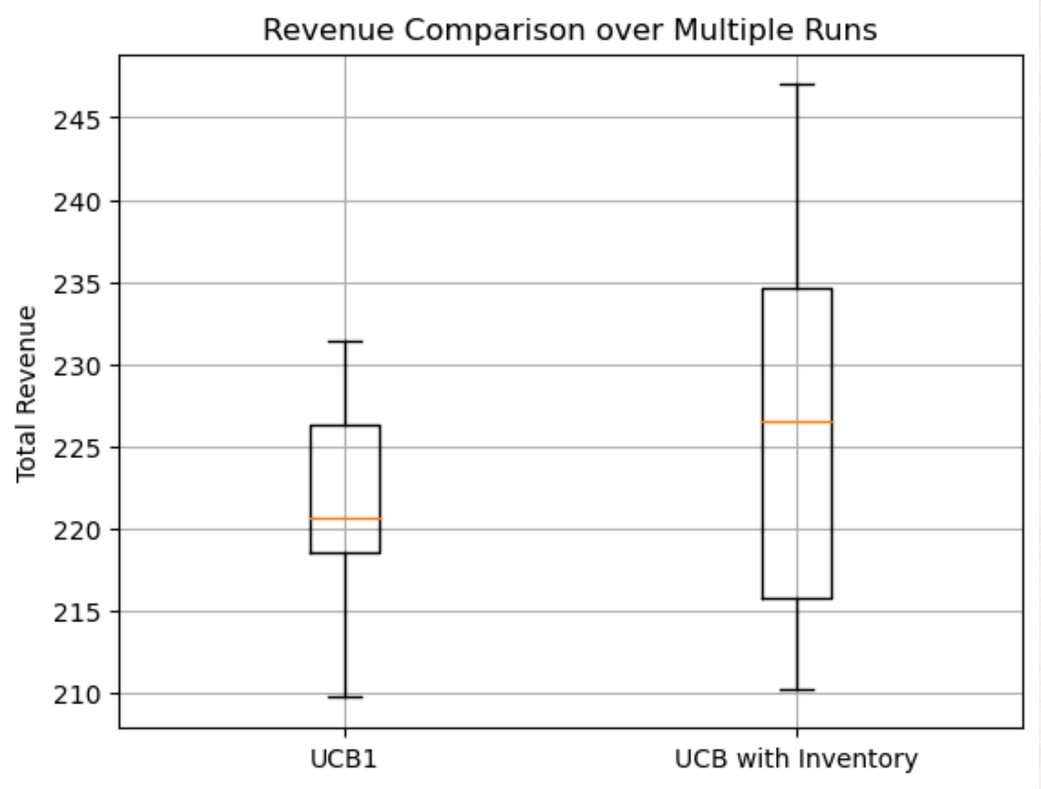
Requirement 1 :
Single product and stochastic environment

Observations:

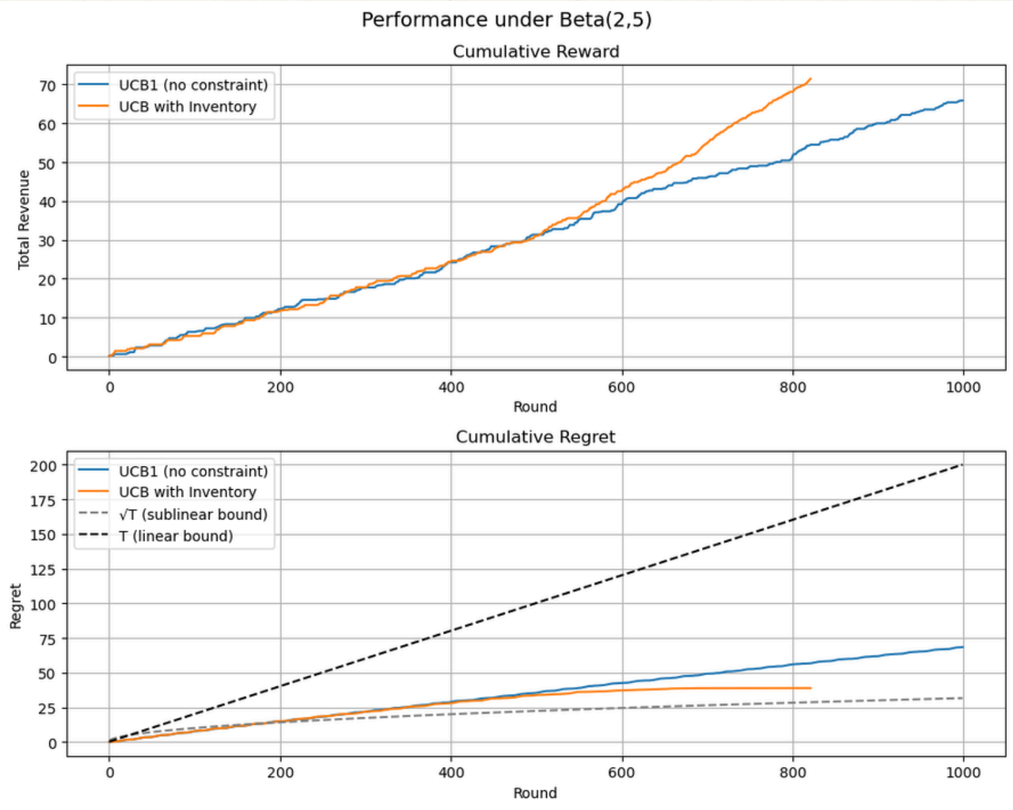
- UCB1 sometimes outperforms when the budget isn't binding.
- Inventory-aware UCB1 is safer and more robust when the budget is tight.

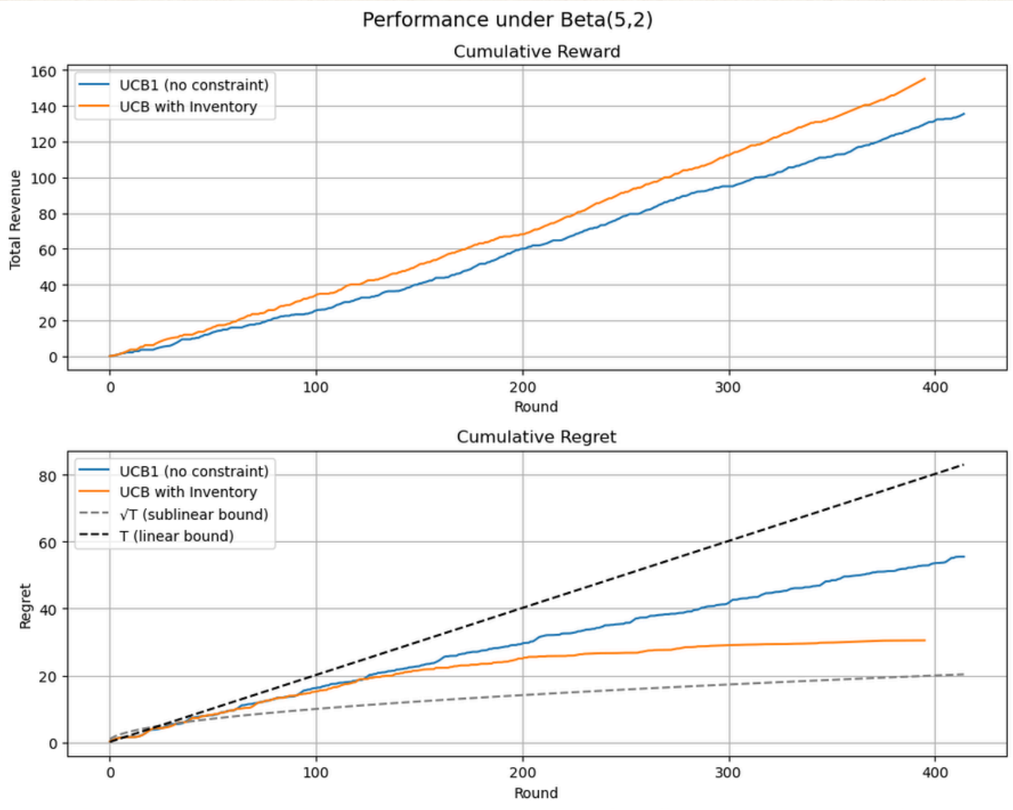
Settings:

- Budget **B = 1000**
- Horizon **T = 10000**
- Prices: **{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0}**
- Simulations: **20**



Requirement 1 :
Single product and stochastic environment





Requirement 2 :
Multiple products and stochastic environment

Environment:

- Heterogeneous valuation distributions per product.

Algorithms implemented:

- Combinatorial-UCB with inventory constraint.

Evaluation:

- Cumulative Reward
- Cumulative Regret vs Oracle
- Price Histograms

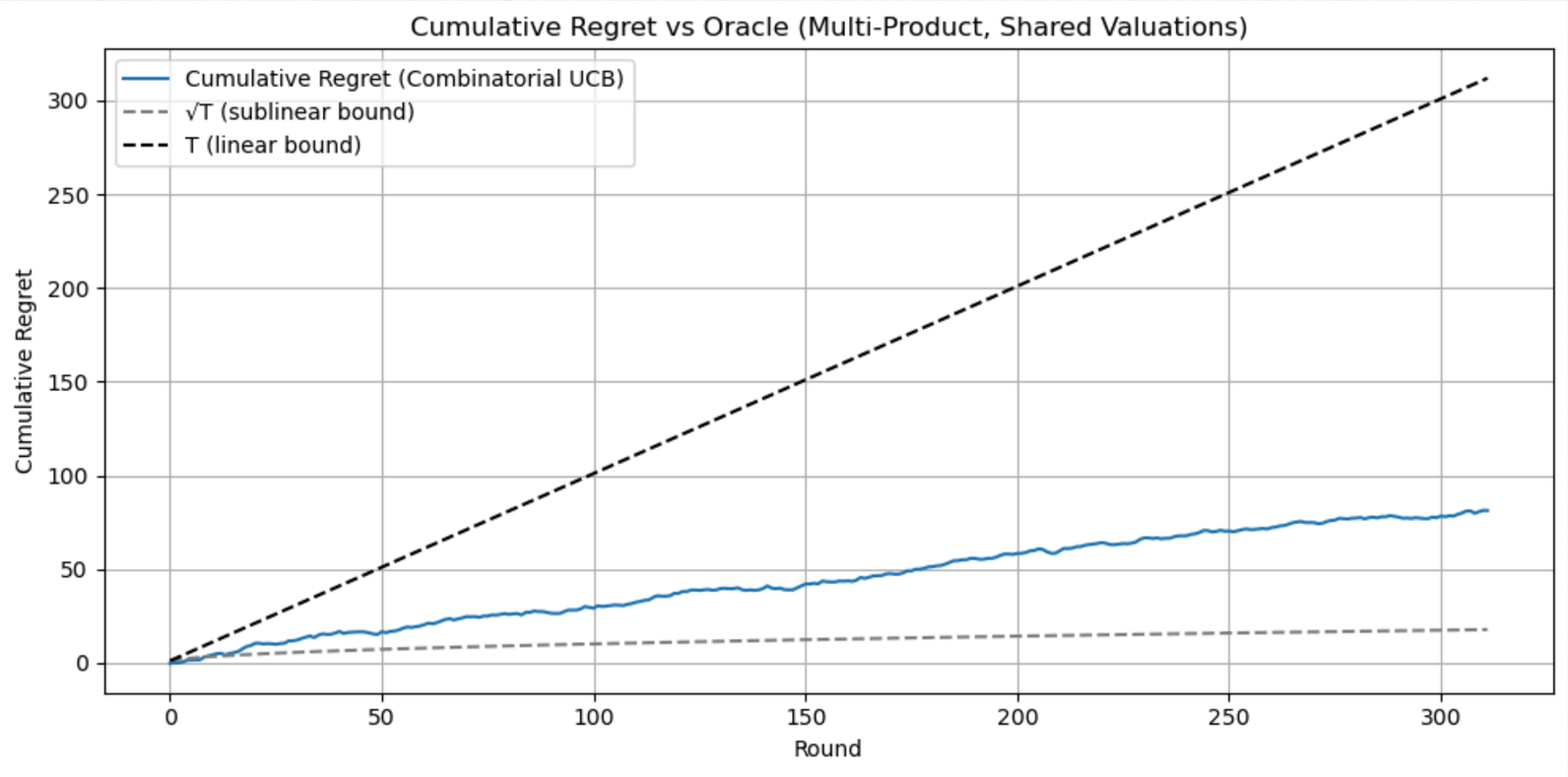
Observations:

- Combinatorial-UCB adapts independently per product.
- Doesn’t outperform the Greedy baseline in total revenue. ***

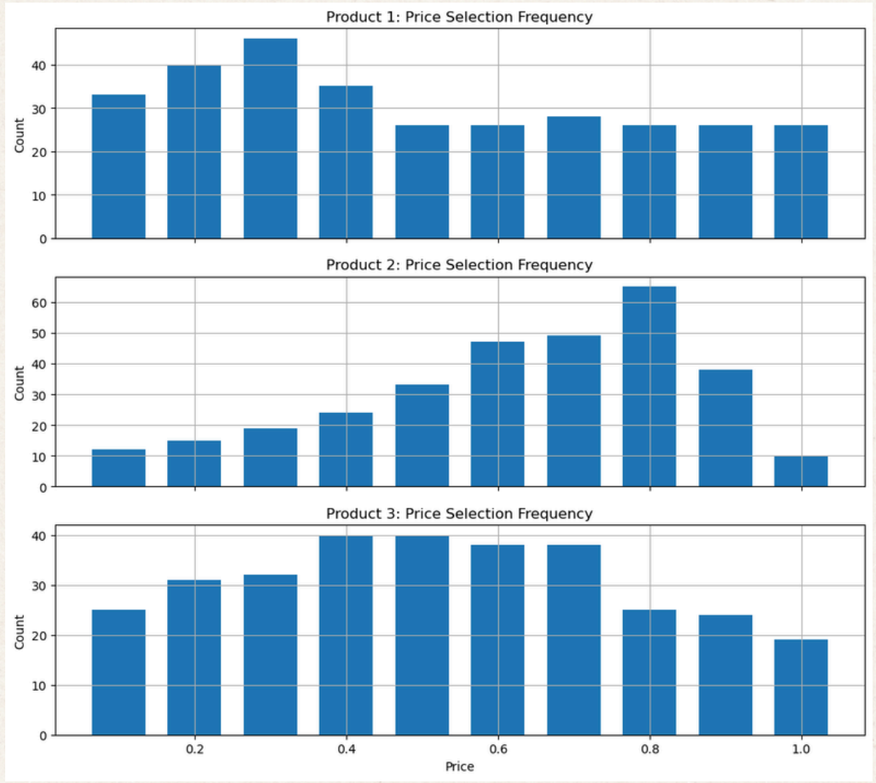
Settings:

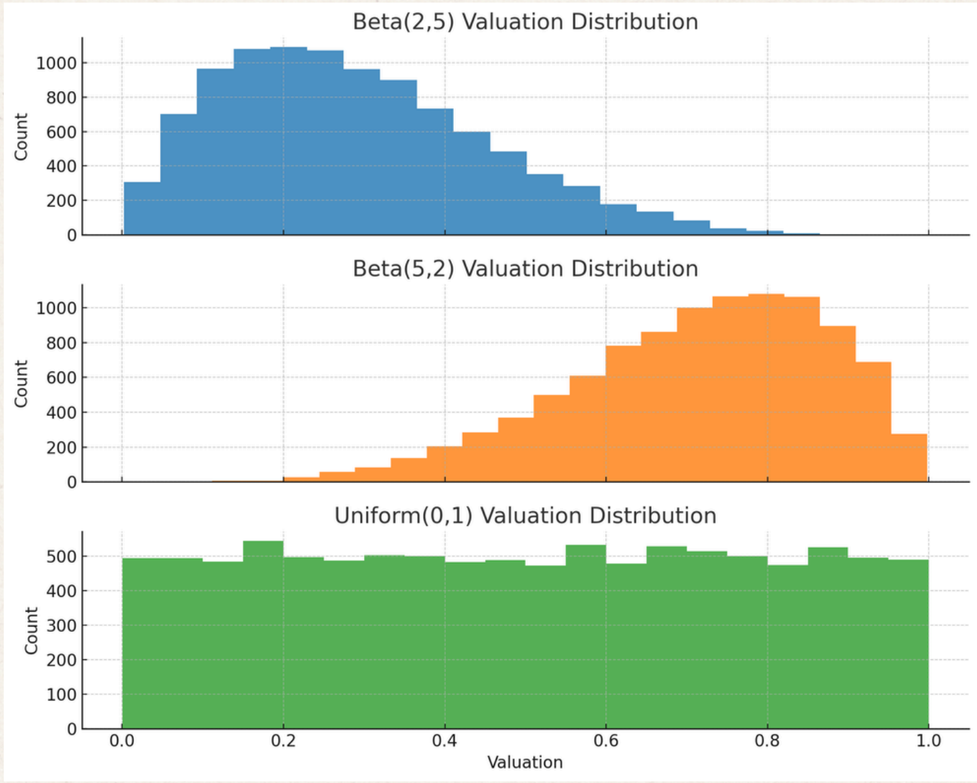
- Budget **B = 500**
- Horizon **T = 1000**
- Prices: **{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0}**

Requirement 2 :
Multiple products and stochastic environment

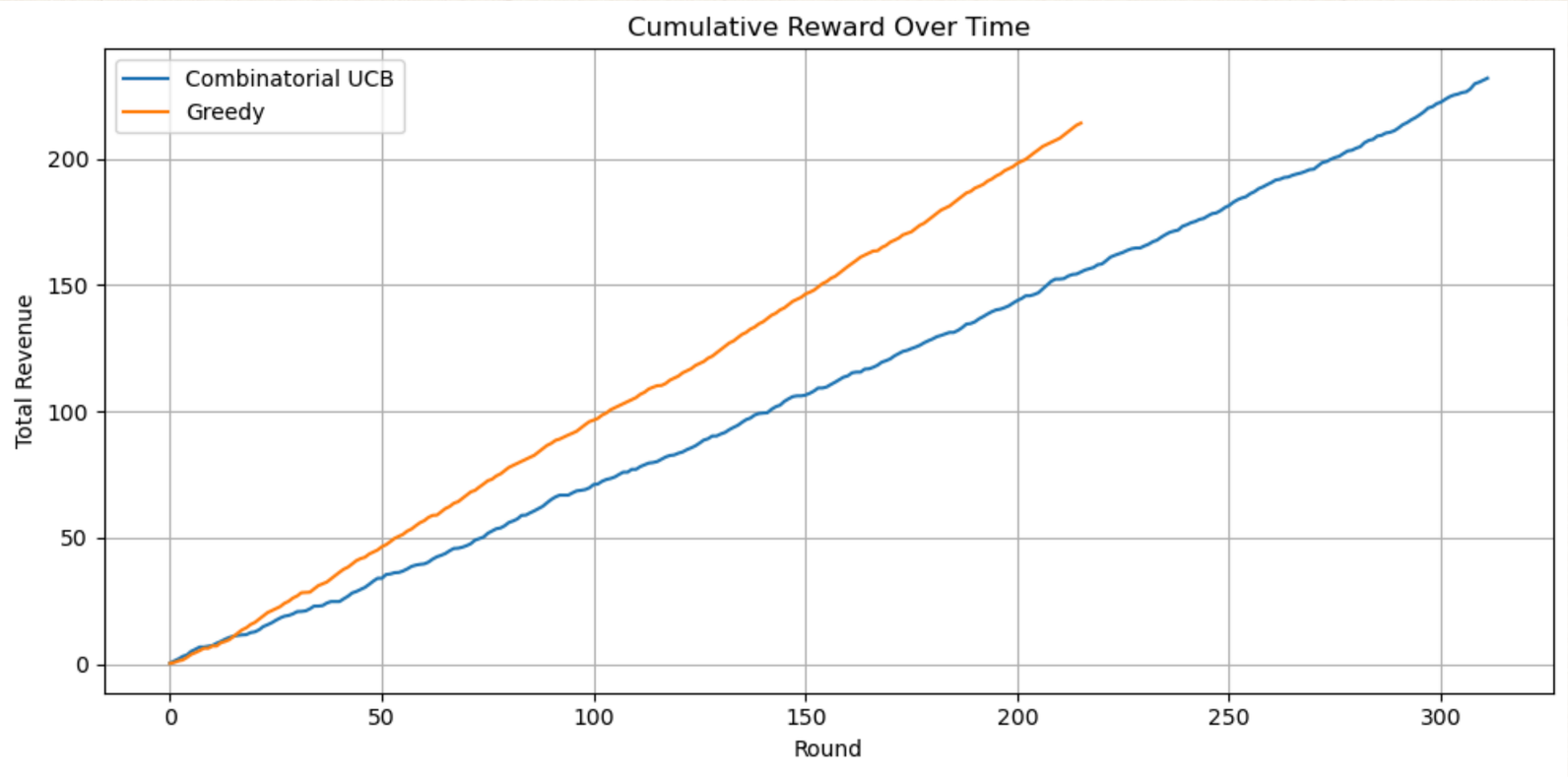


Requirement 2 :
Multiple products and stochastic environment





Requirement 2 :
Multiple products and stochastic environment



Requirement 3:
Best-of-Both-Worlds and single product

Environment:

- Highly non-stationary valuation distribution.

Algorithms implemented:

- Primal-Dual pricing strategy.

Method:

- Dual variable λ updated as:

$$\lambda_{t+1} = \left[\lambda_t + \eta \cdot \left(x_t - \frac{B}{T}\right)\right]^+$$

- Prices chosen as:

$$p_t = \arg\max_{p \in P} (p - \lambda_t) \cdot \mathbb{I}(v_t \geq p)$$

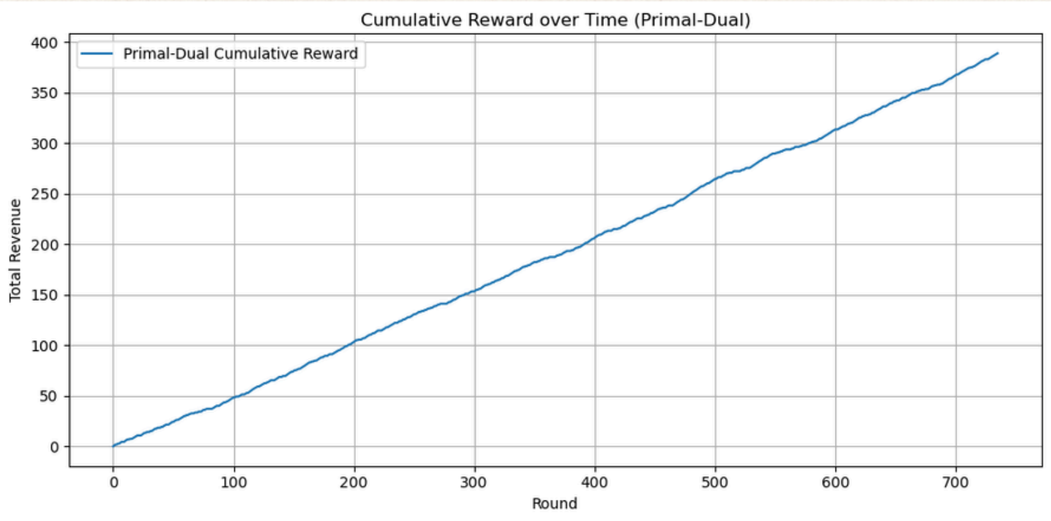
Observations:

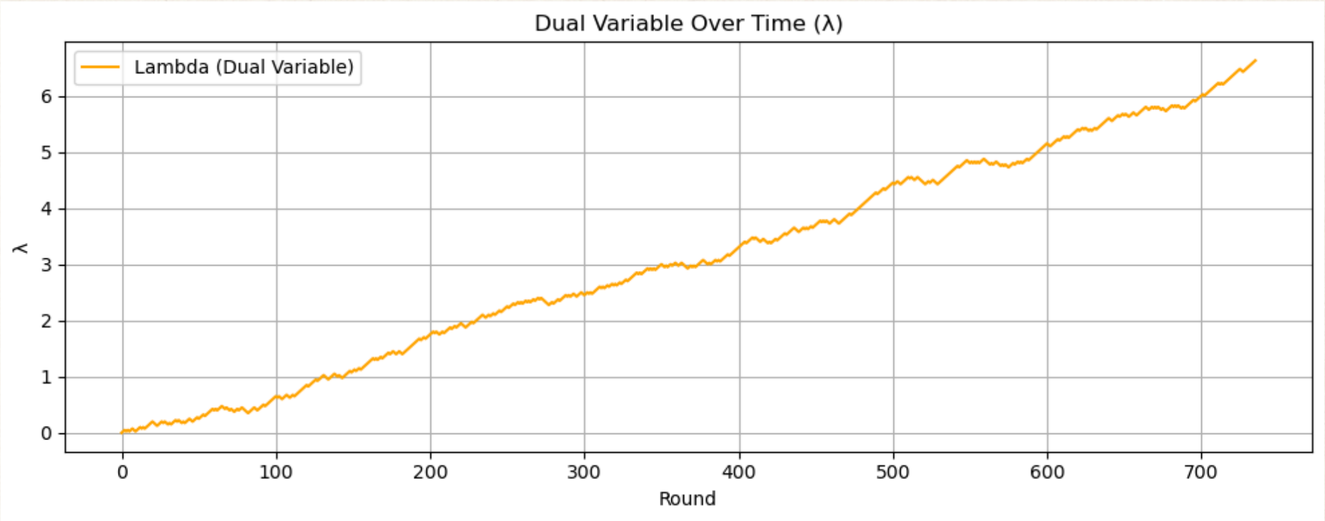
- Primal-Dual adapts to non-stationary shifts.
- λ stabilizes over time, controlling inventory usage.

Settings:

- Budget **B = 500**
- Horizon **T = 1000**
- Prices: **{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0}**

Requirement 3:
Best-of-Both-Worlds and single product





Requirement 4:
Best-of-Both-Worlds and multiple products

Environment:

- Highly non-stationary valuation distribution.
- Joint distribution over valuations.

Algorithms implemented:

- Primal-Dual method with inventory constraint.
- Separate regret minimiser per product.

Evaluation:

- Cumulative Regret Comparison.
- Cumulative Revenue Comparison.

Settings:

- Budget **B = 8000**
- Horizon **T = 2000**
- Prices: **{0.1 , 0.23, 0.36, 0.5, 0.63, 0.76, 0.9 }**

Requirement 5: Slightly non-stationary environments with multiple products

Environment:

- Highly non-stationary valuation distribution.
- Joint distribution over valuations.

Algorithms implemented:

- Combinatorial-UCB with sliding window.

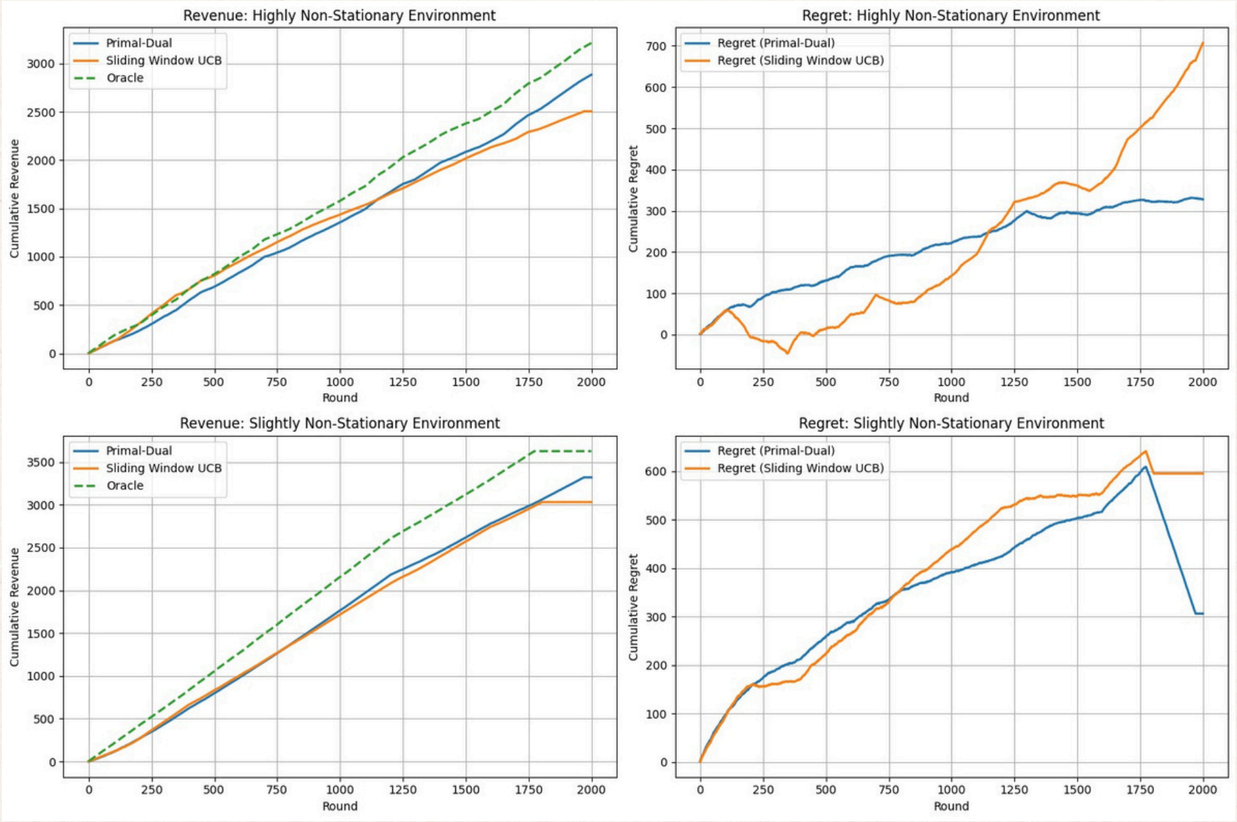
Evaluation:

- Cumulative Regret Comparison.
- Cumulative Revenue Comparison.

Settings:

- Budget **B = 8000**
- Horizon **T = 2000**
- Prices: **{0.1 , 0.23, 0.36, 0.5, 0.63, 0.76, 0.9 }**

Requirement 4/5 results:



Thank you for your attention!