Online Learning Application Project Presentation

Gottschling Daniel - 11123625

Floris Fabio Marco - 10811227

Parenti Carolina – 10797066

Roberto Sonzini Gobbi - 10794845

Singh Karanbir - 10865124

Overview

Pricing problem with multiple products and inventory constraint

At each round $t \in T$:

- The company chooses which types of product to sell and set price pi for each type of product
- A buyer with a valuation for each type of product arrives
- The buyer buys a unit of each product with price smaller than the product valuation

Requirement 1.1
Single product stochastic environment with no inventory constraint

AGENT
UCB1 agent

PARAMETERS
Time horizon: T = 10000
Prices: pi ∈ [0,1]

VALUATION DISTRIBUTION
At each round the valuation is sampled from: $X \sim \mathcal{N}(0{,}5,\ 1)$

BASELINE
Price chosen by solving the following linear programming:

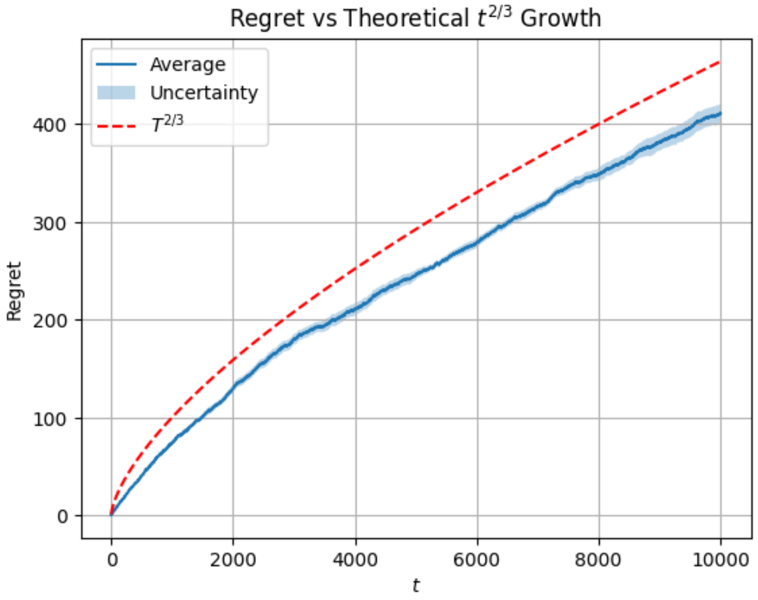
$$p_j = 1 - F(v_j)$$

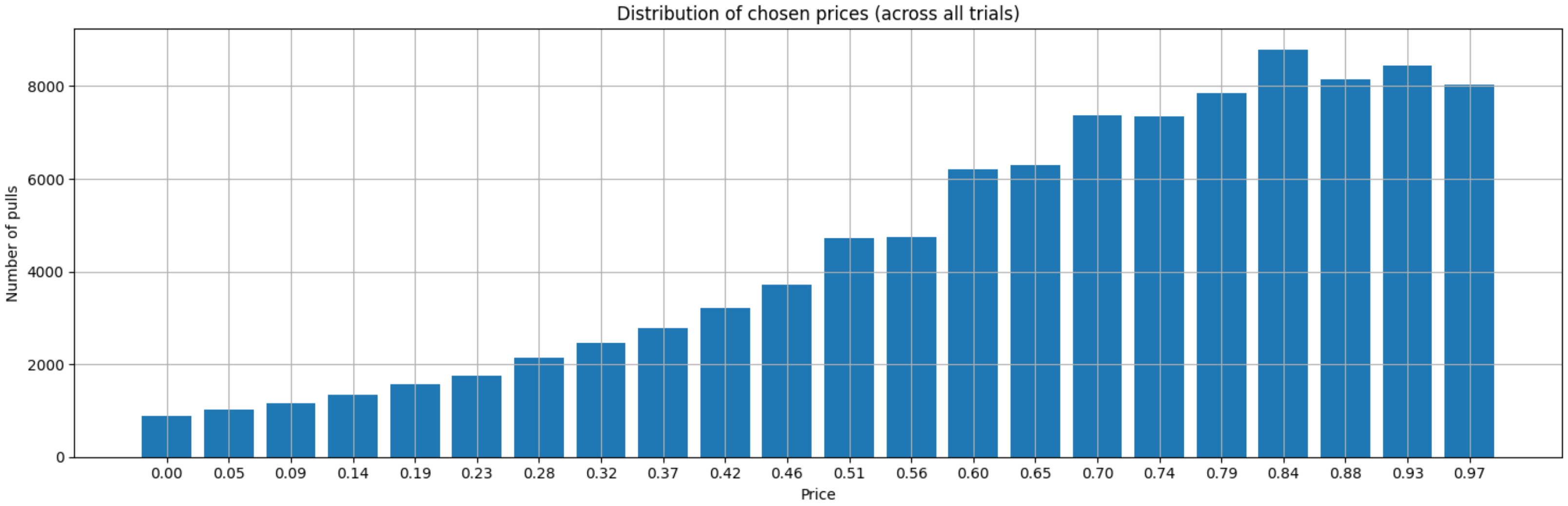
$$\max_{x_j \in [0,1]} \sum_{j=1}^{K} v_j \cdot p_j \cdot x_j$$

$$\sum_{j=1}^{K} x_j = 1$$

Requirement 1.1

Single product stochastic environment with no inventory constraint





Requirement 1.2

Single product stochastic environment with inventory constraint

PARAMETERS

- Time horizon: T = 10000
- Prices: pi ∈ [0,1]
- Budget: 45% T

VALUATION DISTRIBUTION

At each round the valuation is sampled from

$$X \sim \mathrm{Beta}(1,1)$$

AGENT

Pulls the arm that solve the following LP:

$$\max_{x_j \in [0,1]} \sum_{j=1}^{P} \mathrm{UCB}_j \cdot x_j$$

$$\sum_{j=1}^{P} \mathrm{LCB}_j \cdot x_j \leq \rho$$

$$\sum_{j=1}^{P} x_j = 1$$

BASELINE

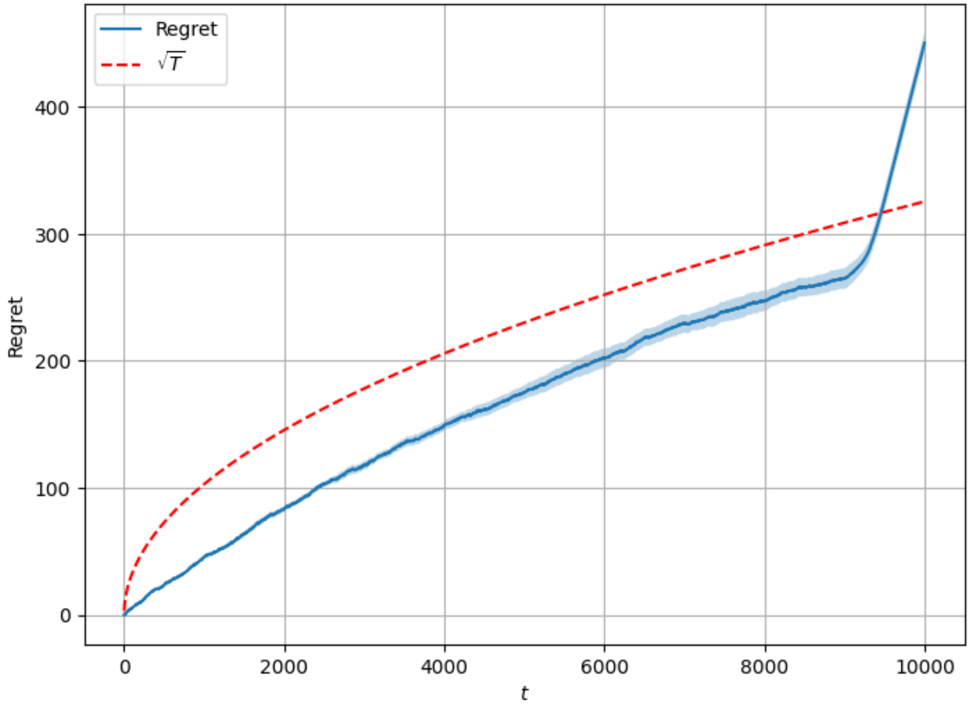
Price chosen by following the following LP:

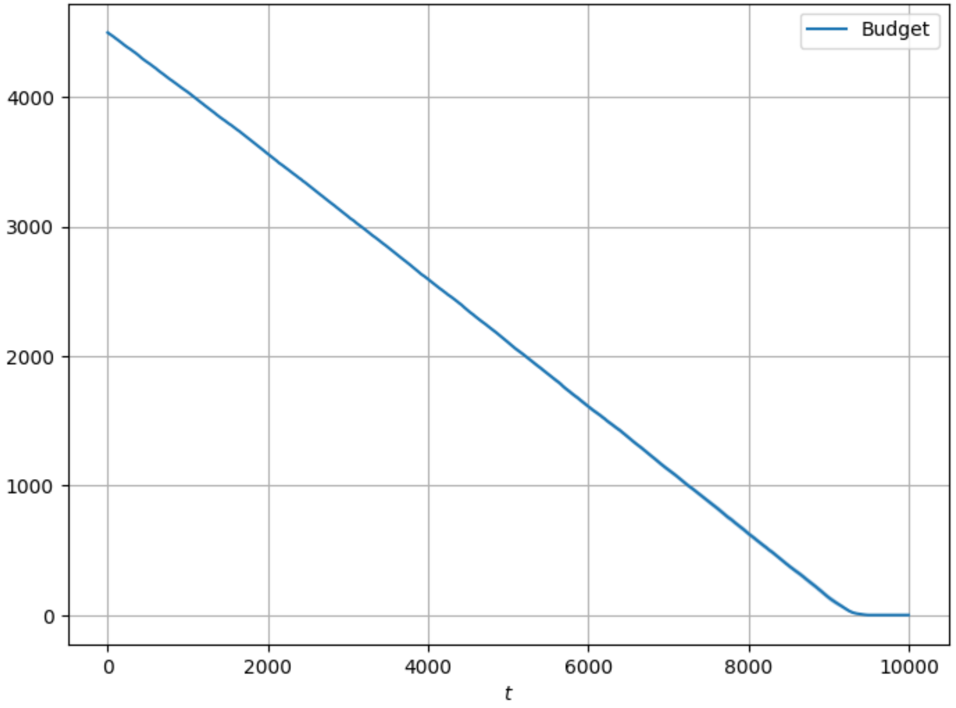
$$\max_{x_j \in [0,1]} \sum_{j=1}^{K} v_j \cdot p_j \cdot x_j$$

$$\sum_{j=1}^{K} x_j = 1$$

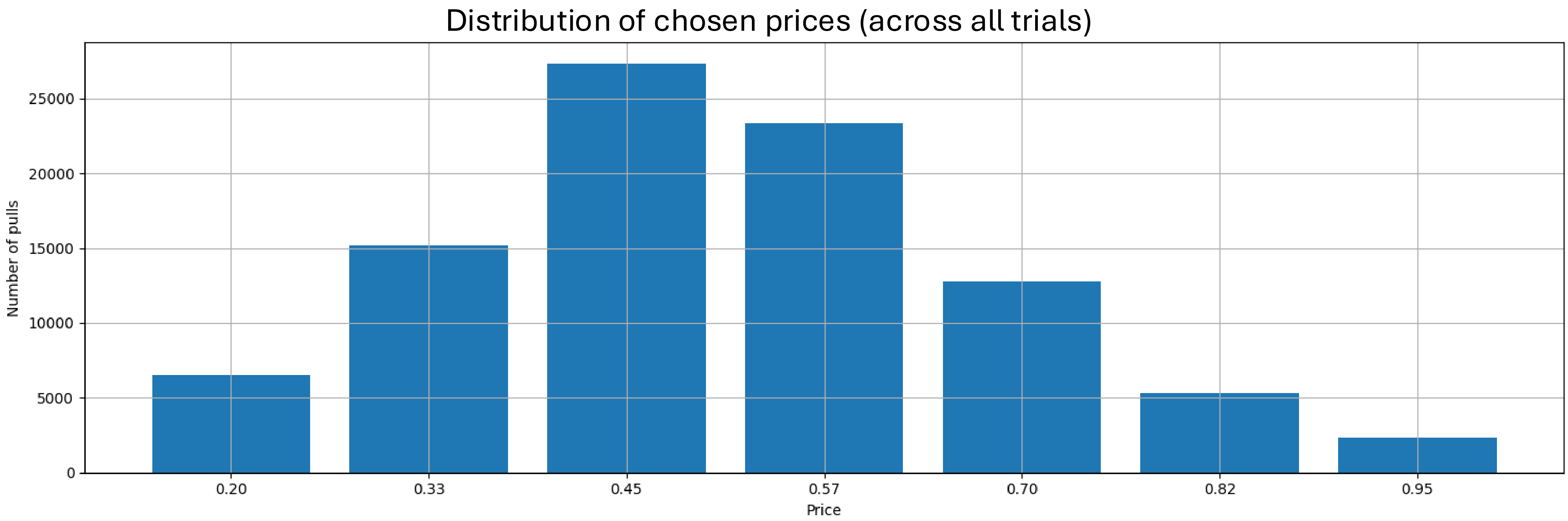
$$\sum_{j=1}^{K} p_j \cdot x_j \leq \rho$$

Requirement 1.2
Single product stochastic environment with inventory constraint





Requirement 1.2
Single product stochastic environment with inventory constraint



Requirement 2

Multiple product stochastic environment with inventory constraint

AGENT

- Combinatorial- UCB
- To extend the inventory constraint we modified the standard Combinatorial-UCB with UCB-like

PARAMETERS

- Time horizon: T = 10000
- Prices: pi ∈ [0,1]
- Budget: 75% T
- Number of products: N=3

BASELINE

- Price chosen by solving the following LP

$$\max_{x_{ij}\in[0,1]} \sum_{i=1}^{N}\sum_{j=1}^{P} v_j \cdot p_{ij} \cdot x_{ij}$$

$$\sum_{j=1}^{P} x_{ij} = 1 \quad \text{per } i = 1, \dots, N$$

$$\sum_{i=1}^{N}\sum_{j=1}^{P} p_{ij} \cdot x_{ij} \le \rho$$

VALUATION DISTRIBUTION

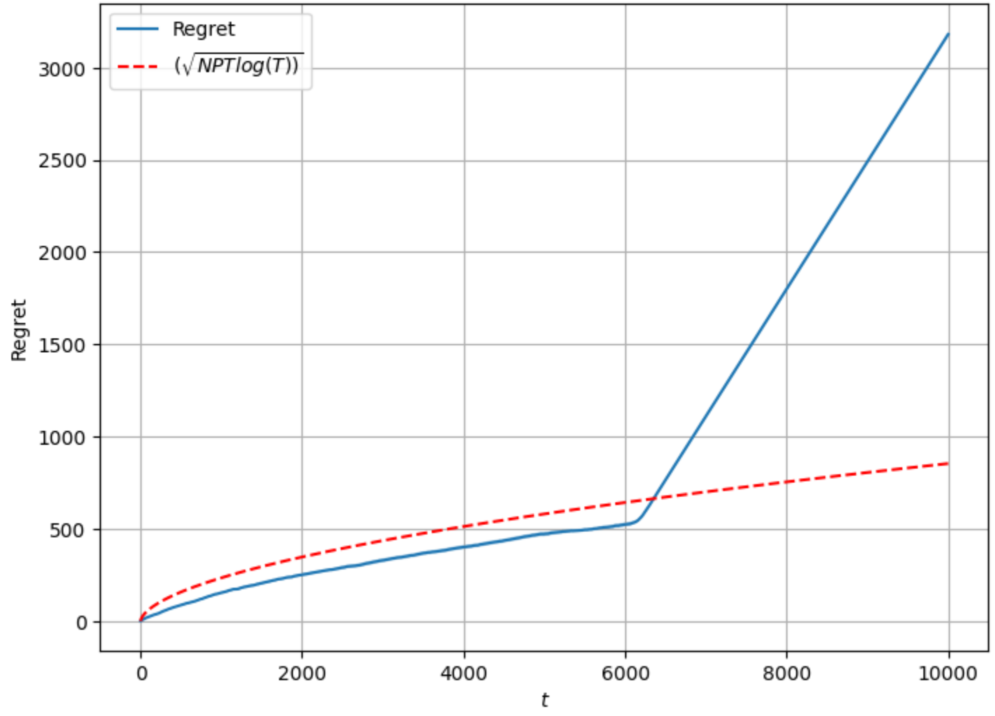
At each round the valuation is sampled from

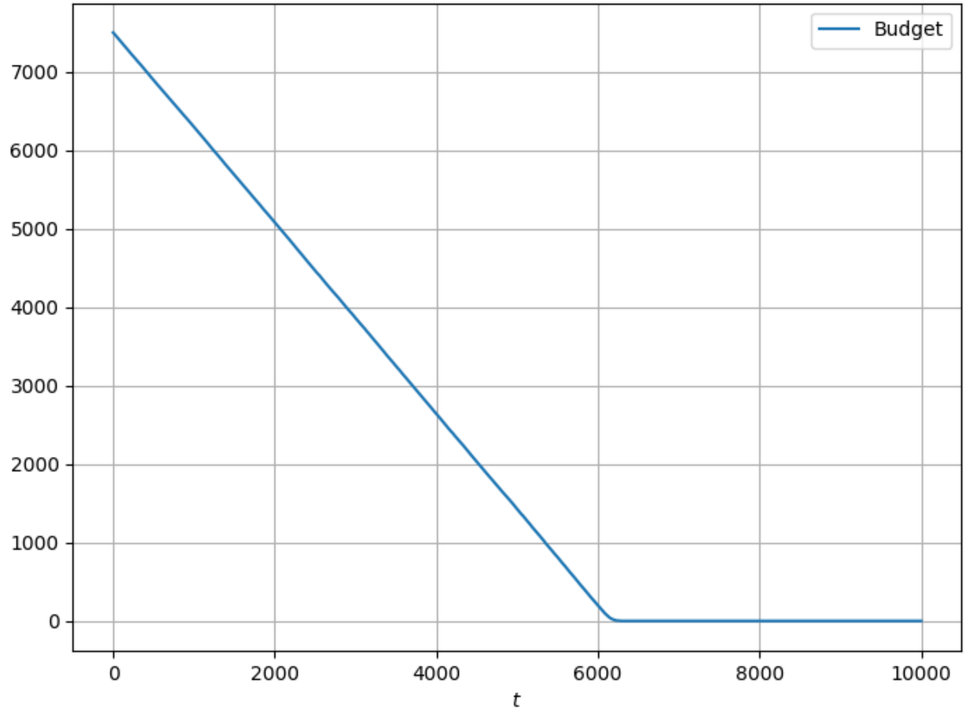
$$X \sim \mathcal{N}(0.5, 1^2)$$

$$Y \sim \mathrm{Beta}(3, 2)$$

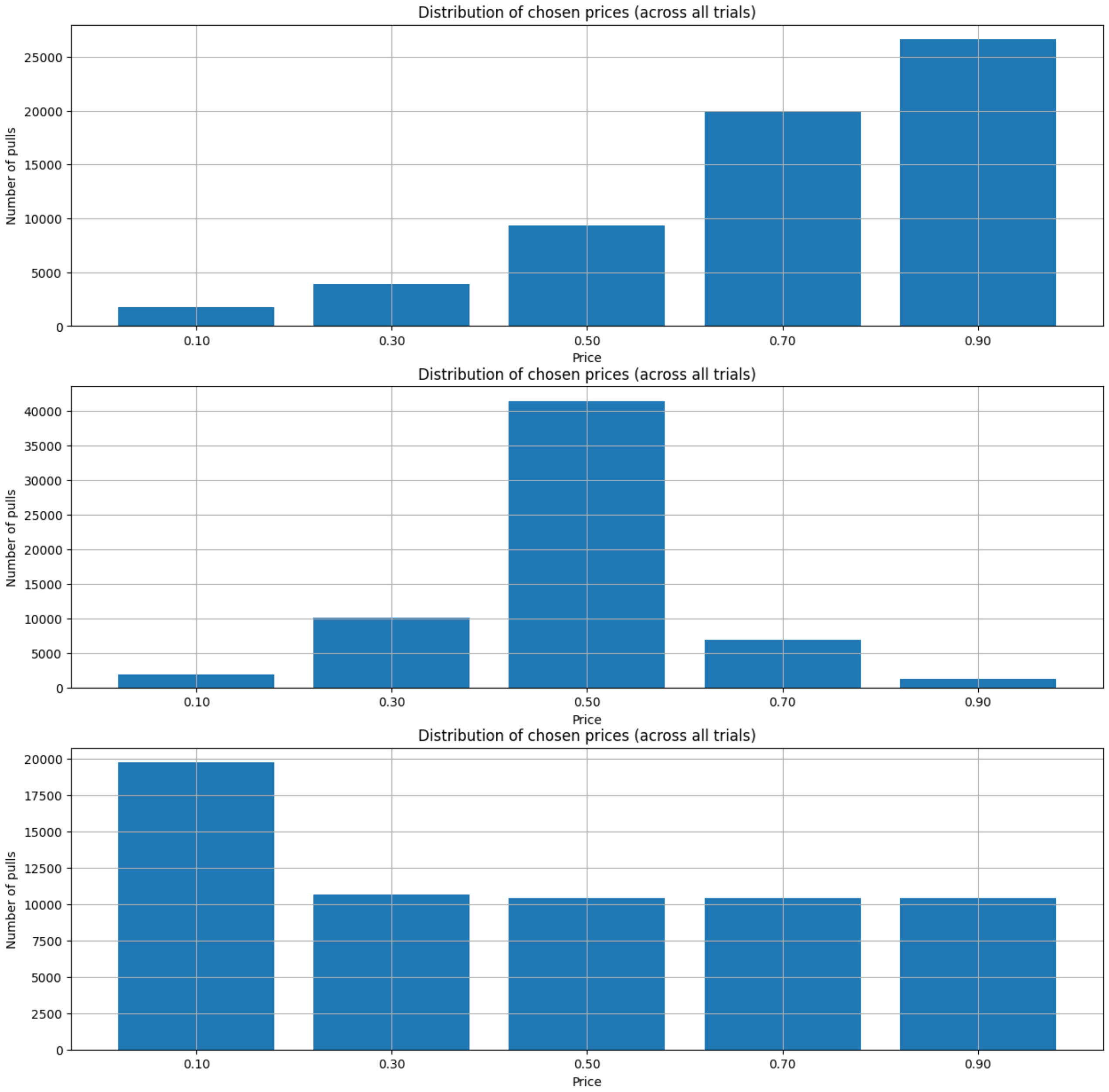
$$Z \sim \mathrm{Beta}(2, 20)$$

Requirement 2
Multiple product stochastic environment with inventory constraint





Requirement 2 Multiple product stochastic environment Inventory constraint



Requirement 3

Best-of-both-worlds algorithms with a single product with inventory constraint

AGENT

- Multiplicative Pacing
- EXP3 as regret minimizer

BASELINE

- Price chosen by solving the following LP

$$\bar{F}(v_j) = \frac{1}{w} \sum_{t'=t-w+1}^{t} F_{t'}(v_j)$$

$$p_j = 1 - \bar{F}(v_j)$$

$$\min_{x \in [0,1]^K} - \sum_{j=1}^{K} v_j \cdot p_j \cdot x_j$$

$$\sum_{j=1}^{K} x_j = 1$$

$$\sum_{j=1}^{K} p_j \cdot x_j \leq \rho$$

PARAMETERS

- Time horizon: T = 10000
- Prices: pi ∈ [0,1]
- Budget: 50% T
- Learning rate EXP3: $\sqrt{\frac{\log K}{KT}}$
- Learning rate Multiplicative pacing: $\frac{1}{\sqrt{T}}$

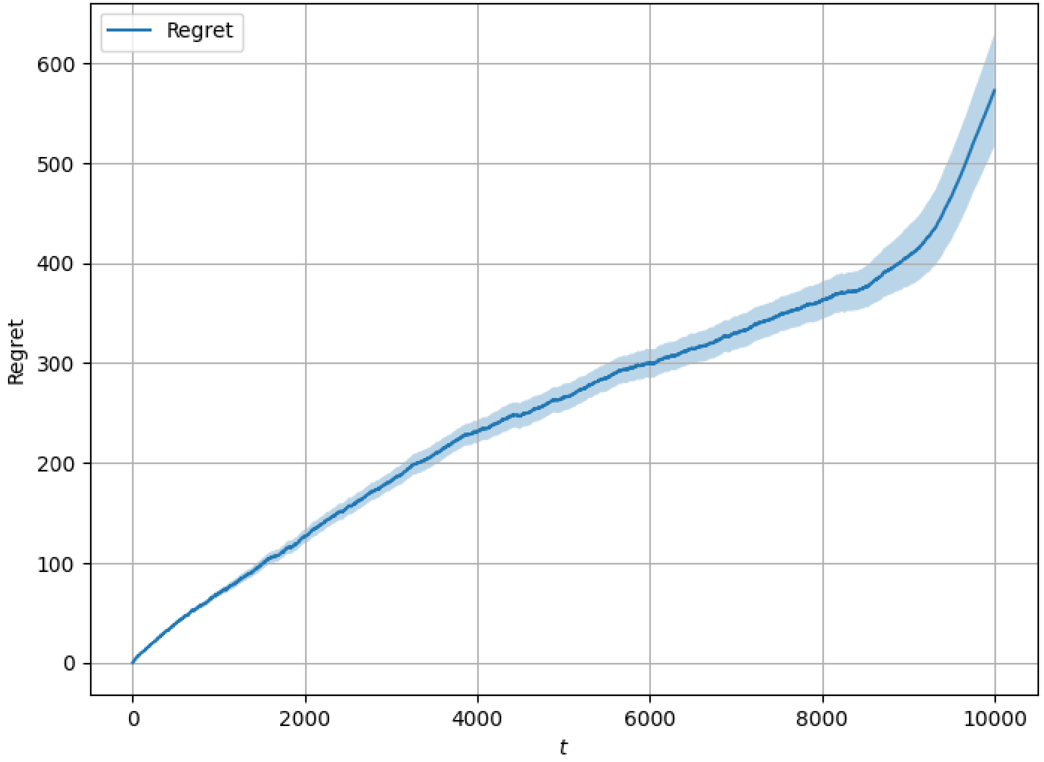
VALUATION DISTRIBUTION

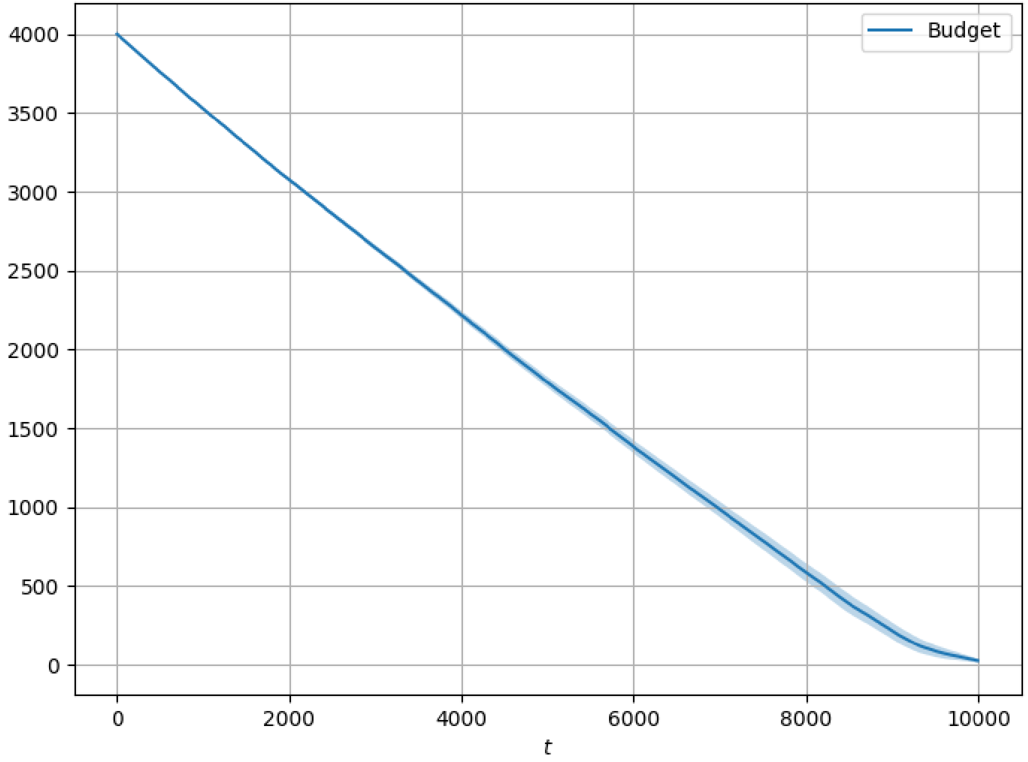
At each round the valuation is sampled from

Adversarial	*Stationary*
$X \sim \mathrm{Beta}(a, b)$	$X \sim \mathcal{N}(0{,}5,\ 1)$

Requirement 3
Best-of-both-worlds algorithms with a single product with inventory constraint

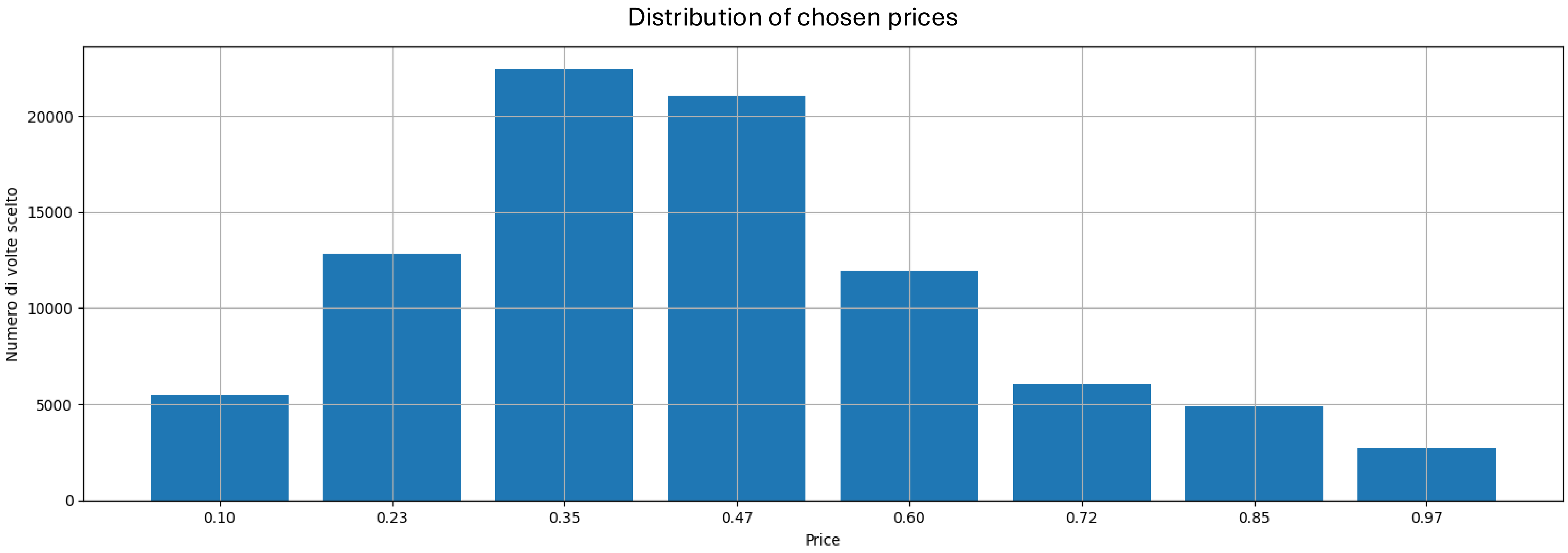
ADVERSARIAL ENVIRONMENT





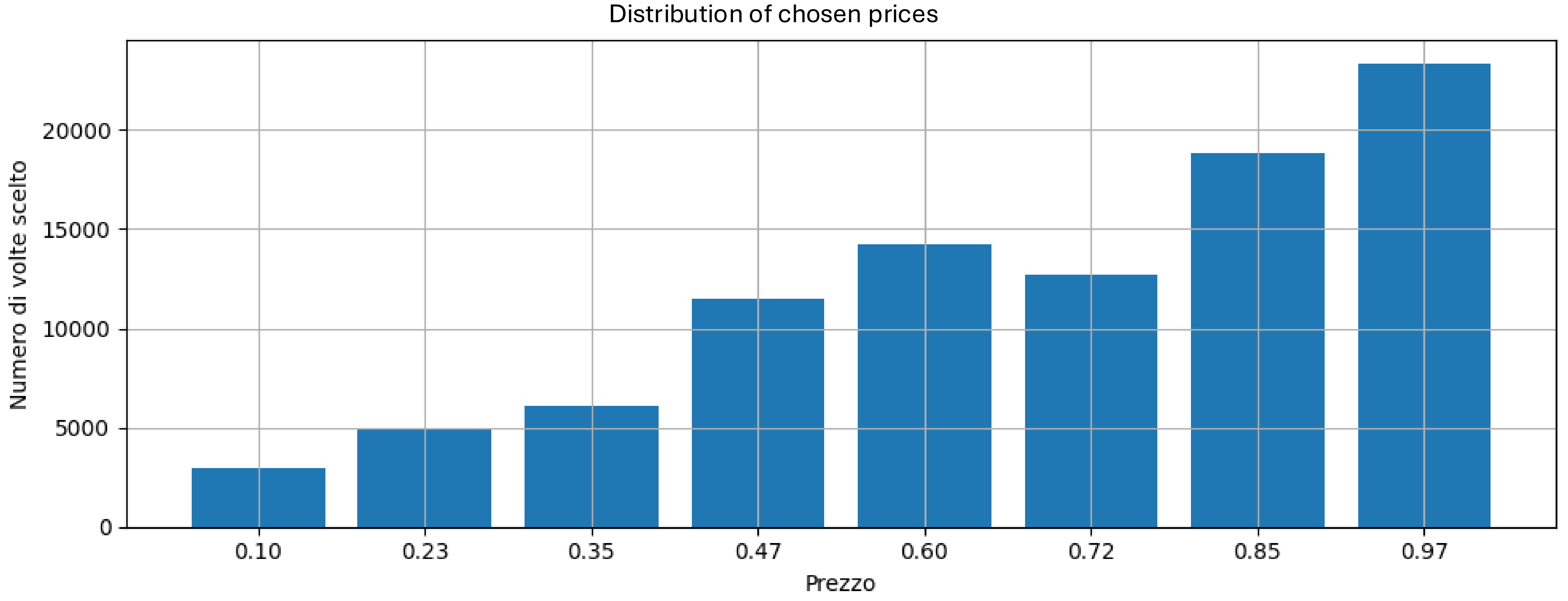
Requirement 3
Best-of-both-worlds algorithms with a single product with inventory constraint

ADVERSARIAL ENVIRONMENT

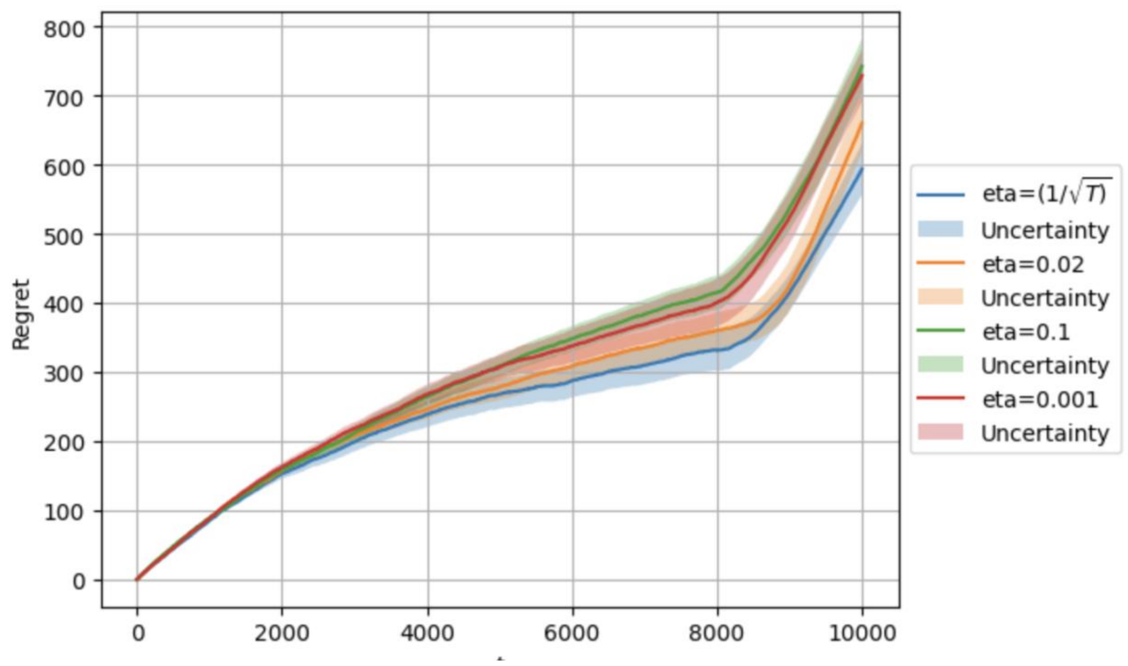


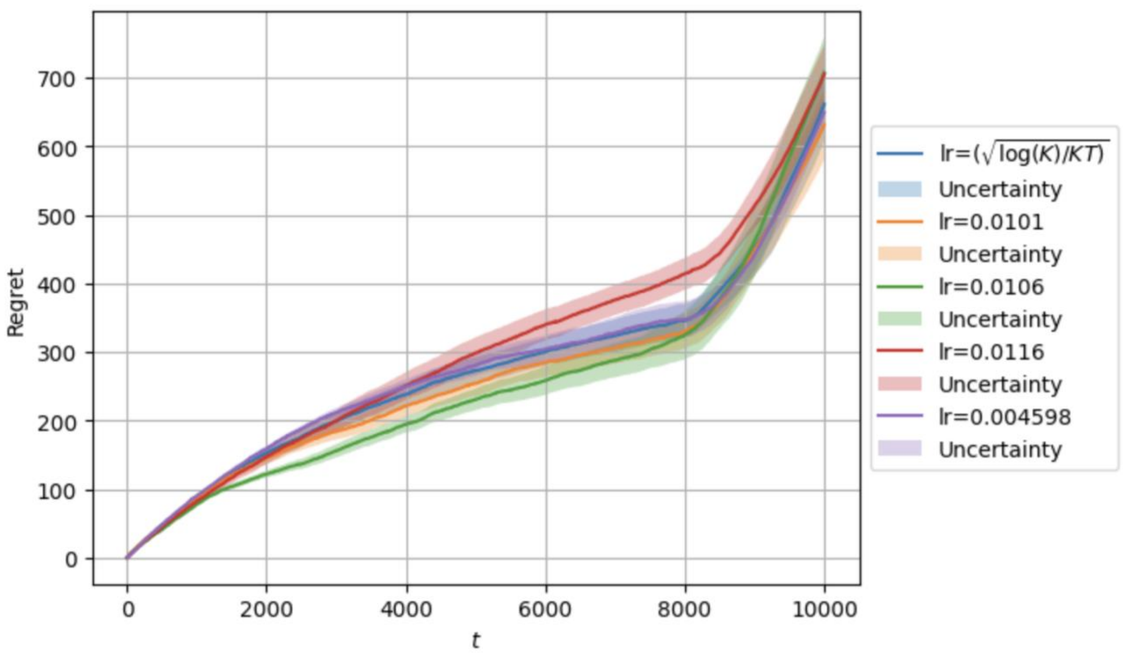
Requirement 3
Best-of-both-worlds algorithms with a single product with inventory constraint

STATIONARY ENVIRONMENT



Requirement 3
Best-of-both-worlds algorithms with a single product with inventory constraint





Tests on EXP3 parameters

Requirement 4
Best-of-both-worlds with multiple products with inventory constraint

PARAMETERS

- Time horizon: T = 10000
- Prices: pi ∈ [0,1]
- Budget: 75% T
- Number of products: N=3

STATIONARY ENVIRONMENT

At each round the valuation is sampled from

$$X \sim \mathcal{N}(0.5, 1^2)$$
$$Y \sim \mathrm{Beta}(3, 2)$$
$$Z \sim \mathrm{Beta}(2, 20)$$

HIGHLY NON-STATIONARY ENVIRONMENT

A sequence of correlated valuations for each type of product that changes quickly over time

At each round the valuation is sampled from

$$\mu \sim \mathcal{U}[0.4, 1.0]^N$$
$$A \sim \mathrm{Uniform}[0, 1]^{N \times N}$$
$$\Sigma = AA^\top + 0.05 \cdot I_N$$
$$X \sim \mathcal{N}(\mu, \Sigma)$$

Requirement 4
Best-of-both-worlds with multiple products with inventory constraint

AGENT

- Multiplicative pacing agent for multiple product
- A different EXP3 agent for each product is used as a regret minimizer

BASELINE

At every round the chosen price is the one that maximises the following LP:

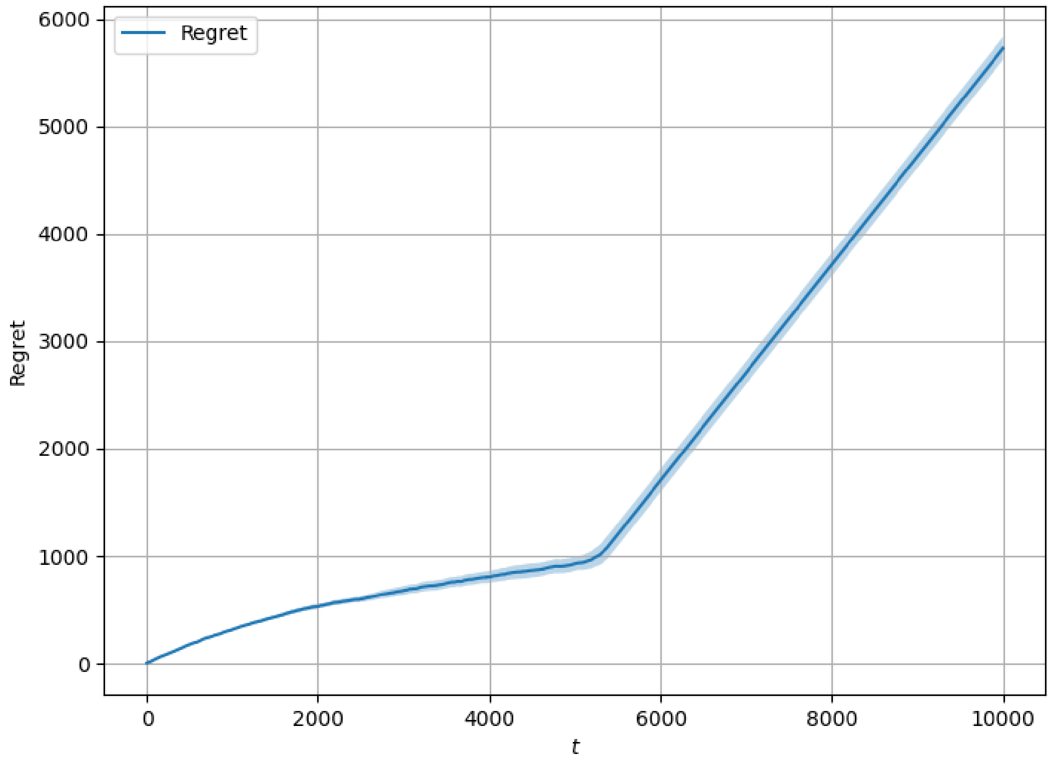
$$\max_{x_{ij}\in[0,1]} \sum_{i=1}^{N}\sum_{j=1}^{P} v_j \cdot \bar{p}_j \cdot x_{ij}$$

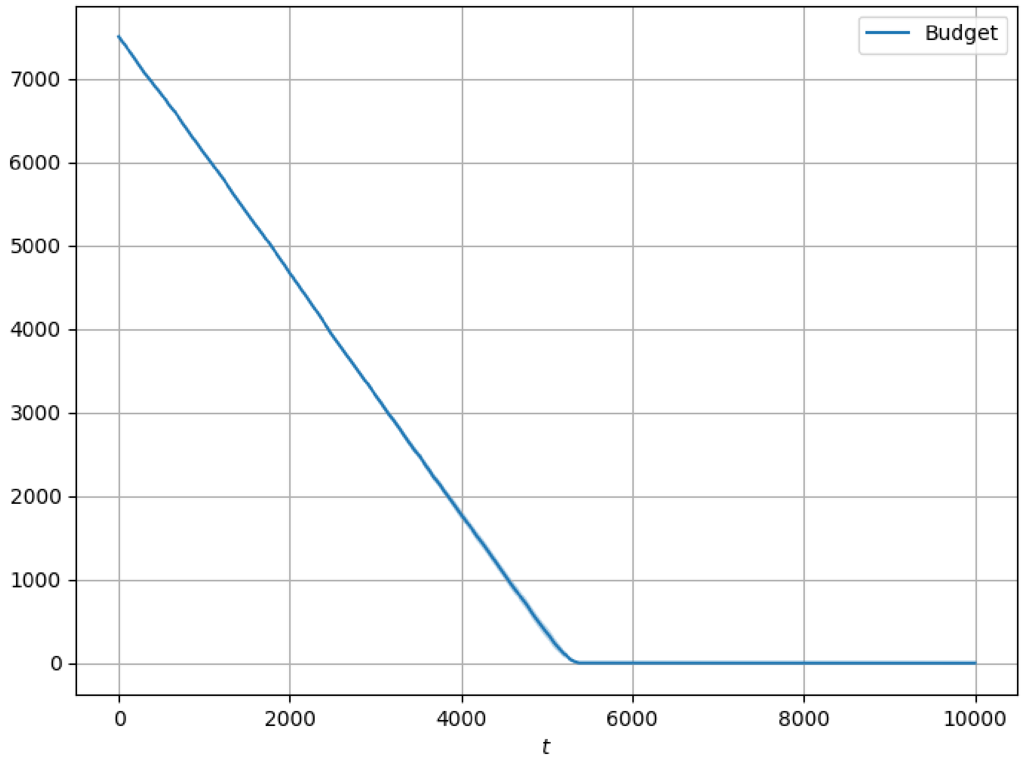
$$\sum_{j=1}^{P} x_{ij} = 1 \quad \forall i = 1,\ldots,N$$

$$\sum_{i=1}^{N}\sum_{j=1}^{P} \bar{p}_j \cdot x_{ij} \leq \rho$$

Requirement 4
Best-of-both-worlds with multiple products with inventory constraint

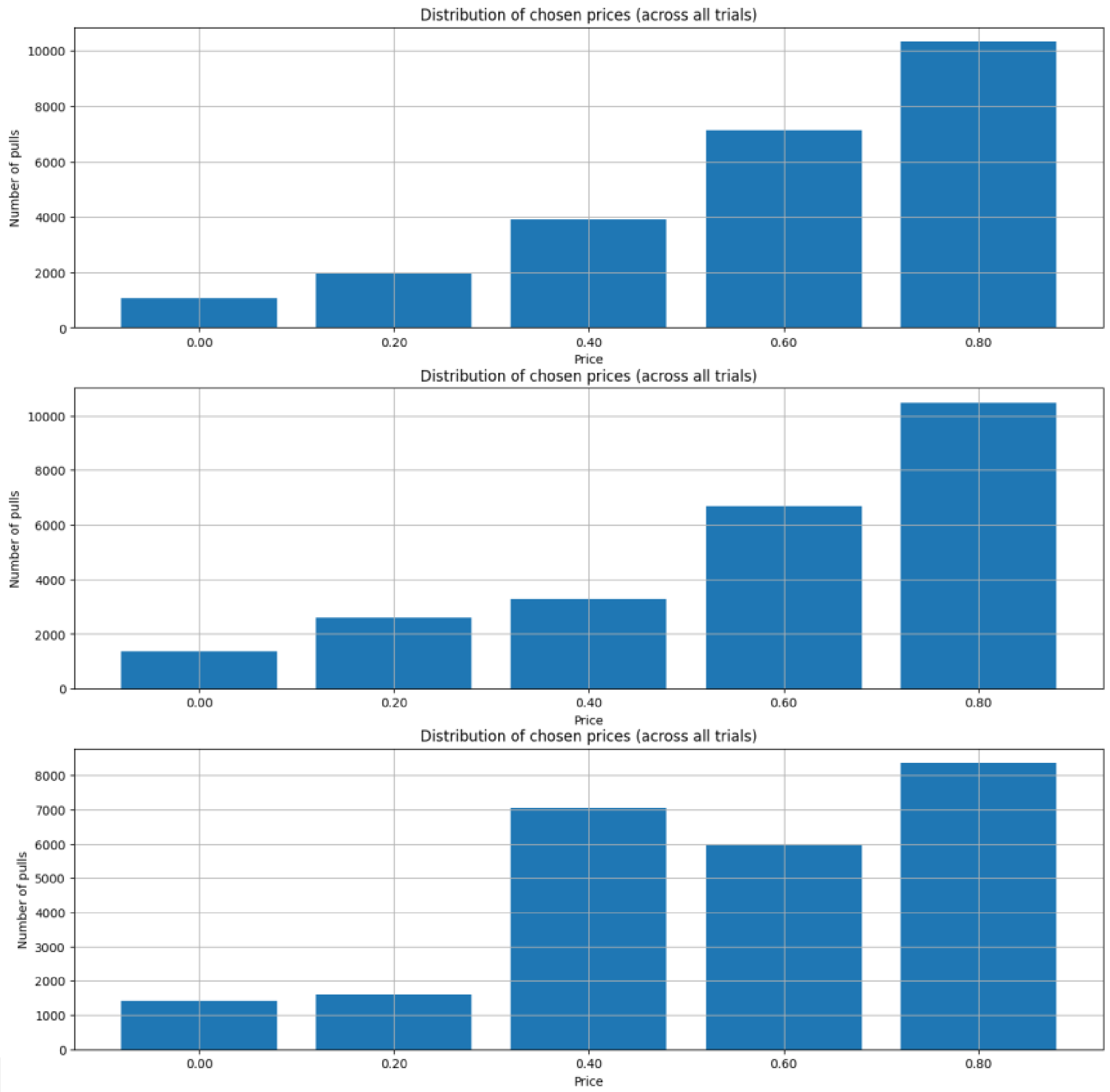
ADVERSARIAL ENVIRONMENT





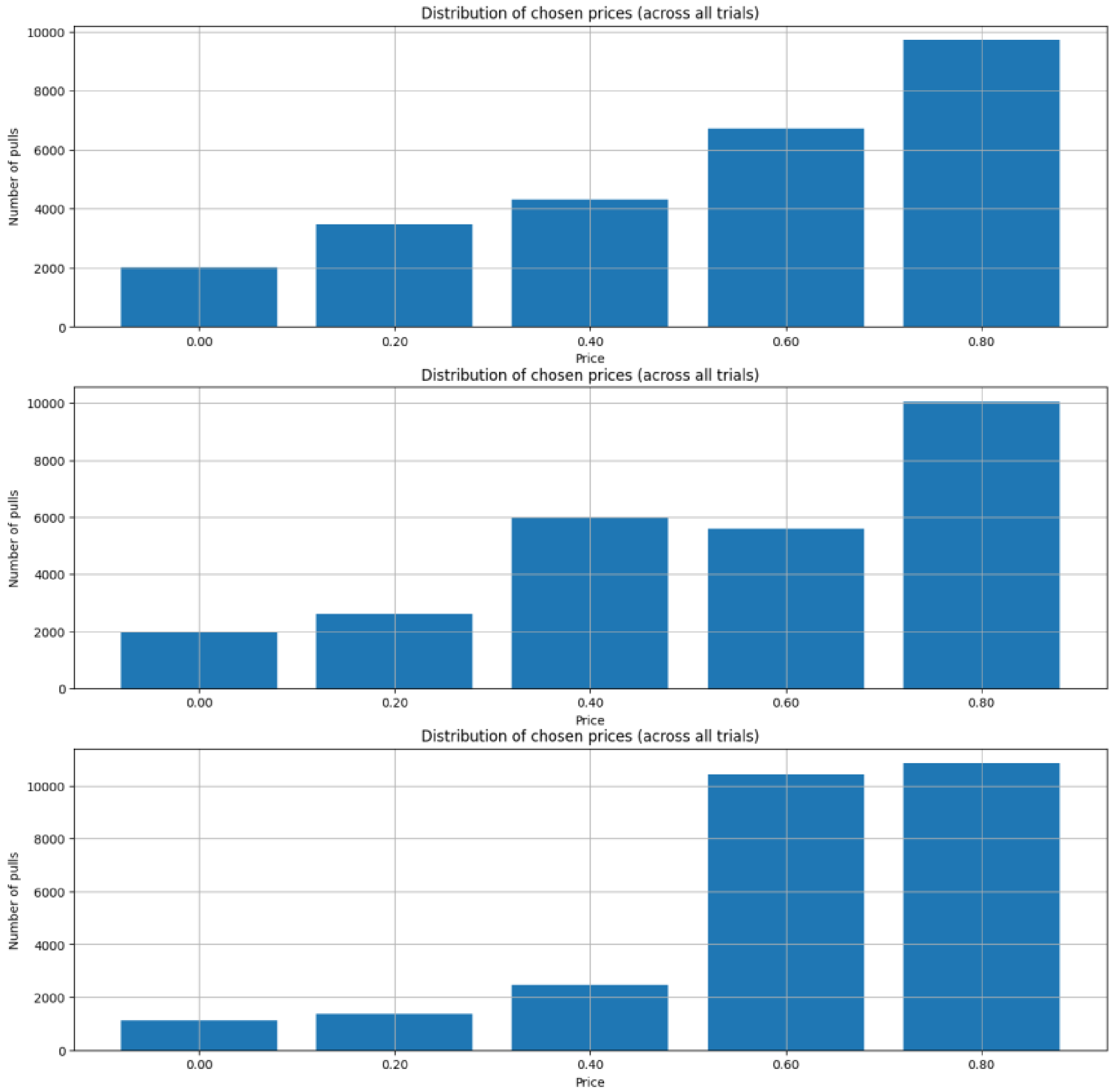
Requirement 4 Best-of-both-worlds with multiple products with inventory constraint

ADVERSARIAL ENVIRONMENT



Requirement 4 Best-of-both-worlds with multiple products with inventory constraint

STATIONARY ENVIRONMENT



Requirement 5
Slightly non-stationary environments with multiple products and inventory constraint

SLIGTHLY NON-STATIONARY ENVIRONMENT

- Rounds are partitioned in intervals of different sizes
- In each interval the distribution of products valuations is chosen randomly between the following:

$$X_1 \sim \mathcal{U}(0,1)$$

$$X_2 \sim \mathrm{Beta}(4,2)$$

$$X_3 \sim \mathrm{Beta}(2,4)$$

$$X_4 \sim \mathcal{N}(0.5, 1^2)$$

PARAMETERS

- Time horizon: T = 10000
- Prices: pi ∈ [0,1]
- Budget: 75% T
- Number of products: N=3
- Window size: w= $50\sqrt{T}$

Requirement 5

Slightly non-stationary environments with multiple products and inventory constraint

AGENT

- Combinatorial- UCB with sliding window
- To extend the inventory constraint we modified the standard Combinatorial-UCB with UCB-like

BASELINE

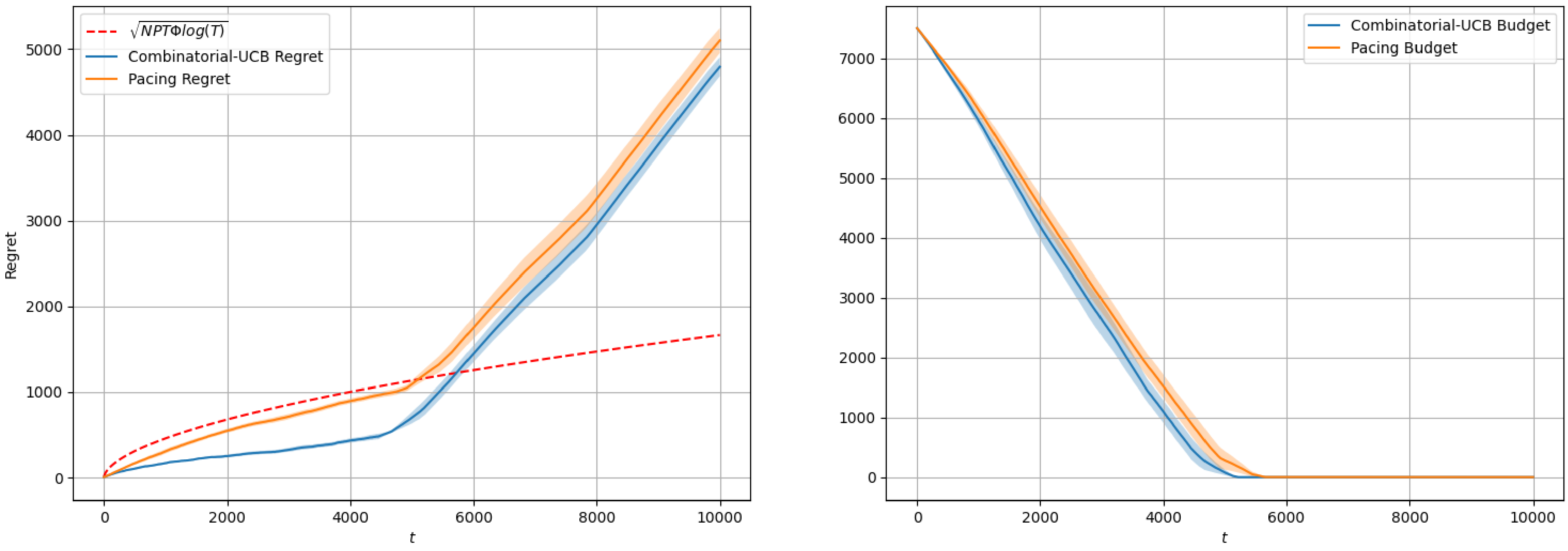
For ***each interval*** *chooses* the price that maximises the following LP:

$$\max_{x_{ij}\in[0,1]} \sum_{i=1}^{N} \sum_{j=1}^{P} v_j \cdot p_{ij} \cdot x_{ij}$$

$$\sum_{j=1}^{P} x_{ij} = 1 \quad \text{per } i = 1, \ldots, N$$

$$\sum_{i=1}^{N} \sum_{j=1}^{P} p_{ij} \cdot x_{ij} \leq \rho$$

Comparison in slightly non-stationary environment



Requirement 5
Slightly non-stationary environments with multiple products and inventory constraint