LEARN HOW TO SELL MULTIPLE TYPES OF PRODUCTS UNDER BUDGET CONSTRAINT

ONLINE LEARNING APPLICATION PROJECT

PROJECT REQUIREMENTS

Requirement 1

Single product and stochastic environment

Requirement 2

Multiple products and stochastic environment

Requirement 3

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

Requirement 4

Best-of-both-worlds with multiple products

Requirement 5

Slightly non-stationary environments with multiple products

SETTING

A company has to choose prices dynamically.

The goal of the company is to maximize profit in different selling scenarios with specific environment settings and according to different buyers behavior.

PARAMETERS AND INTERACTION

Given:

- A time horizon of **T** rounds
- The number of products **N**
- The set of possible prices **P**
- The production capacity **B** (expressed as the total number of products that the company can produce)
- The valuation v_i of the buyer for each type of product

At each round:

1. The company chooses the types of product to sell and set the price for each type of product
2. A buyer with a valuation for each type of product arrives
3. The buyer buys a unit of product if the price smaller than his valuation
4. If the product is sold, the budget of the company is decreased

REQUIREMENT 1

1.1 - Single product and Stochastic environment without Budget constraint

1.2 - Single product and Stochastic environment with Budget constraint

ENVIRONMENT

Requirement 1.1

COMPANY

- Single product selling
- No budget constraints

BUYER

- Has a distribution over the valuation of a single product
- Modelled as a Gaussian distribution

SOLUTION

Requirement 1.1

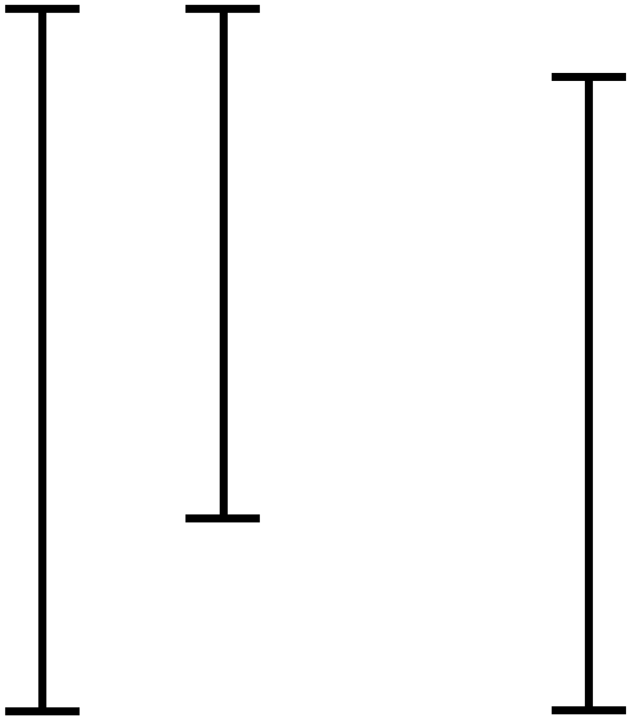
UCB1 approach:

1. Compute UCB for all the arms (prices)
2. Choose the arm with the highest UCB
3. Update the agent

Baseline computation:

Expected rewards calculated weighting the prices vector with the conversion probability

$$\{p_1, \quad p_2, \quad \ldots, \quad p_n\}$$



SIMULATION

Requirement 1.1

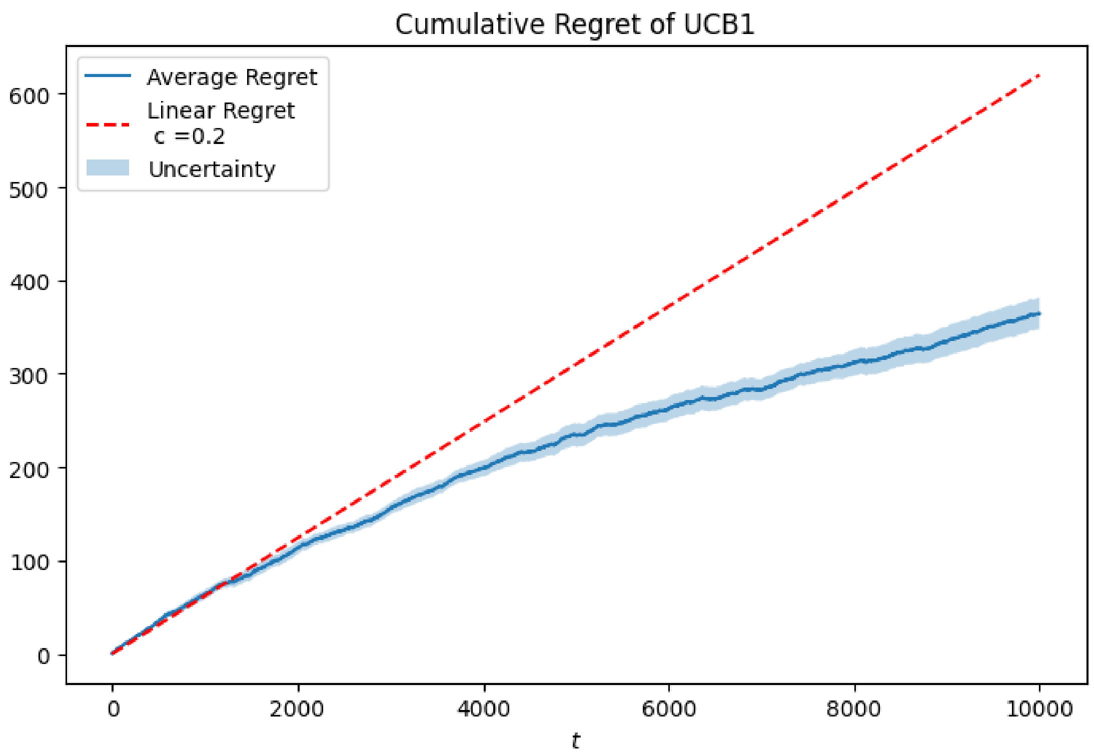
We provide results for a simulation with the following parameters:

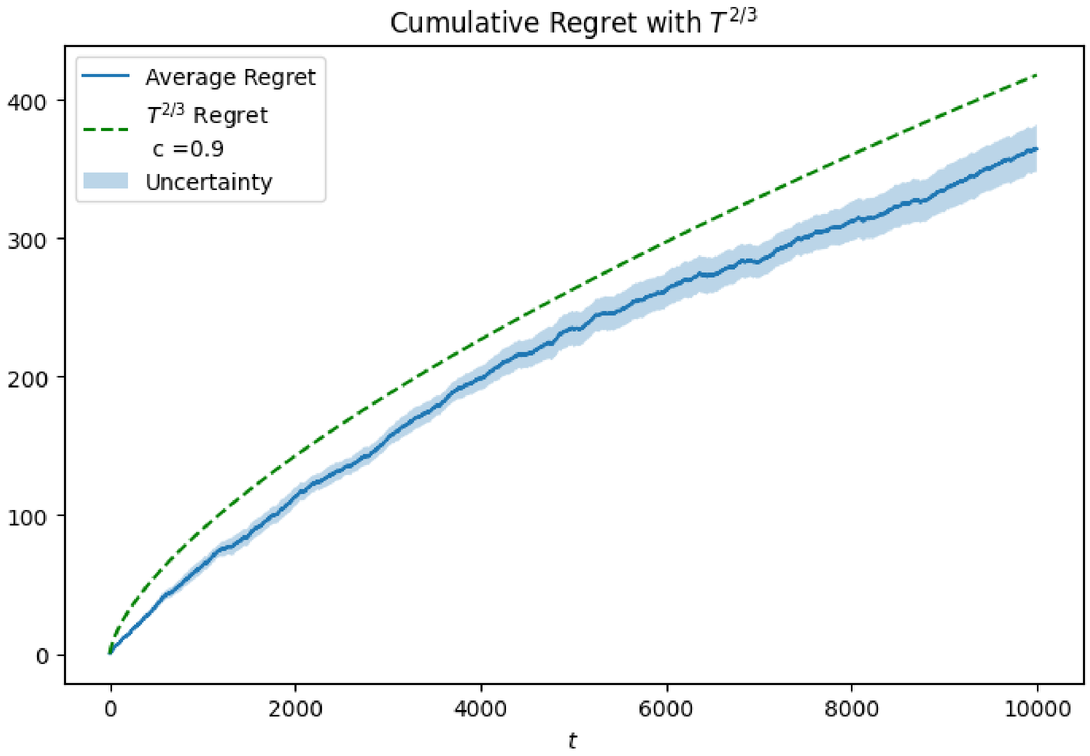
- **Time horizon** T = 10000
- **Price set P** on the interval [0, 1]
- **Gaussian distribution *N*(0.5, 1.0)** for the buyer distribution

For measuring the uncertainty on the result the simulation is executed over 10 trials

RESULTS

Requirement 1.1





ENVIRONMENT

Requirement 1.2

COMPANY

- Single product selling
- Budget constraints

BUYER

- Has a distribution over the valuation of a single product
- Modelled as a Gaussian distribution

SOLUTION

Requirement 1.2

UCB1-like approach:

1. Compute UCB for rewards and LCB for costs
2. Solve the linear program to find the optimal probabilities
3. Draw an arm from the computed distribution
4. Get the reward and the cost (unit sold)
5. Update the agent

Different baseline computation

Linear program for finding the optimal strategy **gamma**

$$OPT_t = \begin{cases} \sup\limits_{\gamma \in \Delta_B} \bar{f}_t^{UCB}(\gamma) \\ \text{s.t. } \bar{c}_t^{LCB}(\gamma) \leq \rho \end{cases}$$

$$\begin{aligned} \max_{\gamma \in \mathbb{R}^K} \quad & \sum_{i=1}^{K} \gamma_i \, \bar{f}_i^{\text{UCB}} \\ \text{s.t.} \quad & \sum_{i=1}^{K} \gamma_i \, \bar{c}_i^{\text{LCB}} \leq \rho, \\ & \sum_{i=1}^{K} \gamma_i = 1, \\ & 0 \leq \gamma_i \leq 1 \quad \forall i = 1, \ldots, K. \end{aligned}$$

SIMULATION

Requirement 1.2

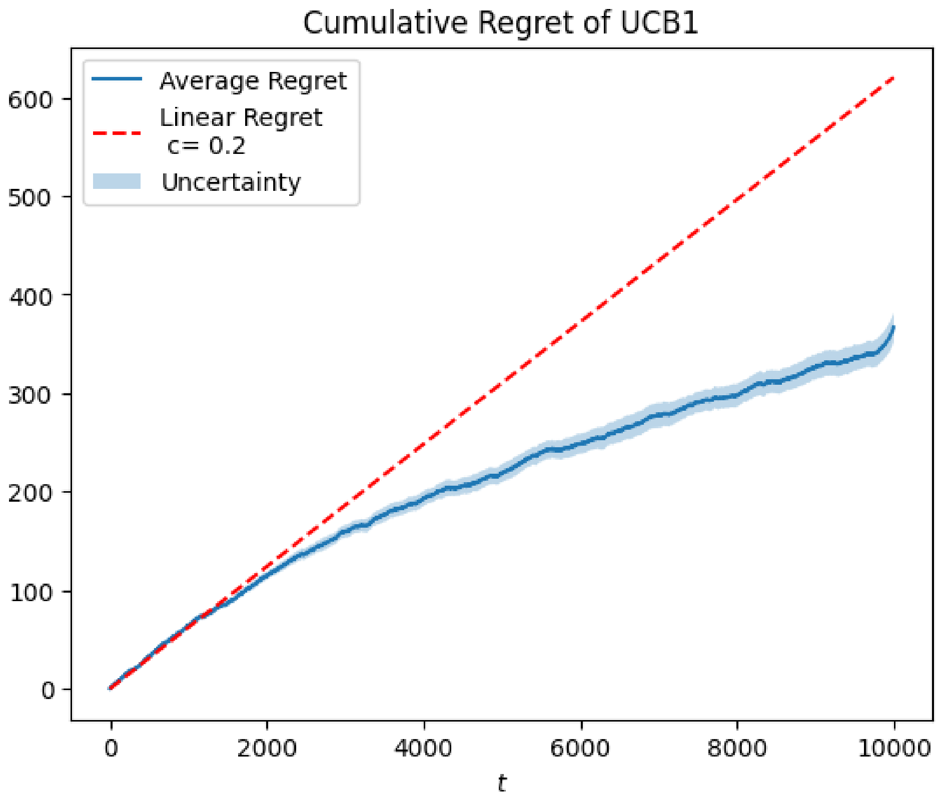
We provide results for a simulation with the following parameters:

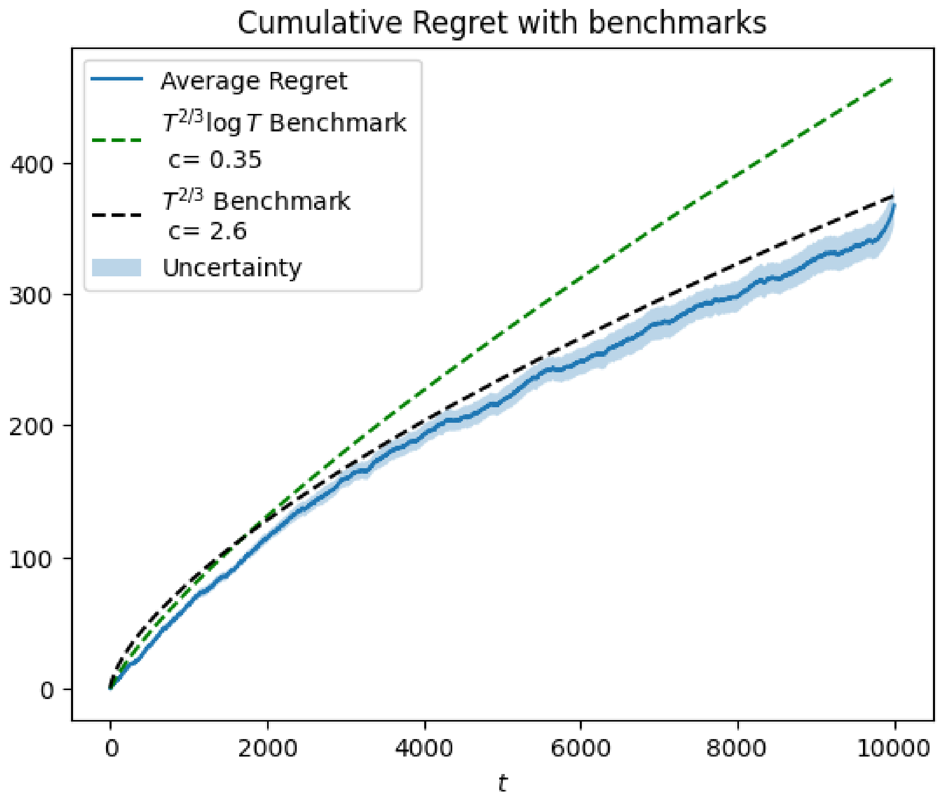
- **Time horizon** T = 10000
- **Budget** B = 4000
- **Price set P** on the interval [0, 1]
- **Gaussian distribution *N*(0.5, 1.0)** for the buyer distribution

For measuring the uncertainty on the result the simulation is executed over 10 trials

RESULTS

Requirement 1.2





REQUIREMENT 2

Multiple products and Stochastic environment

ENVIRONMENT

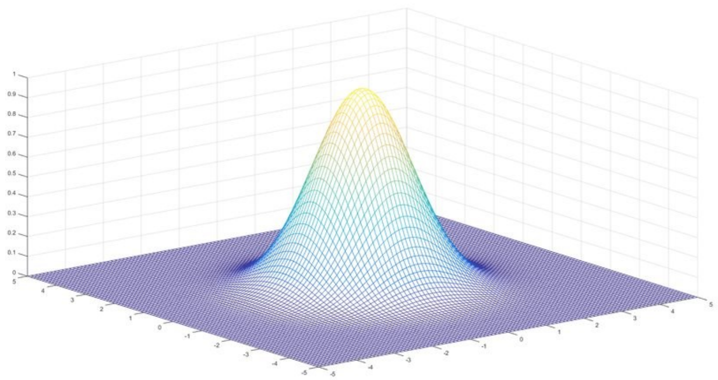
Requirement 2

COMPANY

- Multiple product selling
- Budget constraints

BUYER

- Has a joint distribution over the valuation of the products
- Modelled as a Multivariate Gaussian distribution



PROPOSED SOLUTIONS

Requirement 2

APPROACH 1

Product-wise decomposition with **independent UCB for each product.**

Same approach as Req. 1.2 but for N > 1 products

APPROACH 2

A priori calculation of all **superarms with cartesian product.**

Full combinatorial optimization with linear program solving for joint pricing decisions.

APPROACH 3

Same approach as approach 2 but **greedy**: we don't optimize solving the linear program

Baseline Computation

Linear program for finding the optimal **gamma matrix**

SIMULATION

Requirement 2

We provide results for a simulation with the following parameters:

- **Time horizon** T = 10000
- **Budget** B = 16000
- **Price set P** on the interval [0, 1]
- **Number of Products** 3
- **Multivariate Gaussian distribution** with mean vector [0.5, 0.6, 0.7] and covariance matrix [[0.1, 0.05, 0.02], [0.05, 0.1, 0.03], [0.02, 0.03, 0.1]].

For measuring the uncertainty on the result the simulation is executed over 5 trials

APPROACH 1

Requirement 2

Product-wise UCB1 approach:

1. Compute UCB for rewards and LCB for costs for each product
2. Compute the optimal strategy **gamma** for each product using the **linear program**
3. Generate and pull the superarm using the gamma matrix
4. Get prices and check for units sold
5. Update the agent

SUPERARM

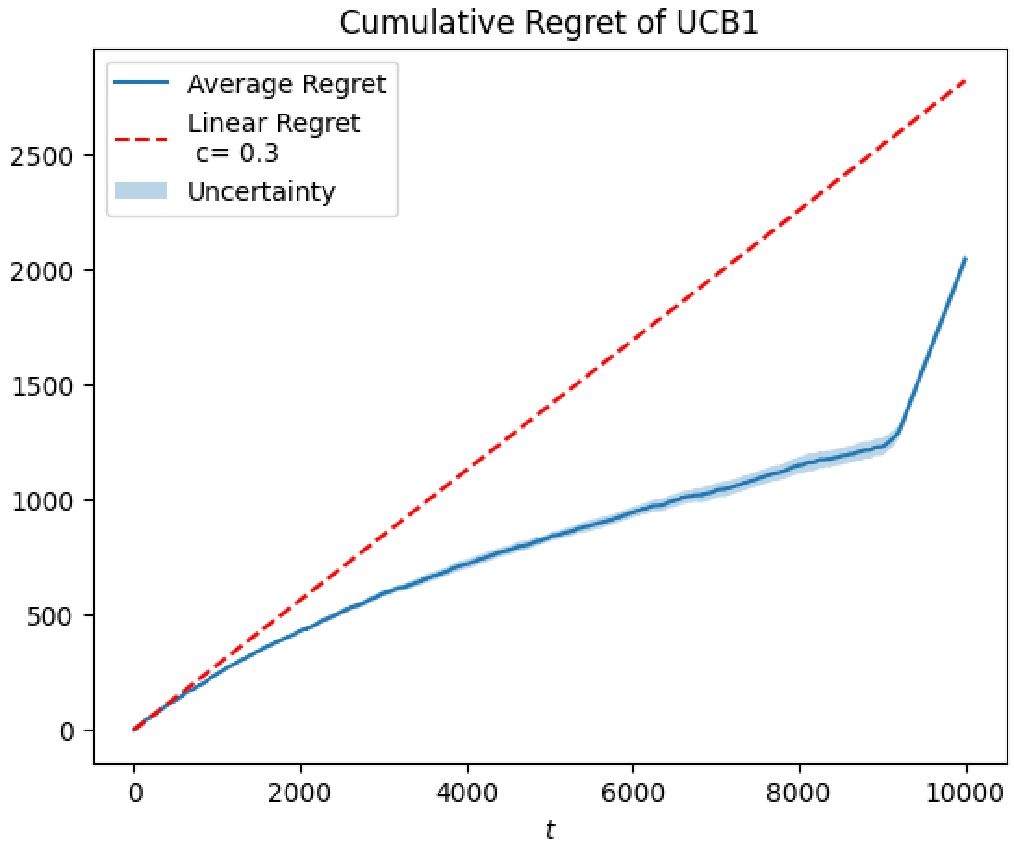
PRODUCT 1	*p1*
PRODUCT 2	*p2*
PRODUCT 3	*p3*

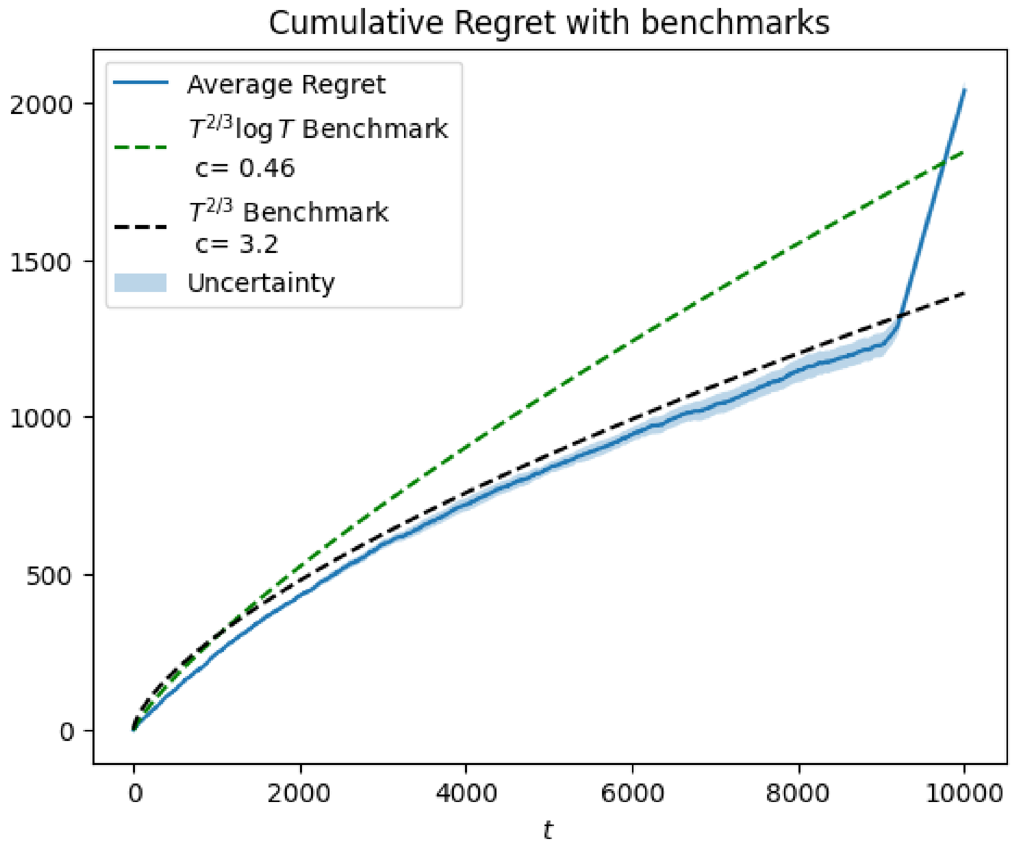
$$
\begin{aligned}
\underset{\{\gamma_{i,p}\}_{p=0}^{K-1}}{\text{maximize}} \quad & \sum_{p=0}^{K-1} \mathrm{UCB}_{i,p}(t)\, \gamma_{i,p} \\
\text{subject to} \quad & \sum_{p=0}^{K-1} \mathrm{LCB}^{c}_{i,p}(t)\, \gamma_{i,p} \leq \rho_t, \\
& \sum_{p=0}^{K-1} \gamma_{i,p} = 1, \\
& 0 \leq \gamma_{i,p} \leq 1, \quad \forall p.
\end{aligned}
$$

$$K = \#prices$$
$$|Y| = \#products$$

RESULTS

Requirement 2 – Approach 1





APPROACH 2

Requirement 2

Full combinatorial UCB1 approach:

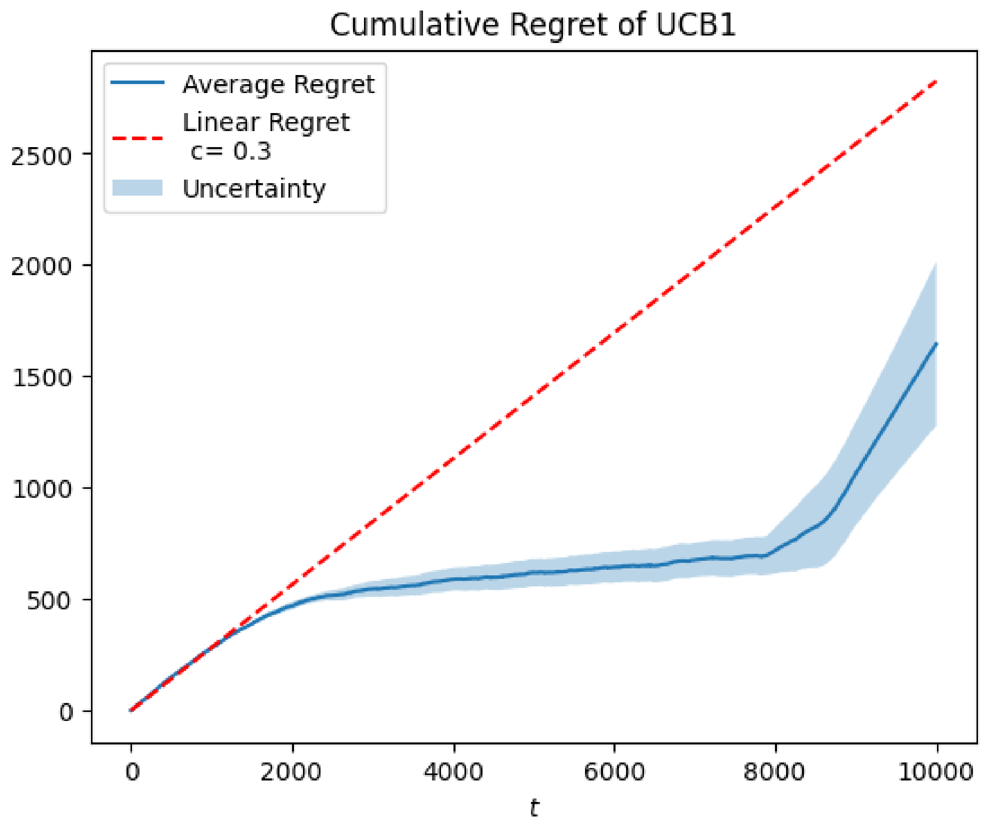
1. Generate all the combination of prices (**superarms**) with cartesian product
2. Compute UCB for rewards and LCB for costs for each superarm
3. Solve the linear program to find the gamma
4. Pull the superarm using the gamma and get the reward and the cost (if sold)
5. Update the agent

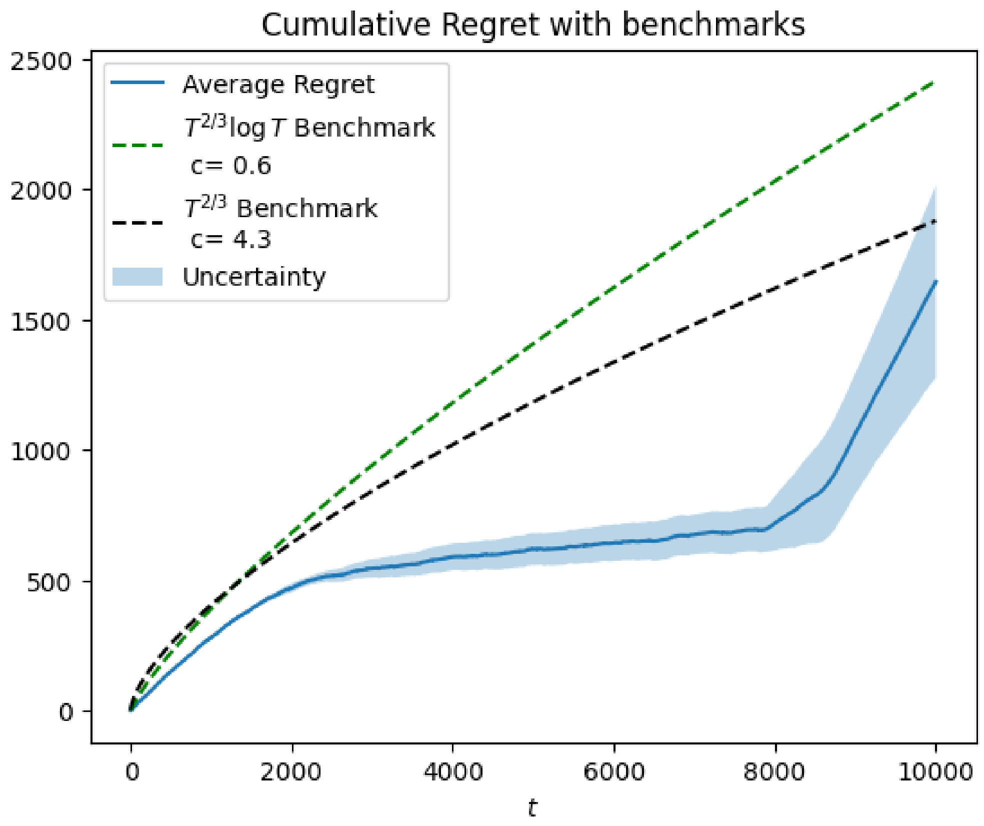
Given S the set of superarms:

$$
\begin{aligned}
\underset{\{\gamma_s\}_{s\in\mathcal{S}}}{\text{maximize}} \quad & \sum_{s\in\mathcal{S}} \mathrm{UCB}_s(t)\,\gamma_s \\
\text{subject to} \quad & \sum_{s\in\mathcal{S}} \mathrm{LCB}^c_s(t)\,\gamma_s \;\le\; \rho, \\
& \sum_{s\in\mathcal{S}} \gamma_s = 1, \\
& 0 \le \gamma_s \le 1, \quad \forall s \in \mathcal{S}.
\end{aligned}
$$

RESULTS

Requirement 2 – Approach 2





APPROACH 3

Requirement 2

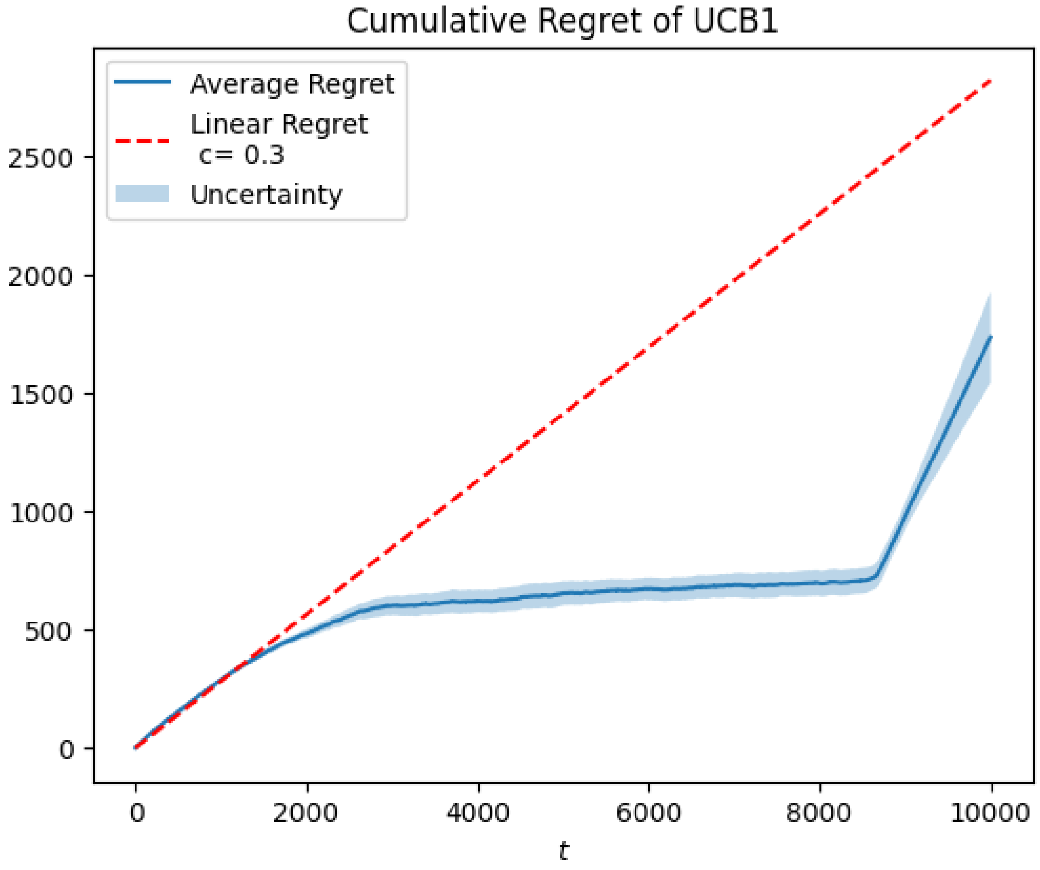
Full combinatorial UCB1 approach, with greedy:

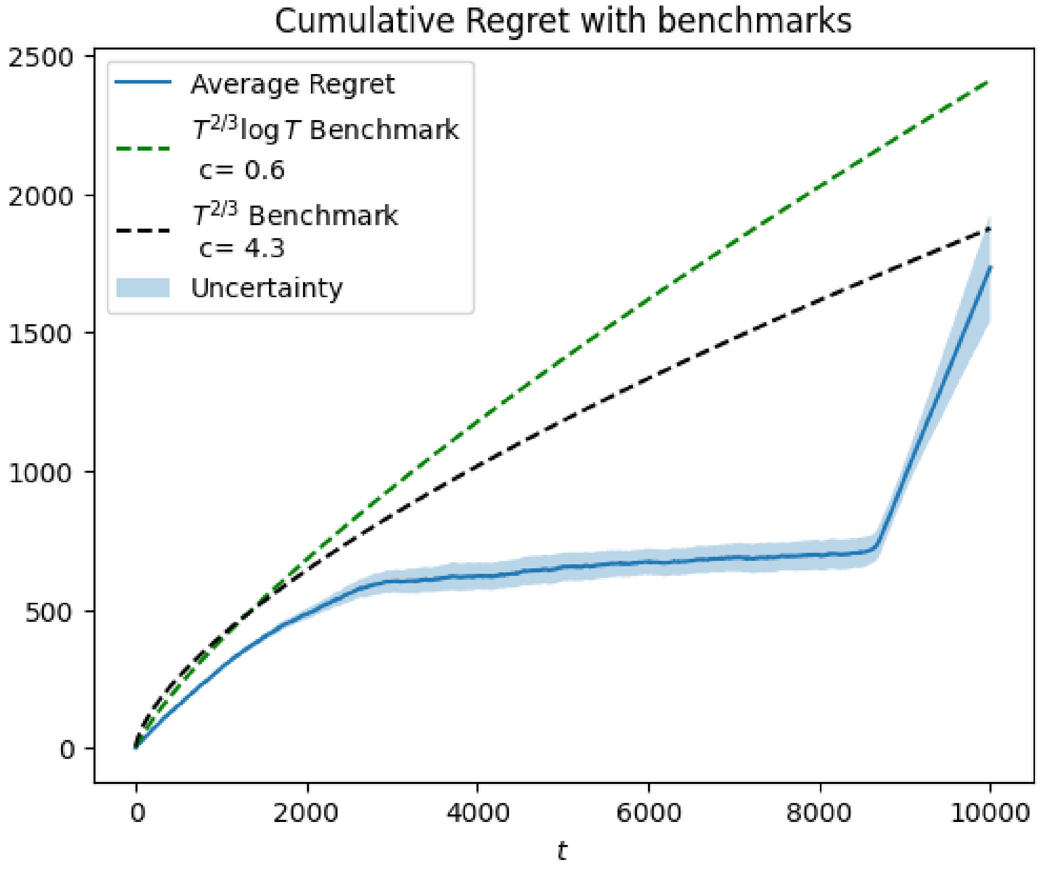
1. Generate all the combination of prices (superarms) with cartesian product
2. Compute UCB for rewards and LCB for costs for each superarm
3. Choose **feasible** superarm which maximize utility, without linear program optimization
4. Pull the superarm and get the reward and the cost (if sold)
5. Update the agent

```
def compute_opt(self, f_ucbs, c_lcbs):
    # dont use the linear program solver, just use a greedy approach
    feasible = c_lcbs <= self.rho
    if not np.any(feasible):
        return np.argmax(f_ucbs)
    return np.argmax(f_ucbs[feasible])
```

RESULTS

Requirement 2 – Approach 3





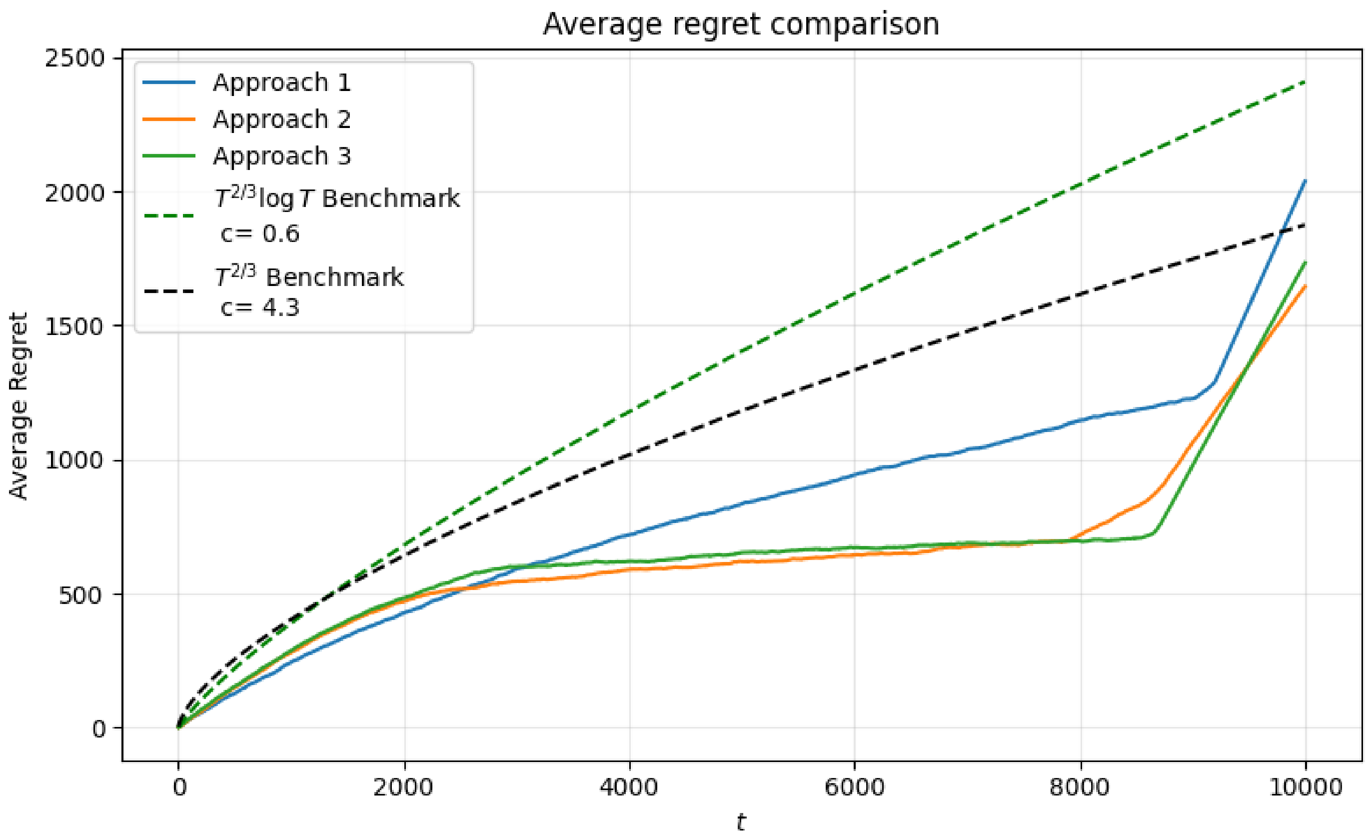
RESULT SUMMARY

Requirement 2

- **Approach 1:**
  - **Less arms** and good learning process, but **worse regret**

- **Approach 2:**
  - **Many arms** (full combinatorial) but **learns well** and achieves **better regret**

- **Approach 3:**
  - Similar to approach 2, but **faster** and **depletes the budget later**



REQUIREMENT 3

Single product and Adversarial environment

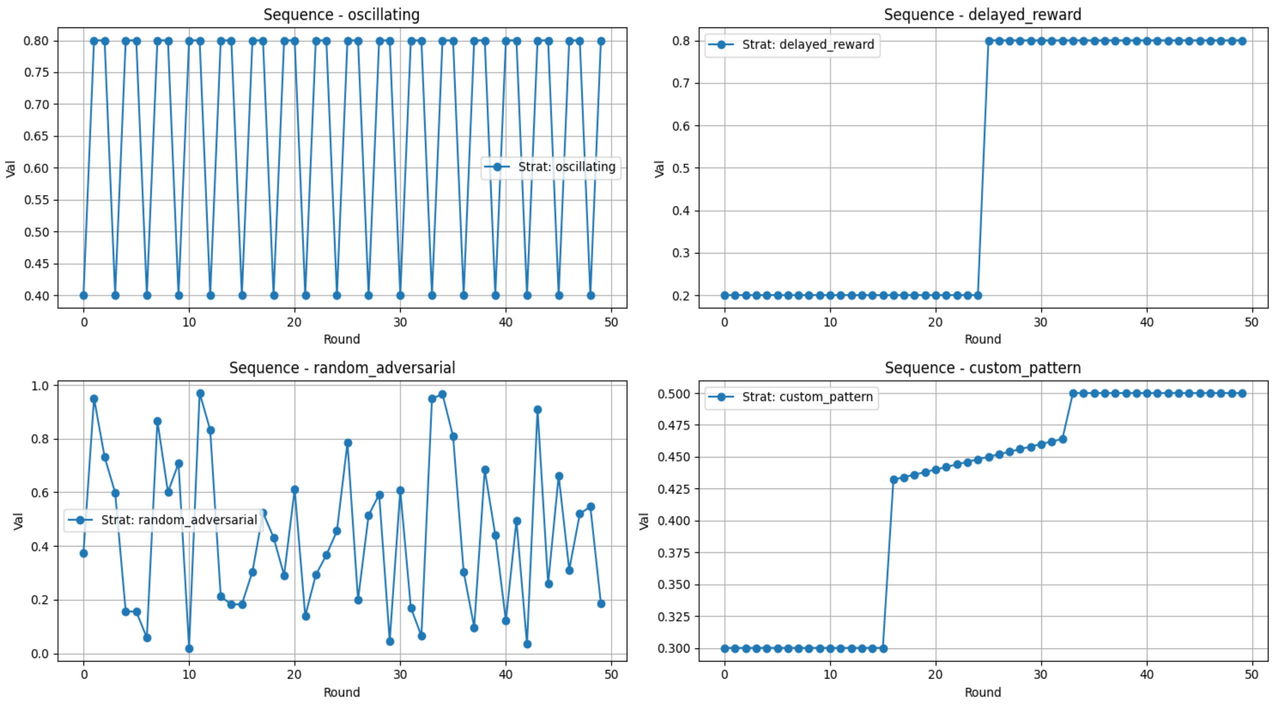
ENVIRONMENT
Requirement 3

COMPANY

- Single product selling
- Budget constraints

BUYER

- **Adversarial valuations** changing the expected value of the gaussian over time



PROPOSED SOLUTIONS

Requirement 3

Using the pacing strategy with a **Lagrangian multiplier λ**.

- If sales exceed ρ, λ increases, **lowering** the next price;
- If sales fall short, λ decreases, **increasing** the next price.

APPROACH 1	APPROACH 2
Bandit Feedback: **EXP3 agent** used as regret minimizer for price selection.	**Full Feedback:** **Hedge agent** used as regret minimizer for price selection.

Baseline Computation

For each price, compute its expected utility and expected cost. Among the prices that satisfy the budget constraint $\mathbf{c \leq \rho}$, choose the one with the highest expected utility (**best fixed arm a priori**).

SIMULATION

Requirement 3

We provide results for a simulation with the following parameters:

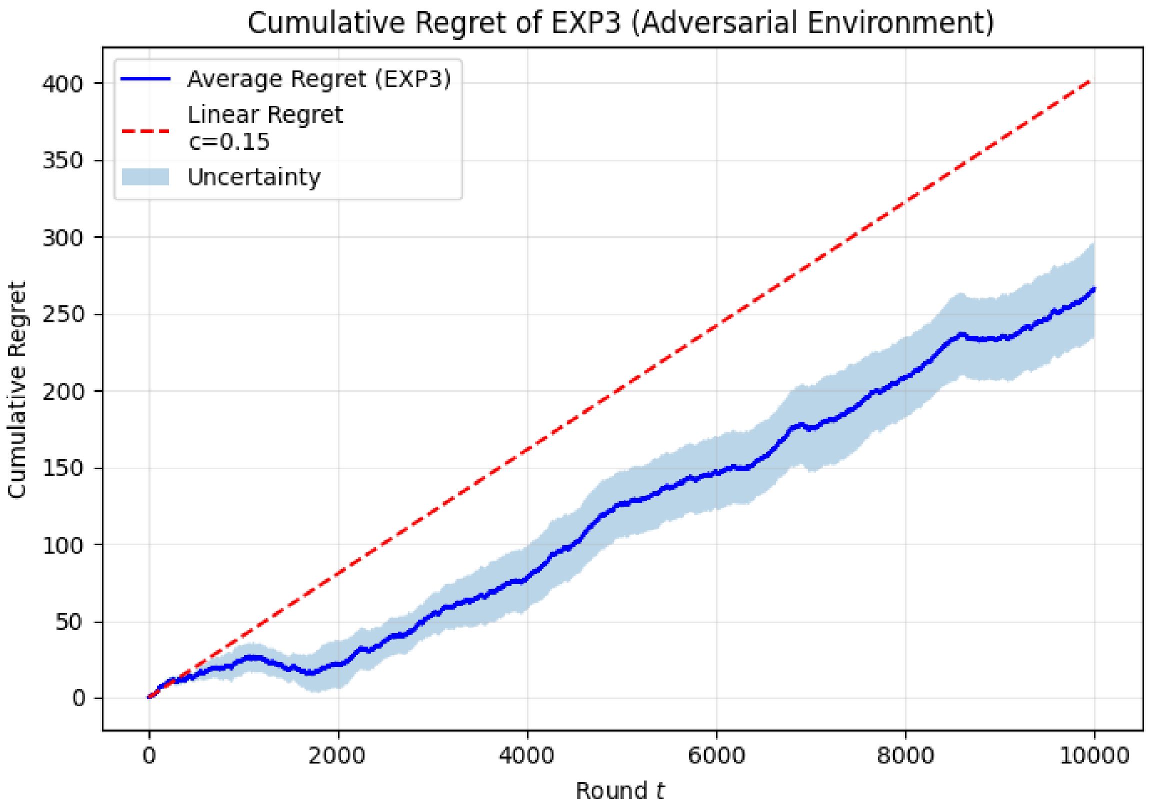
- **Time horizon** T = 10000
- **Budget** B = 5000
- **Price set P** on the interval [0, 1]

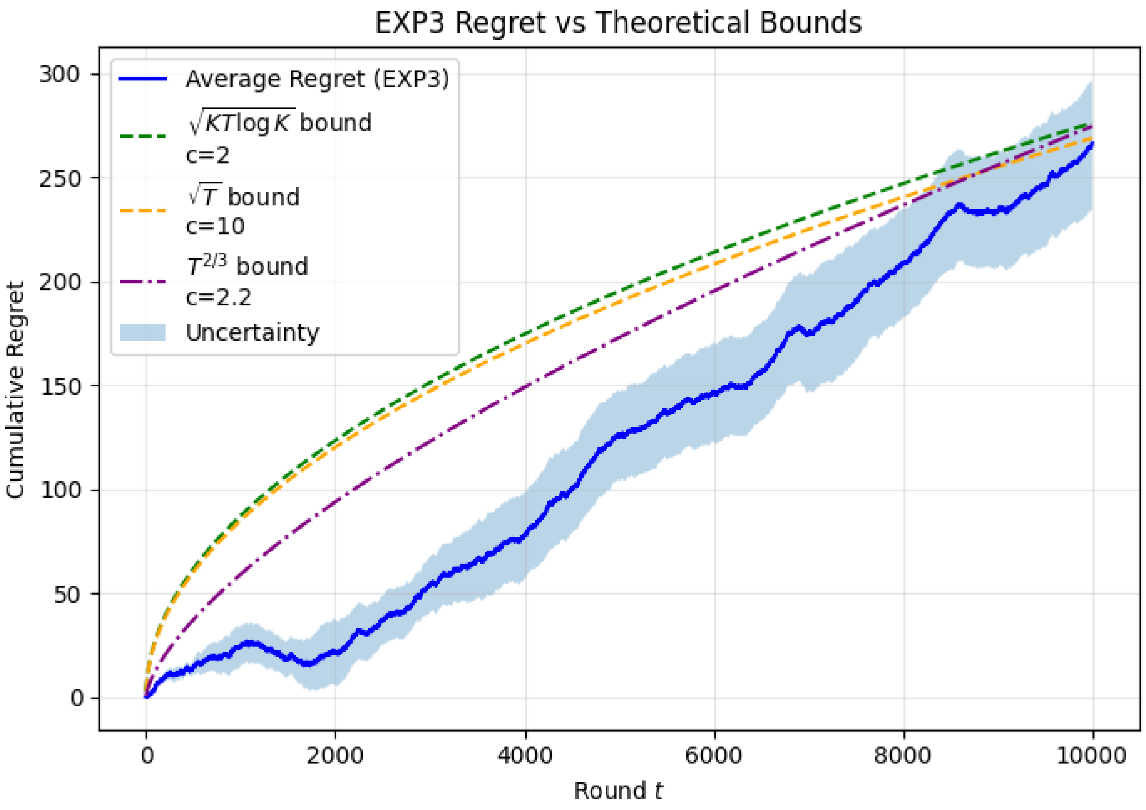
For measuring the uncertainty on the result the simulation is executed over 5 trials

RESULTS

Requirement 3

APPROACH 1 – BANDIT FEEDBACK

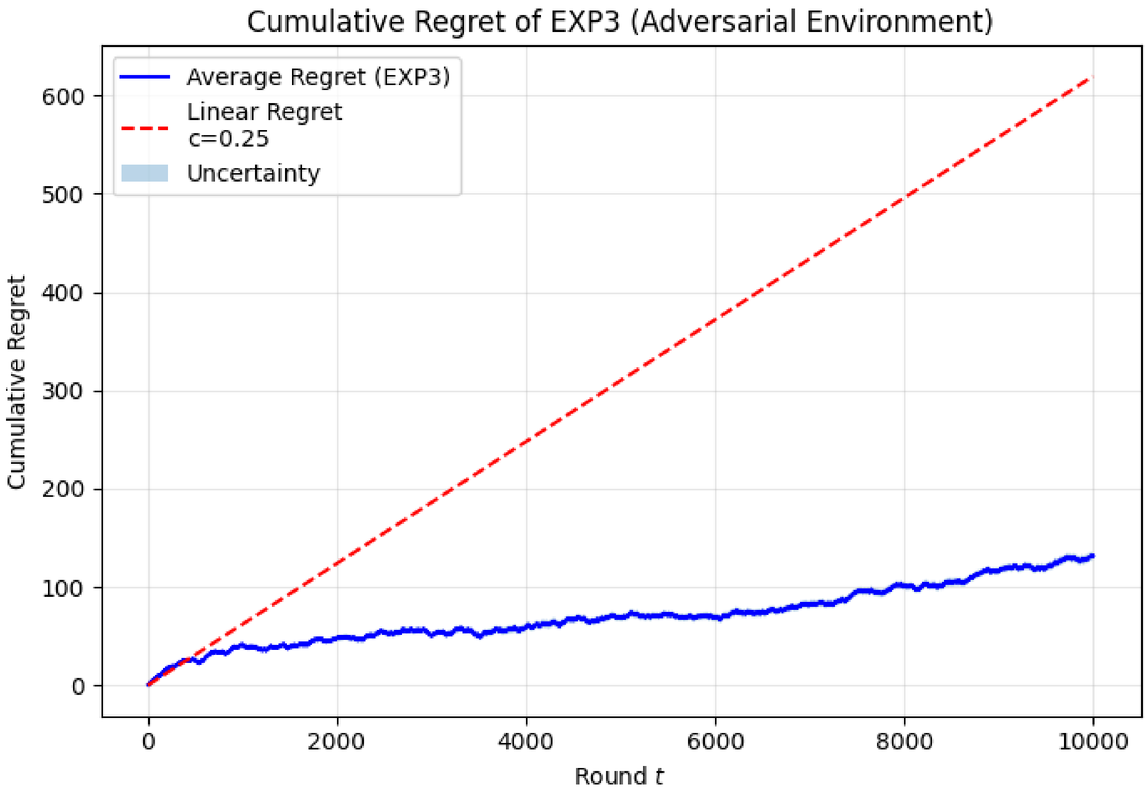


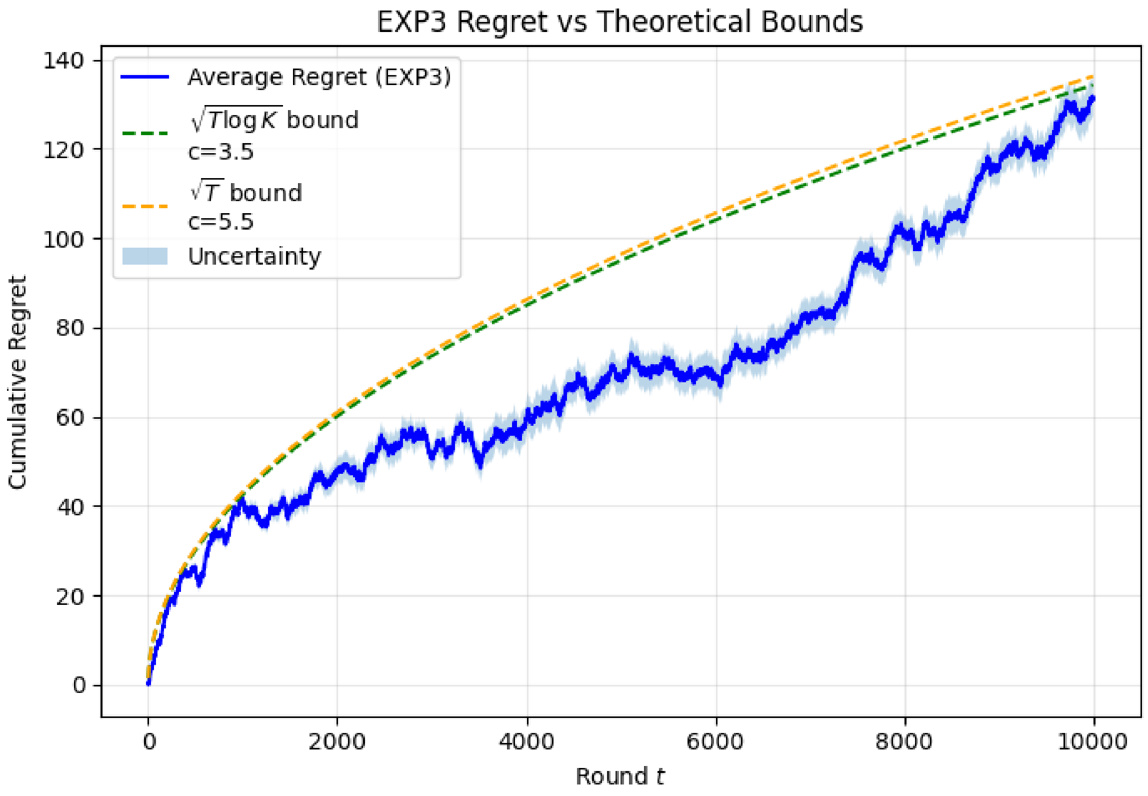


RESULTS

Requirement 3

APPROACH 2 – FULL FEEDBACK





REQUIREMENT 4

Multiple products and Adversarial environment

ENVIRONMENT
Requirement 4

COMPANY

- Multiple product selling
- Budget constraints

BUYER

- Adversarial valuations changing over time:
  - oscillating,
  - delayed reward,
  - random,
  - custom pattern

PROPOSED SOLUTIONS

Requirement 4

Using the pacing strategy with a **Lagrangian multiplier λ**.

- If sales exceed ρ, λ increases, **lowering** the next price;
- If sales fall short, λ decreases, **increasing** the next price.

Bandit Feedback:

EXP3 agent used as regret minimizer for price selection, **for each product**.

Baseline Computation

For each product and price, compute expected utility and cost. Evaluate **all product–price combinations** and select the one with the highest expected utility subject to $\sum c \leq \rho$.

SIMULATION

Requirement 4

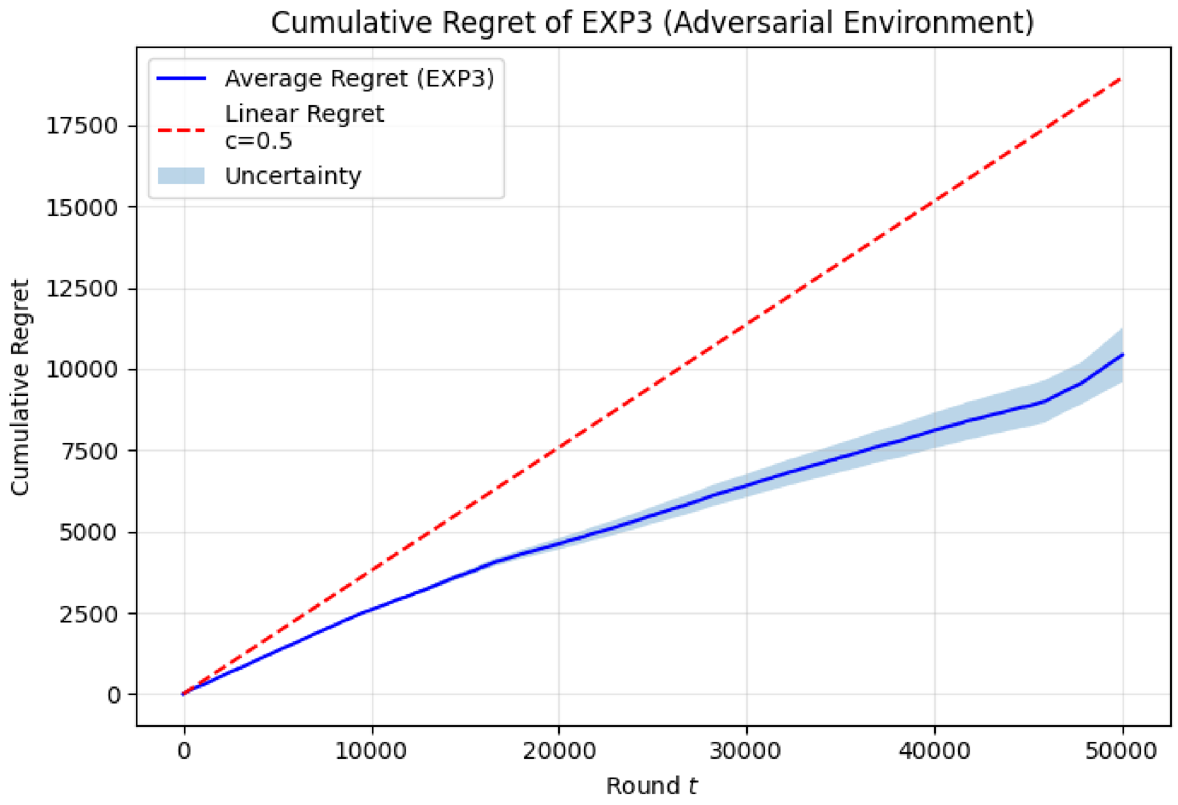
We provide results for a simulation with the following parameters:

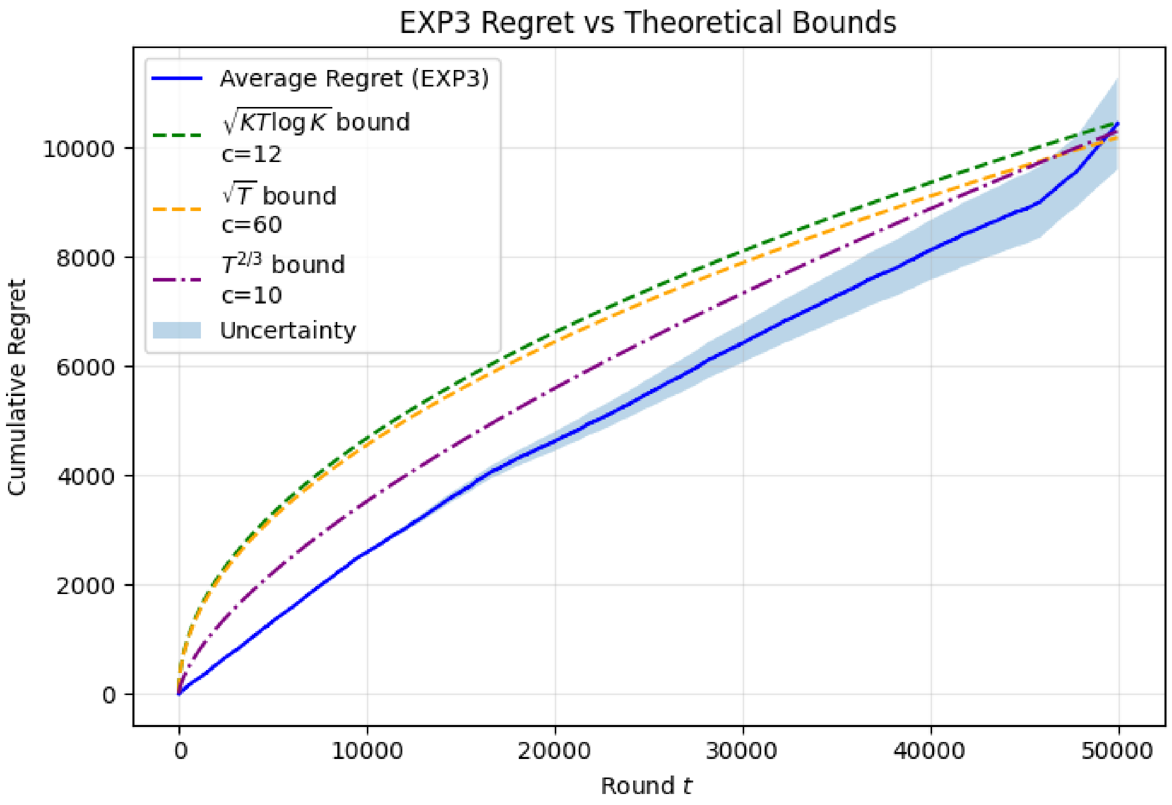
- **Time horizon** T = 50000
- **Budget** B = 80000
- **Price set P** on the interval [0, 1]
- **Number of Products:** 3

For measuring the uncertainty on the result the simulation is executed over 5 trials

RESULTS

Requirement 4





REQUIREMENT 5

Slightly non-stationary environment

ENVIRONMENT
Requirement 5

COMPANY

- Multiple product selling
- Budget constraints

BUYER

- Slightly Non-stationary behavior
- Adversarial valuations changing over time in a fixed, predetermined way.

PROPOSED SOLUTIONS

Requirement 5

Using UCB with **Sliding Window**: we empirically choose a window size W, such that only the most recent W samples are considered when computing the UCB.

Baseline a Priori

Compute the **expected utility** for the initial and the target buyer distributions. Take the **average** of the two optimal utilities as reference benchmark.

Baseline a Posteriori

At **each round t**, **compute the optimal expected utility** given the current buyer distribution **($\mu t, \sigma$)**.

SIMULATION

Requirement 5

We provide results for a simulation with the following parameters:

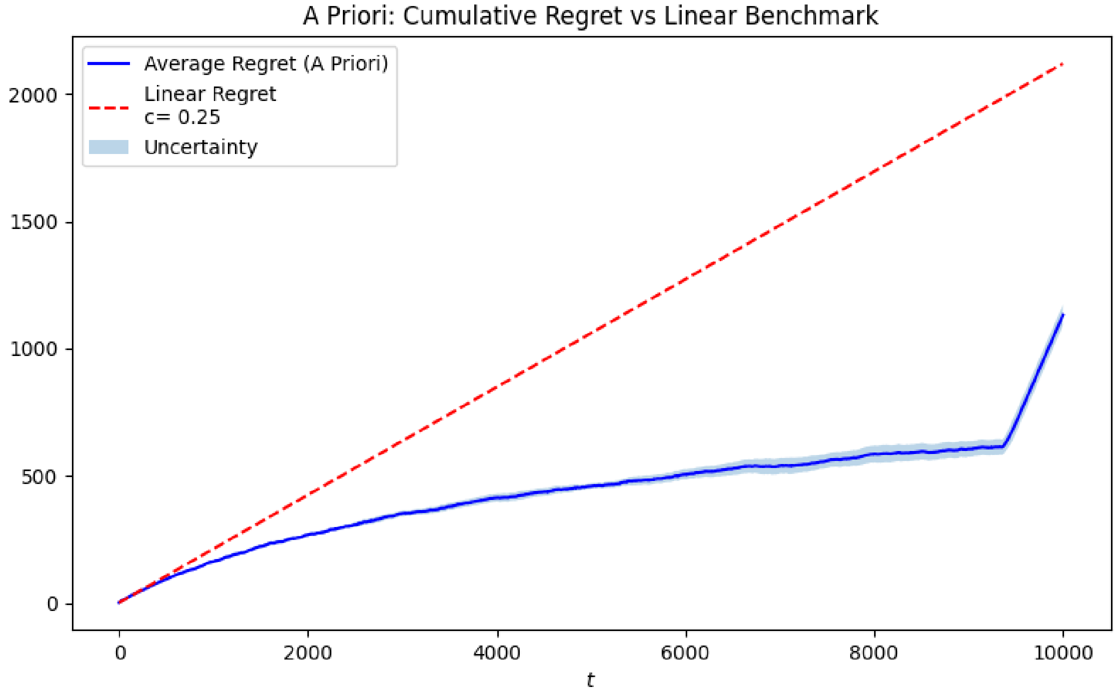
- **Time horizon** T = 10000
- **Budget** B = 16000
- **Price set P** on the interval [0, 1]
- **Number of Products** 3
- **Window size** 2500
- **Covariance Matrix** [[0.1, 0.05, 0.02], [0.05, 0.1, 0.03], [0.02, 0.03, 0.1]], fixed
- **Initial Mean vector** [0.6, 0.5, 0.7]
- **Target Mean vector** [0.4, 0.6, 0.5]

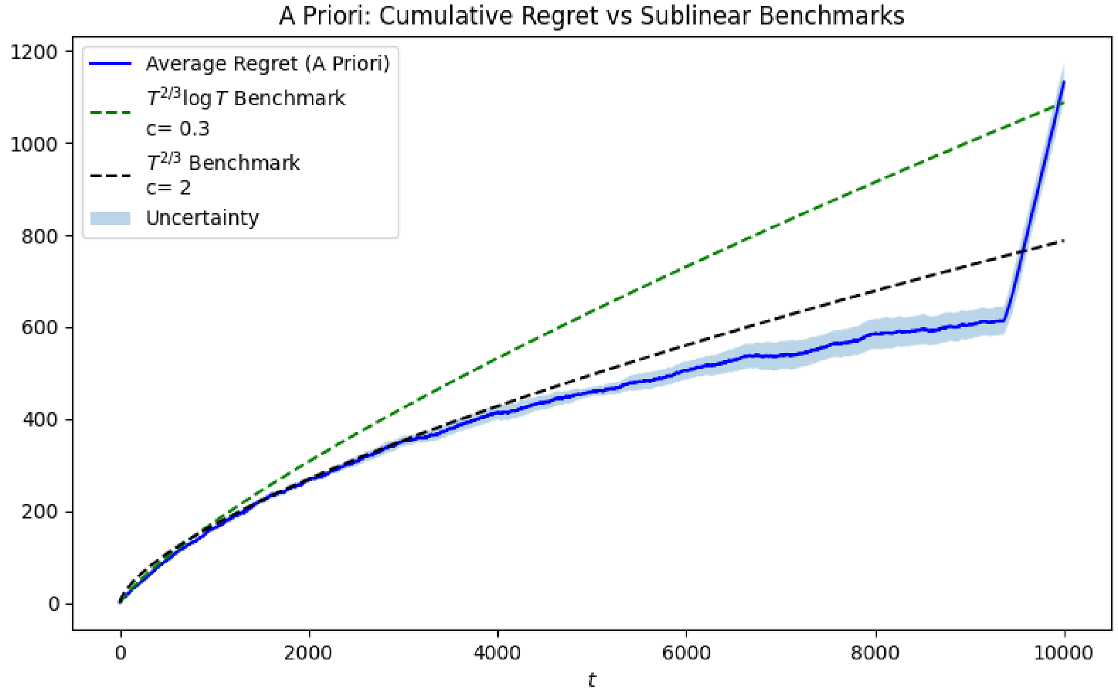
For measuring the uncertainty on the result the simulation is executed over 5 trials

RESULTS

Requirement 5

BASELINE A PRIORI

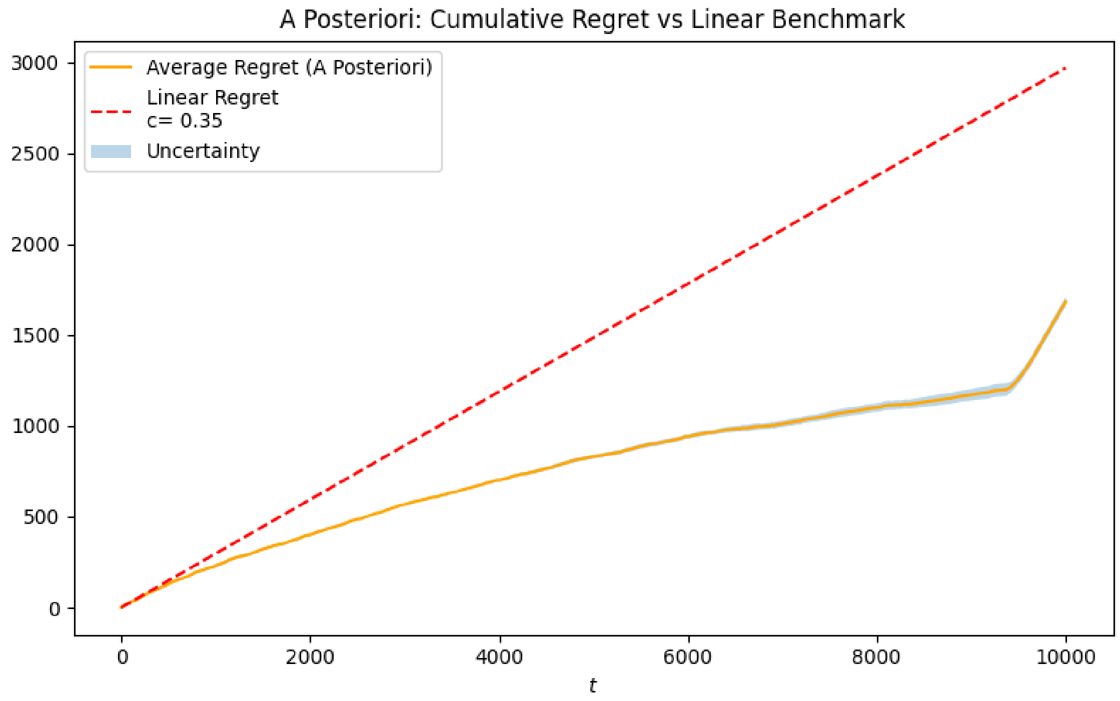


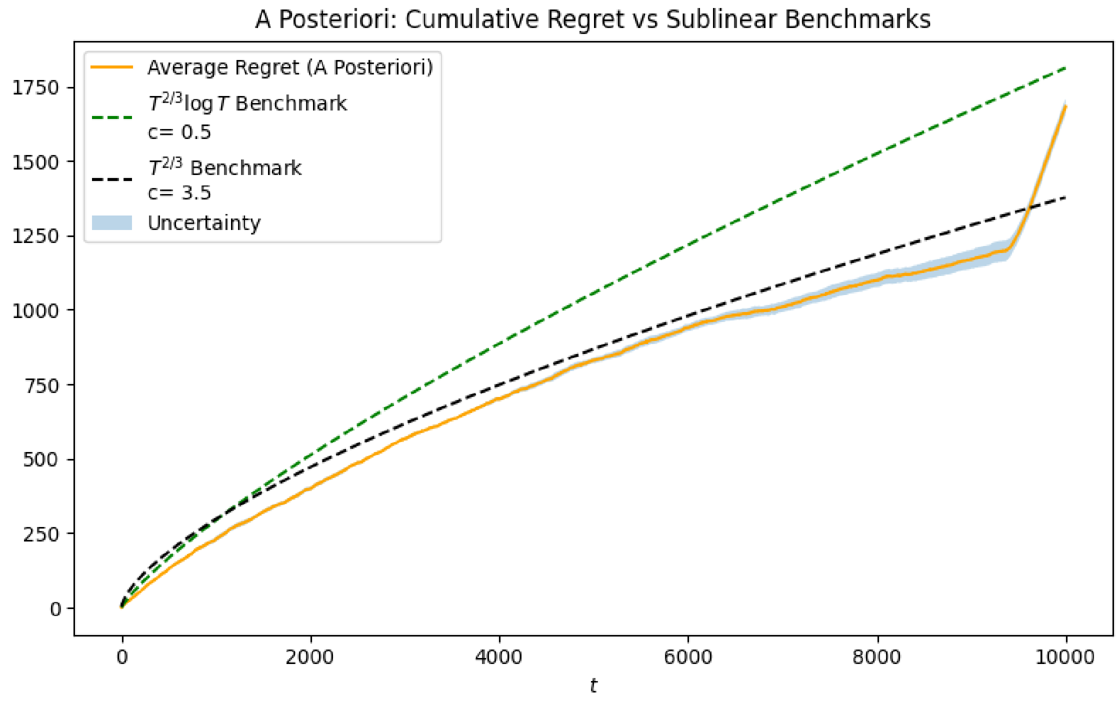


RESULTS

Requirement 5

BASELINE A POSTERIORI

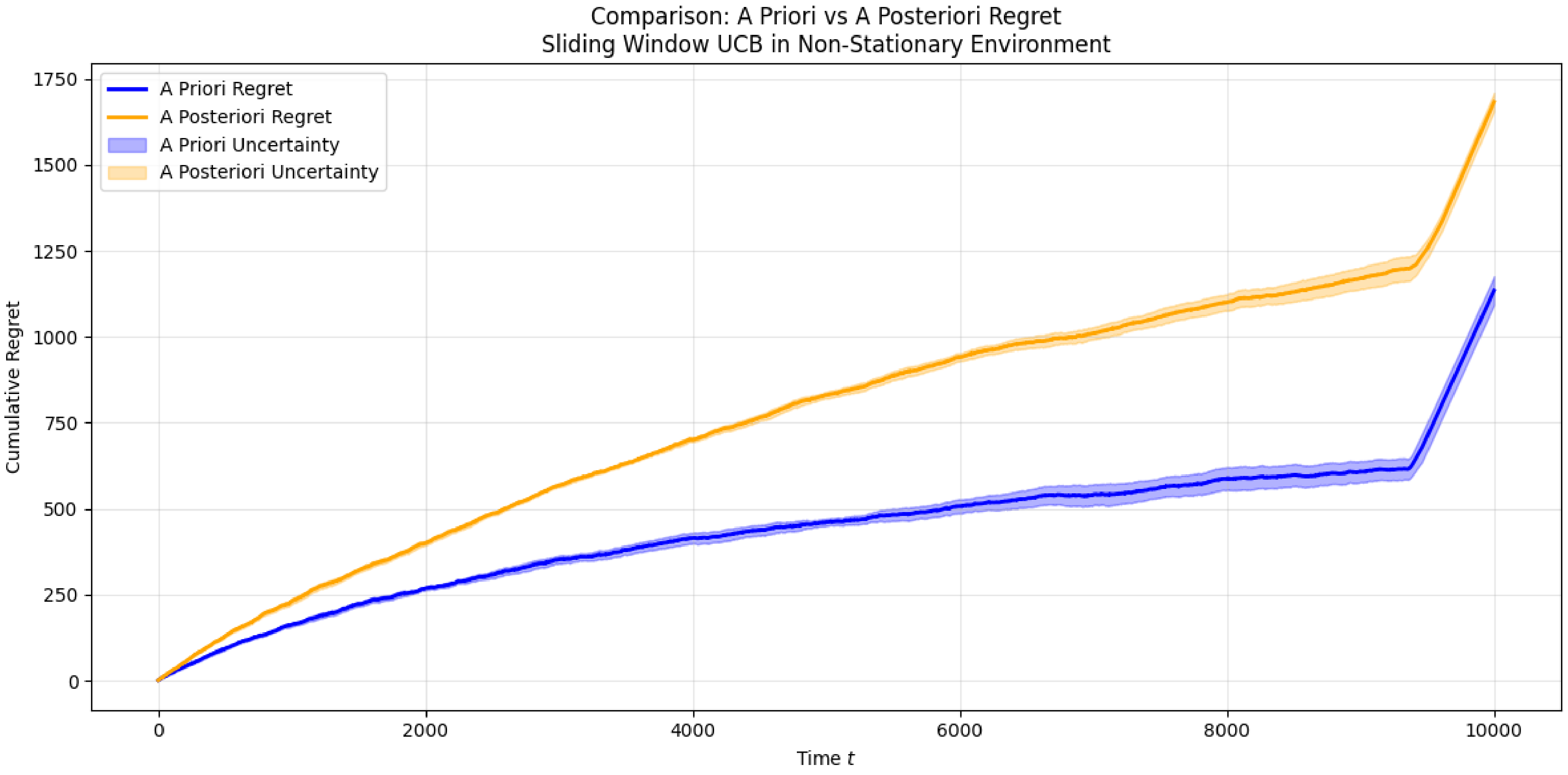




RESULTS

Requirement 5

COMPARISON



CONCLUSIONS

CONCLUSIONS

	Approaches	Regret	Costant	Budget depletion over T
R1	UCB-like	$T^{\frac{2}{3}}$	$0.9 \; - 2.6$	$98\,\% \; - \; 100\,\%$
R2	Combinatorial UCB-like	$T^{\frac{2}{3}}$	$3.2 \; - 4.3$	$91\,\% \; - 96\,\%$
R3	Lagrangian multipliers	$T^{\frac{2}{3}}$	2.2	$94\,\% \; - \; \sim 100\%$
R4	Multiple Lagrangian multipliers	$T^{\frac{2}{3}}$	10	$94\,\%$
R5	SW Combinatorial UCB-like	$T^{\frac{2}{3}}$	$2 \; - 3.5$	$94\,\%$