Online Learning Applications Project

Ameen Hayder Elzaki 10837782
Kalisto Willaey 10987637

Part 1

Implementation Description

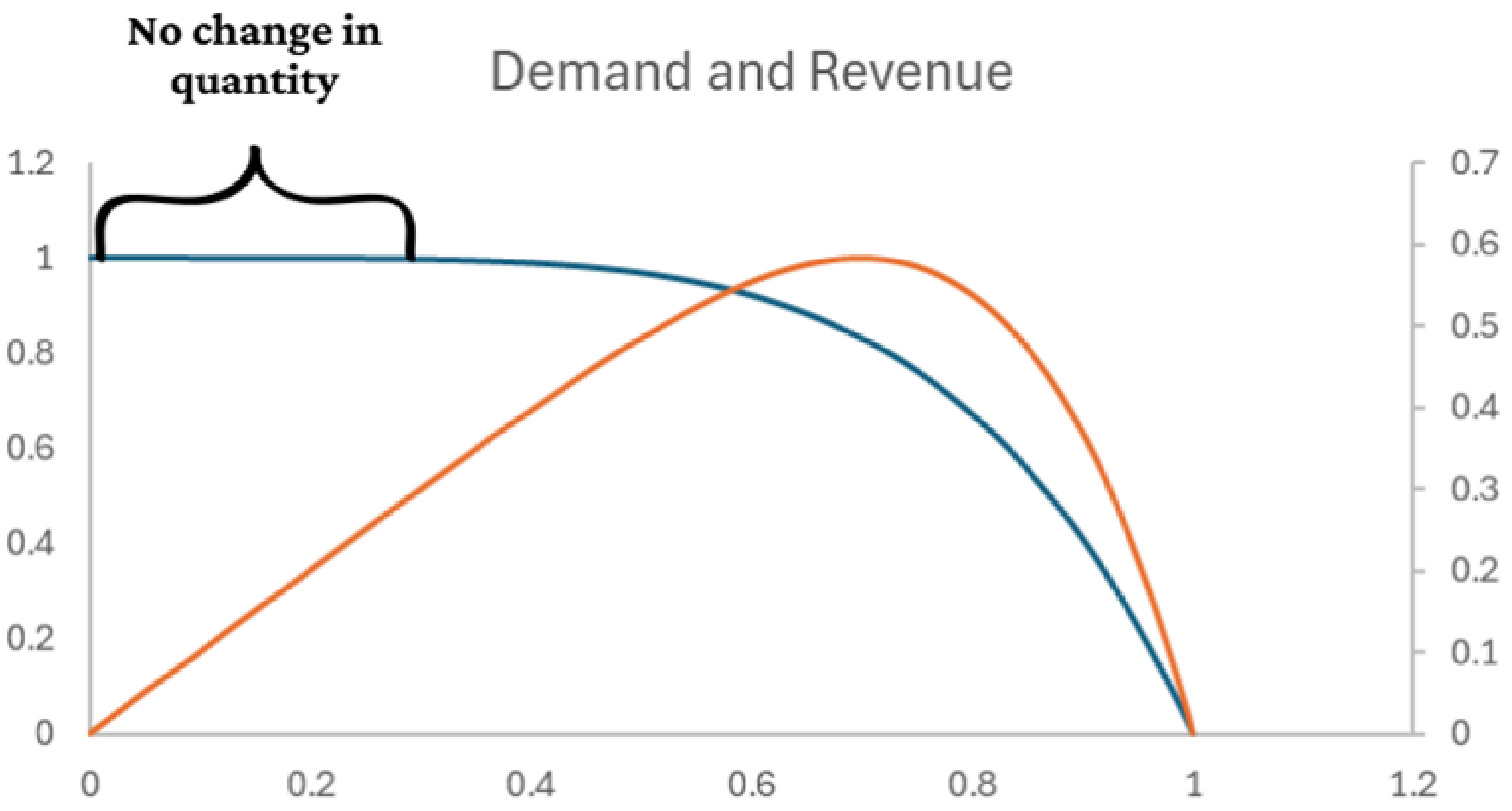
- Pricing Environment Description
- Auction General Description
- Combining Pricing and Advertisement
- Non-stationary advertising
- Non-stationary stochastic pricing environment
- Two-items stochastic pricing environment

Pricing Environment Description

- The choice of the demand function should be monotonically decreasing and confined between the range [0,1].
- The power curve is chosen as it represents scenarios frequently encountered in practice.
- The curve is characterized by regions where the change in price does not lead to significant changes in demand.
- A practical case where this curve is met in airline industries where usually consumers are willing to accept higher prices in certain ranges .

$Q = 1 - P^b$

π /Customer $= Q * (P - c)$



Auction Environment Description

- **Objective:** Maximise the utility, U = valuation - (cost of payment), under a limited budget B by participating in second-price auction.
- **Clairvoyants bid:** Calculate the optimal bid sequence to maximize utility with perfect knowledge of competitors' bids.
- **Competitor behavior:** Other bidders' bids are drawn from a uniform distribution between 0 and 10 * B / n_users simulating competitors with potentially larger budgets but limited depletion.
- **Implemented Algorithms:**
  - **Multiplicative Pacing Algorithm:** Dynamically adjusts bids using a pacing multiplier to balance utility maximization with staying within the budget.
  - **UCB-like Algorithm:** Treats the auction as a multi-armed bandit problem, exploring and exploiting bidding options to maximise cumulative utility while adhering to budget constraints.

Combined Environment:

- Here the interaction is made possible by first adjusting the advertiser evaluation of every auction round to be dependent on the price at that particular day.

$$v_{a,t} = P_t \qquad\qquad P \in [0,1]\,, Q \in [0,1]$$

- For every day $t \in T$, an auction over a number (N) of users takes place and depending on the number of users won in the auction in that particular day, the number of customers been shown the price, is determined.
- The agent rewards depend on the customers he wins everyday and also the costs he incurs for winning the auction. In addition, for sure, to the costs of realizing the good/service.

$$\pi_t = n_t{}' * (P_t - c) - \sum c_{t'} \qquad n_t{}' \sim B(n_t, Q^*)$$

$t' \in [N]$ ... N number of rounds of the auction

n_t *is the number of users seeing the ad*

- The clairvoyant will evaluate the auction at the optimal price in expectation, and will win the optimal number of users (i.e. knowing in advance the sequence of competing bids he will obtain the maximum utility given a certain budget)

$$\pi = n_t * Q^* * (P^* - c) - \min\left(\sum c_{t'}\right) \qquad n_t \sim B(N, Q^*)$$

$t' \in [N]$ ... N number of rounds of the auction

- Finally the regret must be evaluated considering the stochastic nature of the environment. (The expected regret over the randomness of the environment).

Non-Stationary Advertising

- Here the competing bids are sampled from a uniform distribution that has a dynamic range.

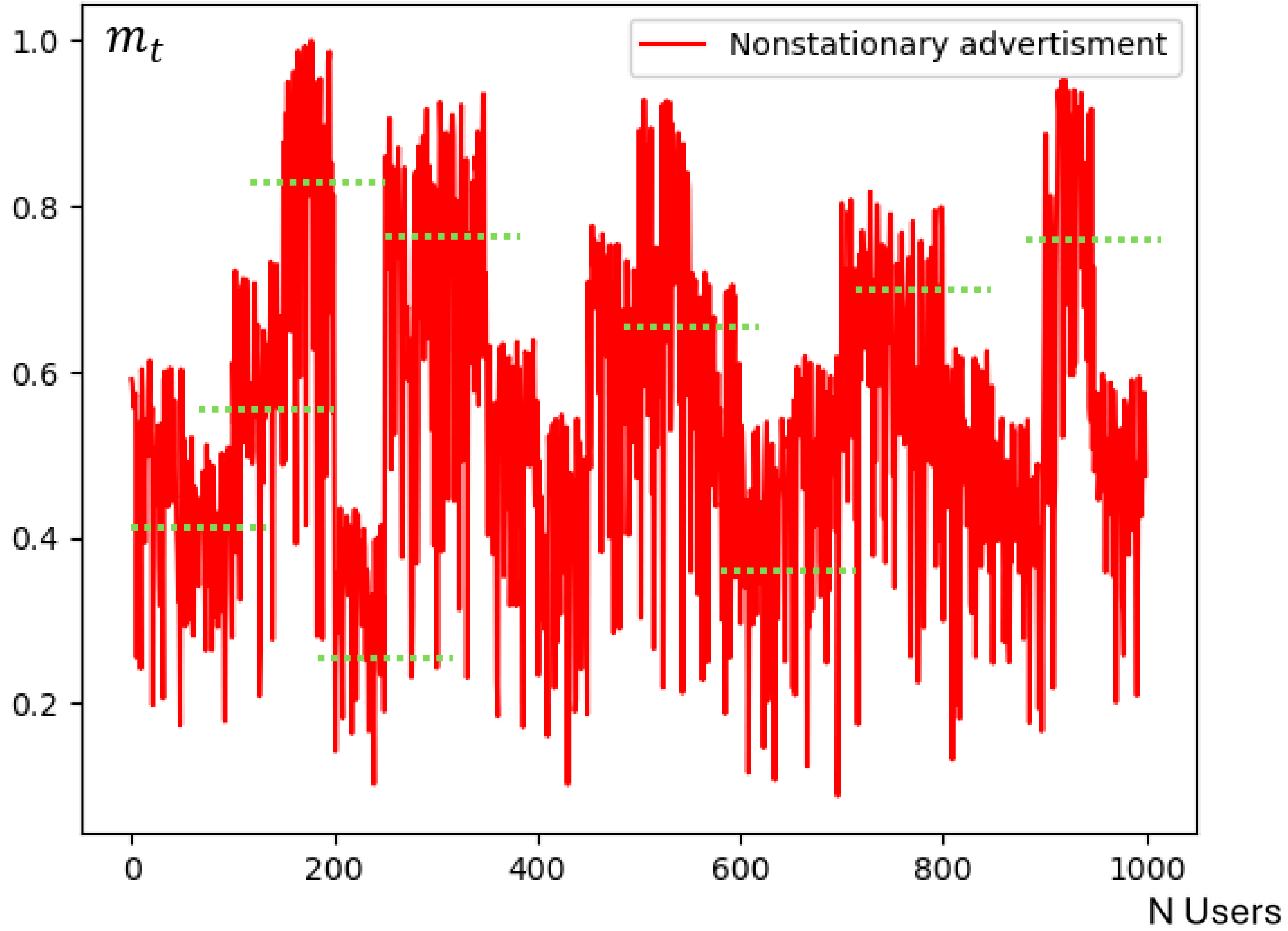
  That is for a competing advertiser his/her bids are sampled from a uniform that changes its range:

$$b_{comp} \sim U(0, b_t)$$

- Therefore the maximum competing bid is also dynamic in nature:

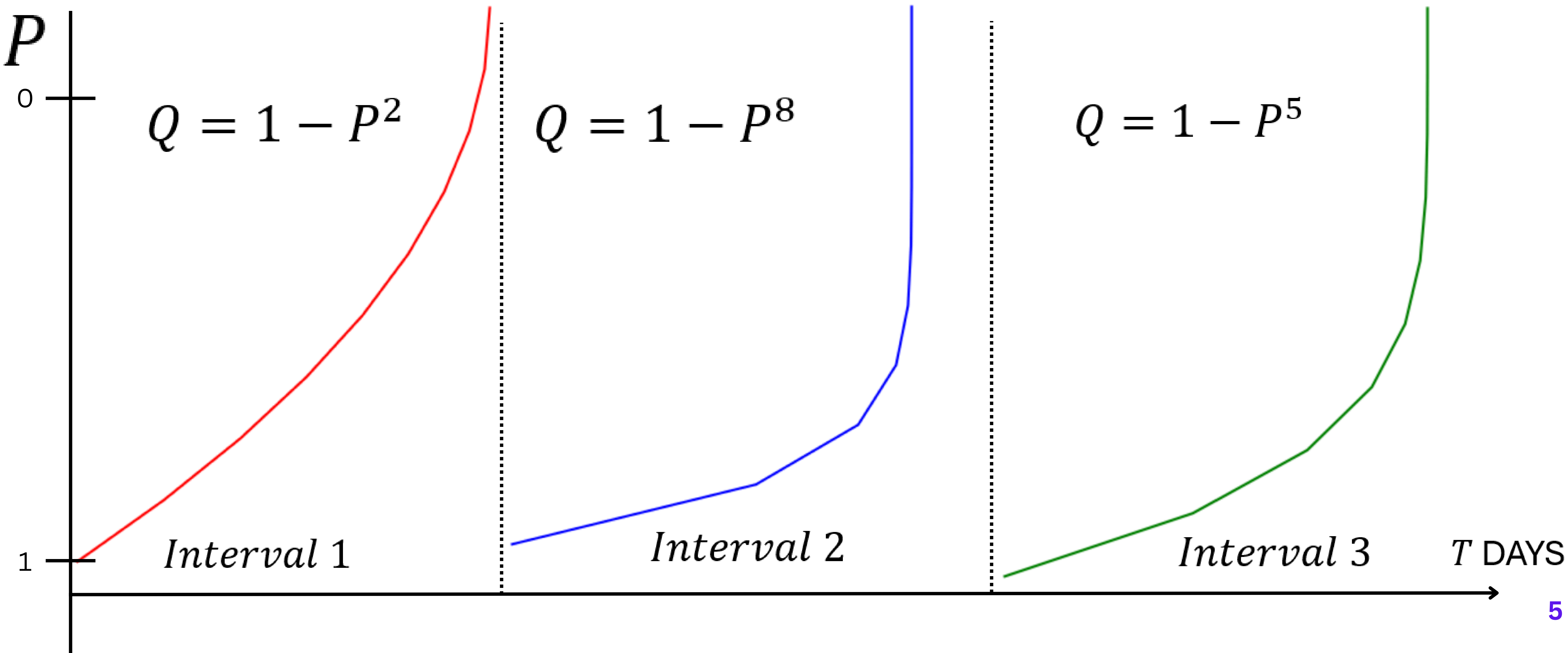
$$m_t \in \max\left(b_{comp}{}^{1}, b_{comp}{}^{2}, \dots, b_{comp}{}^{\#advertisers}\right)$$

- The maximum competing bid appears as a chunks of concentrated shocks. The shocks here follow a level that is varied after the passing of few rounds.



Non-Stationary Stochastic Pricing Environment (1)

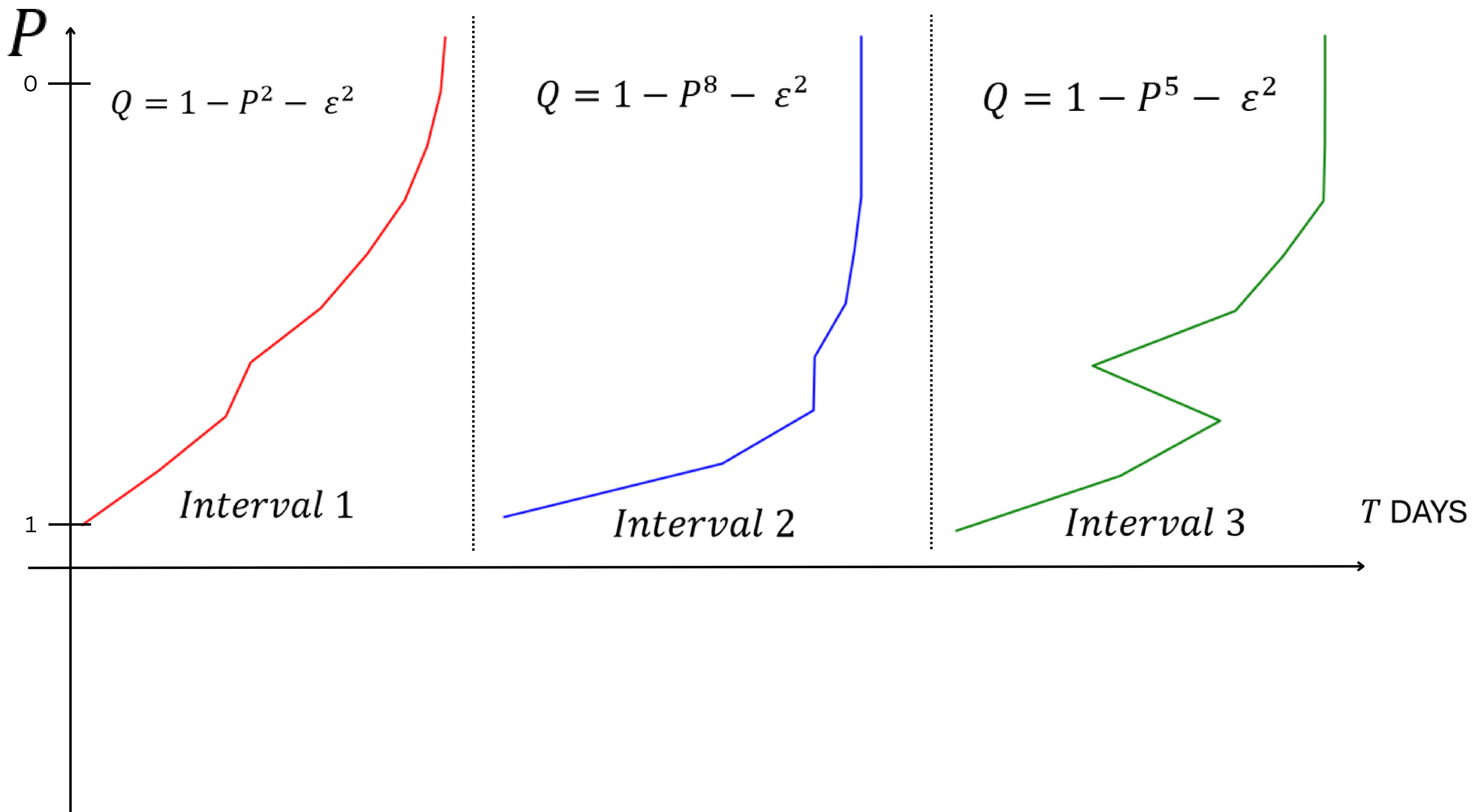
The demand curve changes every while after a few number of days passes, this is typical of seasonal patterns. During the vacations for example there are many customers willing to accept a high price more than they usually do during off-season period.



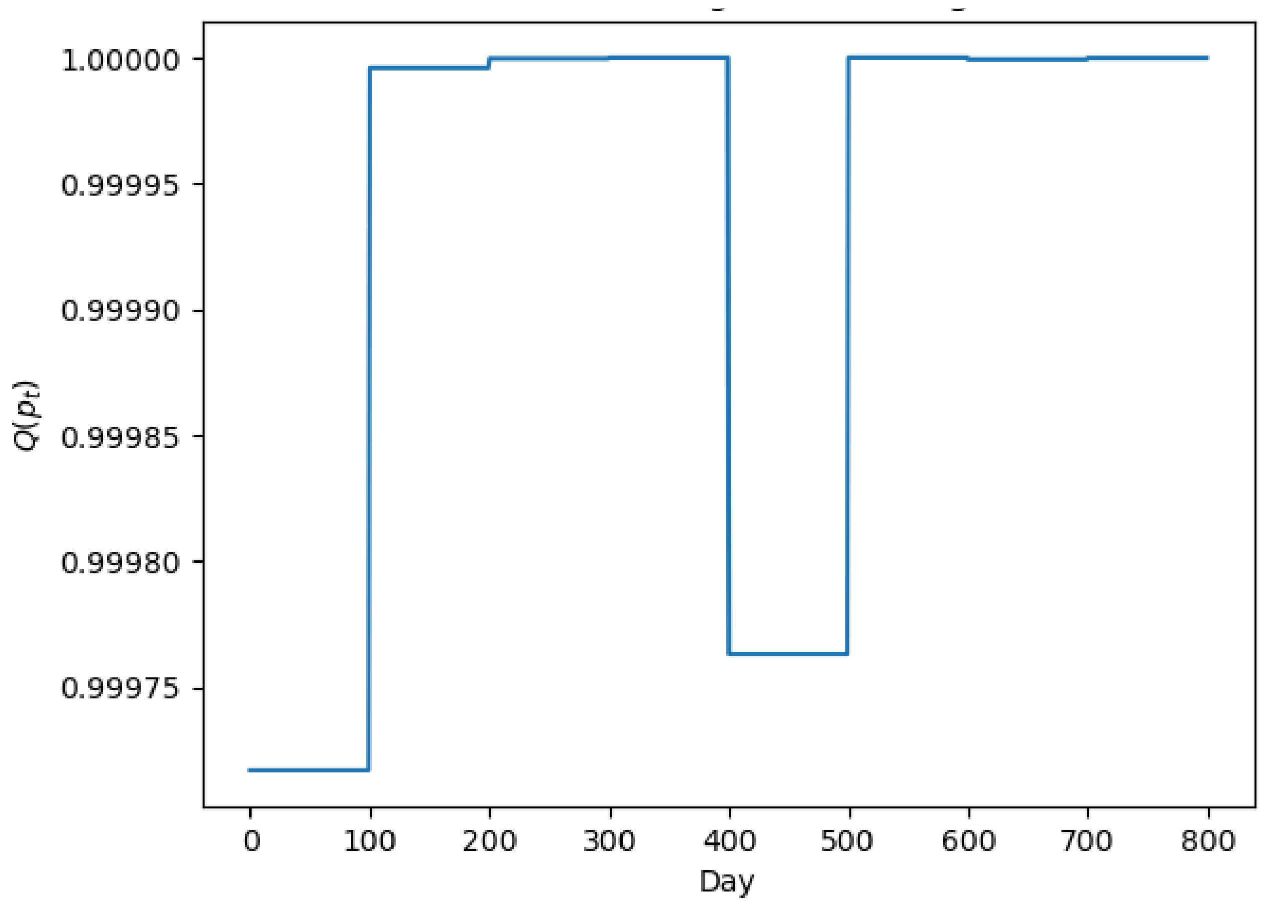
Non-Stationary Stochastic Pricing environment (2)

Accompanying the systematic changes in the demand curve are additional noise that makes the environment closely resembles an actual one. The expected demand at every price deviates from the one expected in the smooth situation.

The assumption behind the noise is that its gaussian with a fixed variance. That is: $\varepsilon \sim N(0, \sigma^2)$



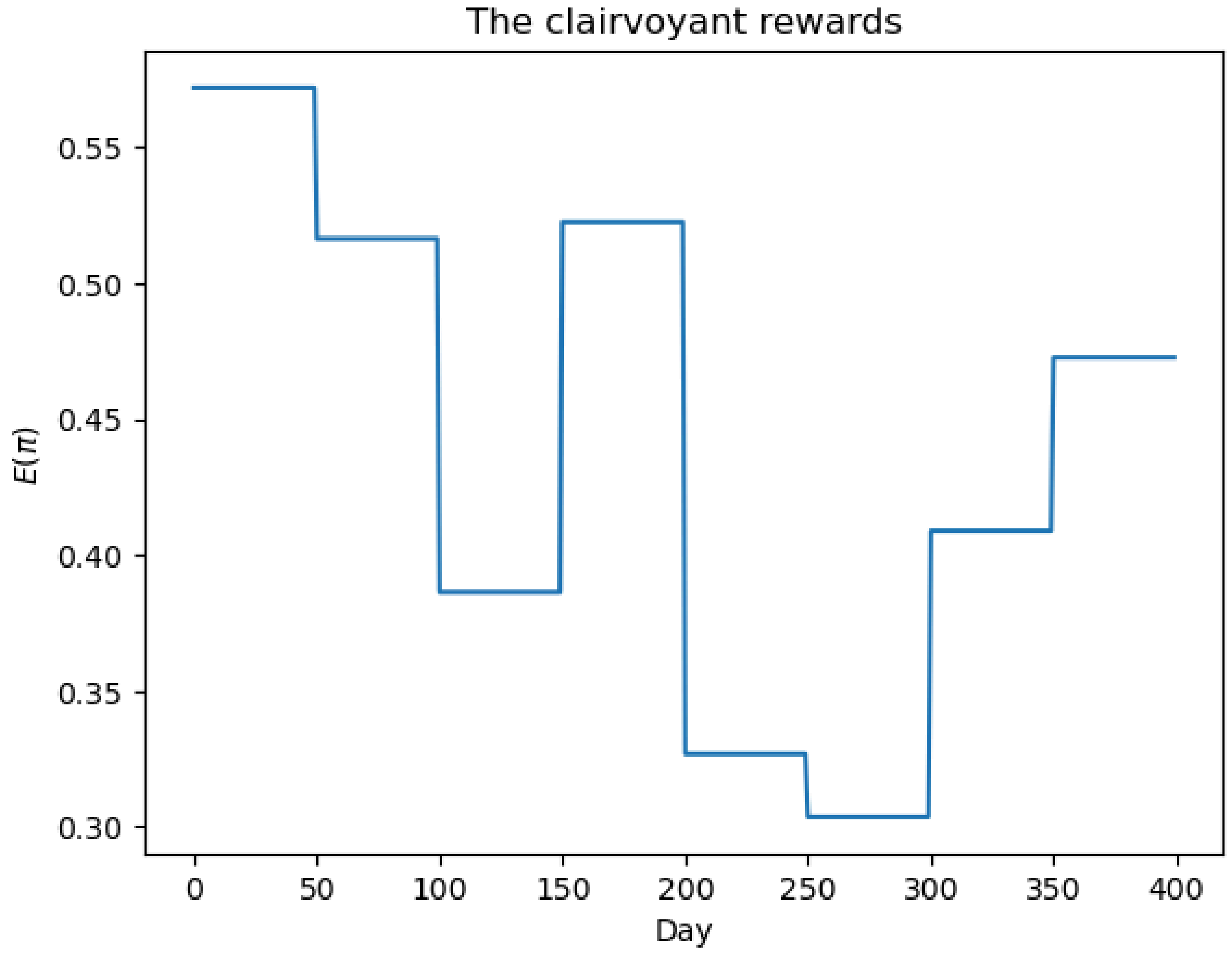
The evolving mean:

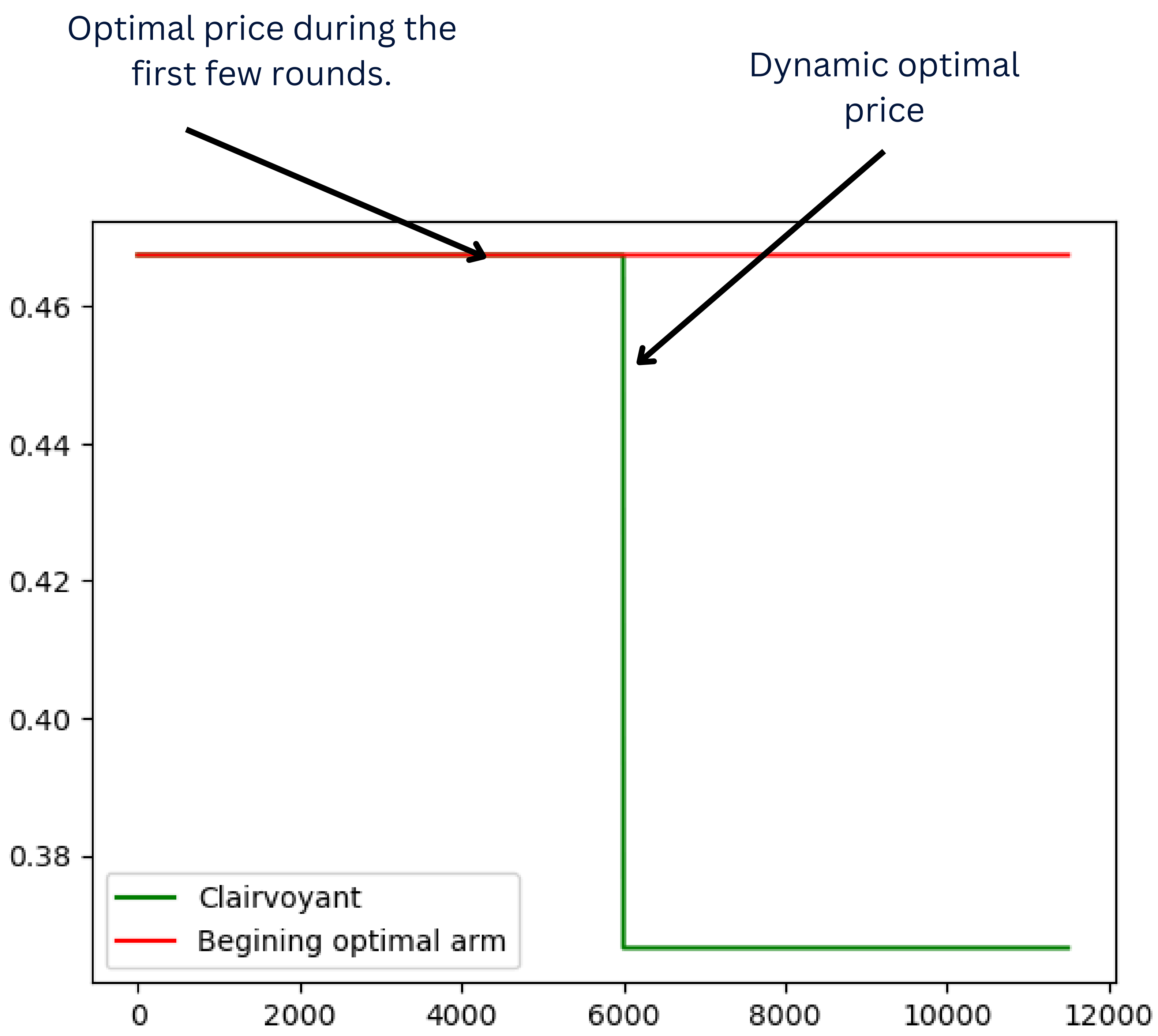


Non-Stationary Stochastic Pricing Environment (3)

The clairvoyant here will be the one pulling always the optimal price in hindsight (the reward is defined in expectation), being the one having access to the true sequence of evolving conversion rates.

The dynamic policy : $$max(\sum_{t \in T} Q * (P - c))$$





Two-Items Stochastic Pricing Environment:

- Here the seller is offering two items to the customer that shows up, its assumed that demand curve of the second item depends not only on its price offered but also on the price offered for the first item. The dependency is created in a way that mimic a well established case known as product-complement.
- The case imagined here is the same airline company offering regular seats and premium seats, obviously the demand on luxury seats can be seen as dependent on the plane regular capacity.
- The same number of customers (N) arrive every day and they are shown both products simultaneously.

Demand on the first item

$$Q_1 = 1 - P_1 - \varepsilon^2$$

$$\varepsilon \sim N(0, \sigma^2)$$

The amount of customers buying product1 depend on the price of the product at that day.

$$n_1 \sim B(N, Q_1)$$

Demand on the second item

$$Q_2 = \sqrt{P_2}(1 - P_1) - \varepsilon^2$$

$$\varepsilon \sim N(0, \sigma^2)$$

The amount of customers buying product2 depend on both the price of the second product at that day and the price of the first one.

$$n_2 \sim B(N, Q_2)$$

The clairvoyant in this setting will try to find the optimal combination of two prices that maximizes its profits (in expectation) and will maintain this combination for the entire time horizon (T) .

$$P_1^*, P_2^* = argmax_{P_1, P_2} E(\pi) = Q_1 * (P_1 - c_1) + Q_2 * (P_2 - c_2)$$

Part 2

Algorithms and results

- Stationary interactive environment
    1. GP-UCB in the combined environment
    2. Results of truthful auction problem
    3. Combining the truthful auction with the pricing
- Non-stationary advertising and pricing environment
    1. Non-truthful pacing agent performance
    2. UCB1 in the combined non-stationary environment
- Non-stationary stochastic pricing and two items ext.
    1. UCB1,SW-UCB and CUSUM-UCB
    2. 2D-UCB and 2D-GPUCB
- Competition among bidding algorithms

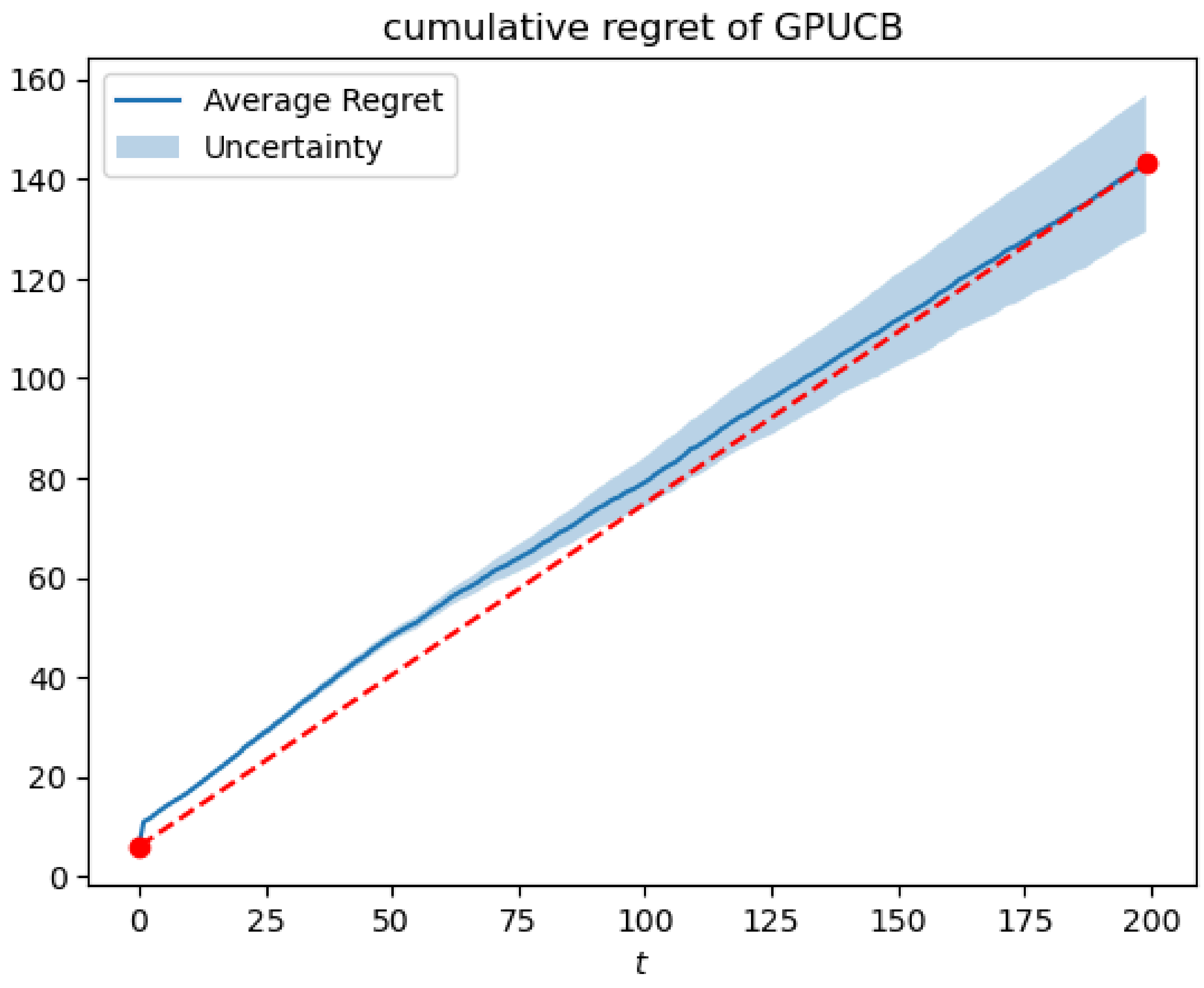
    Bidding like-UCB, FFMP and truthful MP

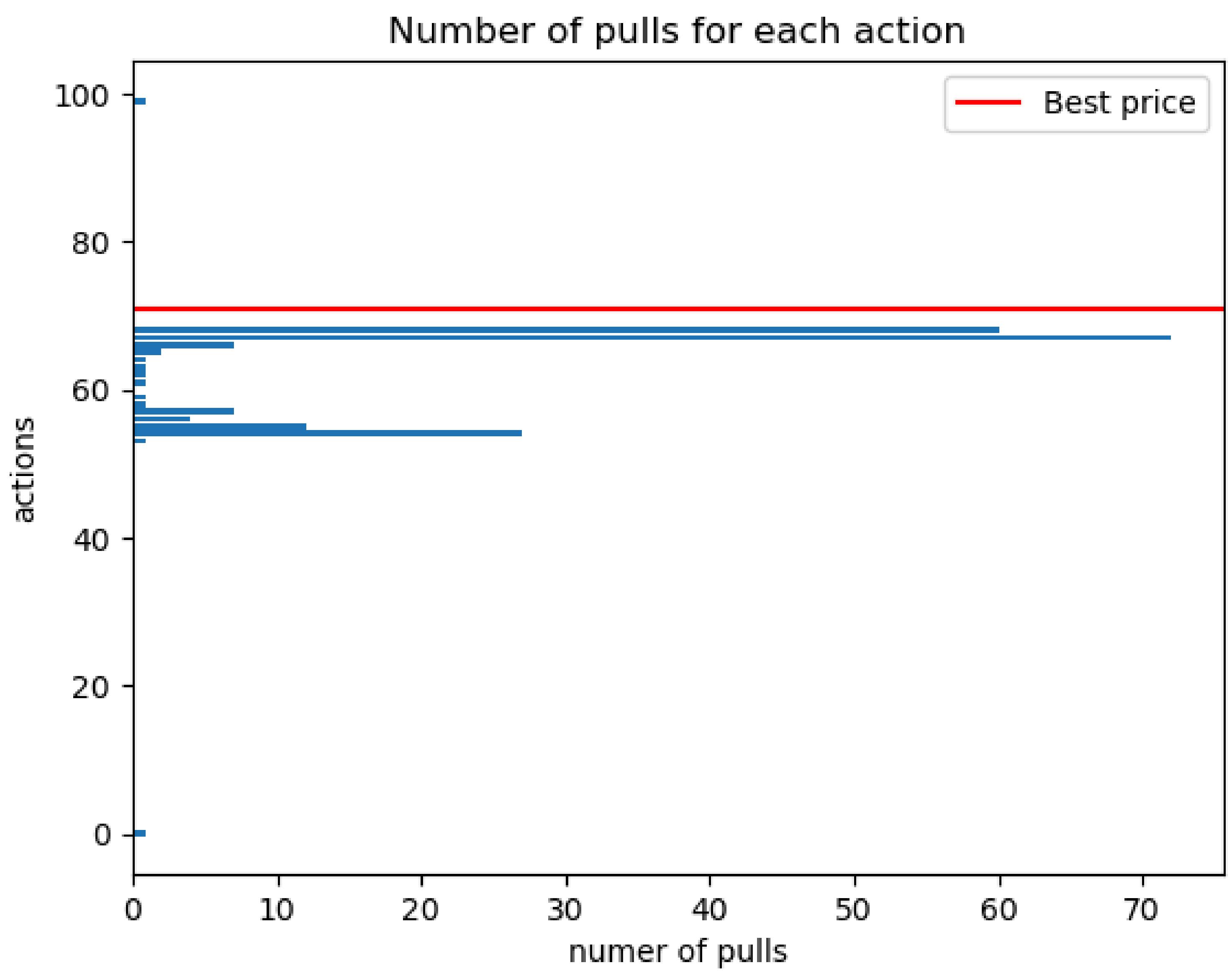
GP-UCB for Pricing In Stochastic Environment

The GPUCB converges to a **sub-optimal** price.

Parameters:

- T = 200
- K = 100
- n_users = 10

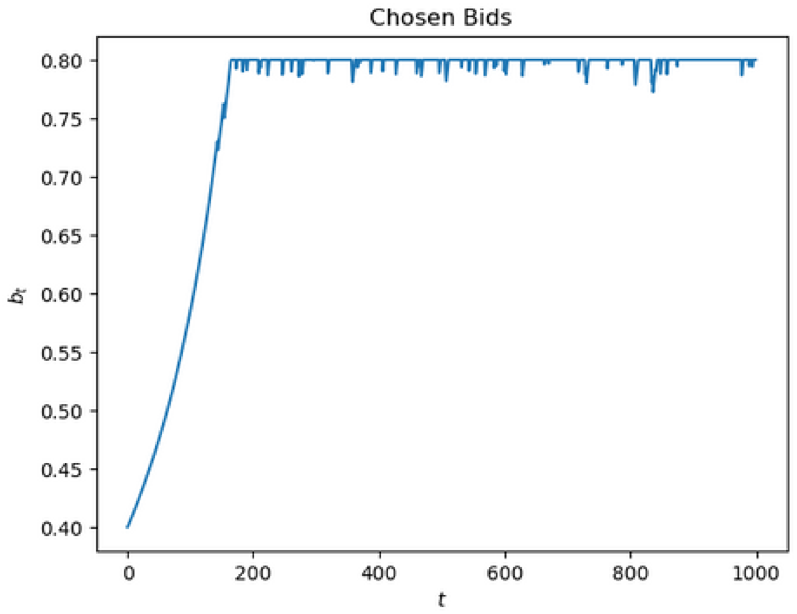


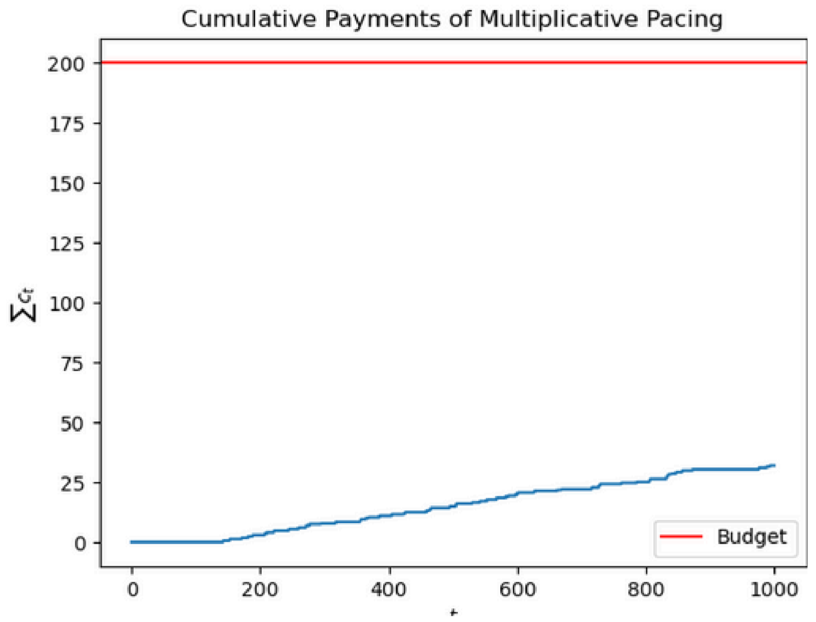


Results of the Truthful Auction Problem

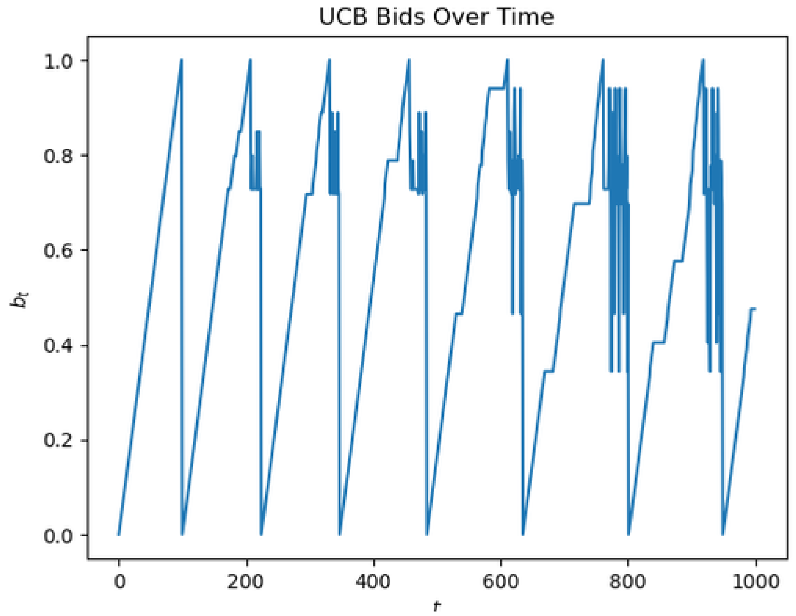
The Multiplicative Pacing Agent successfully converges to the item's price when evaluated under the optimal strategy, assuming a click-through rate of one. In contrast, the UCB-like Agent faces challenges in convergence, potentially due to an excessively large exploration coefficient. Additionally, the second-price auction mechanism (truthful auction) motivates advertisers to bid their true valuations.

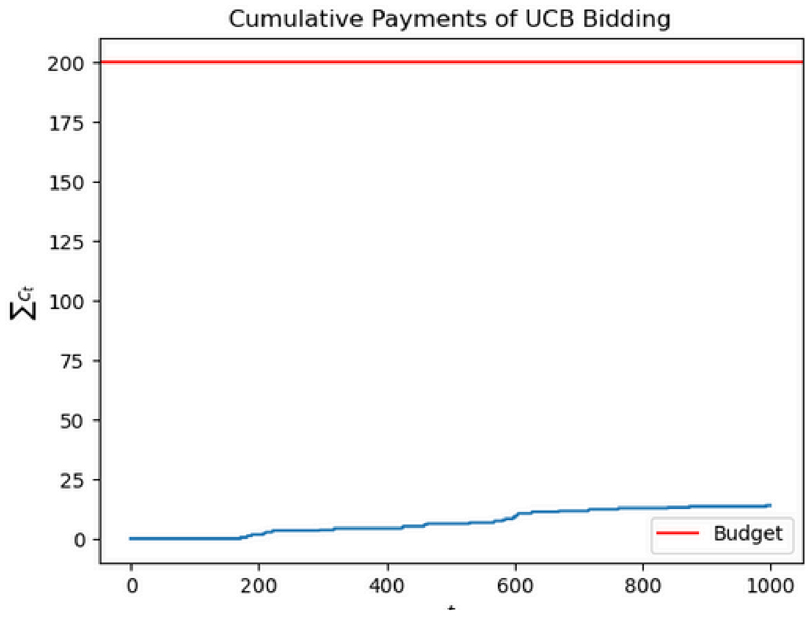
Multiplicative Pacing Agent





UCB-like Agent





Regrets

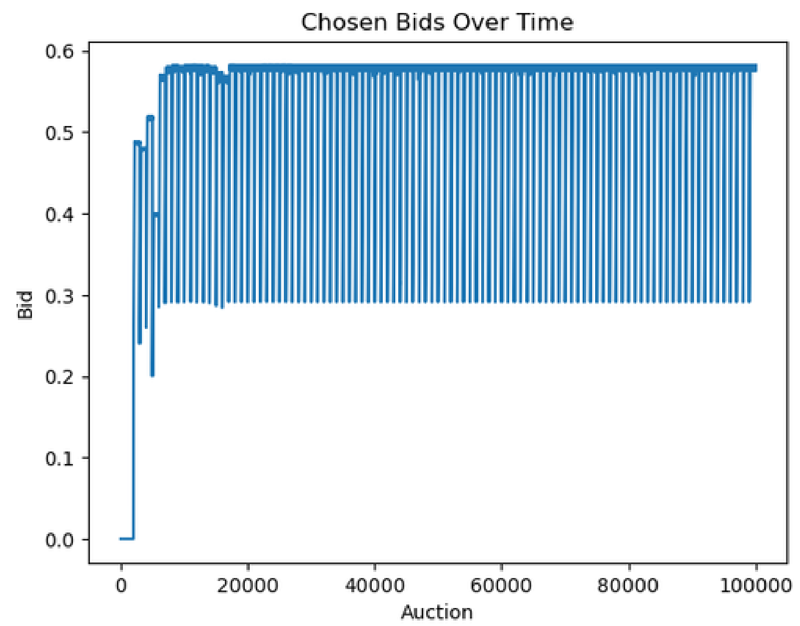
Combining the Truthful Auction and Pricing

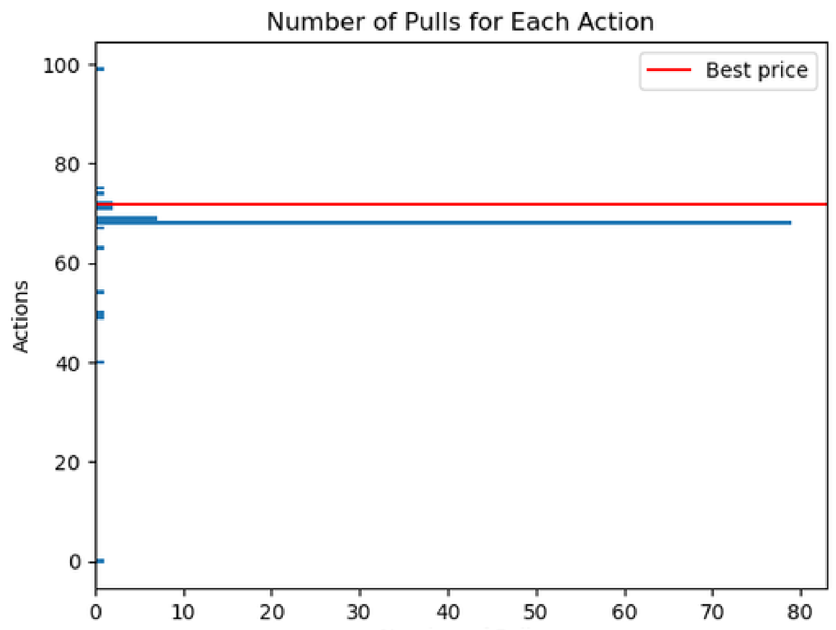
Parameters:

- T = 100
- K = 100
- cost = 0.1
- conversion_probability = lambda p: 1 - p**5
- n_users = 1000

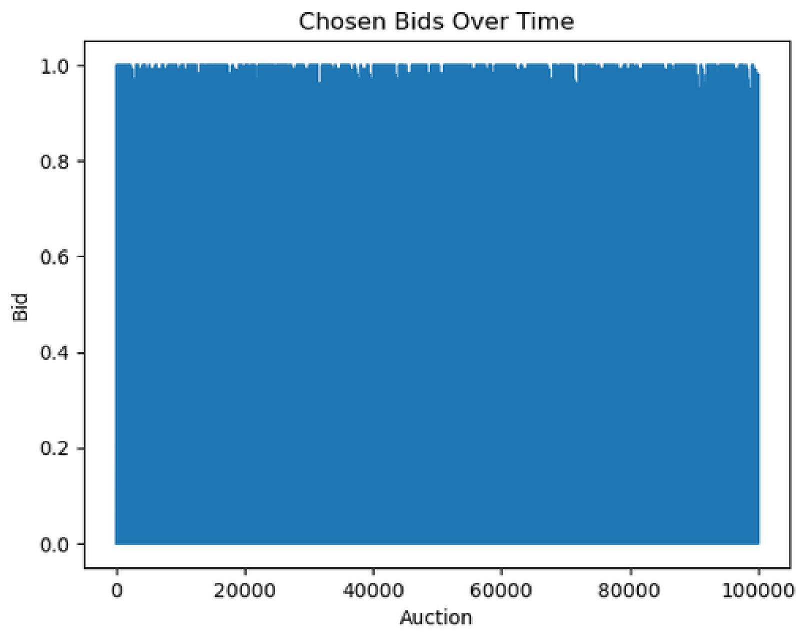
As expected in the pricing problem, the Multiplicative Pacing Agent outperformed the UCB-like Agent. The UCB-like Agent consistently selected a sub-optimal price of 1, likely attempting to minimize losses in each round.

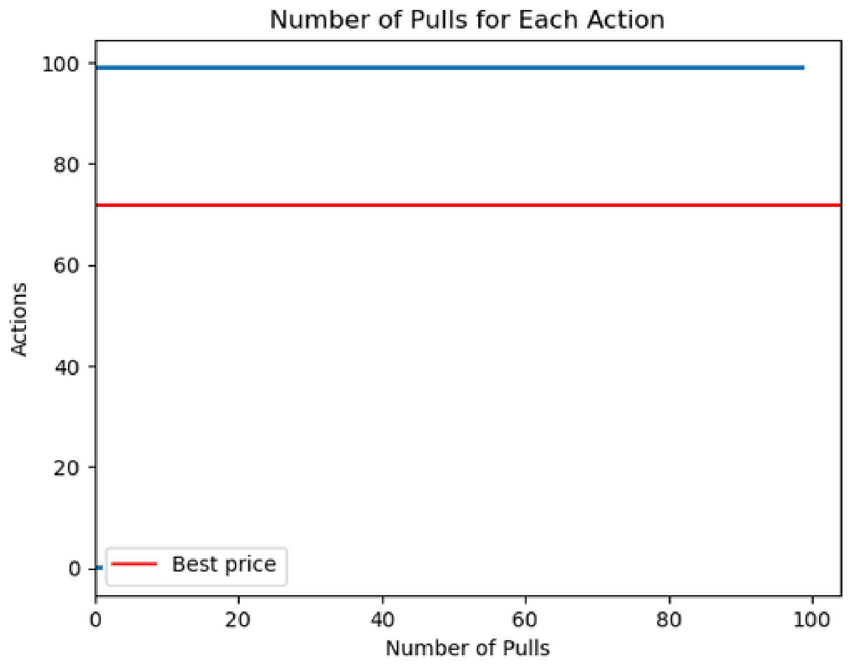
Multiplicative Pacing Agent



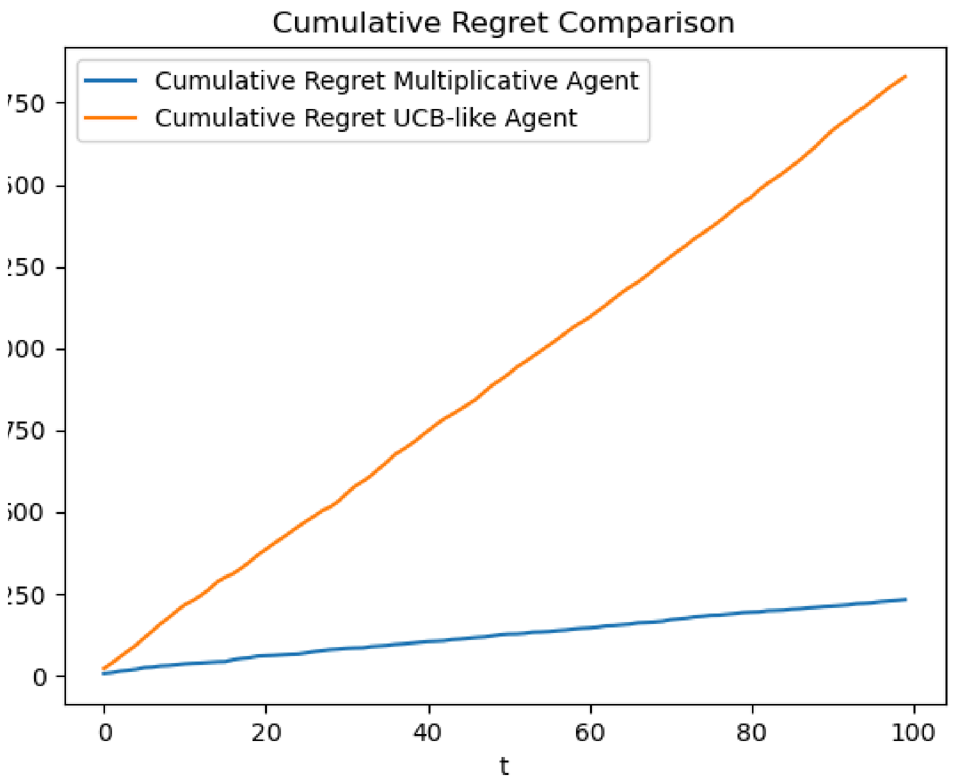


UCB-like Agent



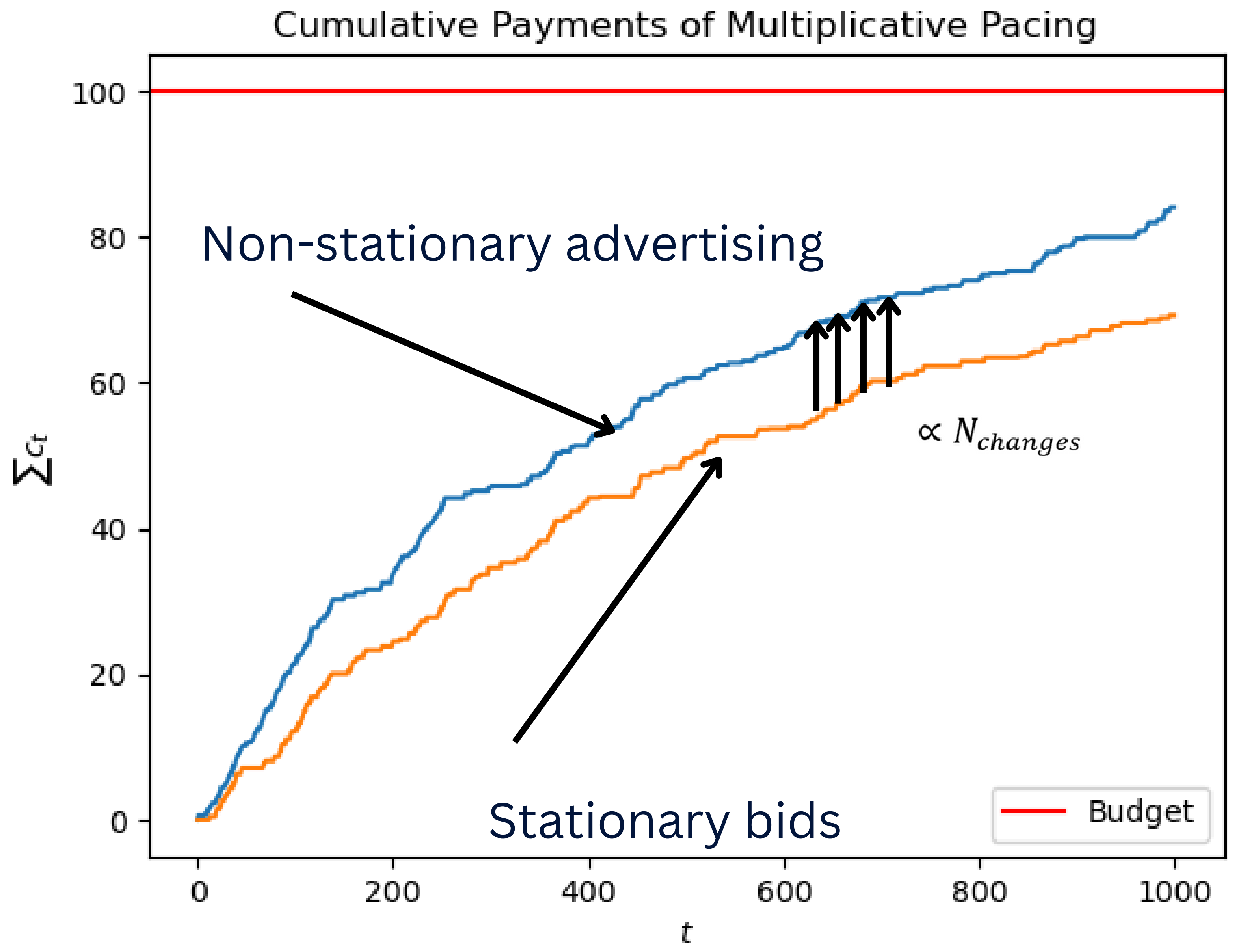


Regrets



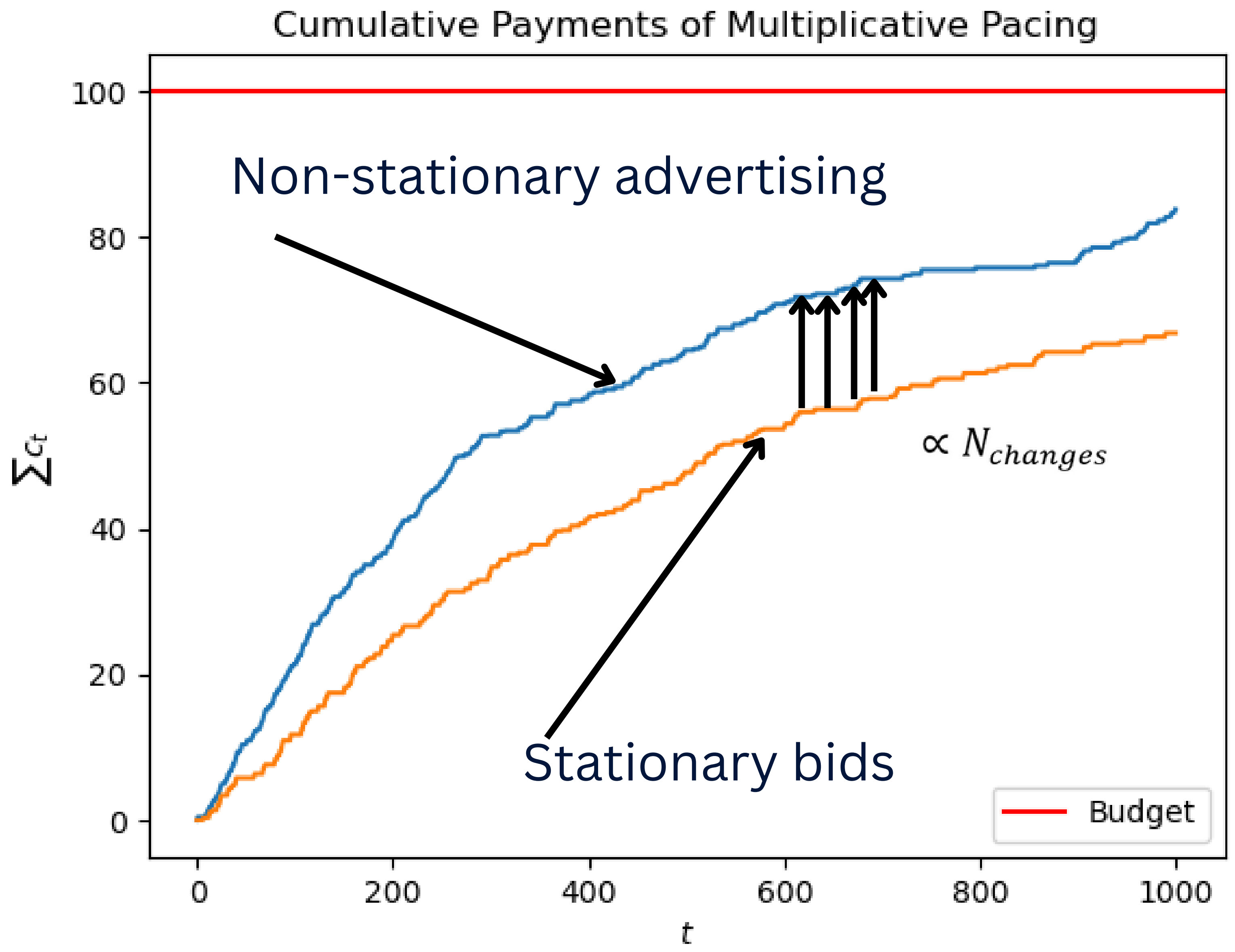
FFMP Agent in Non-Stationary Bidding:

The First price pacing agent is able to handle the non-stationary competing bids at the expense of higher expenditure. Comparing to the stationary advertisement case the gap, which represents the additional spending, is proportional to the number of abrupt changes.



N = 1000

$N_{changes} = 5$



N = 1000

$N_{changes} = 10$

UCB1 (Discretized pricing) Against the Combined Non-Stationary Environment:

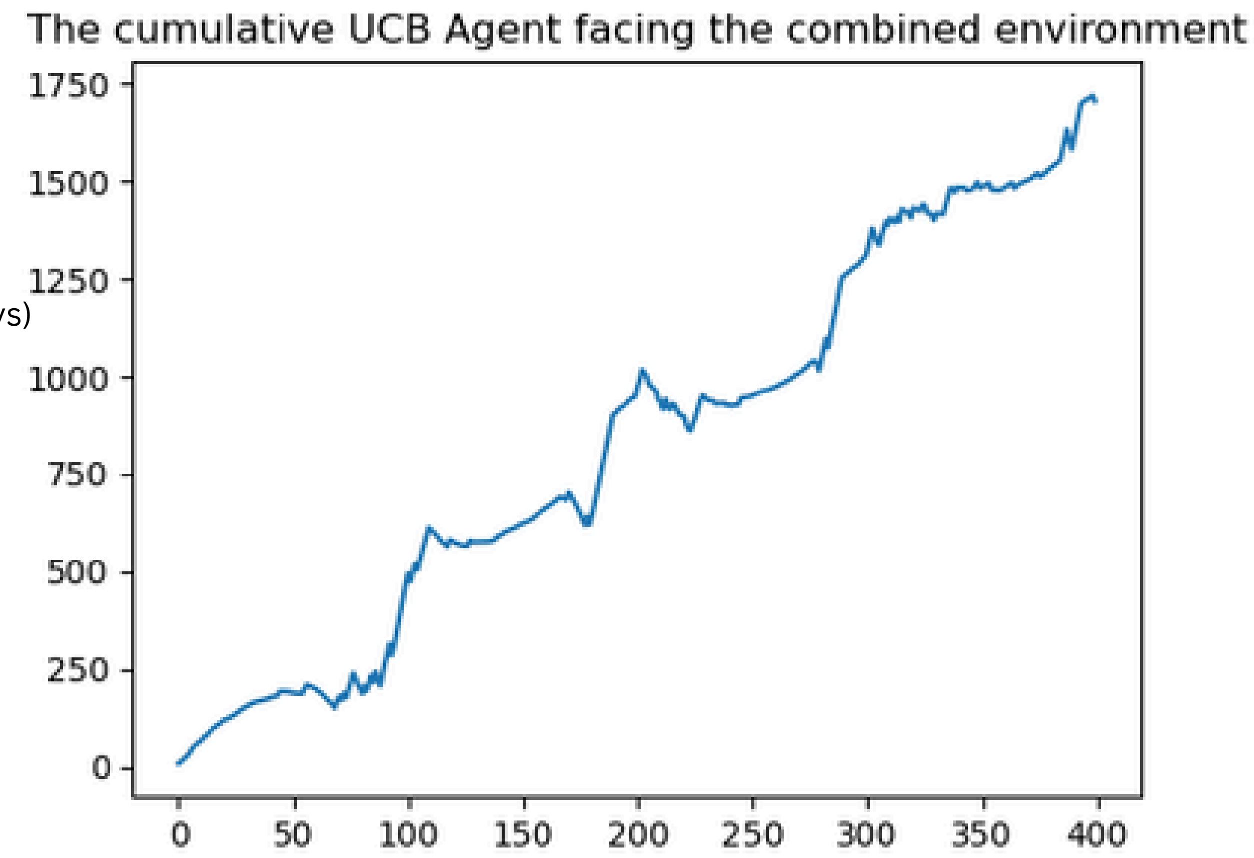
- Here we let UCB1 face the interactive advertising-pricing environment and evaluate its perfomance based on a baseline also facing the combined environment.
- The clairvoyant wins the bids optimally knowing in advance the sequence of competing bids, and finally shows the users the best dynamic optimal price.
- First the advertising agent (FFMP) wins a certain amount of users and that number is the input for the UCB1 algorithm as the number of customers shown a given price on a particular day.

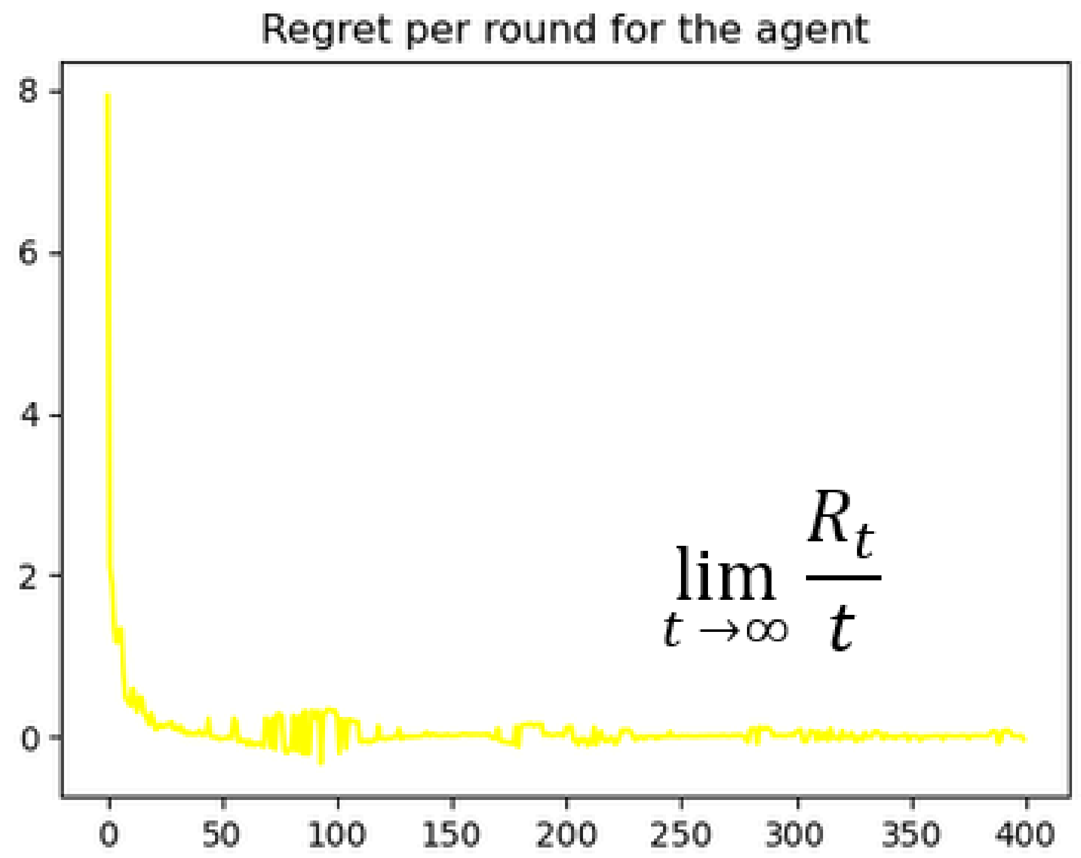
(Budget, Daily users, Days)

B = 100

N = 150

T = 400





$$\lim_{t \to \infty} \frac{R_t}{t}$$

- A combination of FFMP and UCB1 is able to achieve a sub-linear regret against a truthful clairvoyant with optimal dynamic policy.
- A possible reason being the agent winning more users that the clairvoyant at some instances.

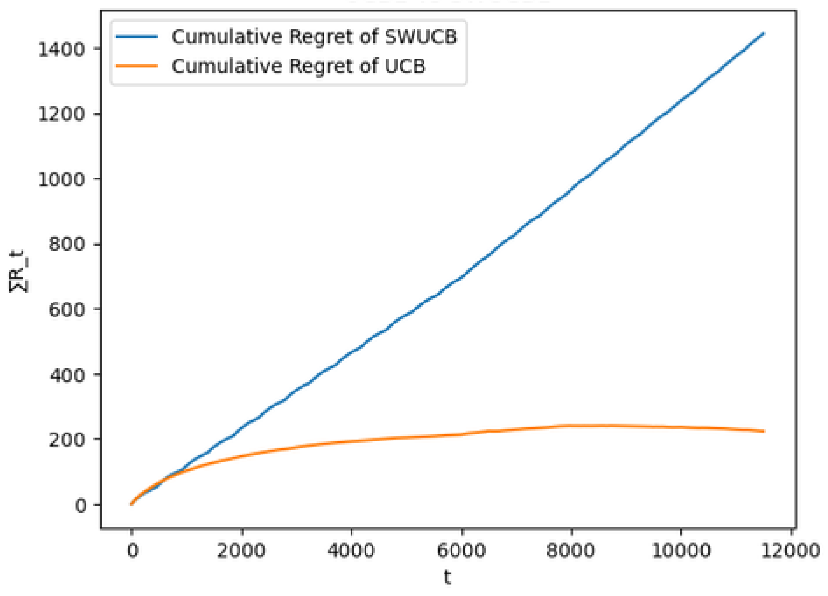
UCB1, SW-UCB, CUSUM UCB:

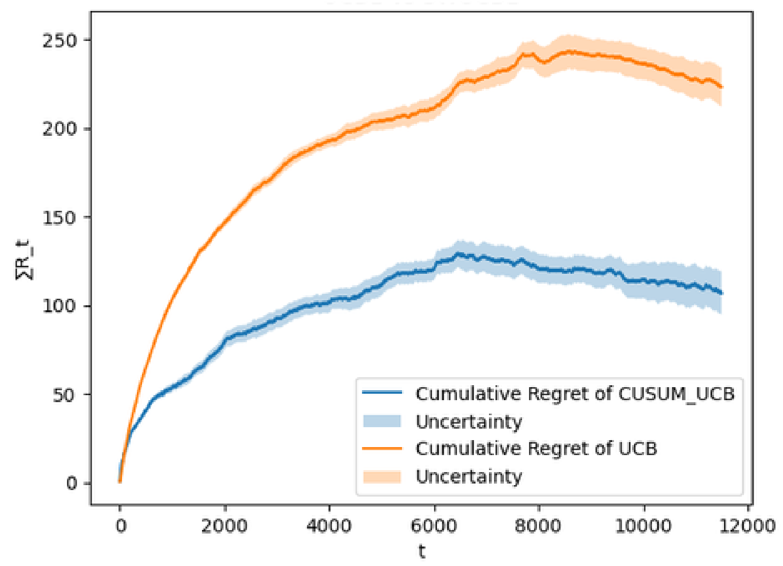
Three algorithms UCB1, Sliding-Window UCB and CUSUM UCB are facing the environment, we would expect that non-stationary algorithms will perform better always, but UCB1 seems to have performed better than SW-UCB. Probably SW-UCB is not able to balance the exploration-exploitation well.

$T = 11500$

$N_{changes} = 2$

K = 30

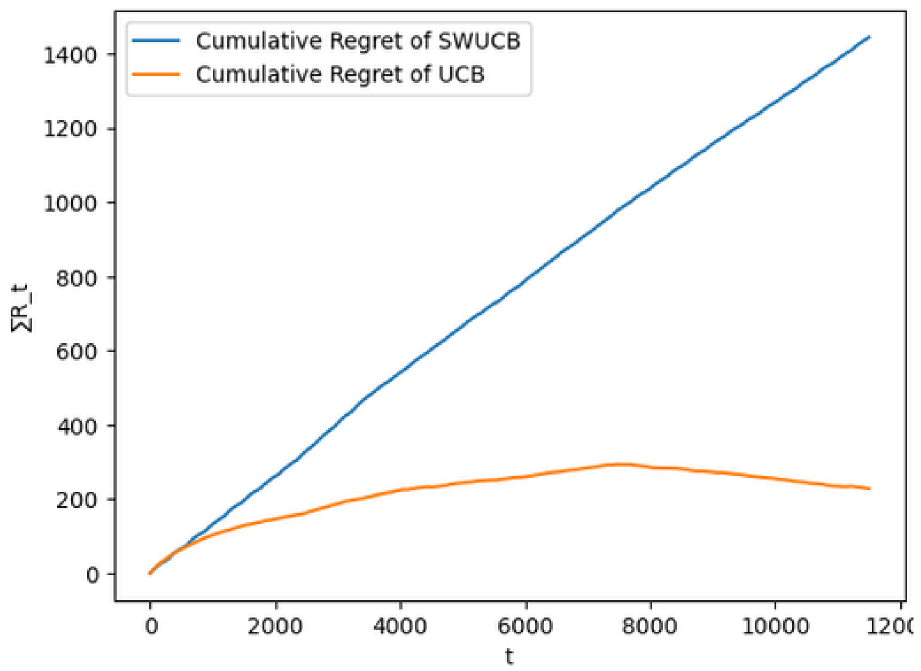


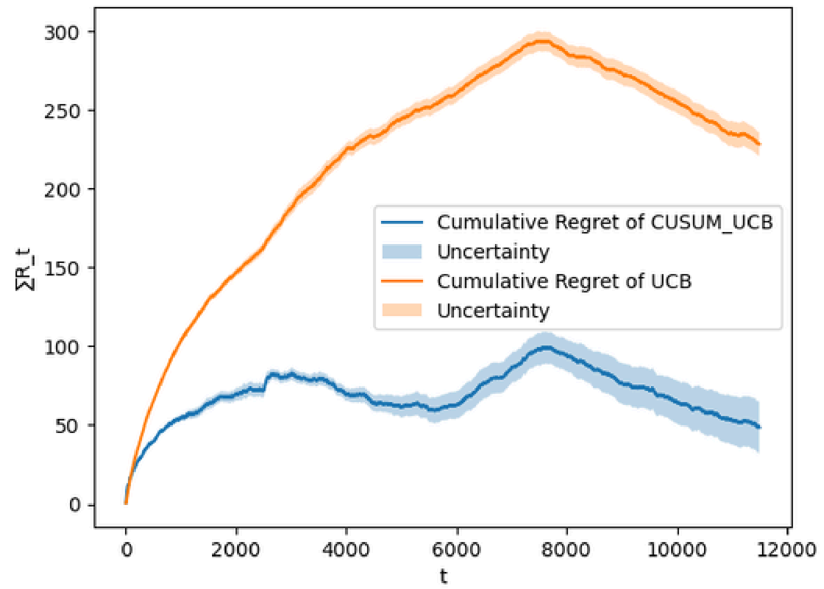


$T = 11500$

$N_{changes} = 5$

K = 30

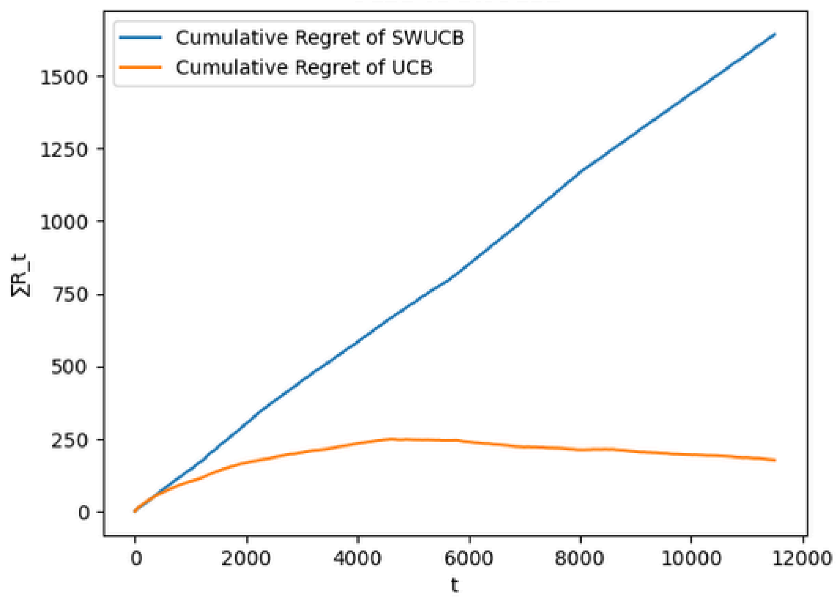


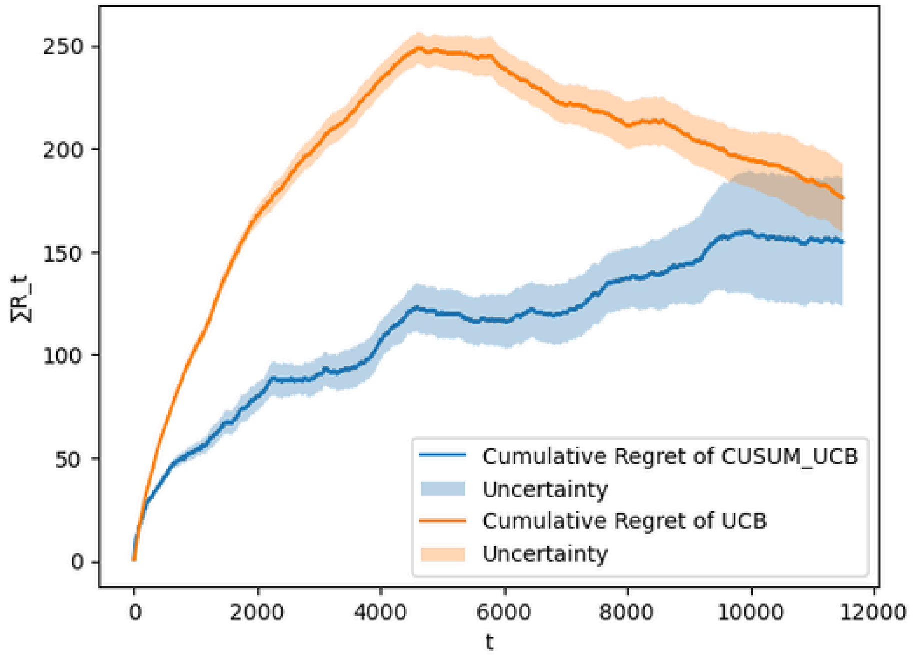


$T = 11500$

$N_{changes} = 10$

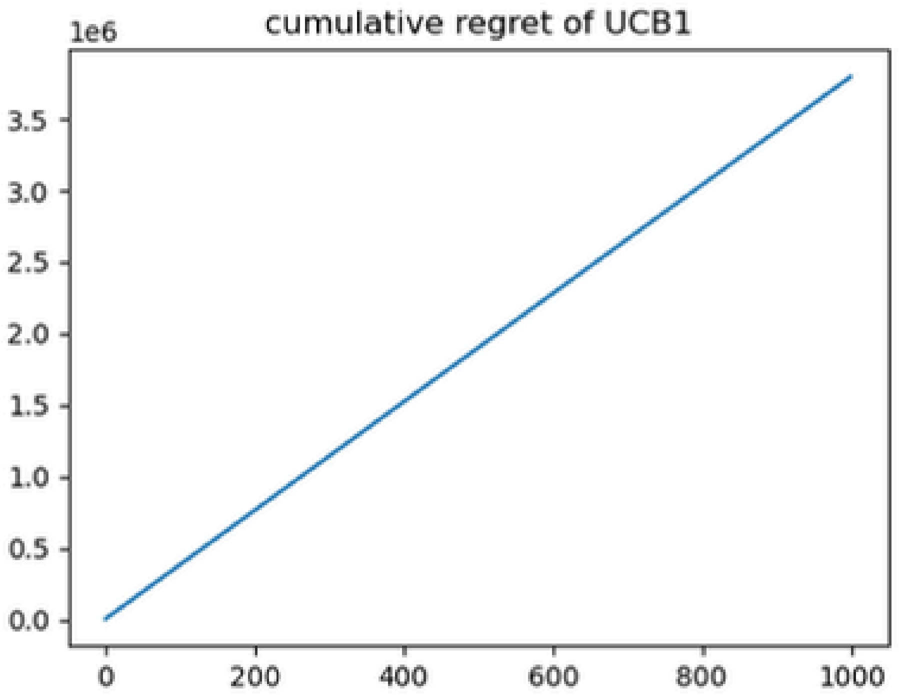
K = 30

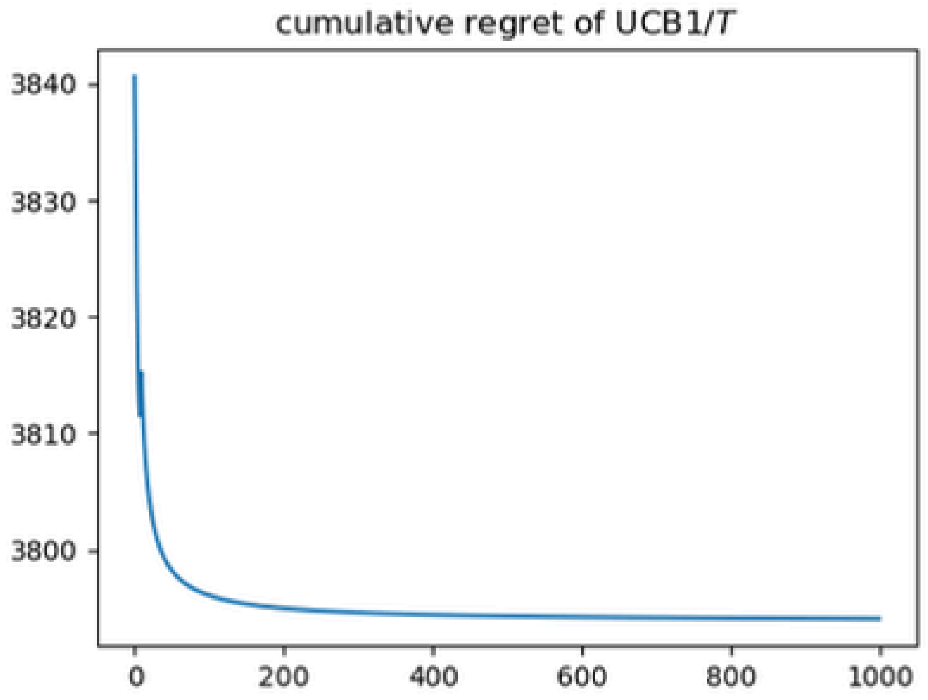


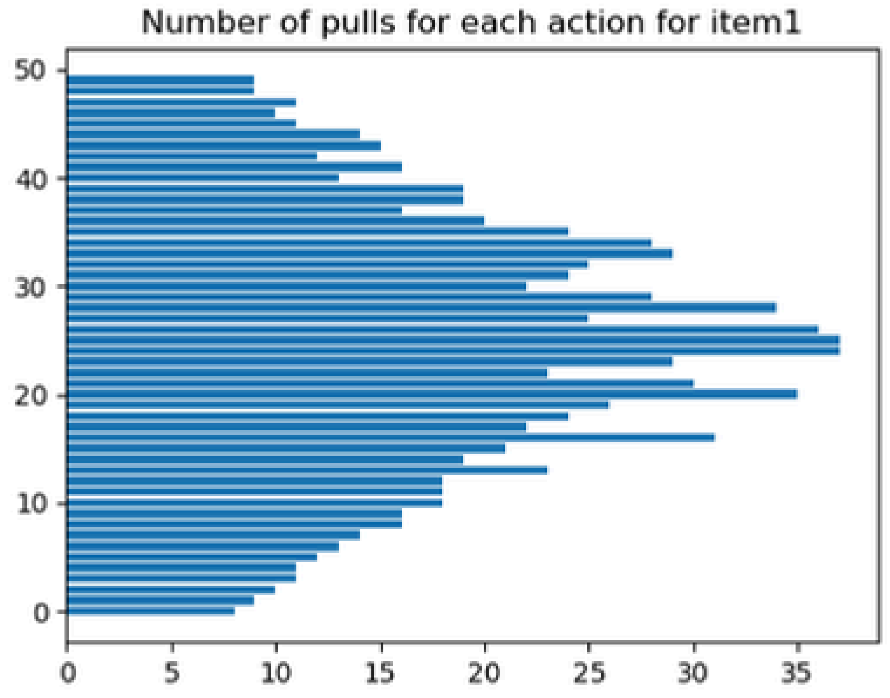


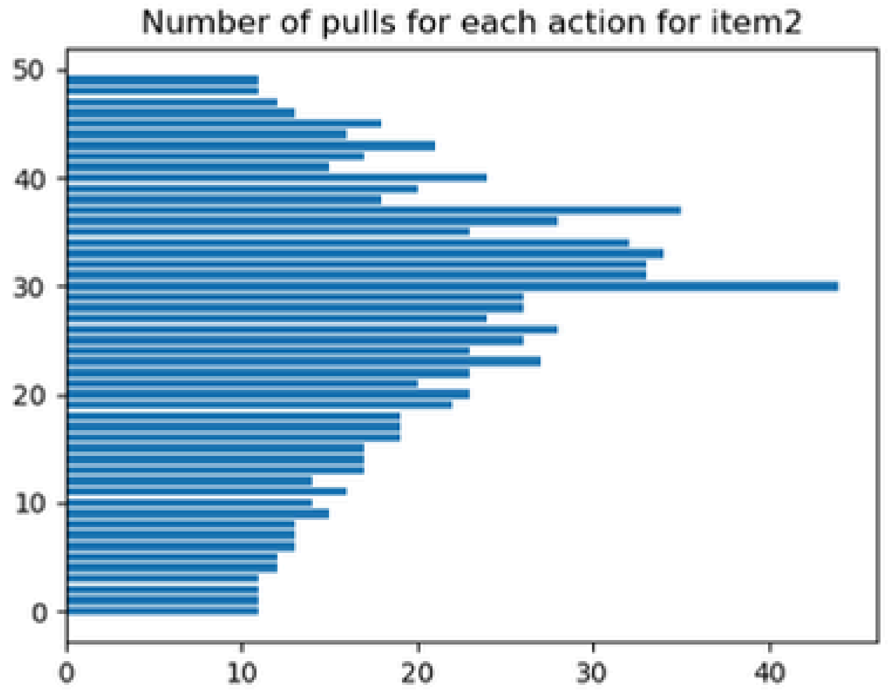
2D-UCB1, 2D- GPUCB:

The 2D UCB1 incurs a linear regret since it will have to explore a 2D space then increases the exploration time, and using lower number of arms makes it also difficult to find the optimal arm.

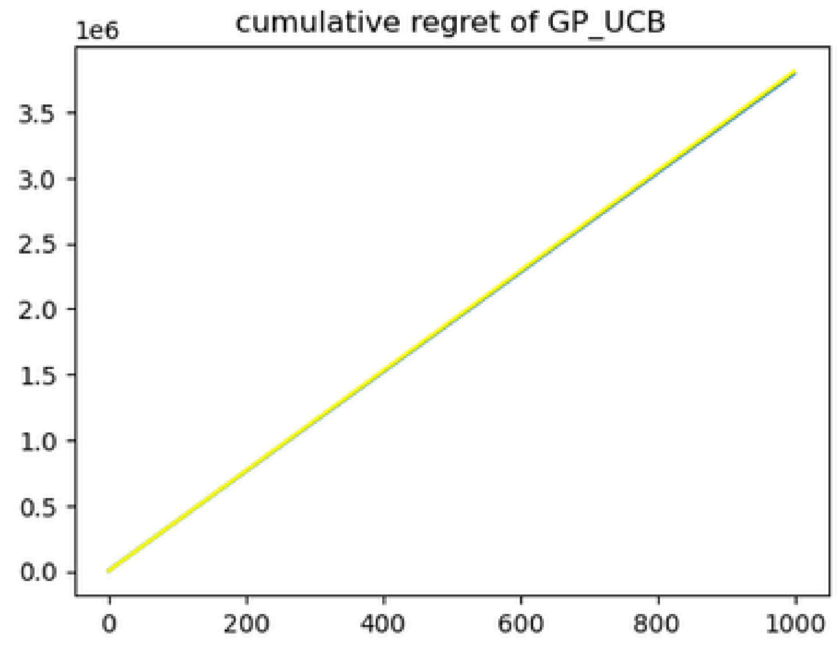




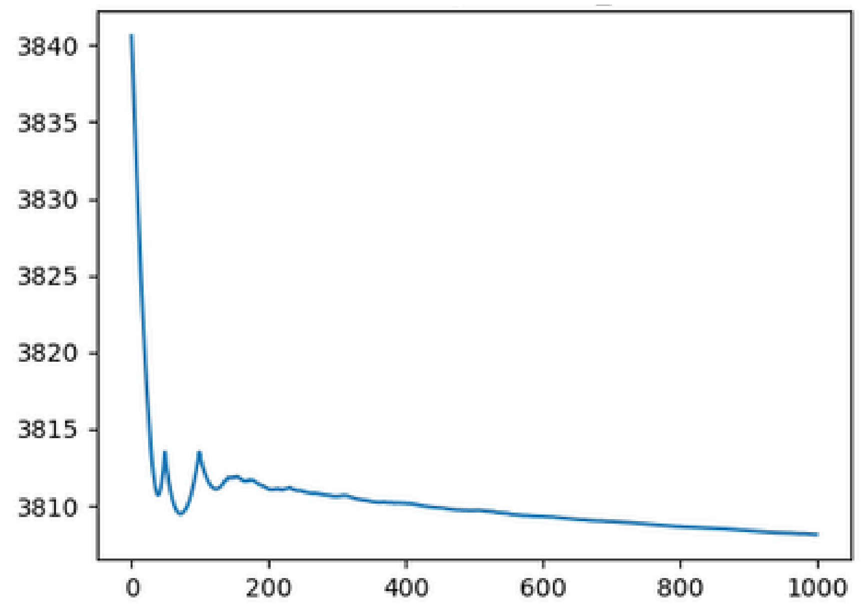


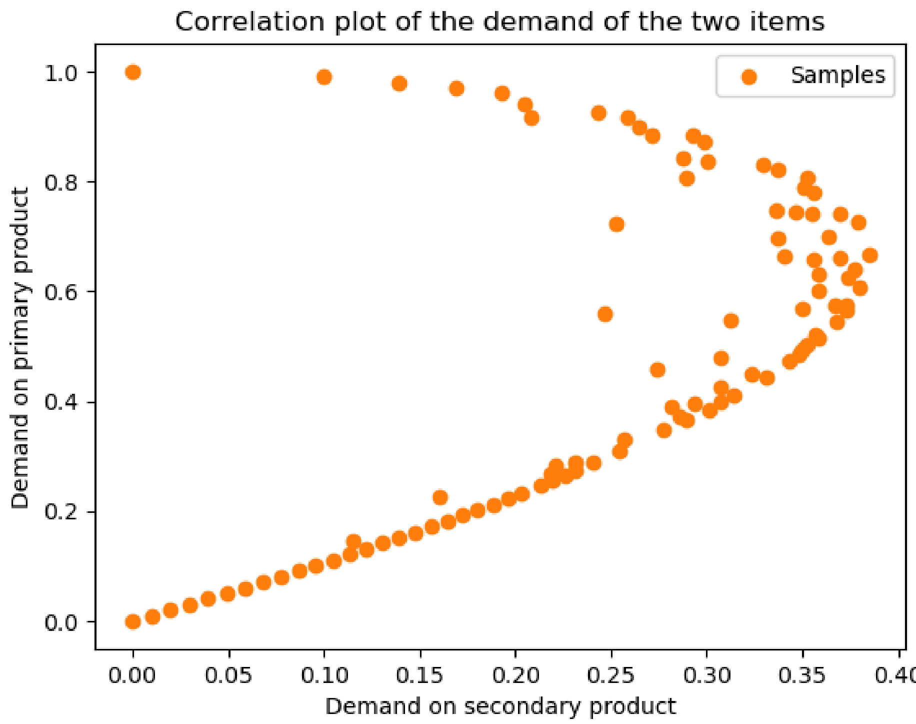


Unfortunately the 2D-GPUCB also incurs a linear regret, The parameter that balance the exploration-exploitation might be chosen differently to the manner we did mimicing in the case of 1D.



Regret per round

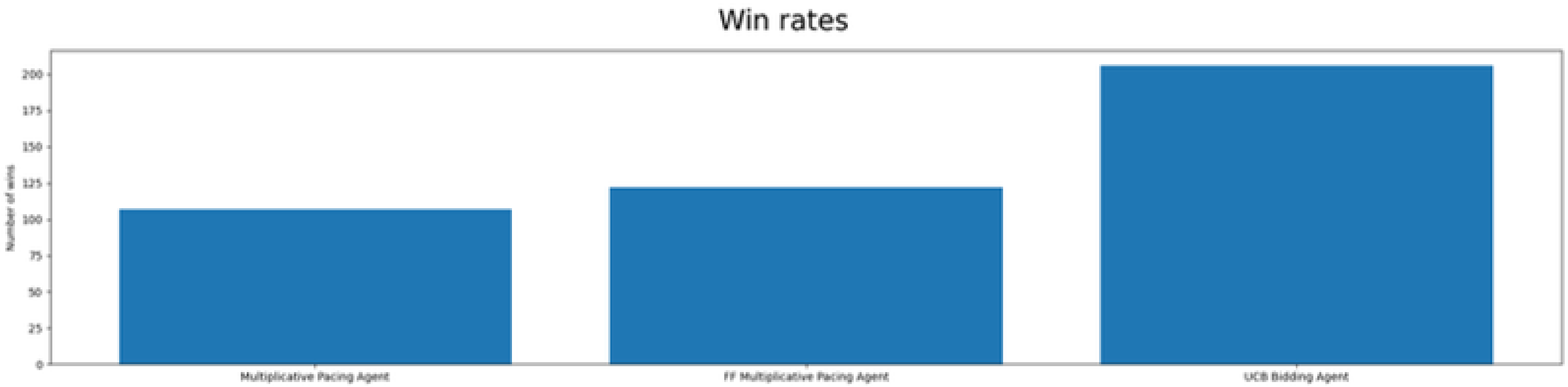




Scatterplot of the correlation between the two items.

Bidding like UCB, FFMP, MP (1):

- In a genaralized first price auction three algorithms are playing against each other to see which one is able to get the hugest number of wins given the same budget.
- UCB for a **large discretization** number and **large number of rounds** is able to win more, since the large discretization number makes UCB slow in depleting its budget.
- In the case of large number of slots no clear results were obtained.



N = 1000
K = 1000

N = 100
K = 1000

Bidding like UCB, FFMP, MP (2):

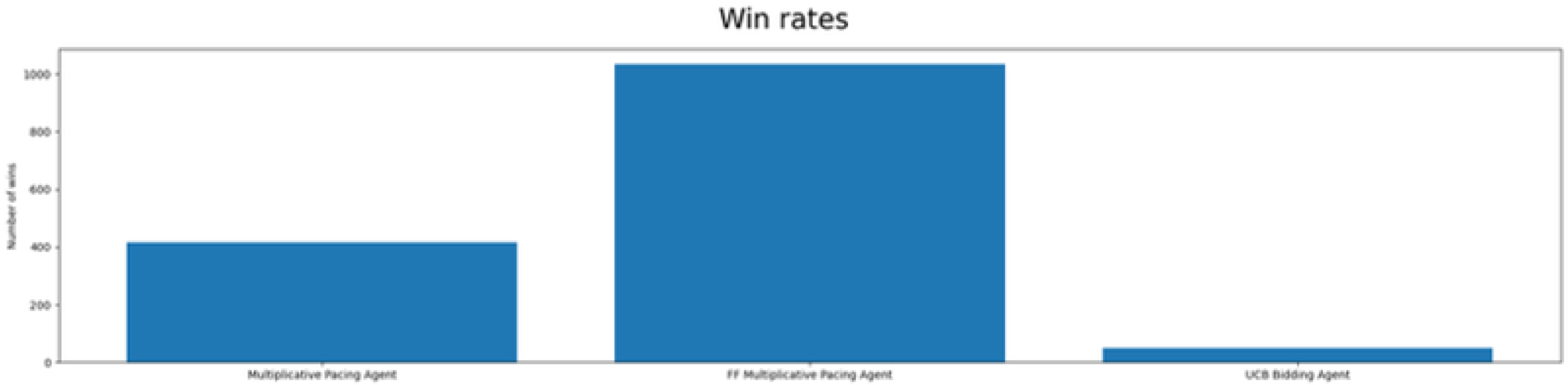
- When the budget is huge compared to the number of rounds the truthful multiplicative pacing agent outperforms.
- This agent is more likely to bid higher and therefore for generous budgets its expected to perform better.

N = 1000

K = 30

B = 1000

- For large budgets and longer auctions the non-truthful auction emerges as the winner.



N = 1500

K = 30

B = 1000

THANK YOU

Q&A