Data Intelligence Applications

Pricing and Matching project – A.Y. 2020/2021

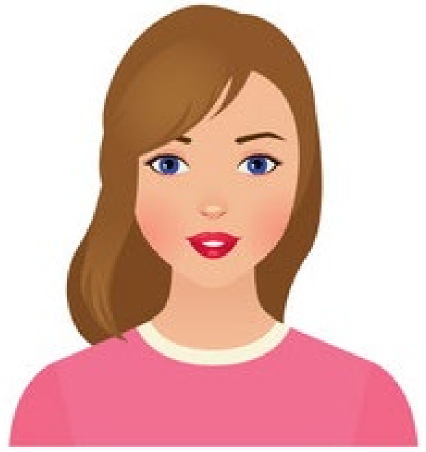
Group: Letizia Brambilla, Francesca Pietrobon, Diego Savoia, Francesco Emanuele Stradi

Scenario: Items

Apple Watch (€300)

Personalized wristband (€50)

Scenario: Customer classes



Class 1: females
up to 35 years old

Class 2: males up
to 35 years old

Class 3: females
older than 35 years old

Class 4: males
older than 35 years old

Scenario: Promos

Promo P0
No discount

Promo P1
10% discount

Promo P2
20% discount

Promo P3
50% discount

Google Form

Quanti anni hai? *

La tua risposta

Sei maschio o femmina? *

- Maschio
- Femmina

Hai comprato o compreresti un Apple Watch (orologio tecnologico)? *

- Sì
- No

Immagina di averlo comprato e che ti venga dato un cinturino base

Compreresti un cinturino personalizzabile e di maggiore qualità a 50 euro? *

- Sì
- No

Se invece avessi uno sconto del 10% (45 euro) sul cinturino personalizzabile, lo compreresti? *

- Sì
- No

Se invece avessi uno sconto del 20% (40 euro) sul cinturino personalizzabile, lo compreresti? *

- Sì
- No

Se invece avessi uno sconto del 50% (25 euro) sul cinturino personalizzabile, lo compreresti? *

- Sì
- No

Google Form

984	31/03/2021 11.21.19	23	Maschio	Sì	No	No	No	Sì
985	31/03/2021 11.23.52	21	Maschio	No	No	No	No	No
986	31/03/2021 11.25.05	26	Maschio	No	No	No	No	Sì
987	31/03/2021 11.25.12	23	Femmina	No	No	No	No	Sì
988	31/03/2021 12.40.50	21	Maschio	No	No	No	No	No
989	01/04/2021 9.05.17	21	Femmina	No	No	No	No	Sì
990	01/04/2021 16.06.01	54	Maschio	Sì	Sì			
991	01/04/2021 20.38.45	48	Femmina	Sì	No	No	No	Sì
992	01/04/2021 23.31.15	18	Maschio	No	No	No	No	No
993	02/04/2021 10.53.46	23	Femmina	No	No	No	No	No
994	03/04/2021 16.37.55	75	Femmina	No	No	No	No	No
995	04/04/2021 14.35.36	26	Maschio	No	No	No	No	No
996	09/04/2021 11.48.46	27	Maschio	No	Sì			
997	09/04/2021 23.16.14	22	Femmina	No	Sì			
998	11/04/2021 12.04.56	25	Maschio	Sì	No	No	No	No
999	15/04/2021 20.53.52	22	Maschio	No	No	No	No	No
1000	15/04/2021 20.54.09	24	Maschio	No	No	No	No	No
1001	15/04/2021 20.54.43	20	Femmina	No	No	No	No	No
1002	15/04/2021 20.54.47	21	Femmina	No	No	No	No	No
1003	15/04/2021 20.54.48	20	Maschio	Sì	No	No	No	No
1004	18/04/2021 22.24.19	31	Maschio	Sì	No	No	No	Sì
1005	22/04/2021 11.15.39	22	Maschio	Sì	No	No	No	Sì

Google Form

Conversion rates of the first item

Conversion rates of the second item

Average number of daily customers

Disclaimer

- Promos are received by all the customers that purchase the first item. Giving promo P0 means that the specific customer does not receive a discount.

- In the project steps we considered vectors of candidate prices (and margins) for both the items, and related vectors of conversion rates, starting from the form data. Therefore, the optimal prices found in the project steps will not be necessarily the ones proposed in the form.

Step 1: Algorithm

```
FOR EACH price_item1:

        FOR EACH price_item2:

                revenue_item2, matching = LINEAR_PROGRAM(margin_item2[price_item2],
                discounts, conversion_rates_item2[price_item2], daily_promos, customers *
                conversion_rate_item1[price_item1])

                total_revenue = revenue_item2 + revenue_item1

RETURN max(total_revenue), (price_item1, price_item2, matching) related to max(total_revenue)
```

Step 1: Linear Program

```
set I;                              # Promo codes (P0, P1, P2, P3)
set J;                              # Customer classes (C1, C2, C3, C4)

param r{I, J};                      # Conversion rates
param margin_item2;                 # Margin of item 2
param P0;                           # Discount P0
param P1;                           # Discount P1
param P2;                           # Discount P2
param P3;                           # Discount P3
param numPromos{I};                 # Number of promo codes available
param numCustomers{J};              # Number of daily customers

var n{I, J} >= 0;

maximize objective: (sum{j in J} (n["P0", j] * r["P0", j] * (1-P0)) +
                     sum{j in J} (n["P1", j] * r["P1", j] * (1-P1)) +
                     sum{j in J} (n["P2", j] * r["P2", j] * (1-P2)) +
                     sum{j in J} (n["P3", j] * r["P3", j] * (1-P3))) * margin_item2;

subject to numberOfPromos{i in I}: sum{j in J} n[i, j] = numPromos[i];

subject to numberOfCustomers{j in J}: sum{i in I} n[i, j] <= numCustomers[j];
```

Step 1: Results

```
Experiment 1: fractions [0.7  0.2  0.07 0.03]

Revenue: 47553.89487156687

Optimal price item 1: 400

Optimal price item 2: 60

Optimal assignment of promos to customer classes:

     Class1  Class2  Class3  Class4

P0      150       6      95       0

P1        0      58       0      12

P2        0       0       0      24

P3        9       0       0       0
```

```
Experiment 2: fractions [0.25 0.25 0.25 0.25]

Revenue: 47609.3637491476

Optimal price item 1: 400

Optimal price item 2: 60

Optimal assignment of promos to customer classes:

     Class1  Class2  Class3  Class4

P0       69       0      19       0

P1        0      65       0      23

P2        0       0      75      13

P3       89       0       0       0
```

Step 2: General idea

For each round:

1. Update the estimation of the conversion rates, optimal prices, and daily customers.
2. Solve the linear program.
3. Normalize the resulting matrix of the matching.
4. Assign promos to the customers of the next day using the previously computed normalized matrix.

Step 2: Environment

For each customer:

1. A new customer of a specific class arrives.
2. Extraction from a Bernoulli distribution with *p=conversion rates for the first item* to understand if the customer buys the first item.
3. Assignment of a promo to the customer using a discrete distribution according to the weights of the matching.
4. Extraction from a Bernoulli distribution with *p=conversion rates for the second item* to understand if the customer buys the second item.

Computation of the daily conversion rates for both the items and the daily revenue.

Step 3: Model

UCB1:

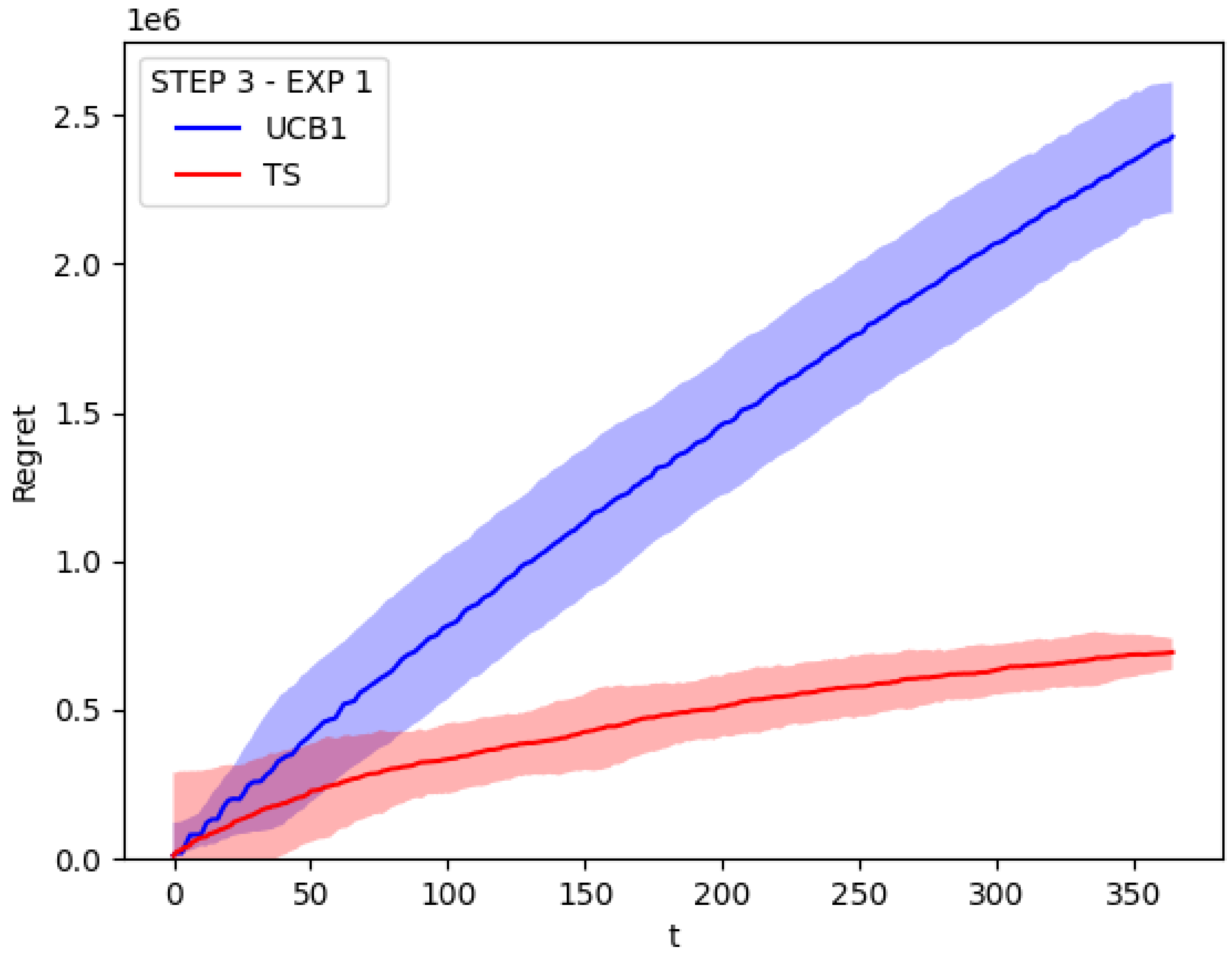
- Arms: margins of the first item.
- Empirical mean: estimation of the conversion rates of the first item.
- Confidence: standard confidence of the UCB1 Bandit.
- Upper bound: revenue given the estimation of the conversion rates of the first item (empirical mean plus confidence) and the known optimal parameters received as inputs.

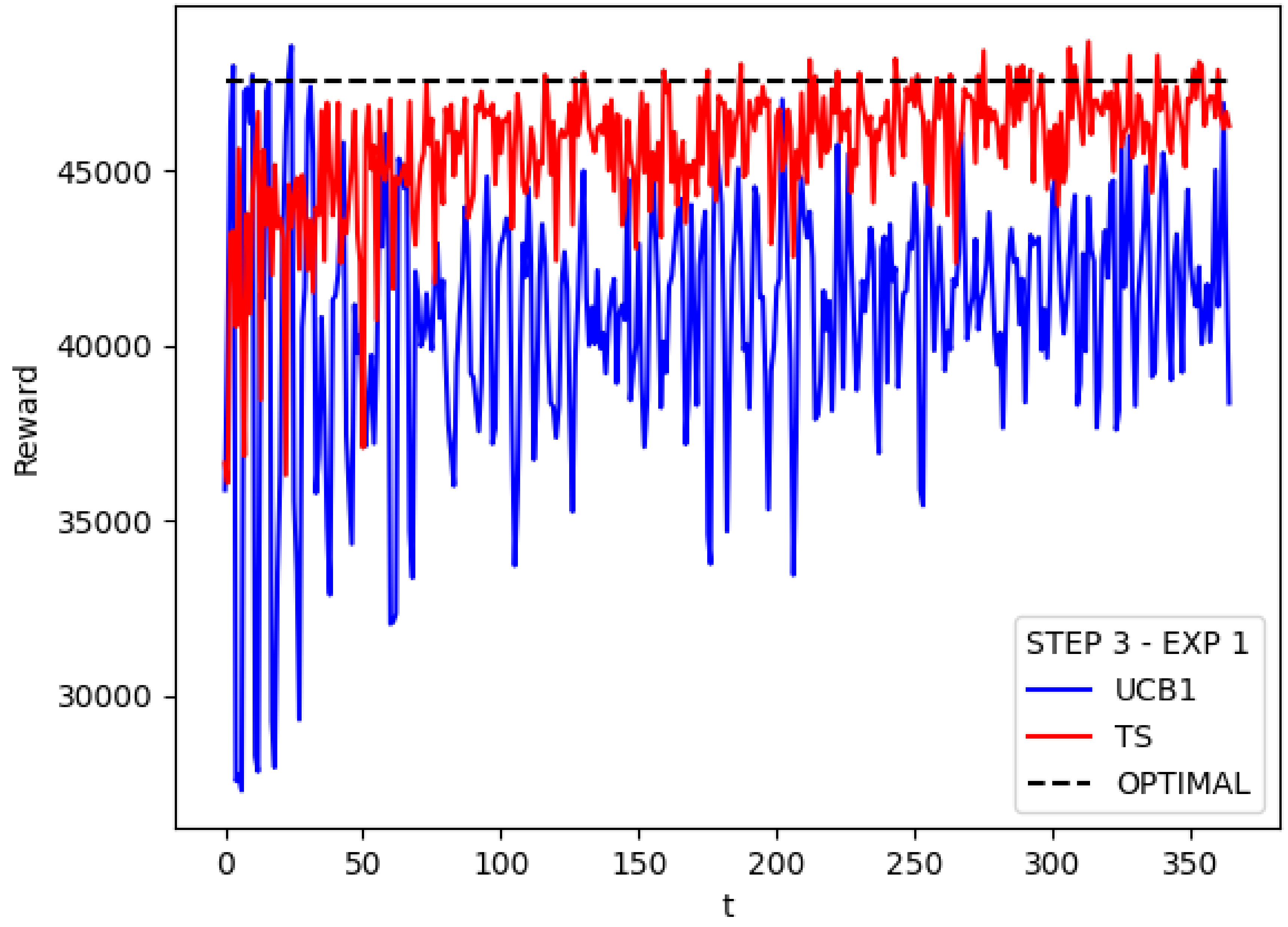
Step 3: Model

Thompson Sampling:

- Arms: margins of the first item.
- Beta distribution: standard Beta distribution of the Thompson Sampling Bandit related to the conversion rates of the first item.
- Pulled arm: based on the revenue given the estimation of the conversion rates of the first item (extraction from Beta distribution) and the known optimal parameters received as inputs.

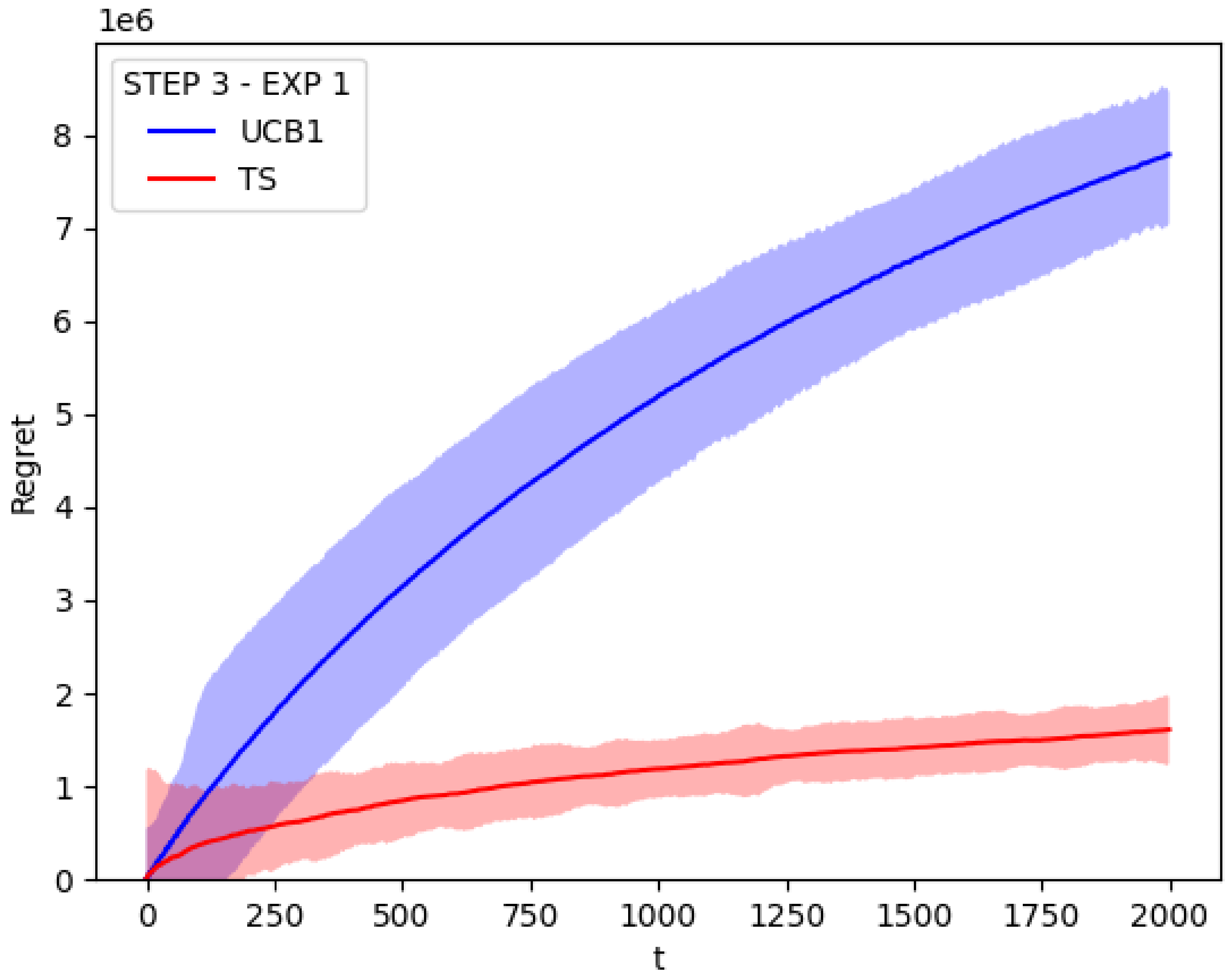
Step 3: Results (first experiment)

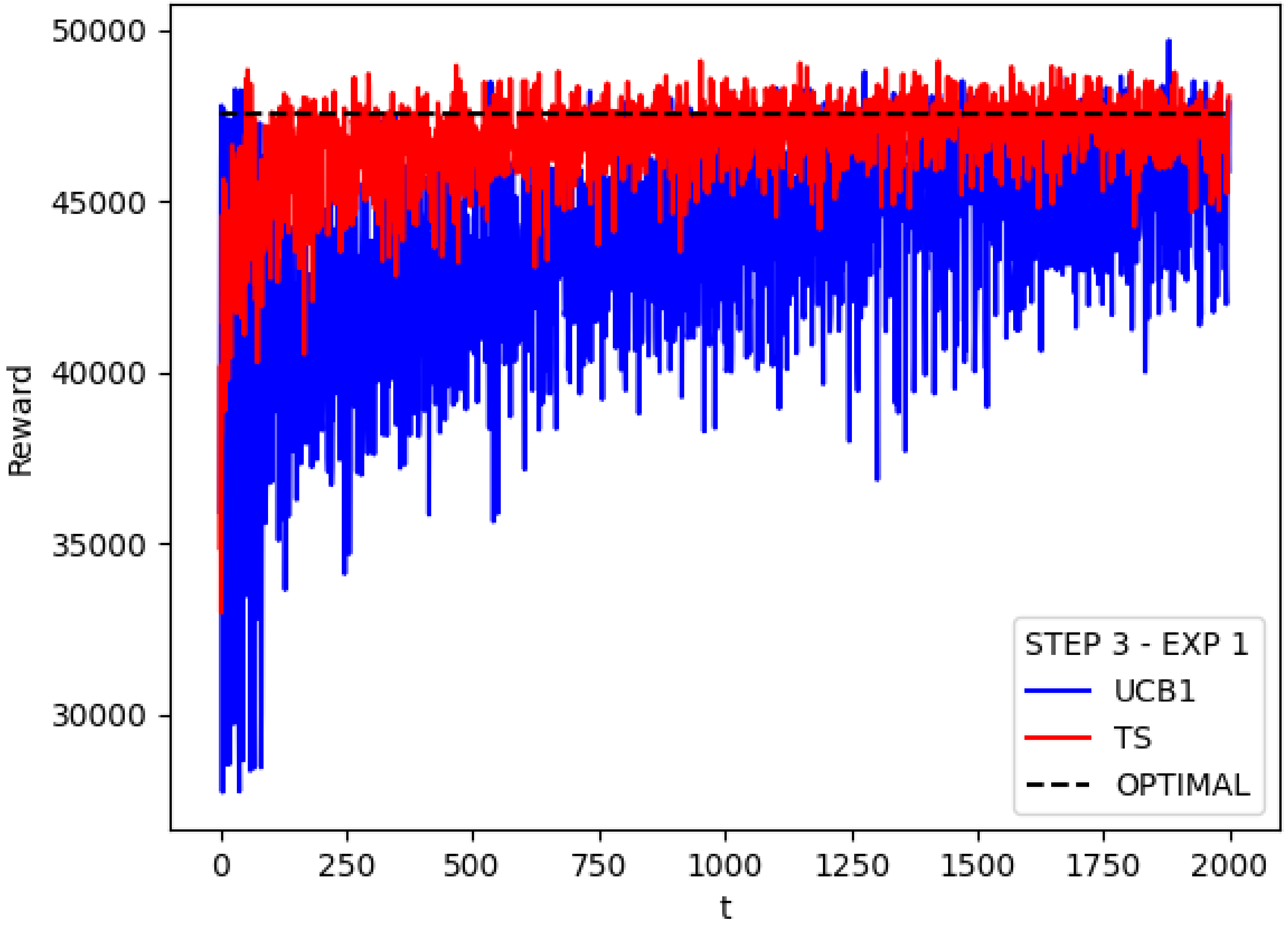




T = 365 days

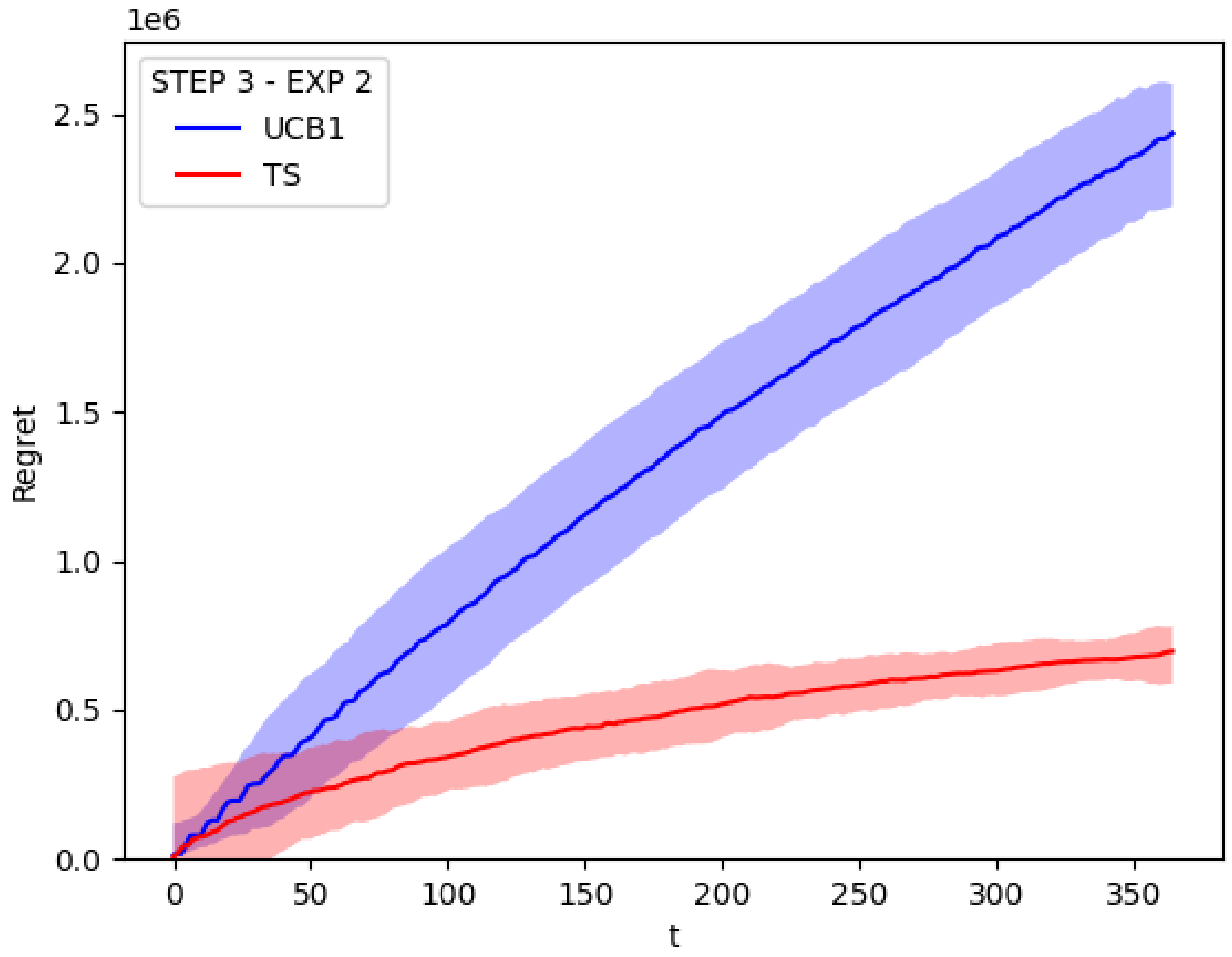
Step 3: Results (first experiment)

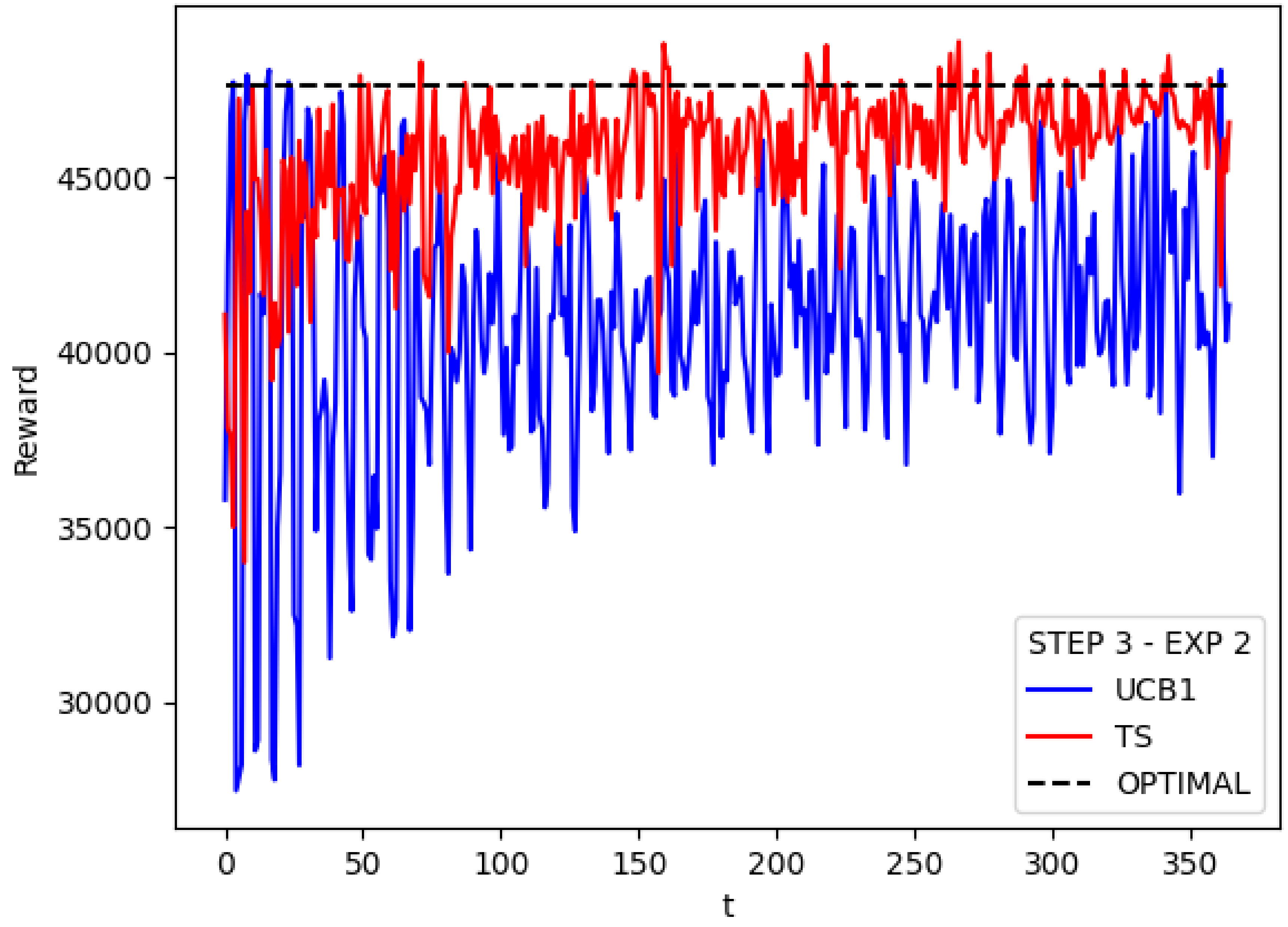




T = 2000 days

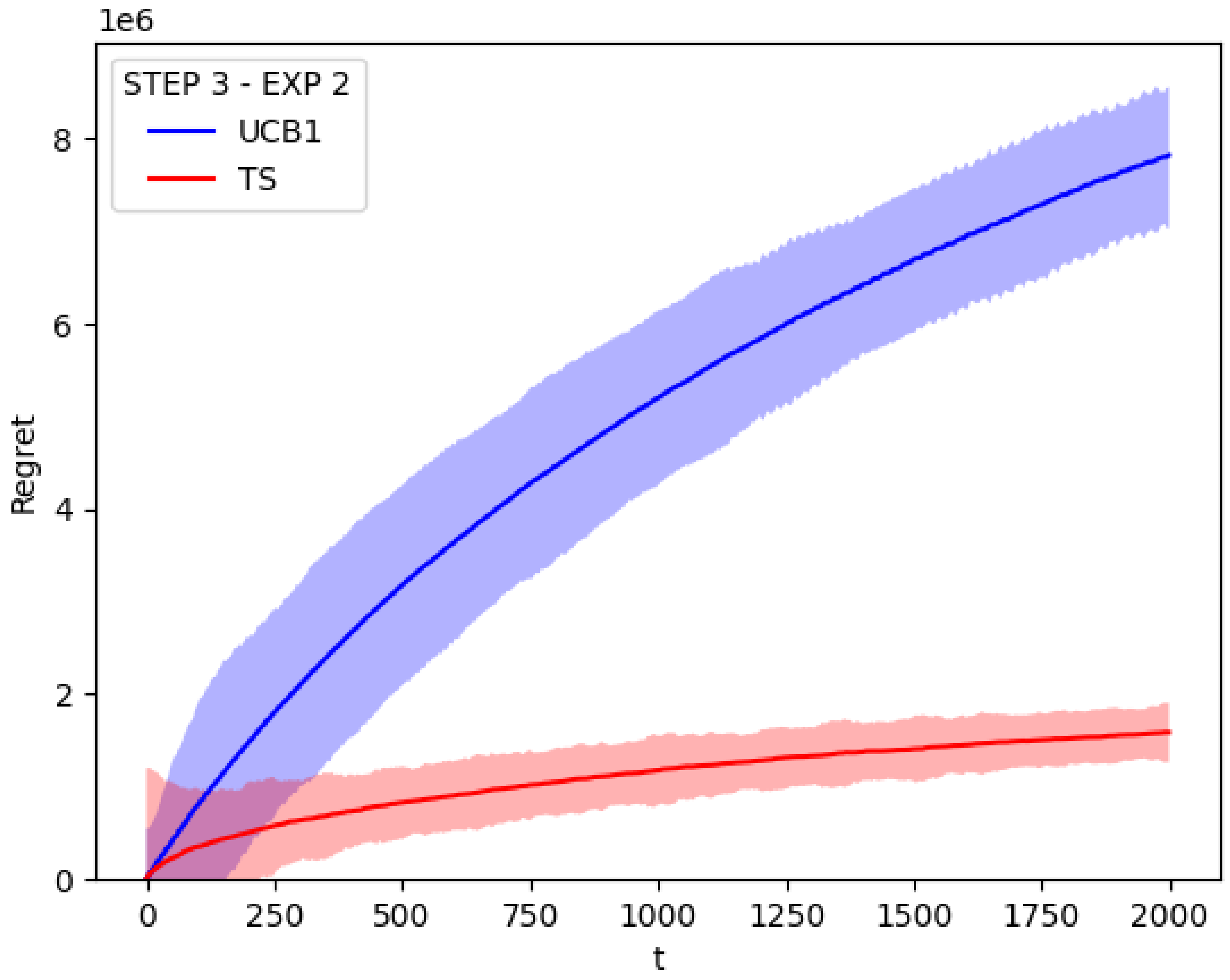
Step 3: Results (second experiment)

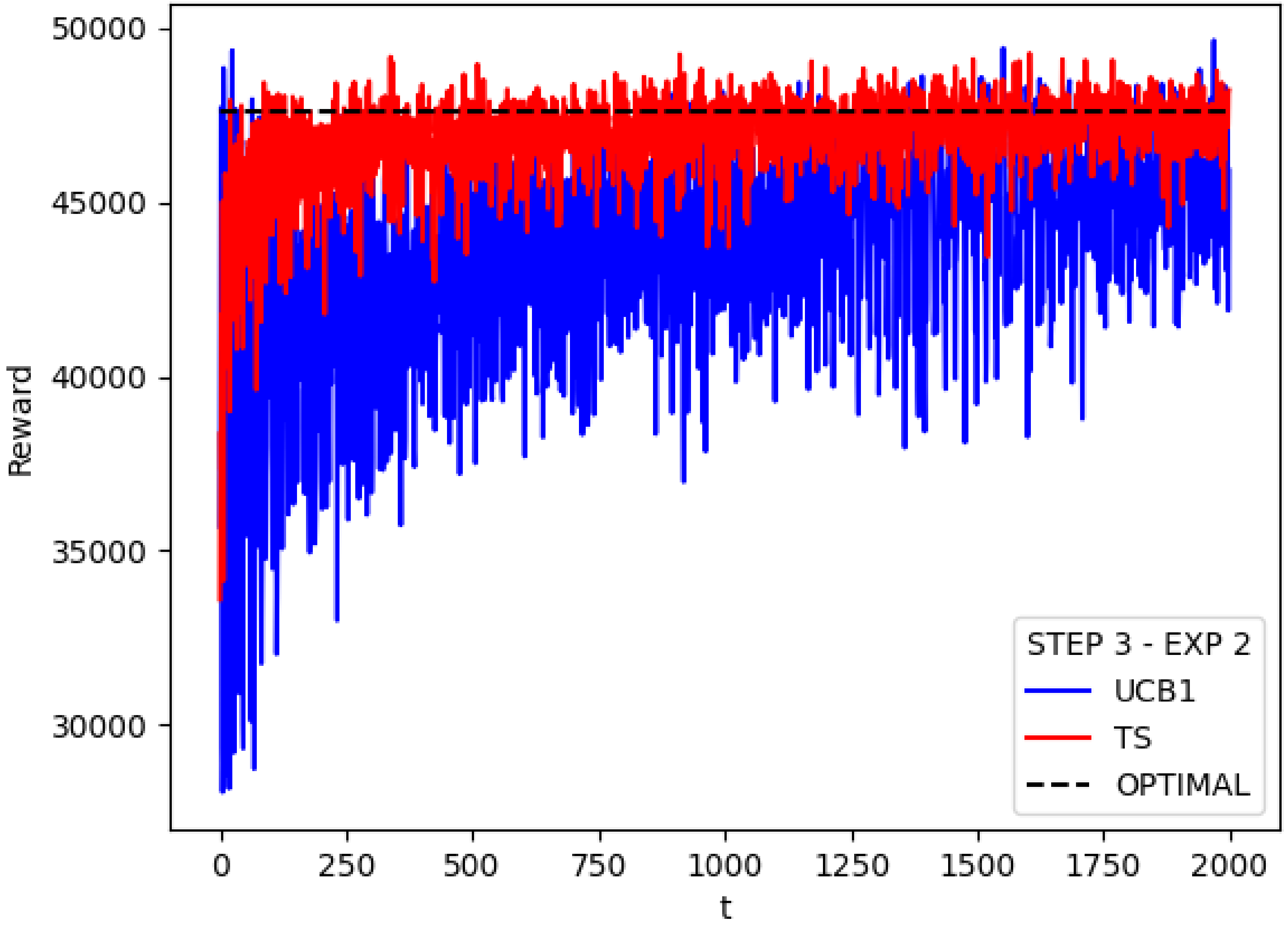




T = 365 days

Step 3: Results (second experiment)





T = 2000 days

Step 4: Model

UCB1:

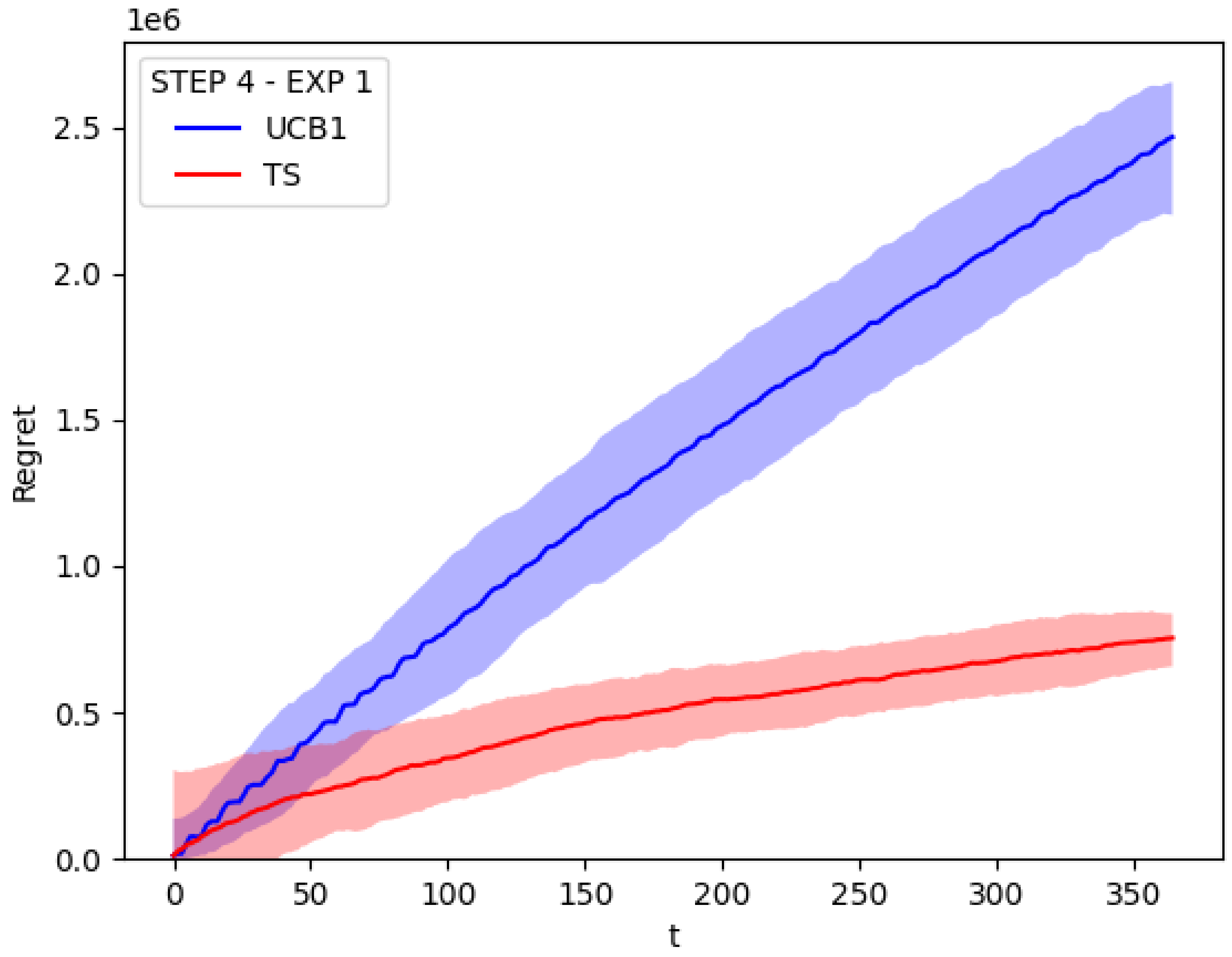
- Arms: margins of the first item.
- Empirical mean: estimation of the conversion rates of the first item.
- Confidence: standard confidence of the UCB1 Bandit.
- Upper bound: revenue given the estimation of the conversion rates of the first item (empirical mean plus confidence), the estimation of the conversion rates of the second item (computing the mean of the daily conversion rates of the previous rounds) and number of customers (computing the mean of the daily customers of the previous rounds), and the known optimal parameters received as inputs.

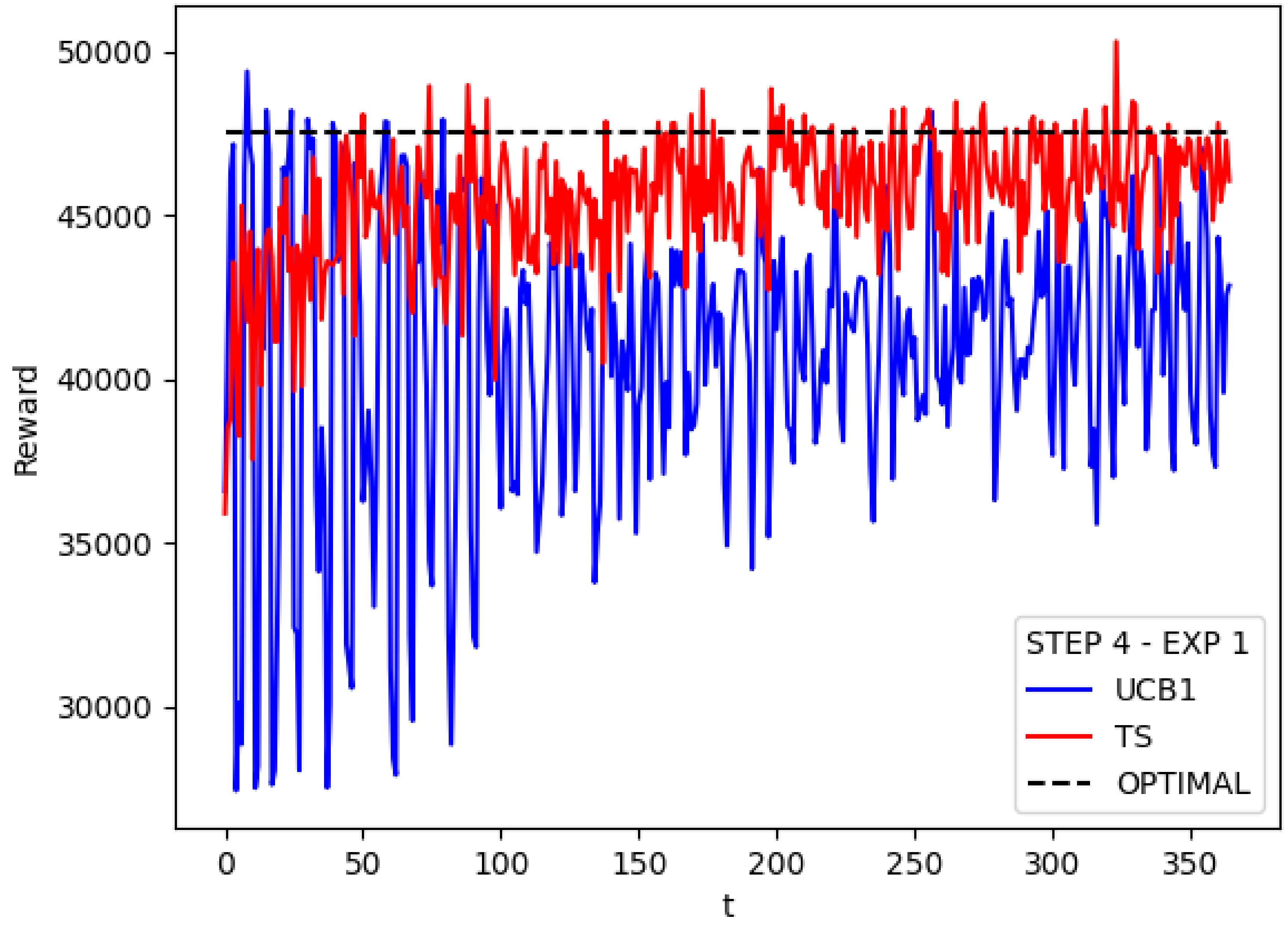
Step 4: Model

Thompson Sampling:

- Arms: margins of the first item.
- Beta distribution: standard Beta distribution of the Thompson Sampling Bandit related to the conversion rates of the first item.
- Pulled arm: based on the revenue given the estimation of the conversion rates of the first item (extraction from Beta distribution), the estimation of the conversion rates of the second item (computing the mean of the daily conversion rates of the previous rounds) and number of customers (computing the mean of the daily customers of the previous rounds), and the known optimal parameters received as inputs.

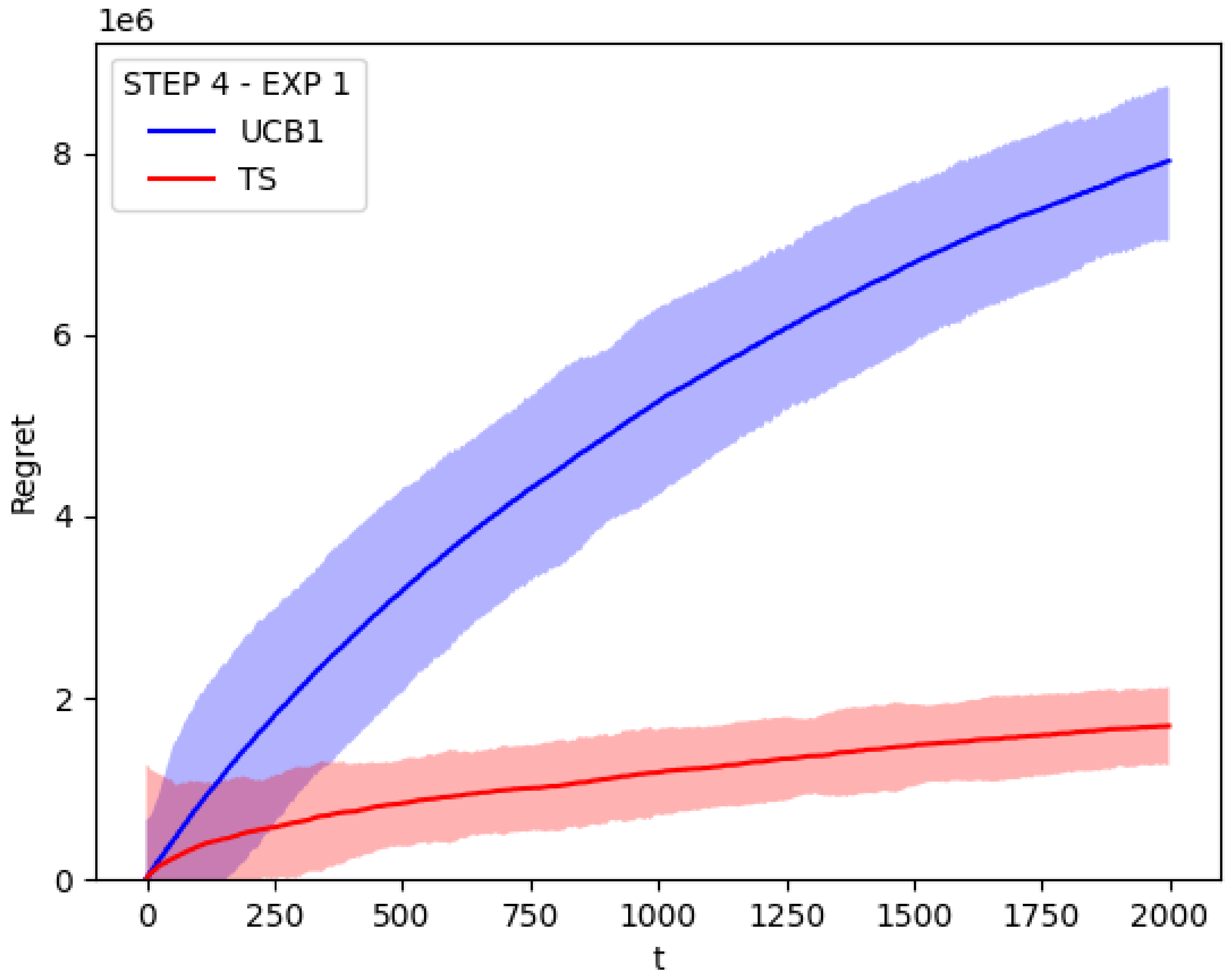
Step 4: Results (first experiment)

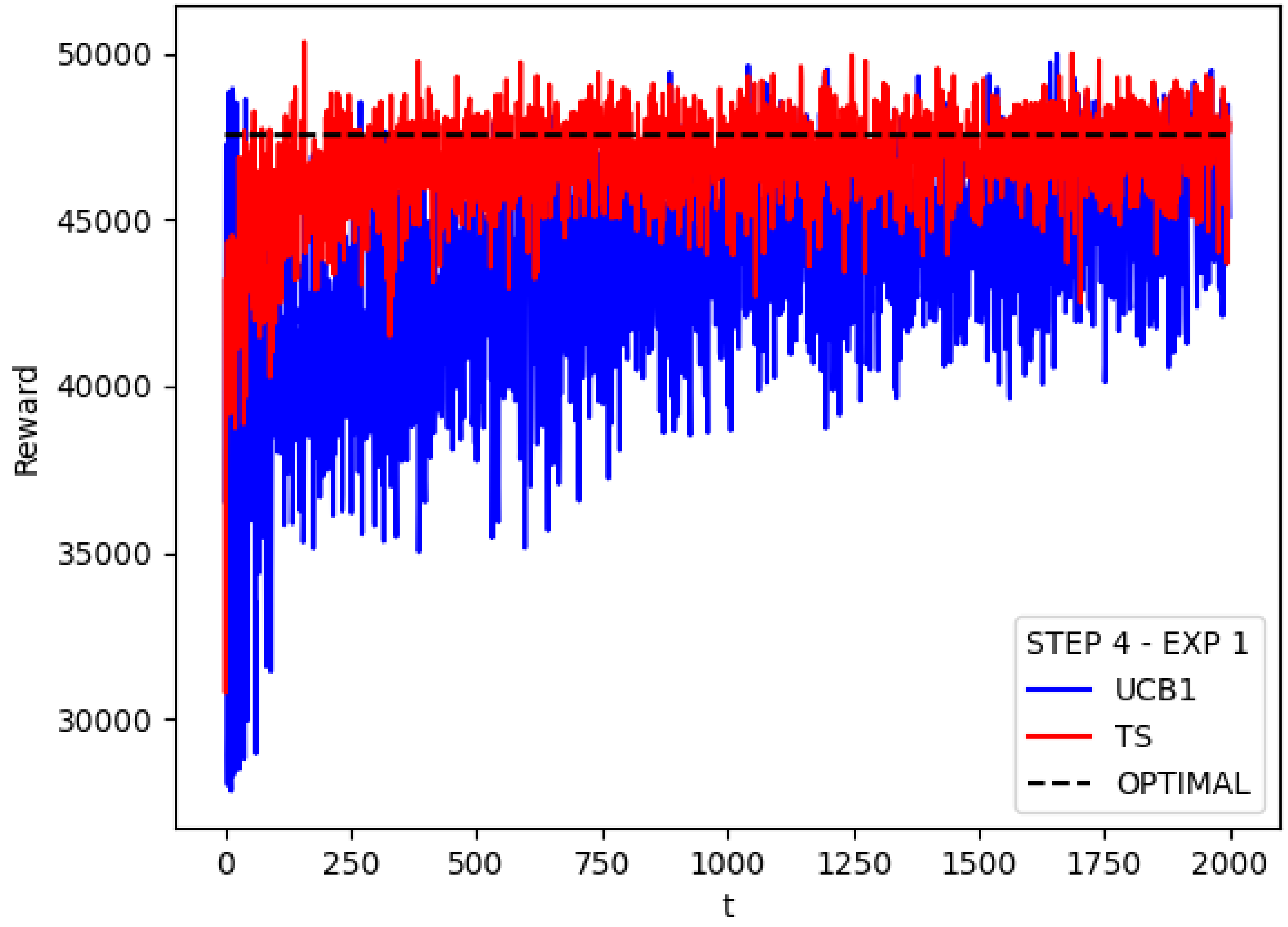




T = 365 days

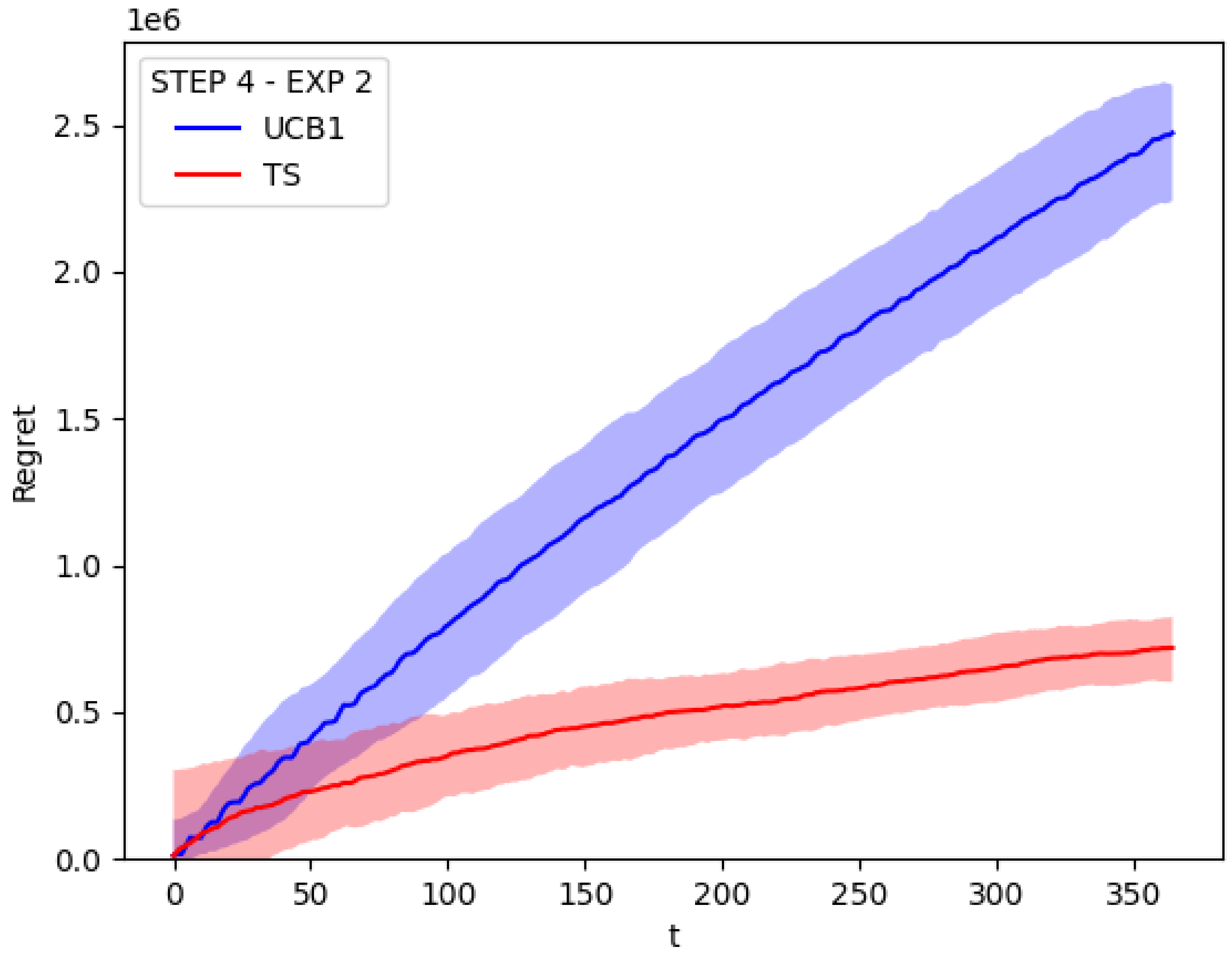
Step 4: Results (first experiment)

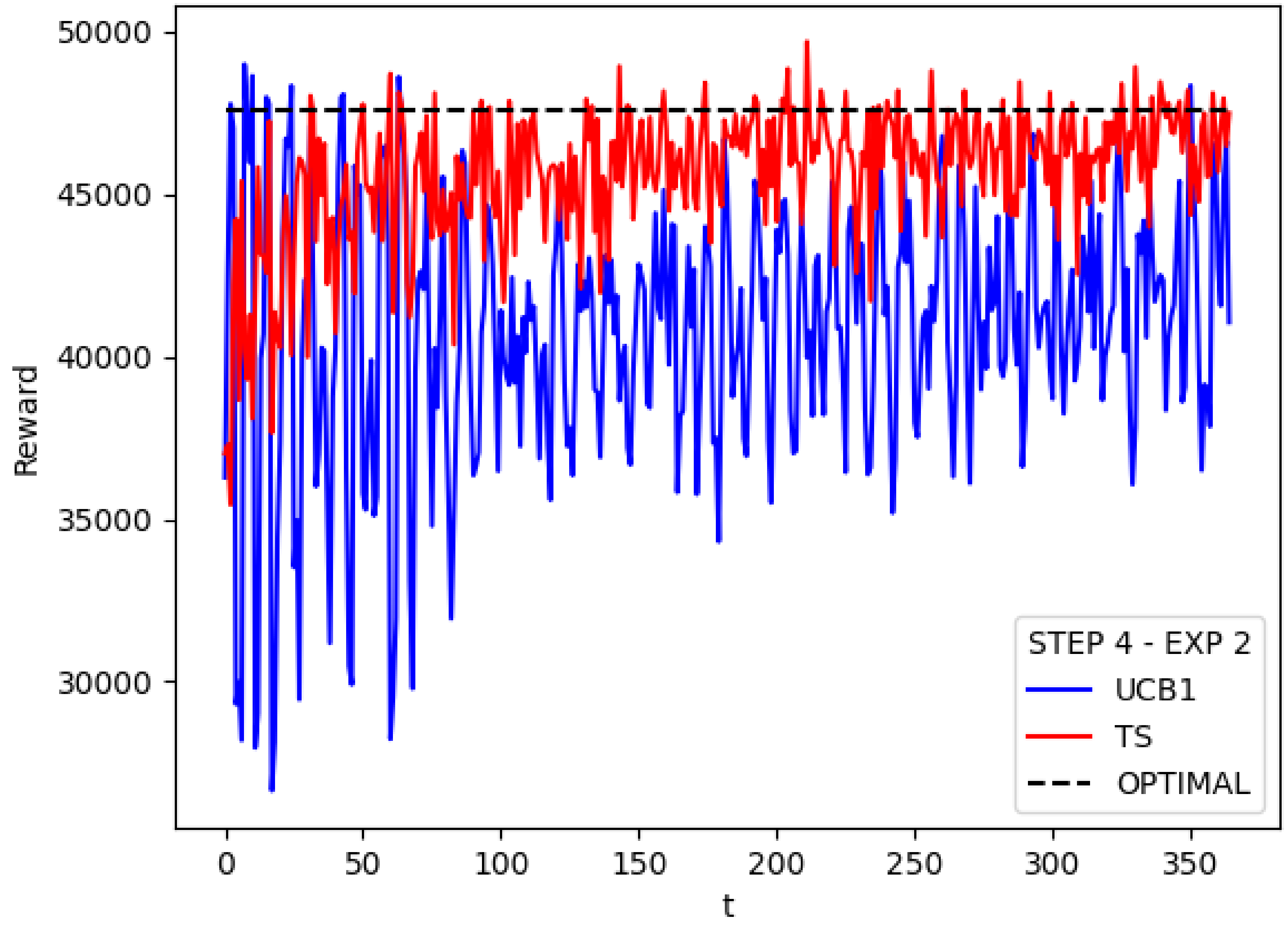




T = 2000 days

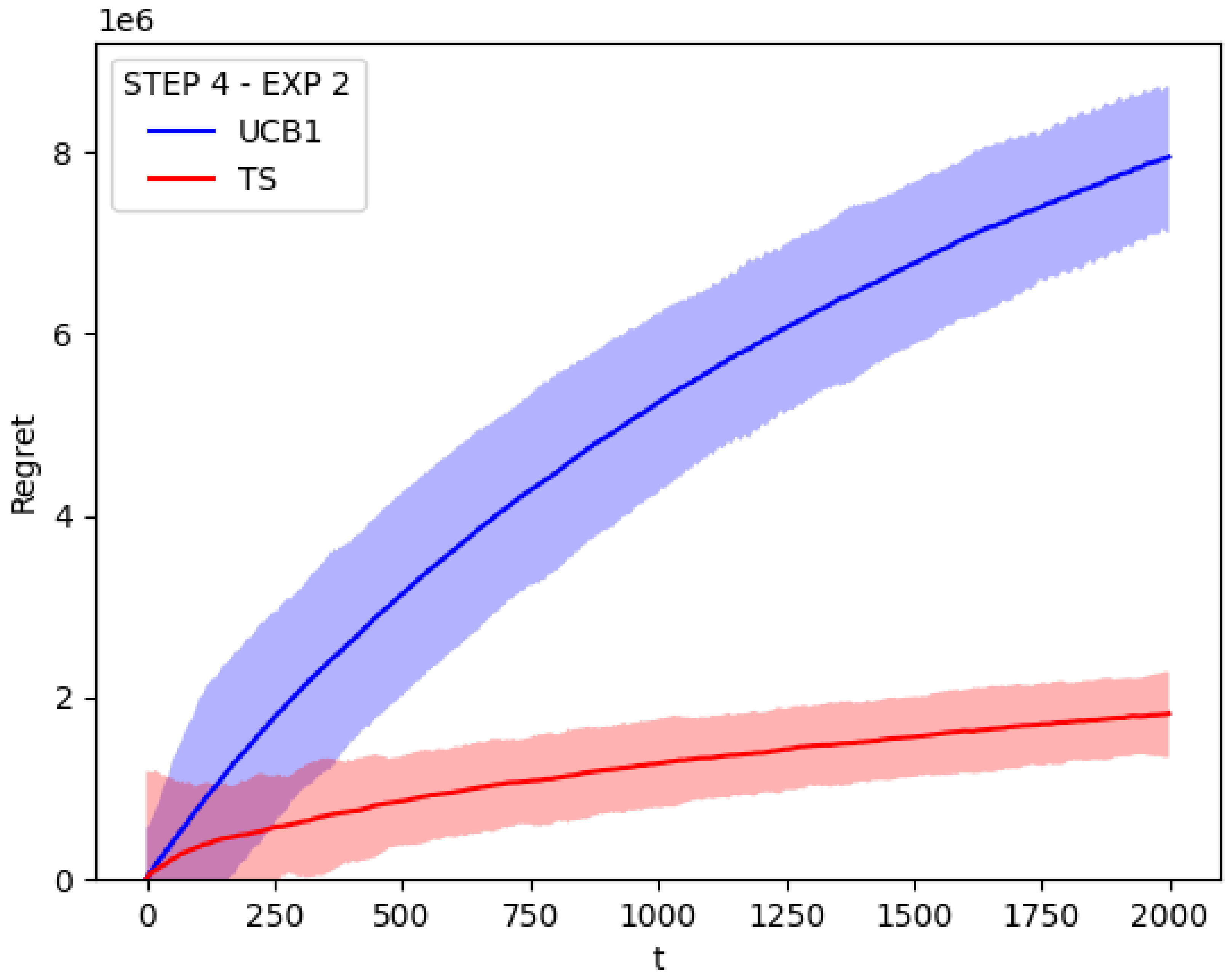
Step 4: Results (second experiment)

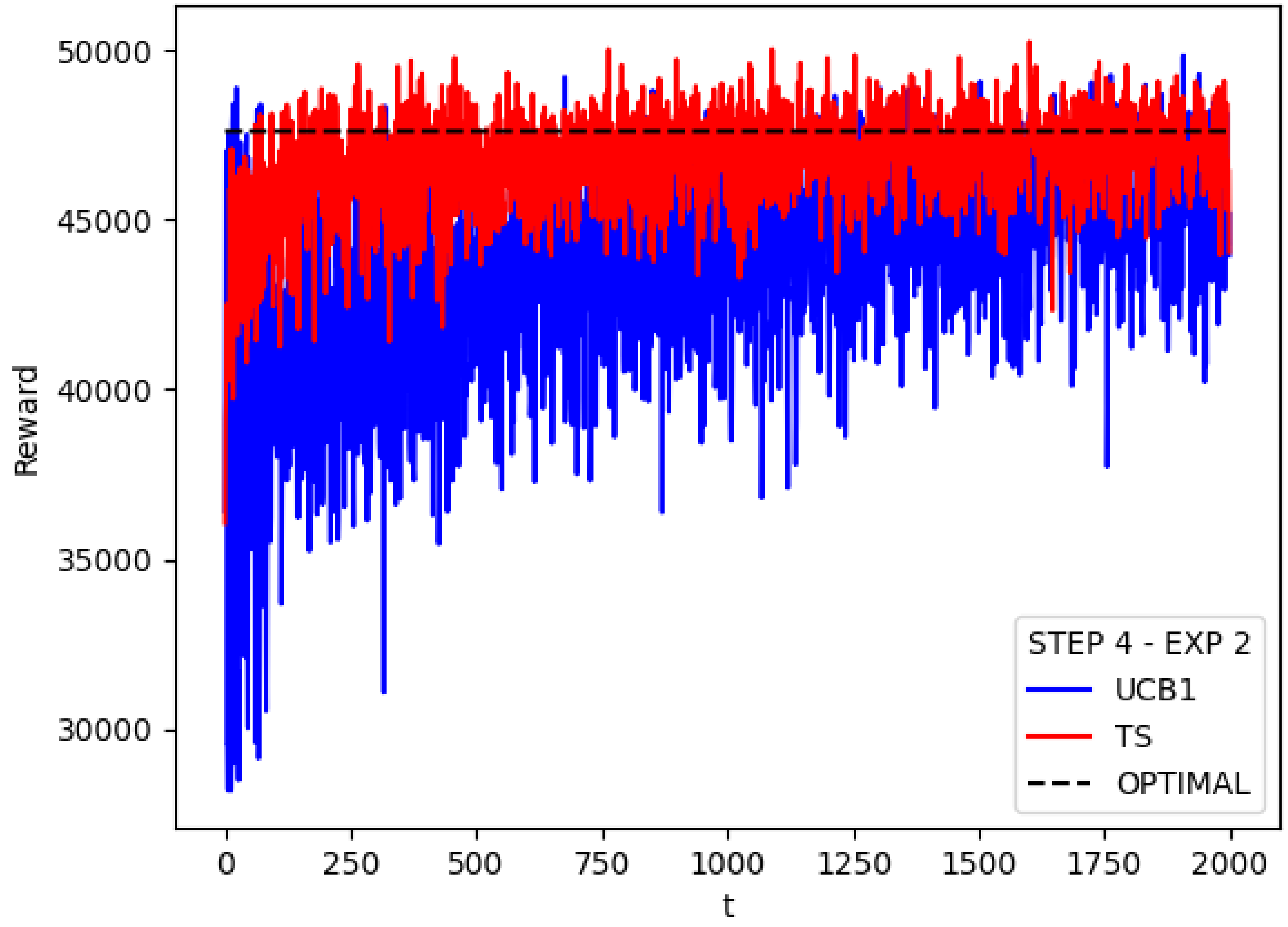




T = 365 days

Step 4: Results (second experiment)



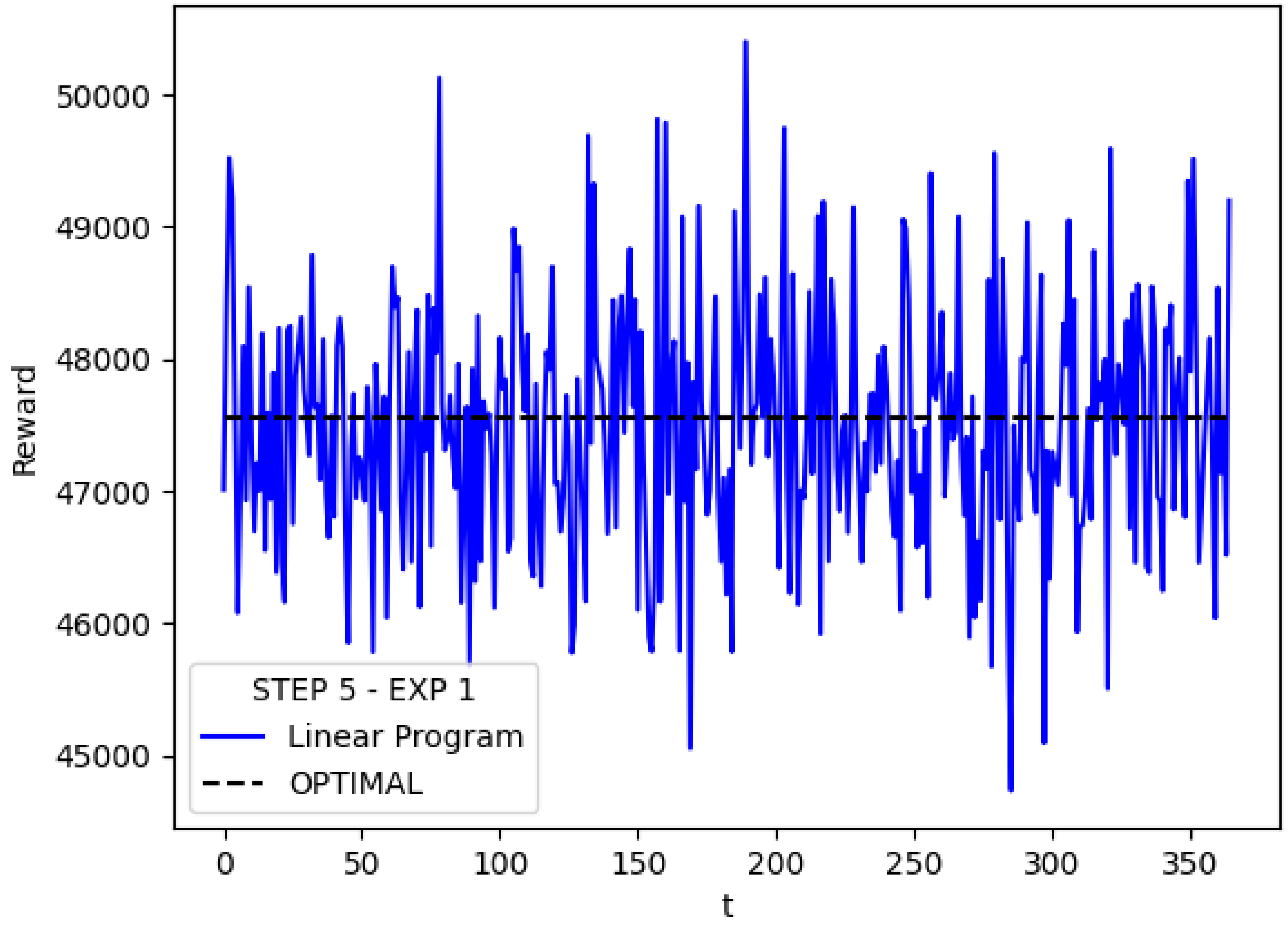


T = 2000 days

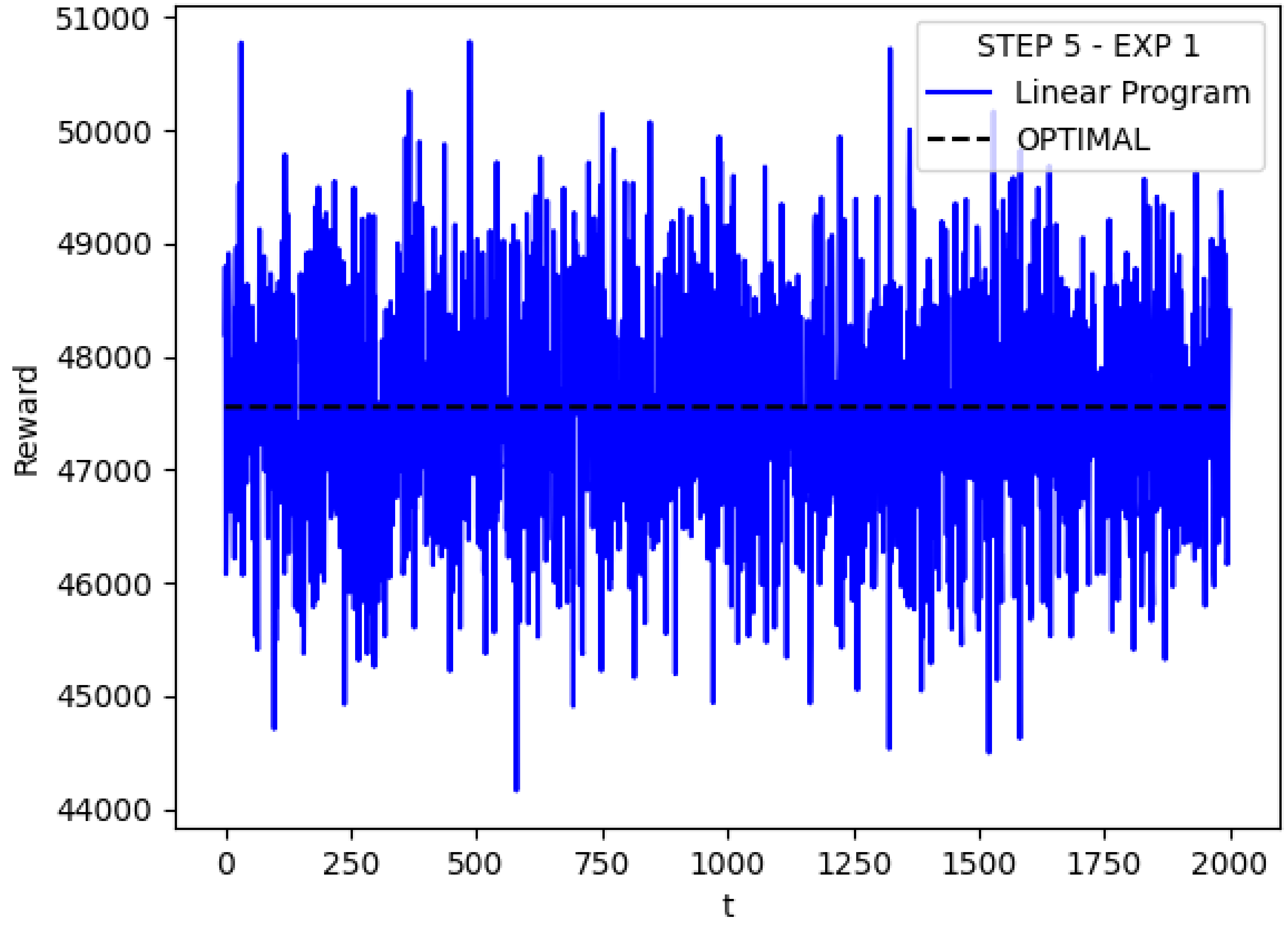
Step 5: Model

- The number of daily customers is learned as the mean of the number of daily customers arrived in the past.
- The conversion rates of the first item are learned as the mean of the conversion rates of the first item observed in the past.
- The conversion rates of the second item are learned as the mean of the conversion rates of the second item observed in the past.
- Every day, the linear program is run using the mean of daily customers and conversion rates and the optimal prices given in input.

Step 5: Results (first experiment)

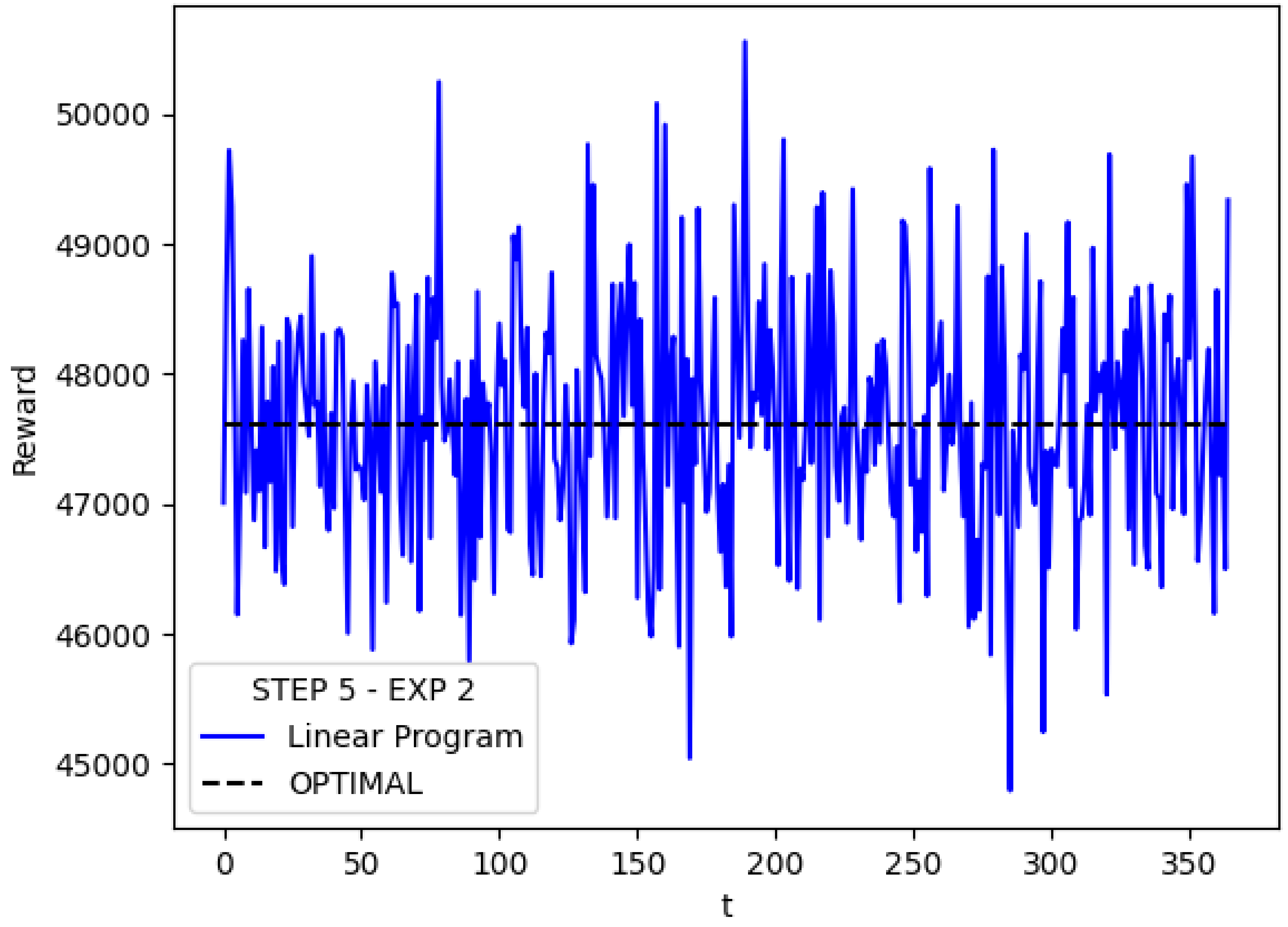


T = 365 days

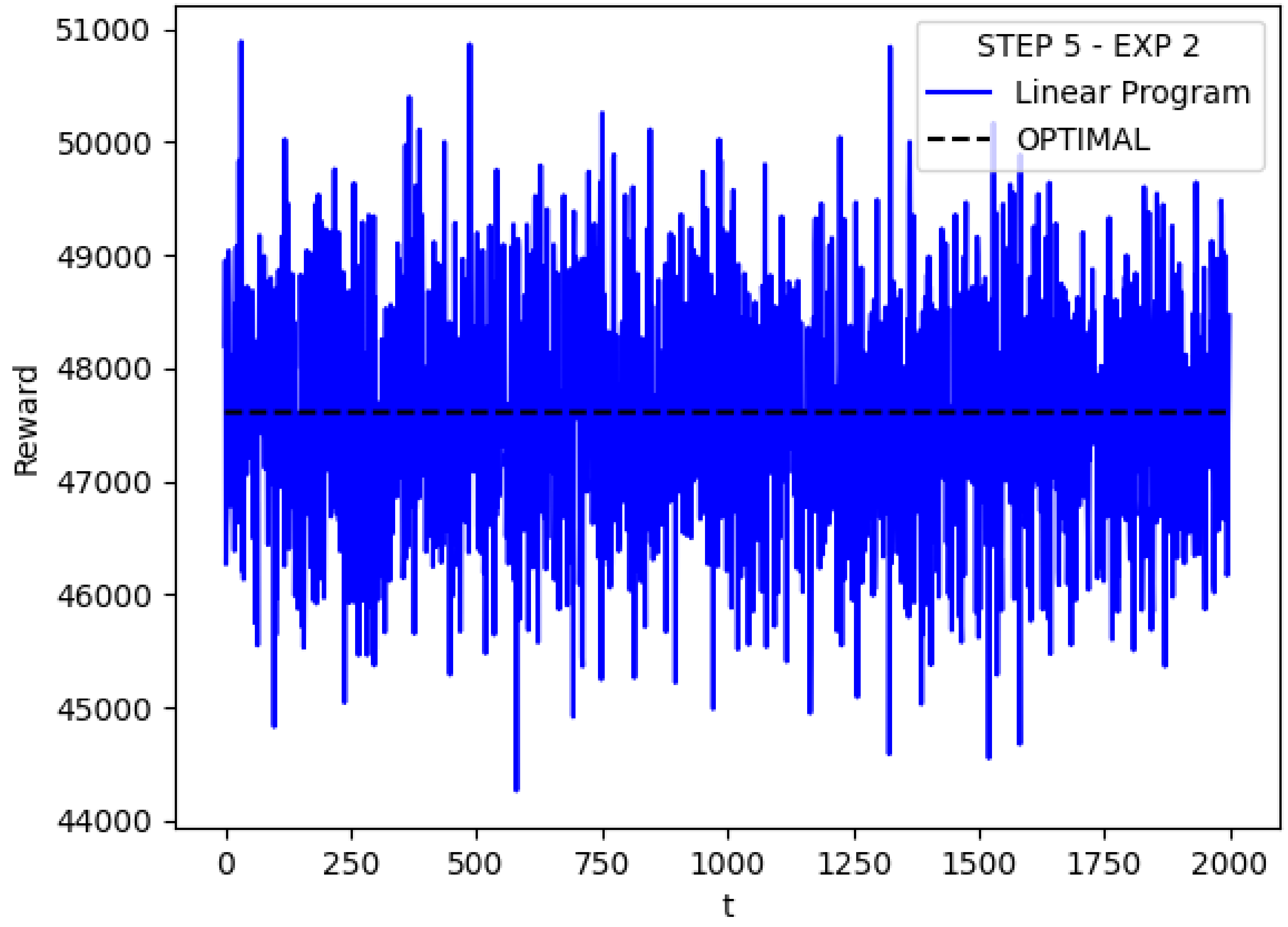


T = 2000 days

Step 5: Results (second experiment)



T = 365 days



T = 2000 days

Step 6: Model

UCB1:

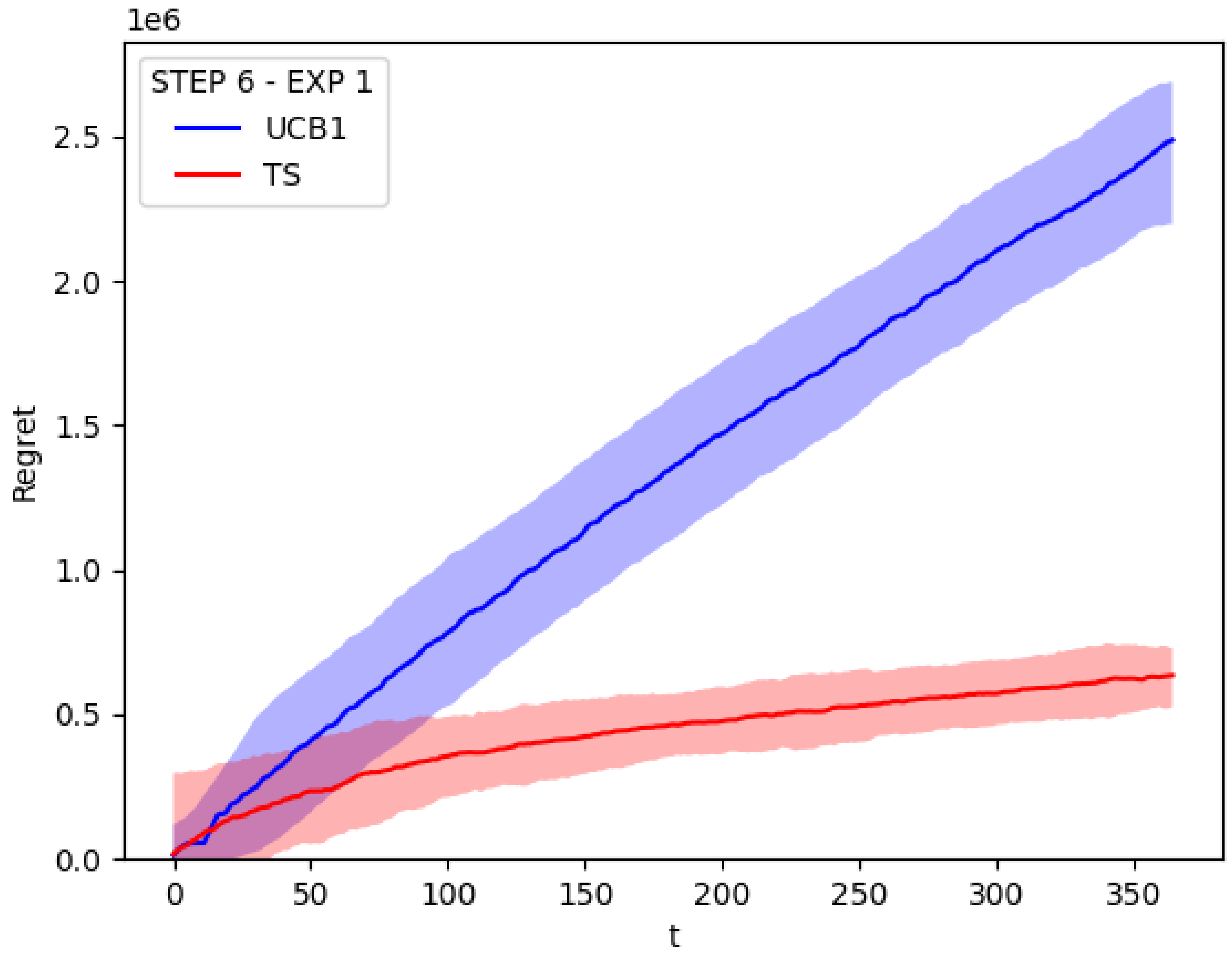
- Arms: cross-product of the margins of the two items.
- Empirical mean first item: estimation of the conversion rates of the first item.
- Empirical mean second item: estimation of the conversion rates of the second item.
- Confidence first item: standard confidence of the UCB1 Bandit related to the first item.
- Confidence second item: standard confidence of the UCB1 Bandit related to the second item.
- Upper bound: revenue given the estimation of the conversion rates of the first item (empirical mean first item plus confidence first item), the estimation of the conversion rates of the second item (empirical mean second item plus confidence second item), the number of customers (computing the mean of the daily customers of the previous rounds) and the assignment computed by the linear program for each pair of prices.

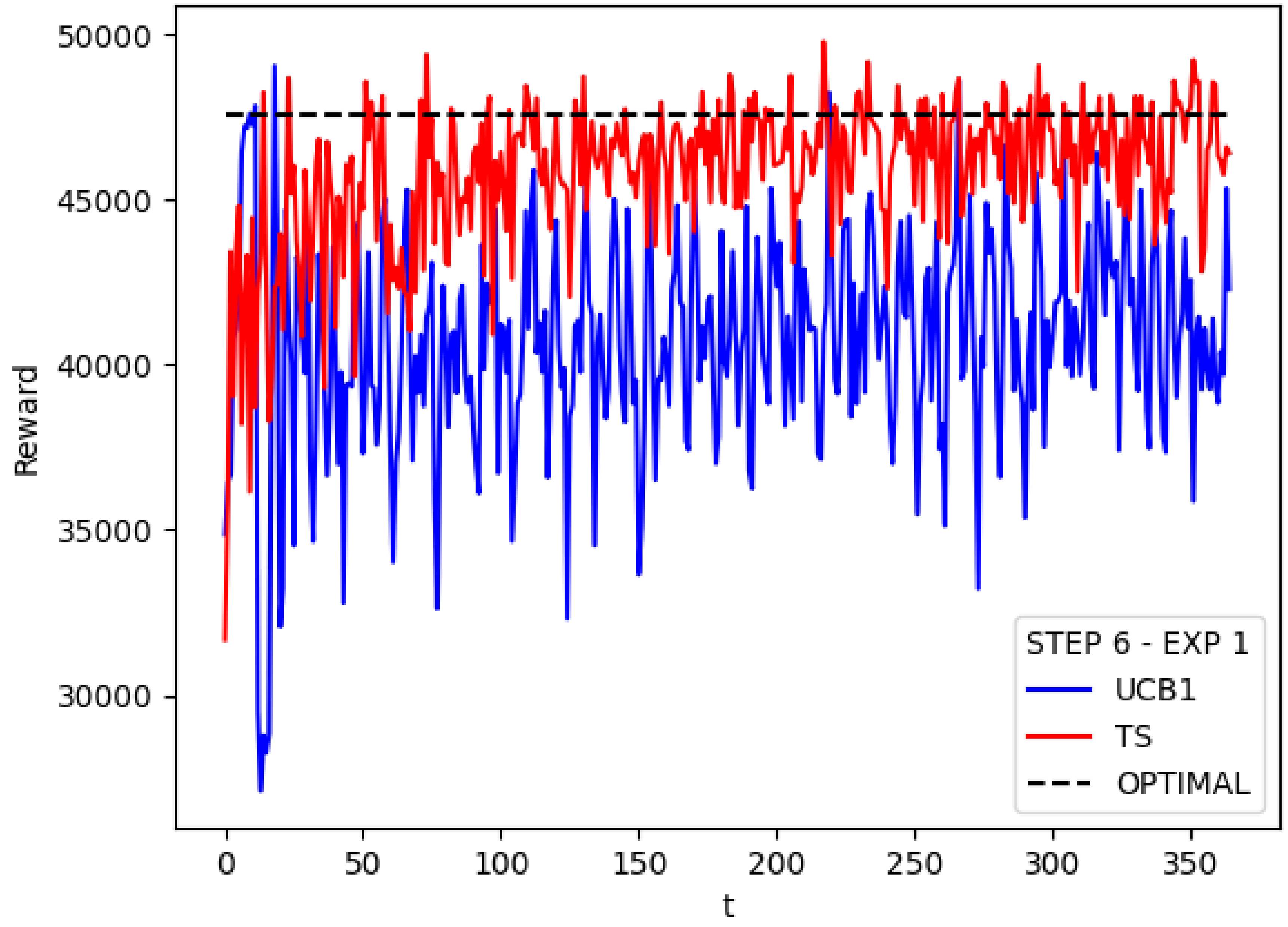
Step 6: Model

Thompson Sampling:

- Arms: cross-product of the margins of the two items.
- Beta distribution first item: standard Beta distribution of the Thompson Sampling Bandit related to the conversion rates of the first item.
- Beta distribution second item: standard Beta distribution of the Thompson Sampling Bandit related to the conversion rates of the second item.
- Pulled arm: revenue given the estimation of the conversion rates of the first item (extraction from Beta distribution first item), the estimation of the conversion rates of the second item (extraction from Beta distribution second item), the number of customers (computing the mean of the daily customers of the previous rounds) and the assignment computed by the linear program for each pair of prices.

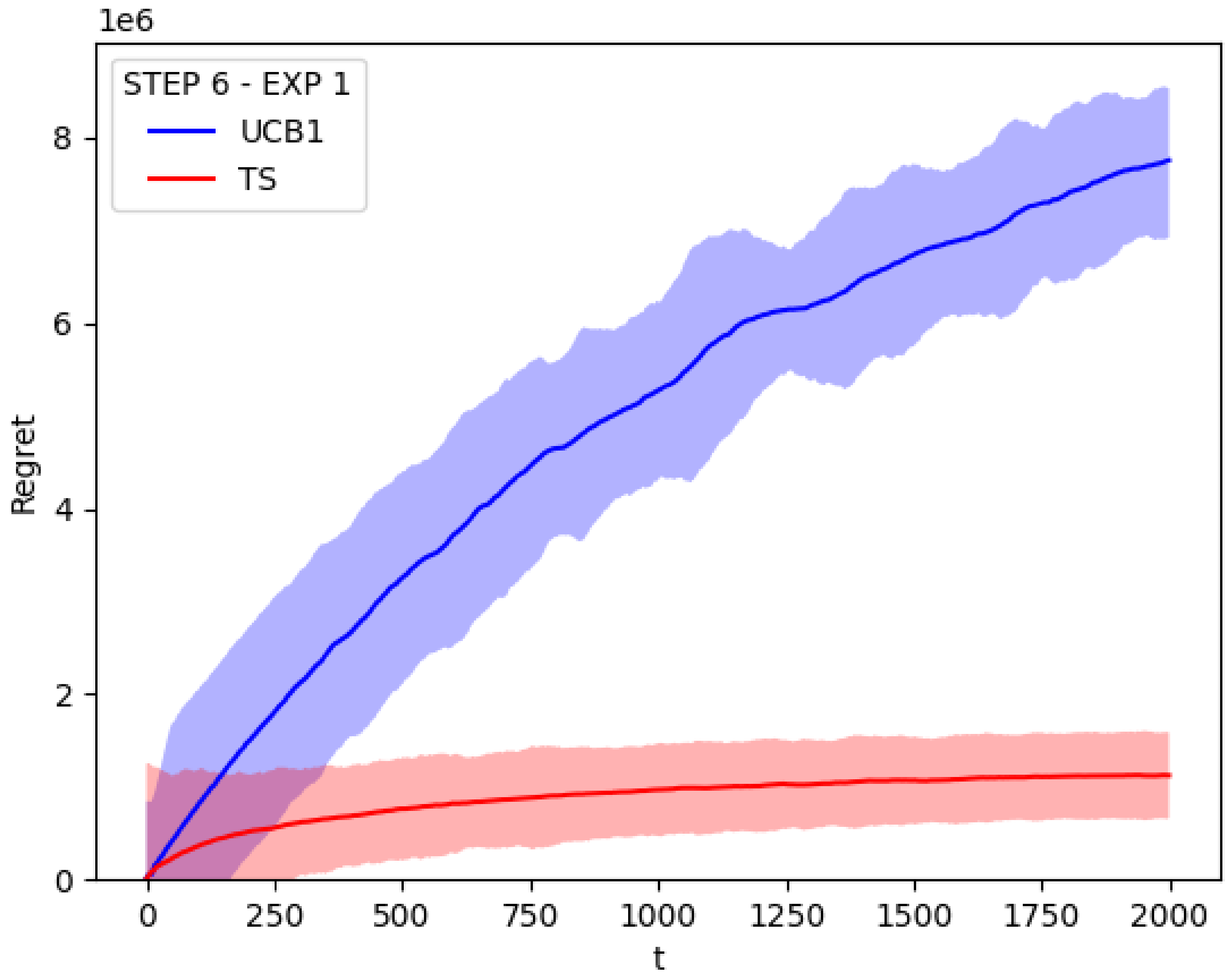
Step 6: Results (first experiment)

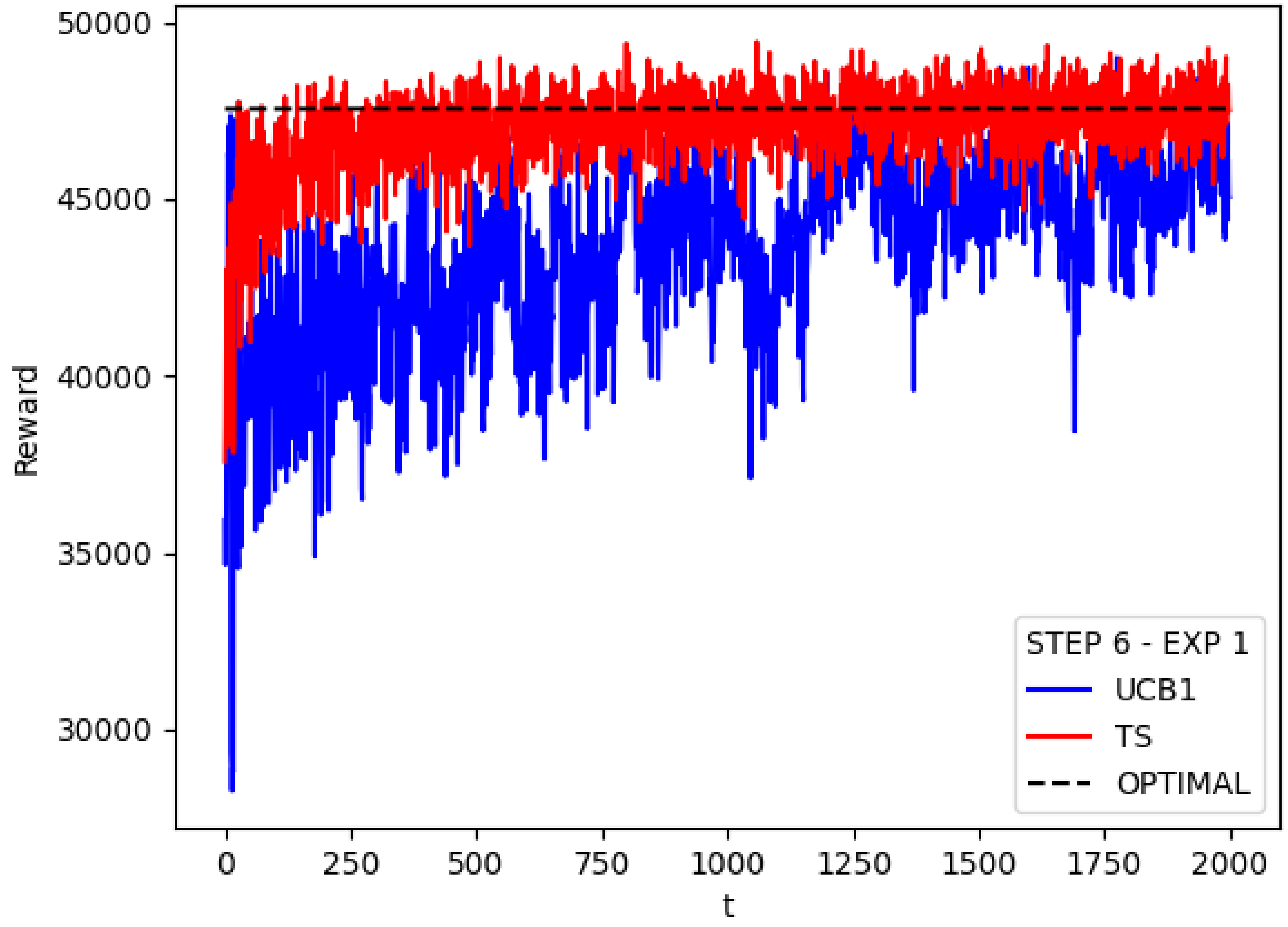




T = 365 days

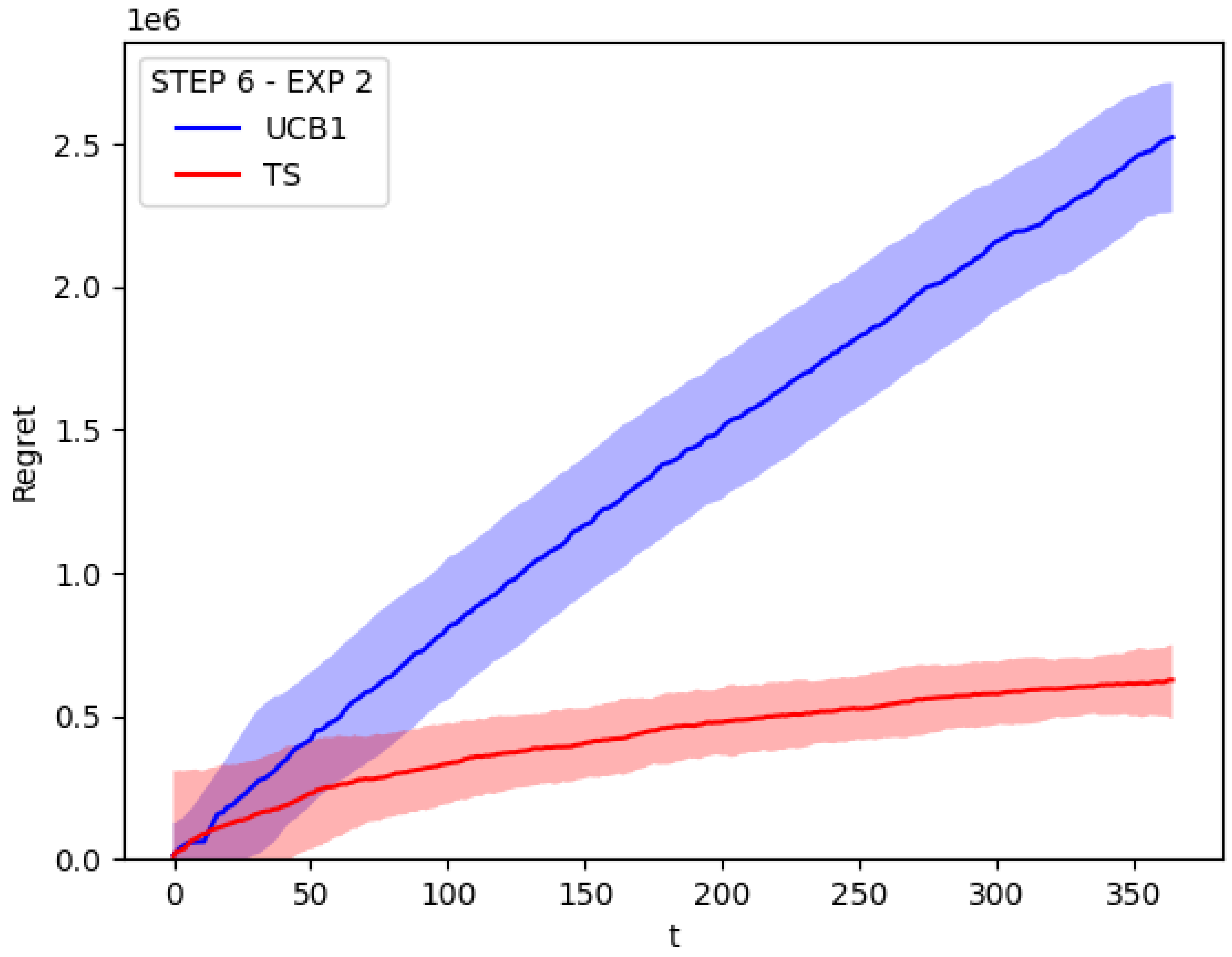
Step 6: Results (first experiment)

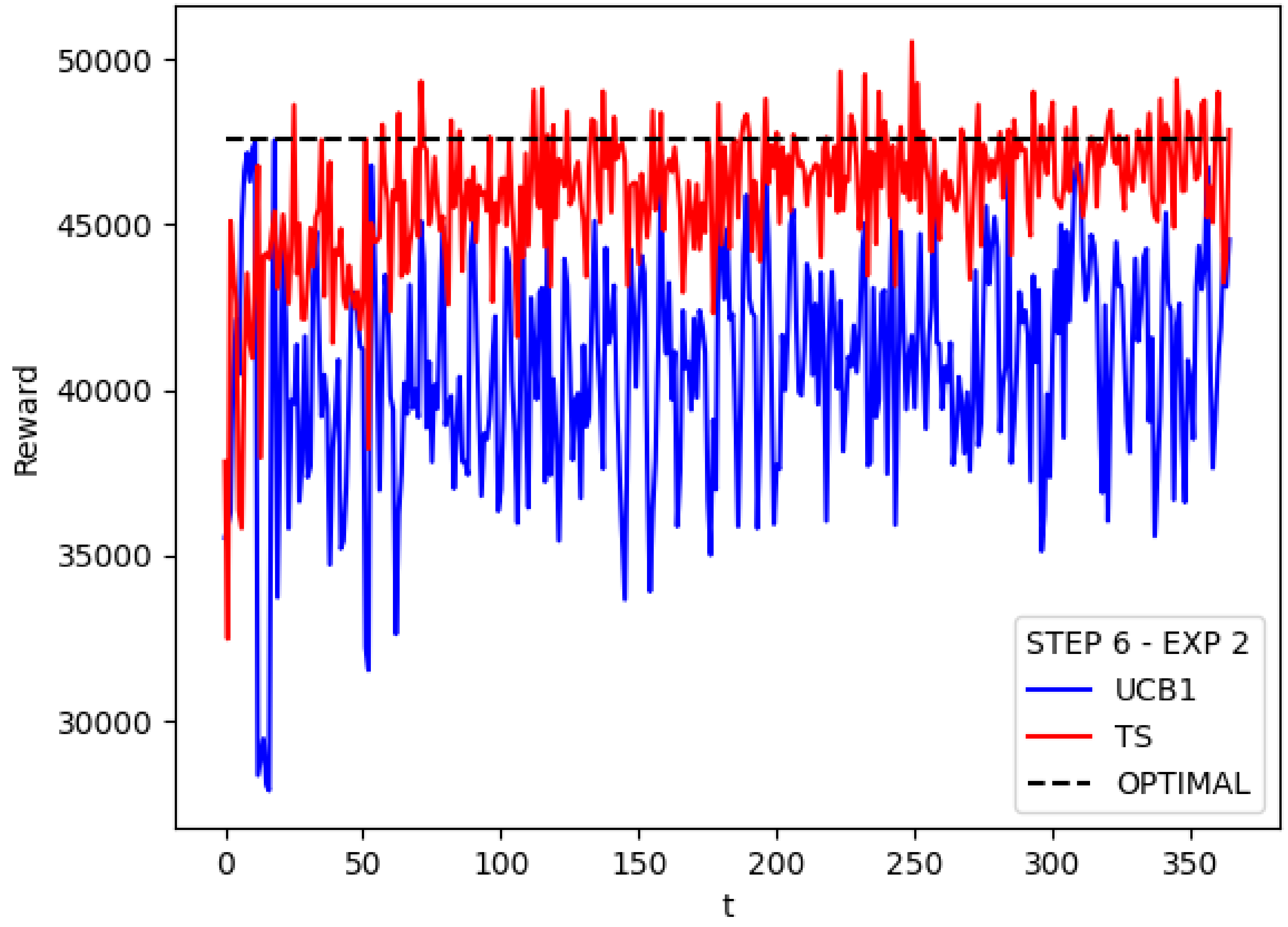




T = 2000 days

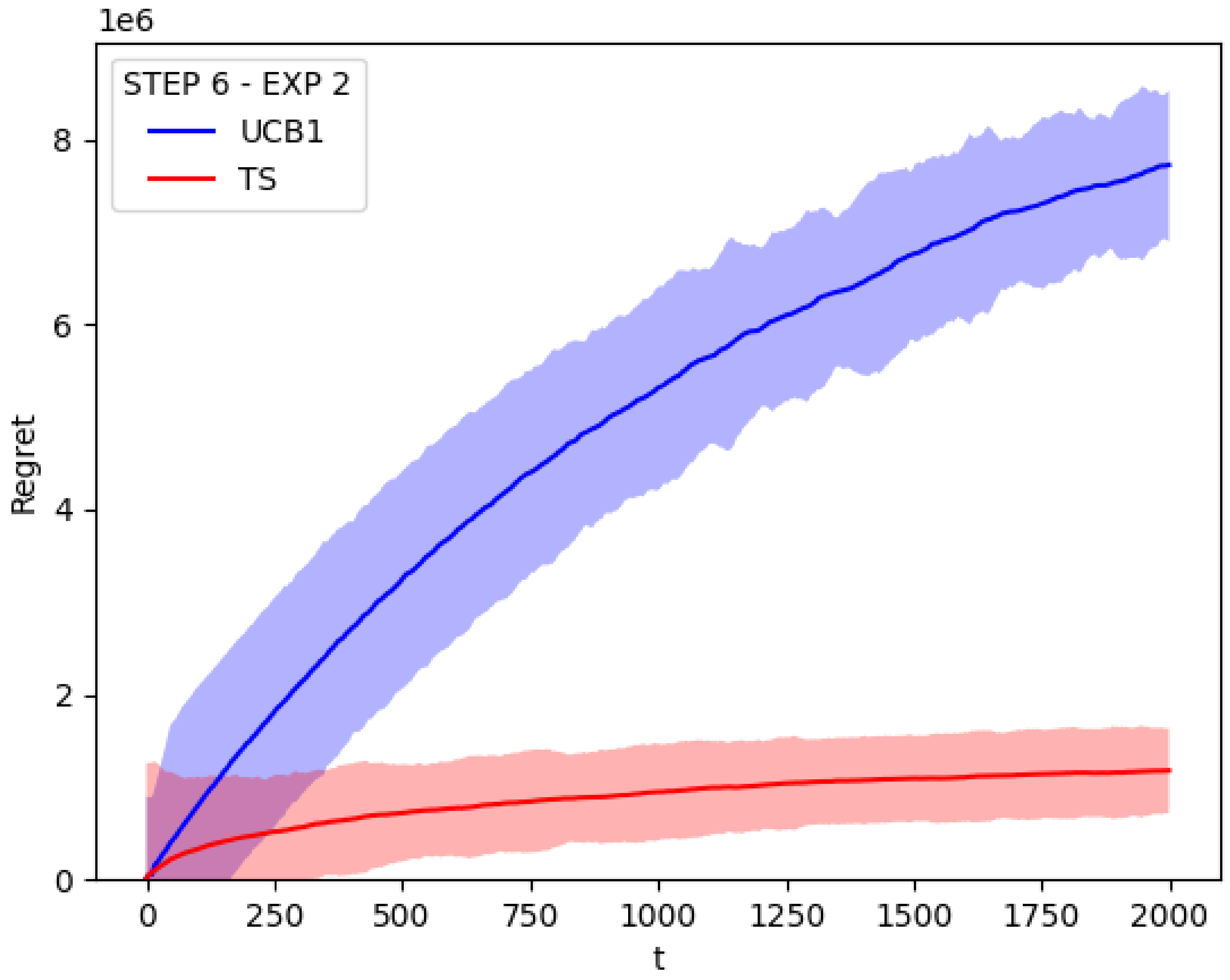
Step 6: Results (second experiment)

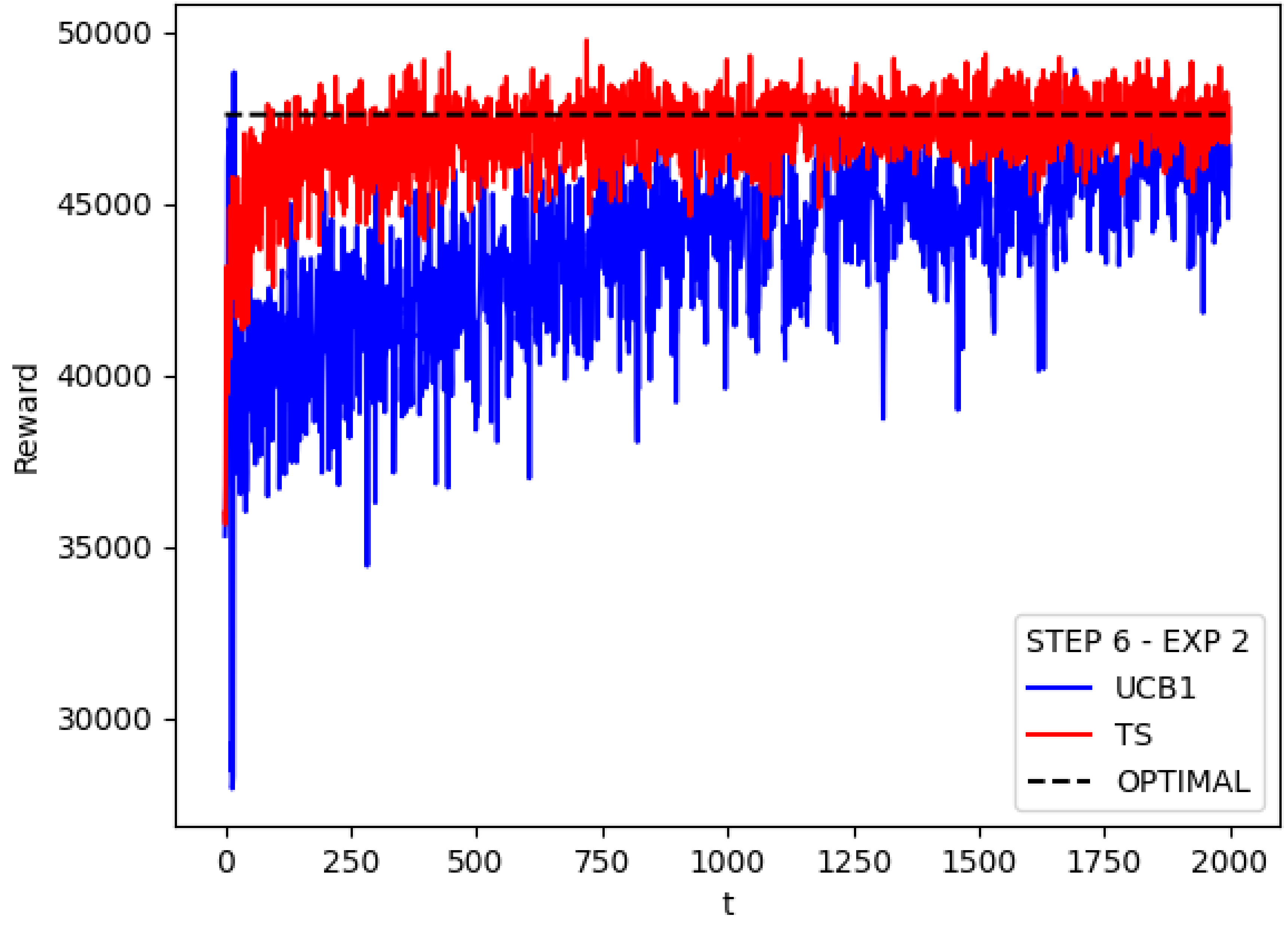




T = 365 days

Step 6: Results (second experiment)





T = 2000 days

Non-Stationary Conversion Rates

Phases:

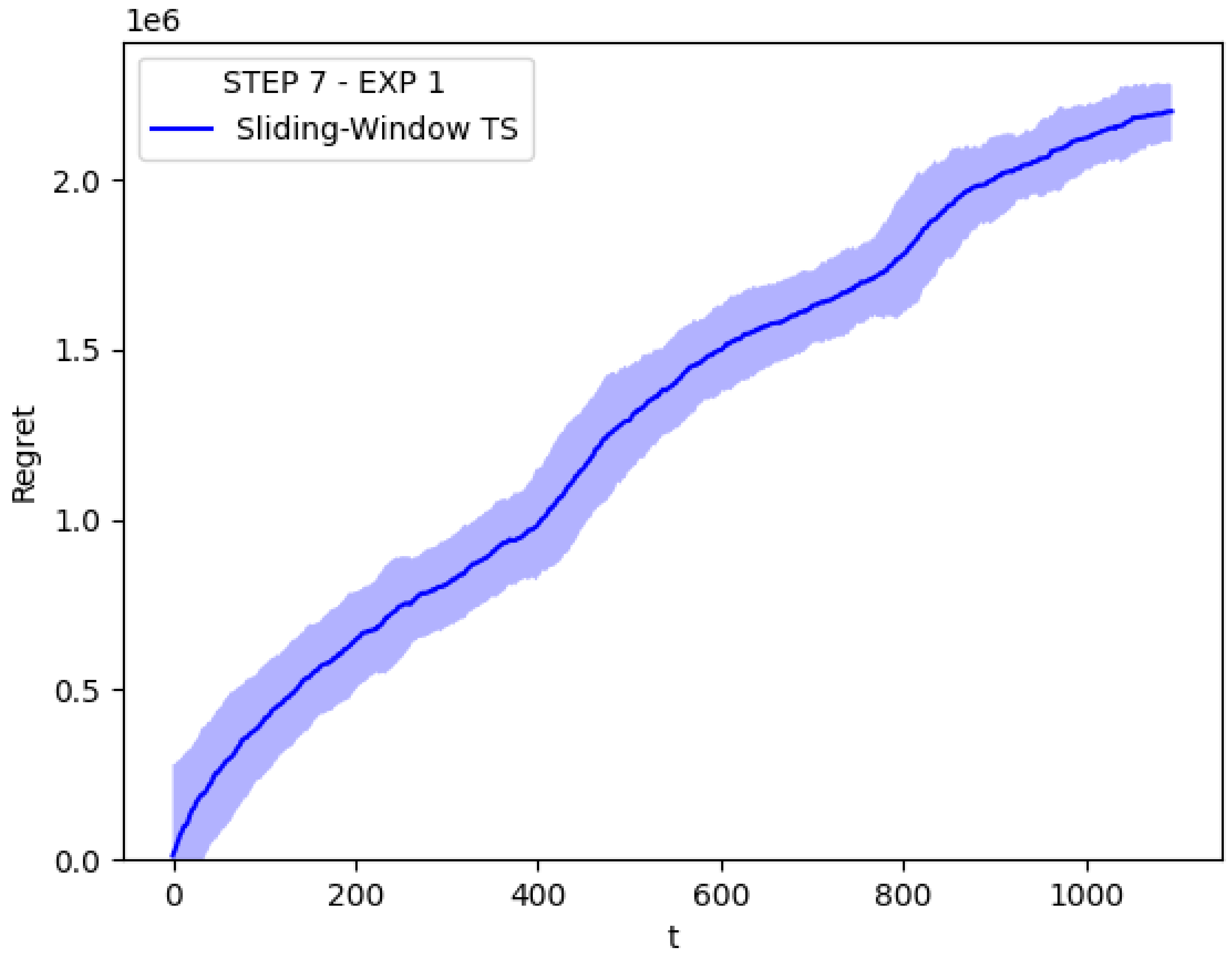
1. The products have just been released on the market.
2. Standard market situation (using the same conversion rates of the stationary steps).
3. The products are becoming obsolete (i.e. a newer model has been released).

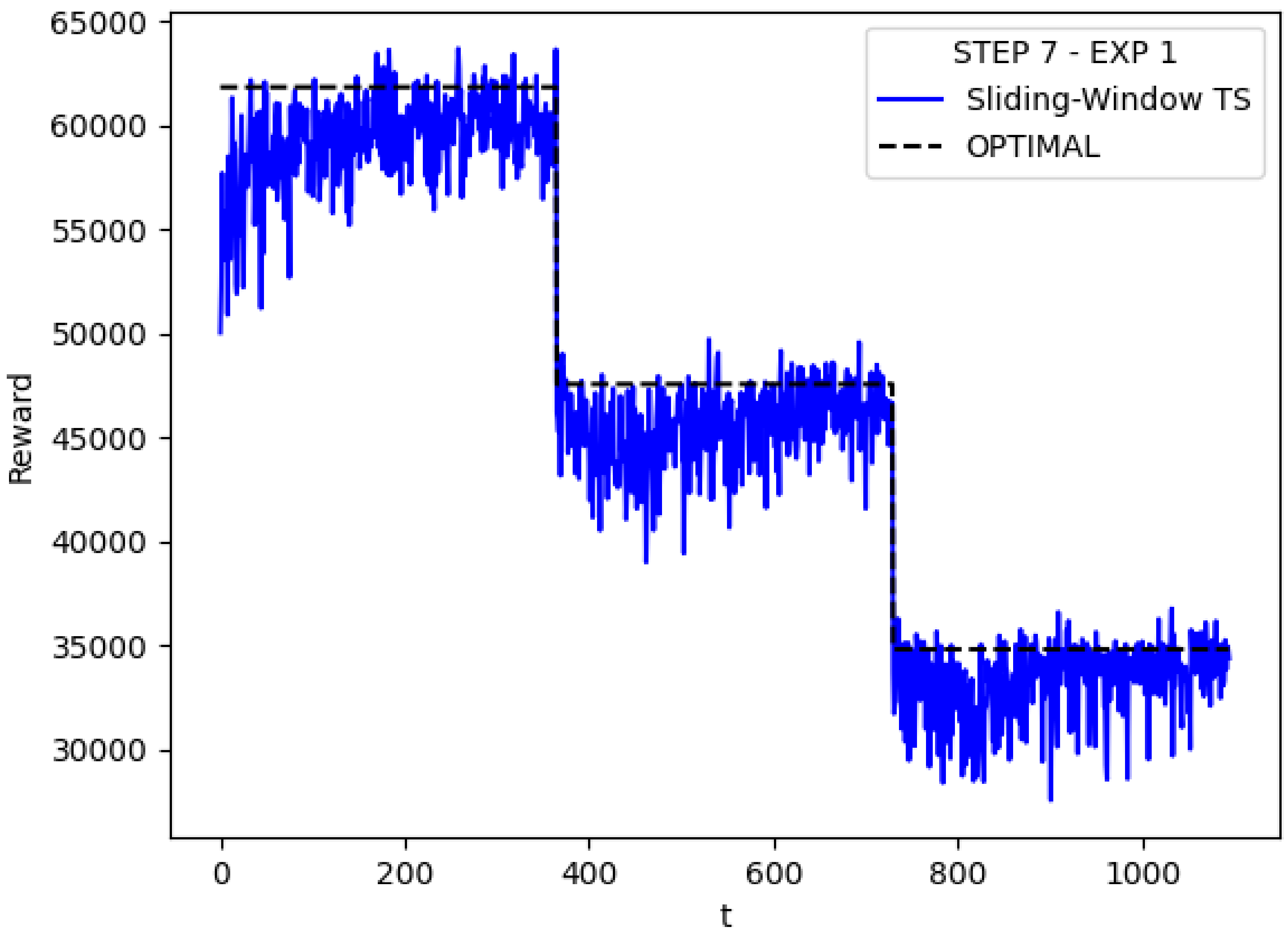
Step 7: Model

SW-TS:

- Arms: cross-product of the margins of the two items.
- Beta distribution first item: standard Beta distribution of the Thompson Sampling Bandit related to the conversion rates of the first item.
- Beta distribution second item: standard Beta distribution of the Thompson Sampling Bandit related to the conversion rates of the second item.
- Pulled arm: revenue given the estimation of the conversion rates of the first item (extraction from Beta distribution first item), the estimation of the conversion rates of the second item (extraction from Beta distribution second item), the number of customers (computing the mean of the daily customers of the previous rounds) and the assignment computed by the linear program for each pair of prices.
- Sliding window data structures, updating the parameters of the Beta distributions considering only the last τ (length of the sliding window) rounds for each time t.

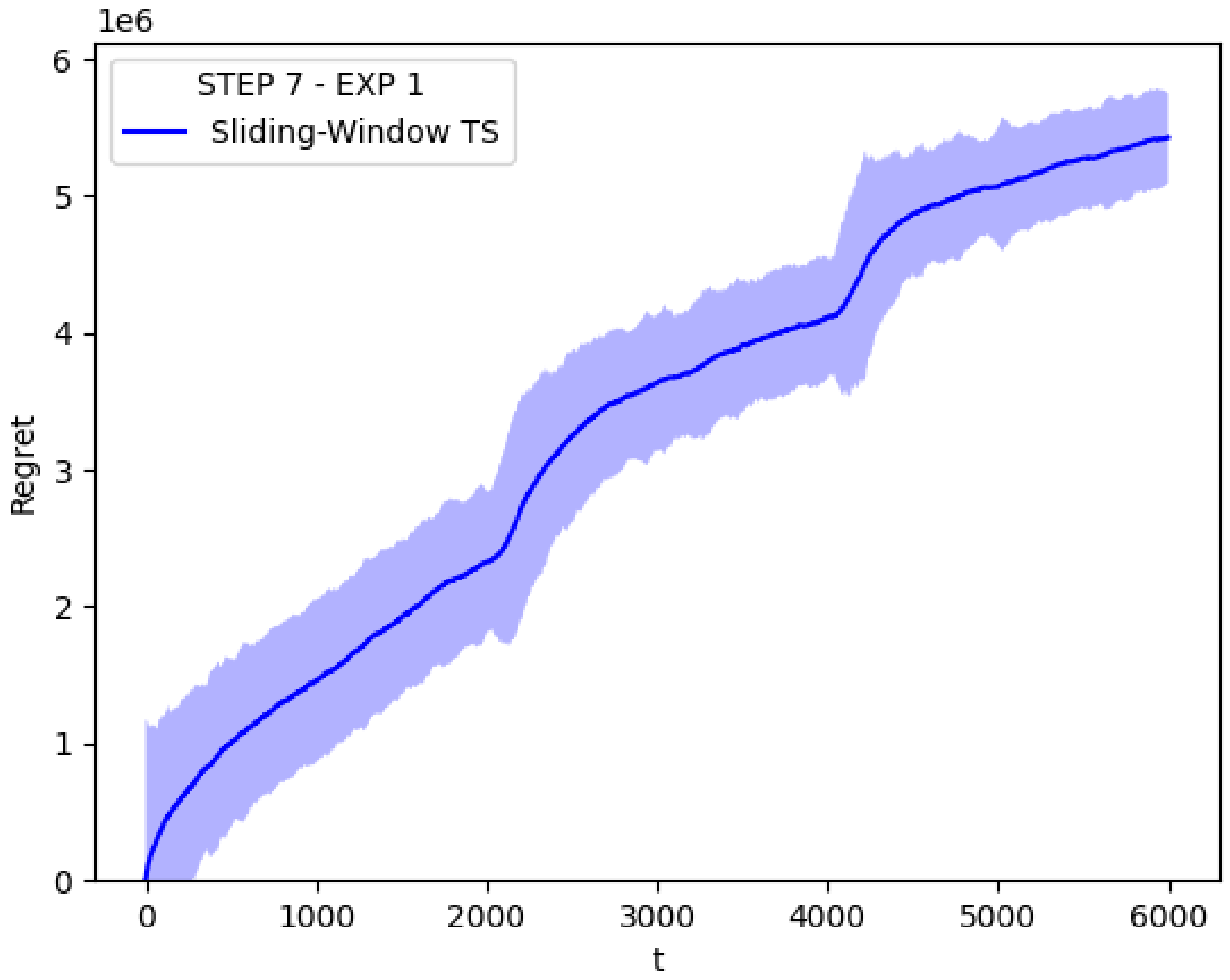
Step 7: Results (first experiment)

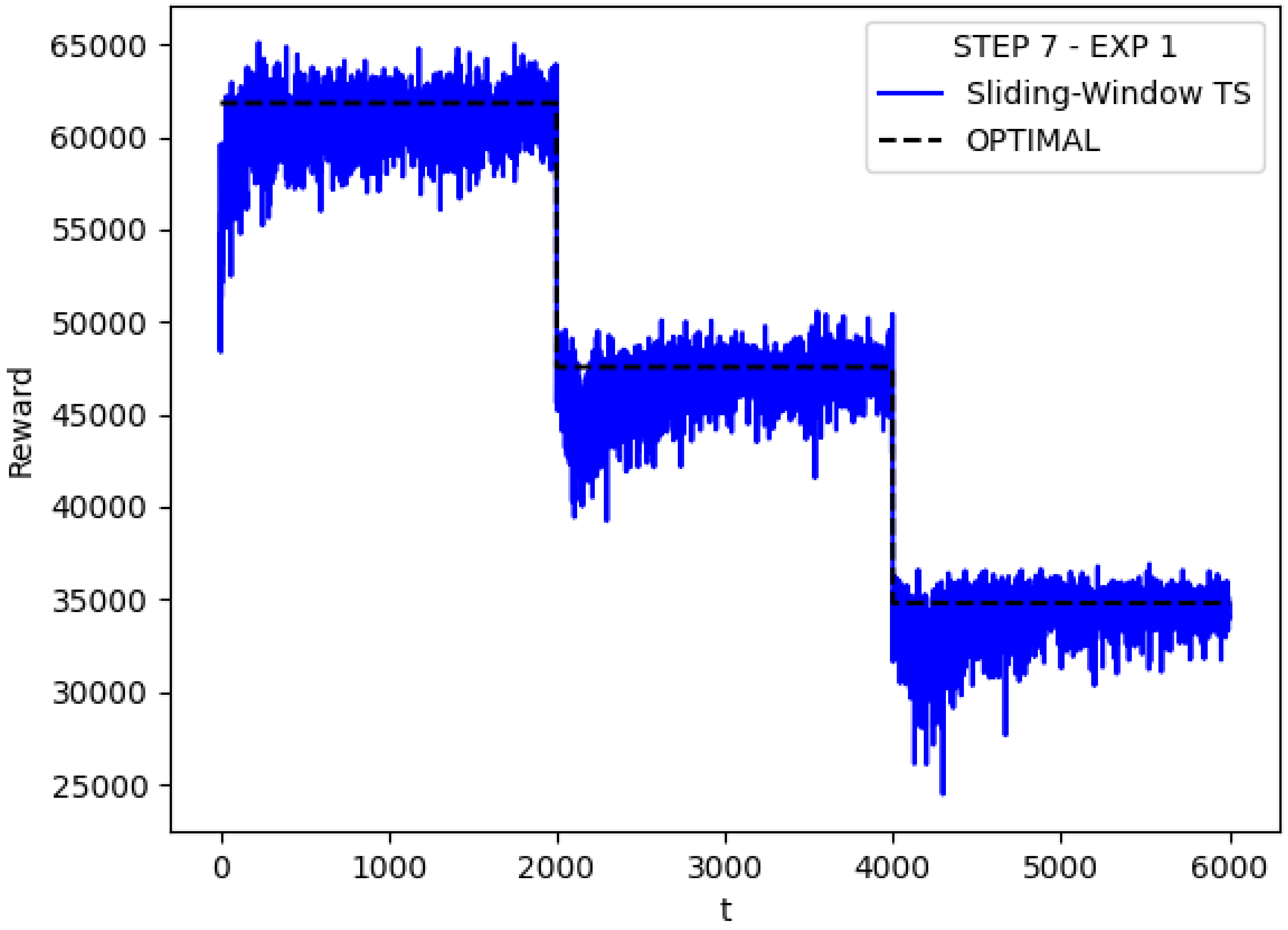




T = 365 days (per phase)

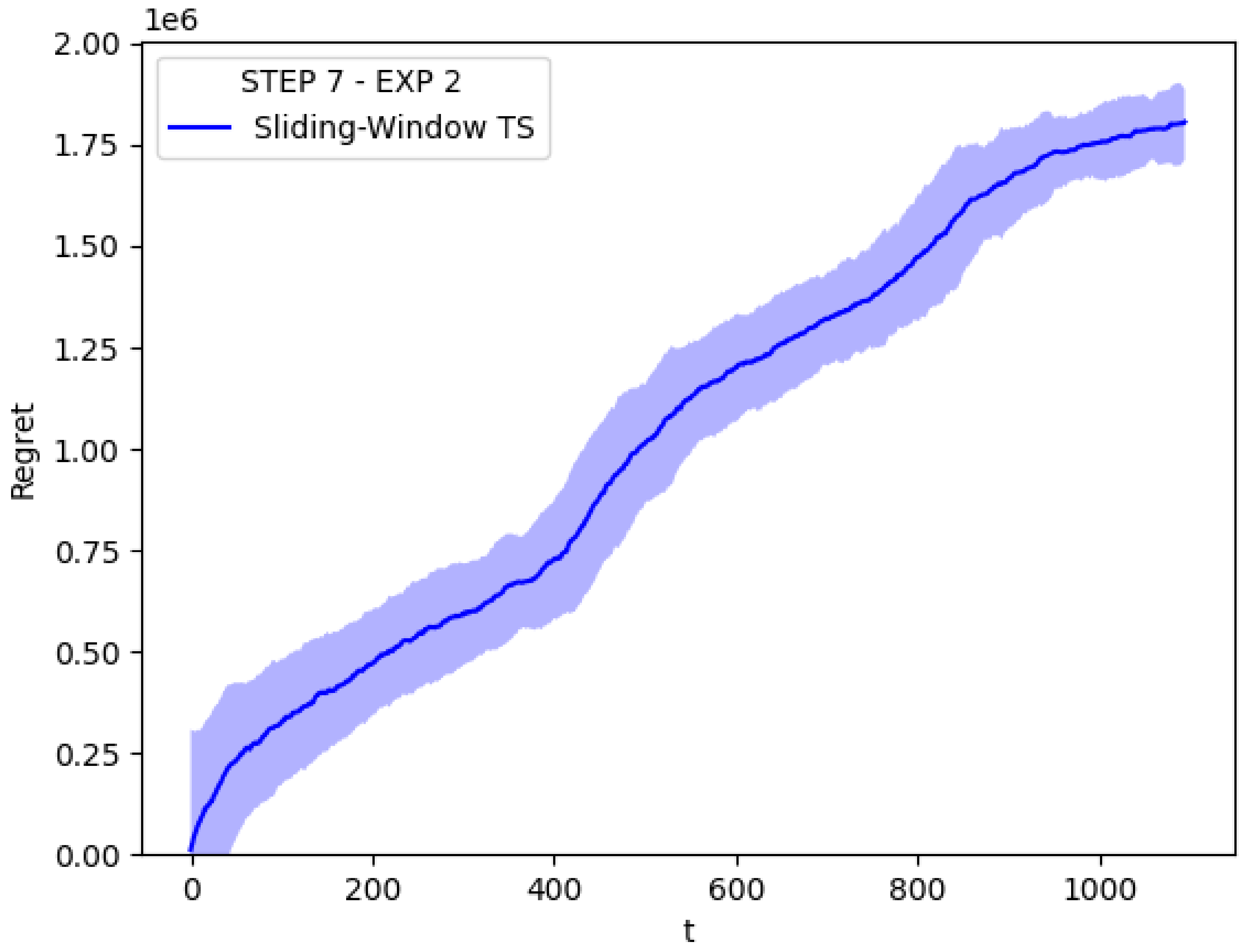
Step 7: Results (first experiment)

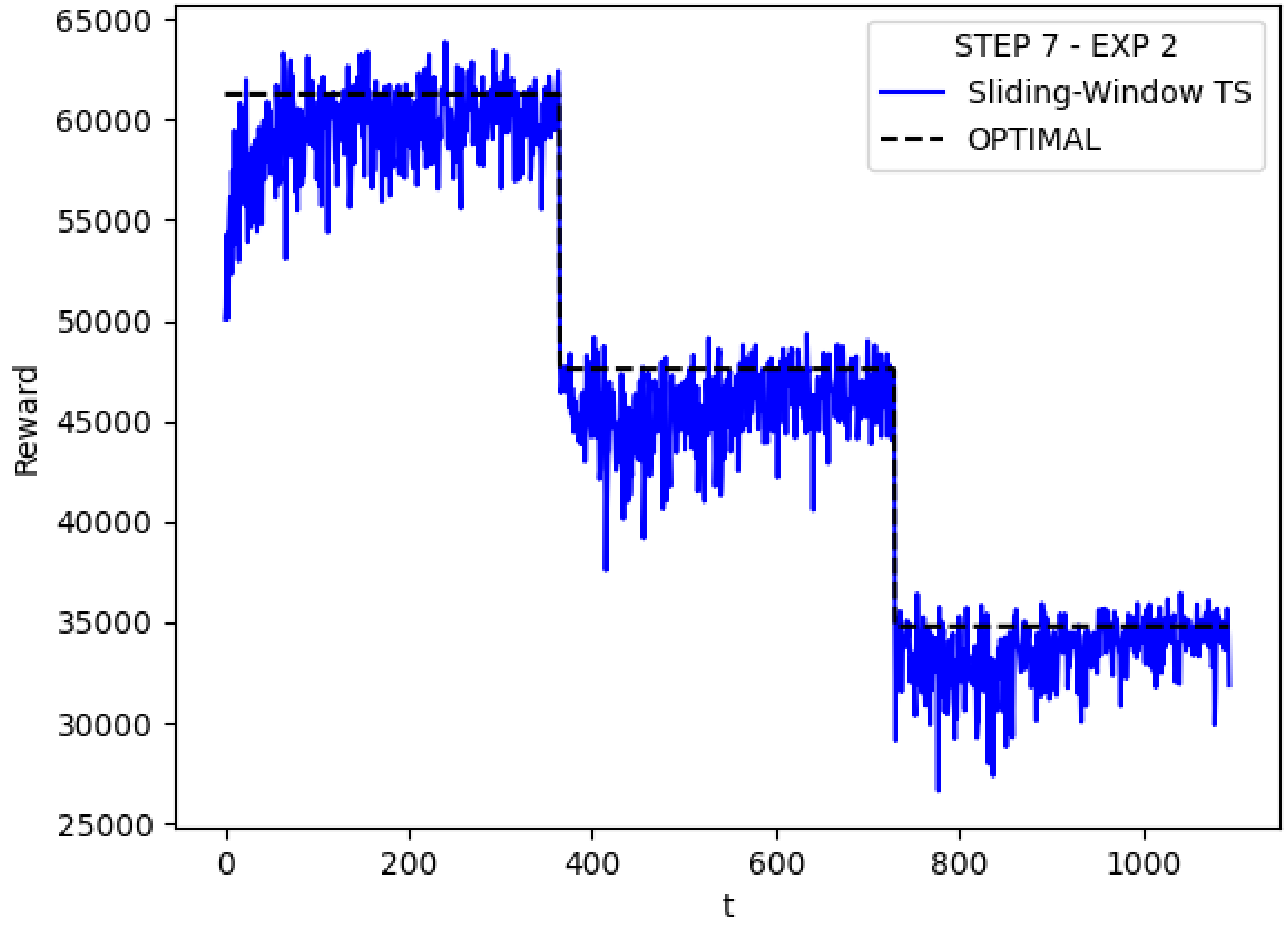




T = 2000 days (per phase)

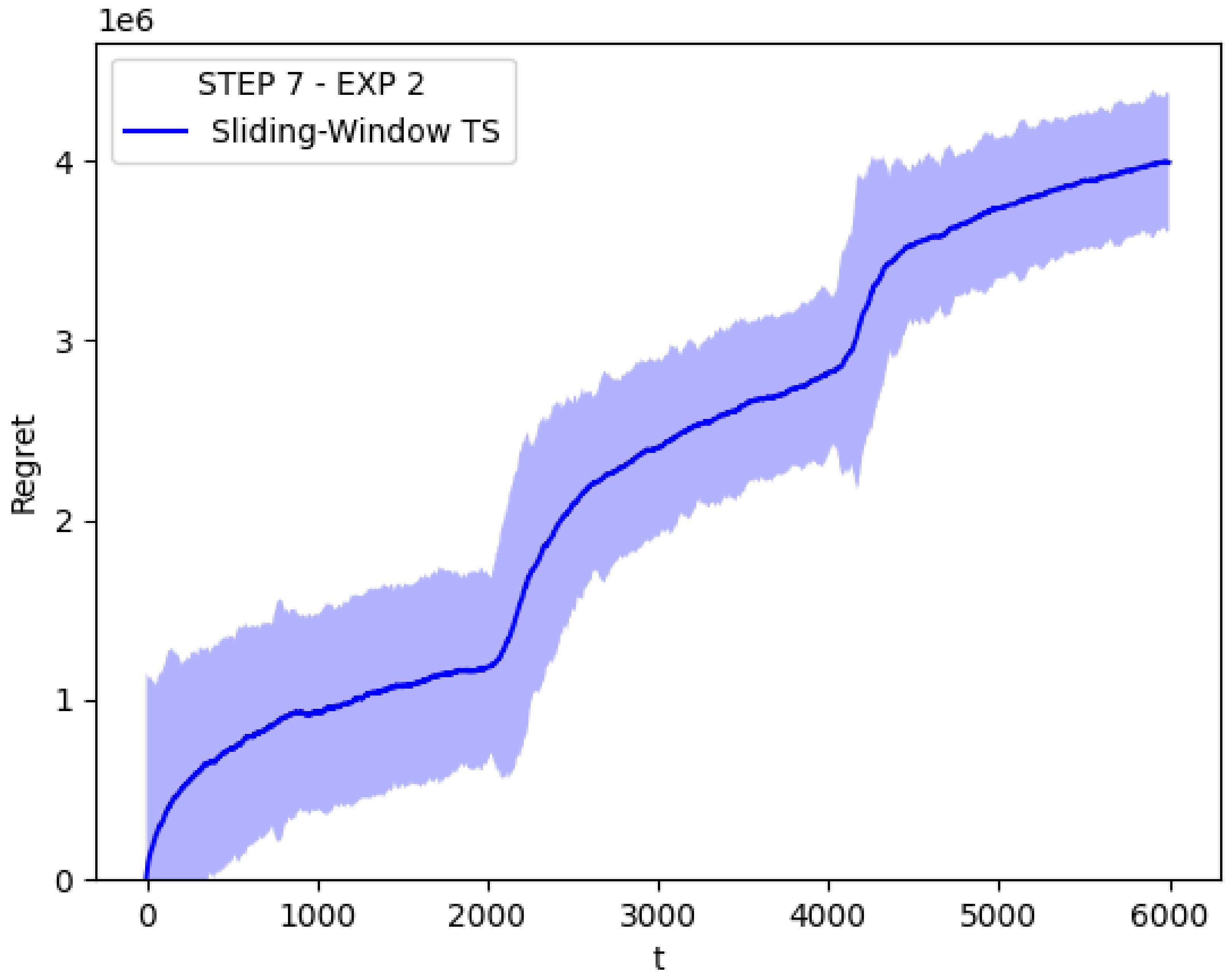
Step 7: Results (second experiment)

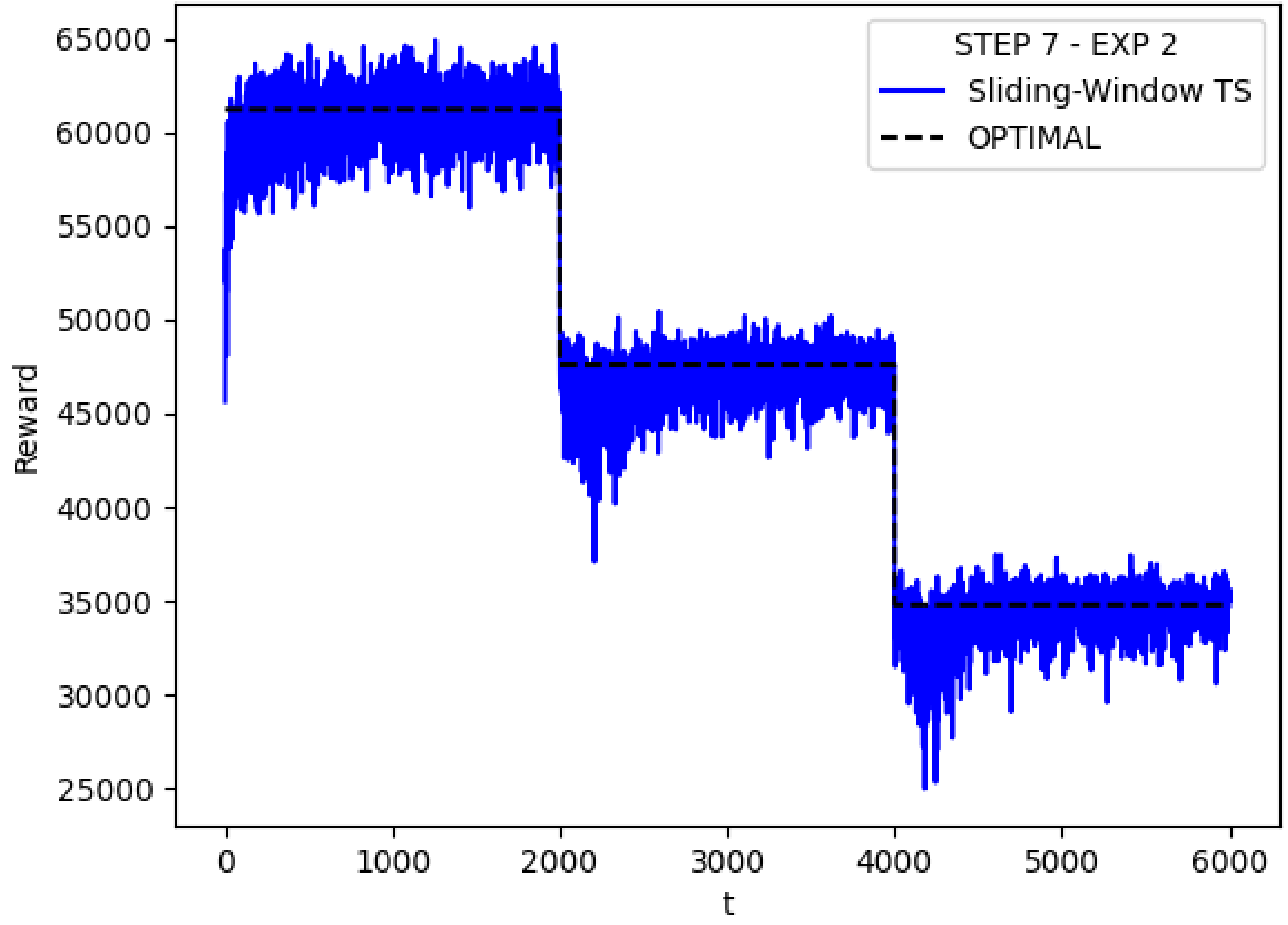




T = 365 days (per phase)

Step 7: Results (second experiment)





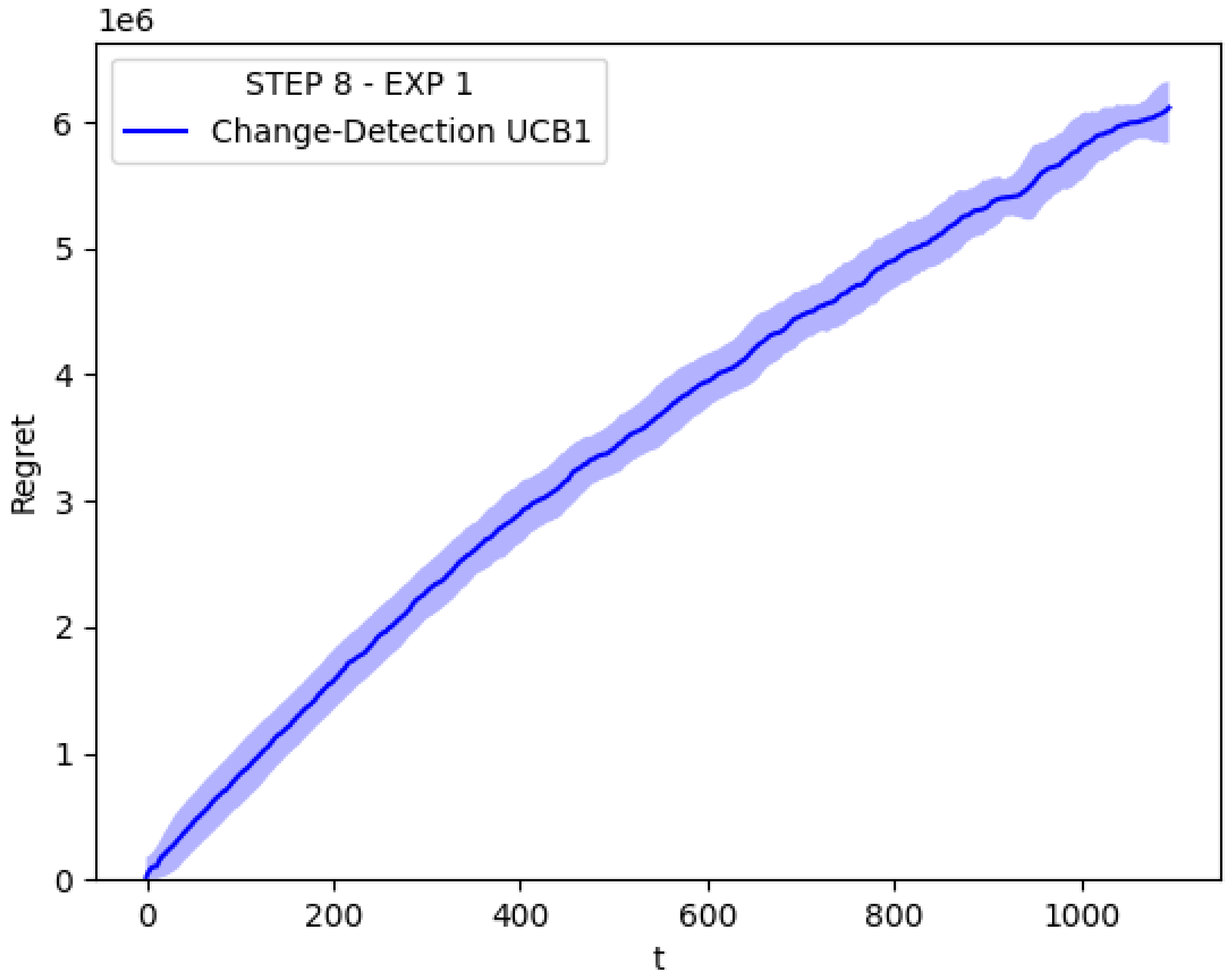
T = 2000 days (per phase)

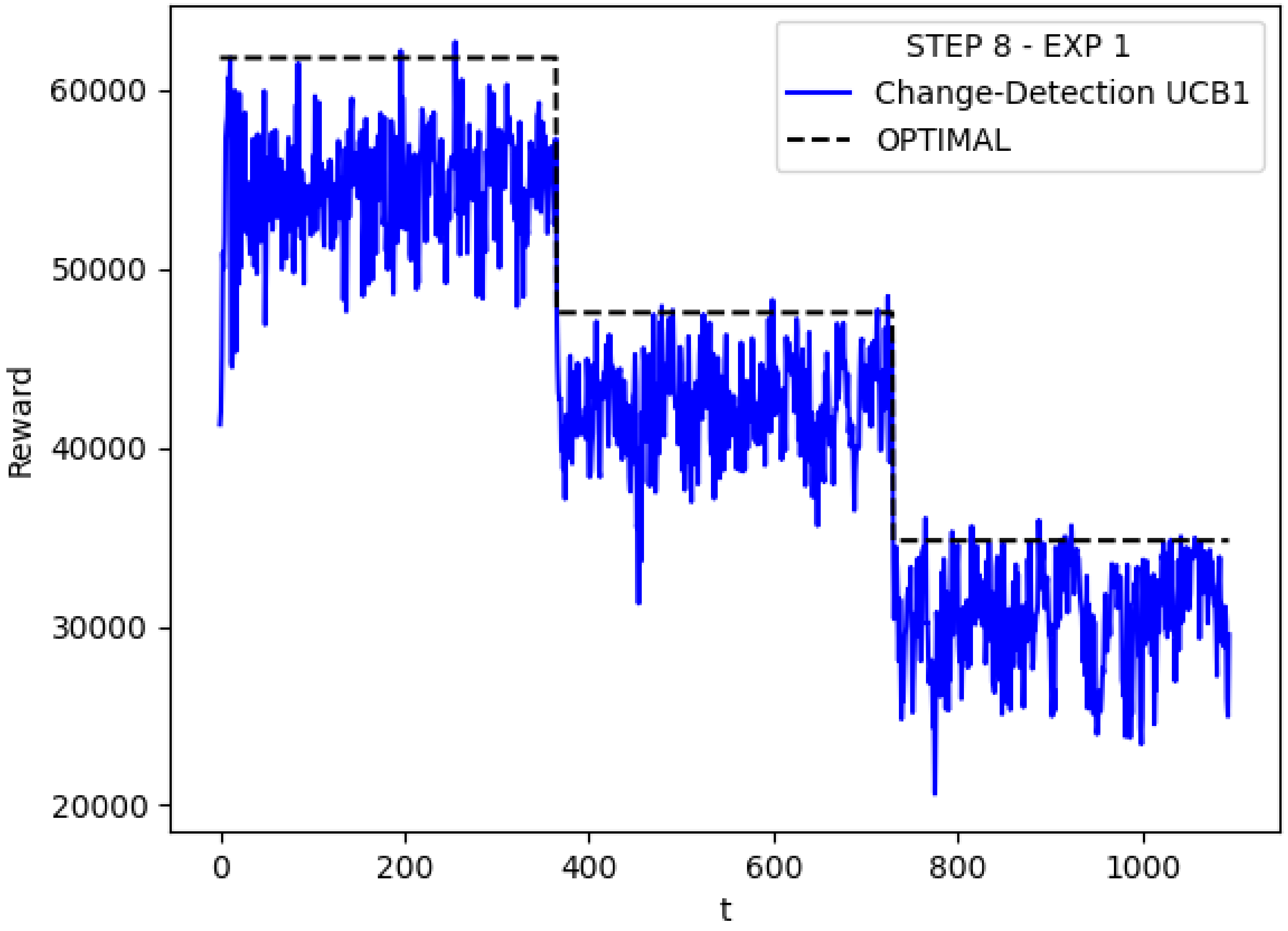
Step 8: Model

UCB1-CUMSUM:

- Arms: cross-product of the margins of the two items.
- Empirical mean first item: estimation of the conversion rates of the first item.
- Empirical mean second item: estimation of the conversion rates of the second item.
- Confidence first item: standard confidence of the UCB1 Bandit related to the first item.
- Confidence second item: standard confidence of the UCB1 Bandit related to the second item.
- Upper bound: revenue given the estimation of the conversion rates of the first item (empirical mean first item plus confidence first item), the estimation of the conversion rates of the second item (empirical mean second item plus confidence second item), the number of customers (computing the mean of the daily customers of the previous rounds) and the assignment computed by the linear program for each pair of prices.
- Change detection data structures to understand when the phases changes, using a cumulative sum approach.

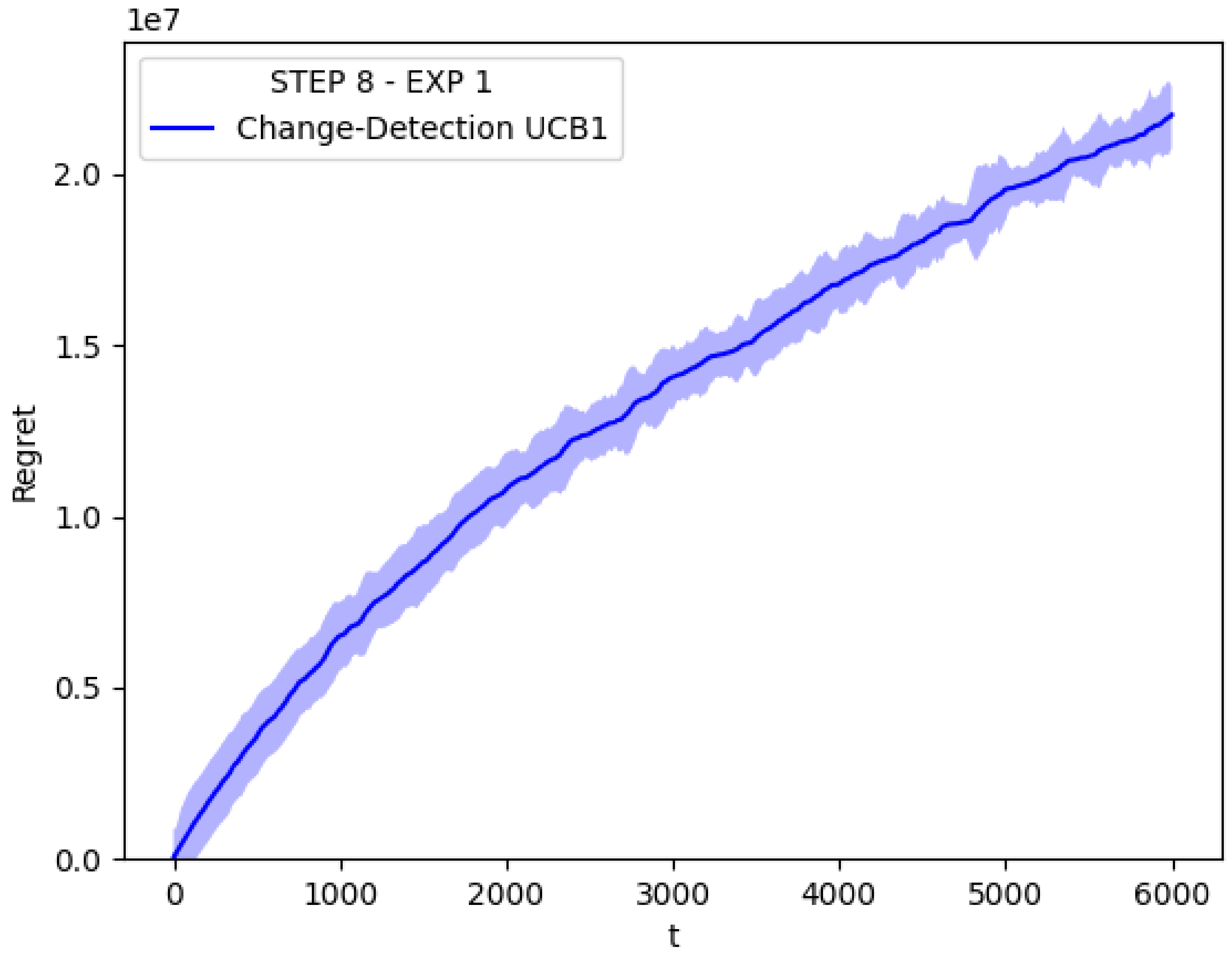
Step 8: Results (first experiment)

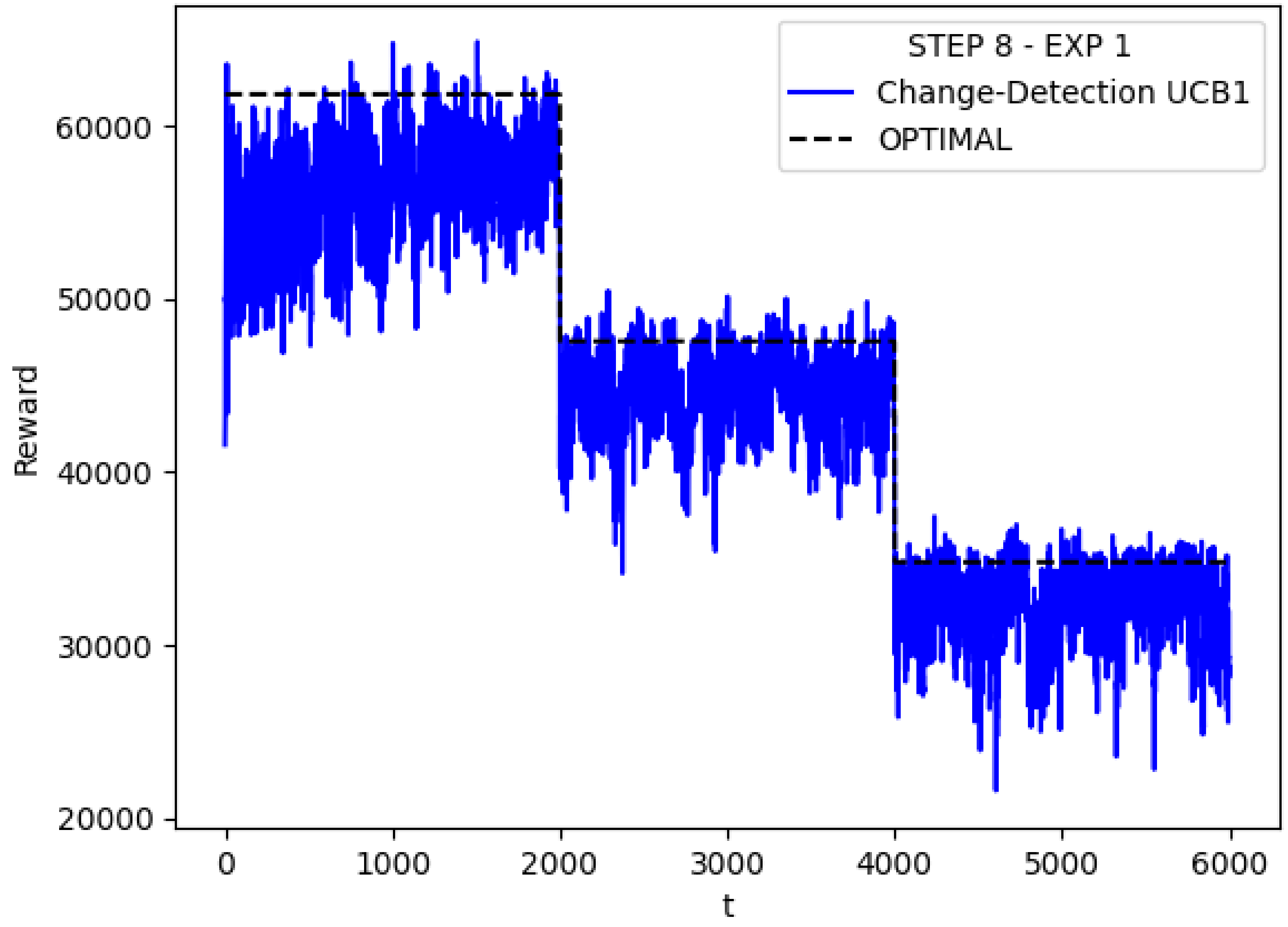




T = 365 days (per phase)

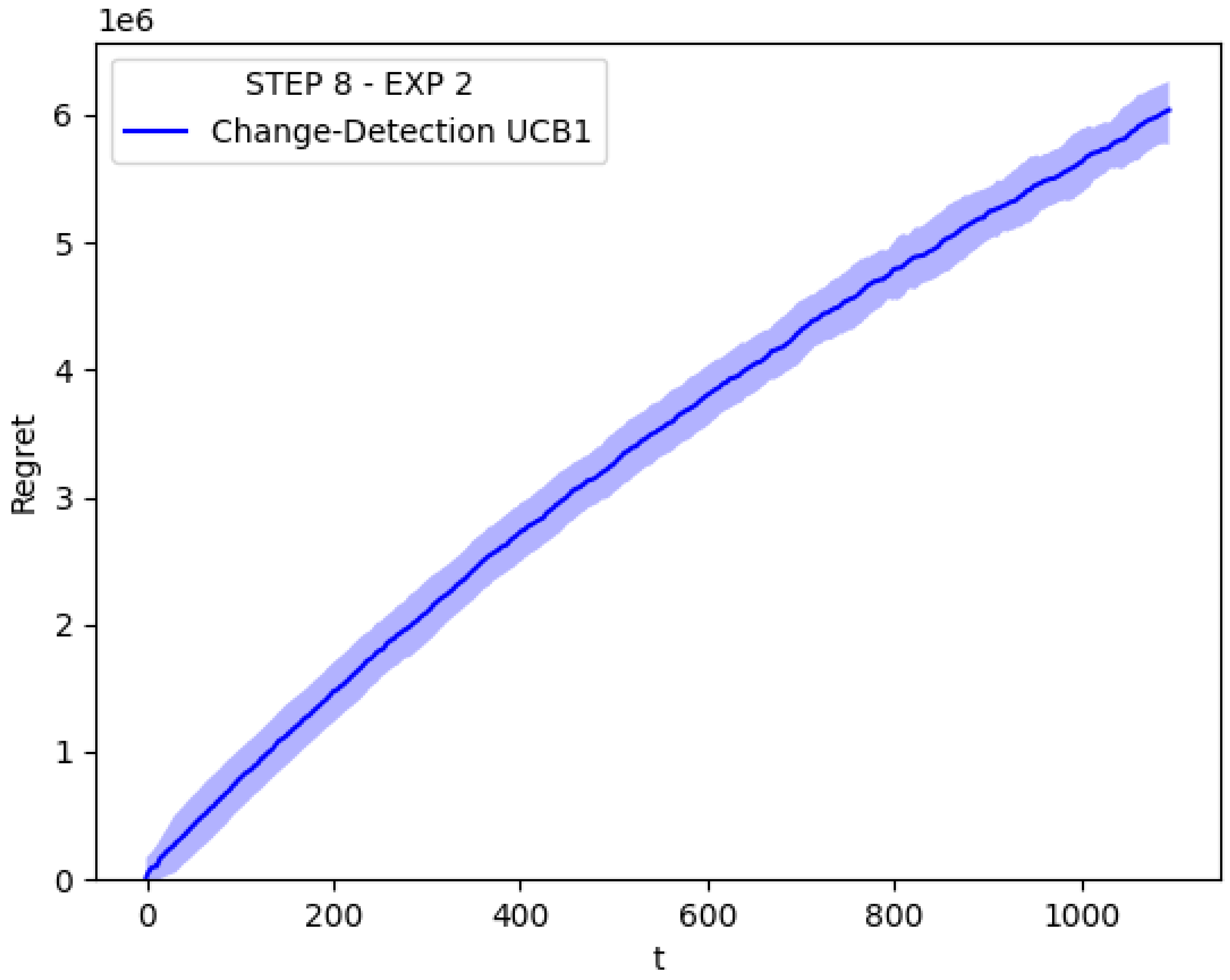
Step 8: Results (first experiment)

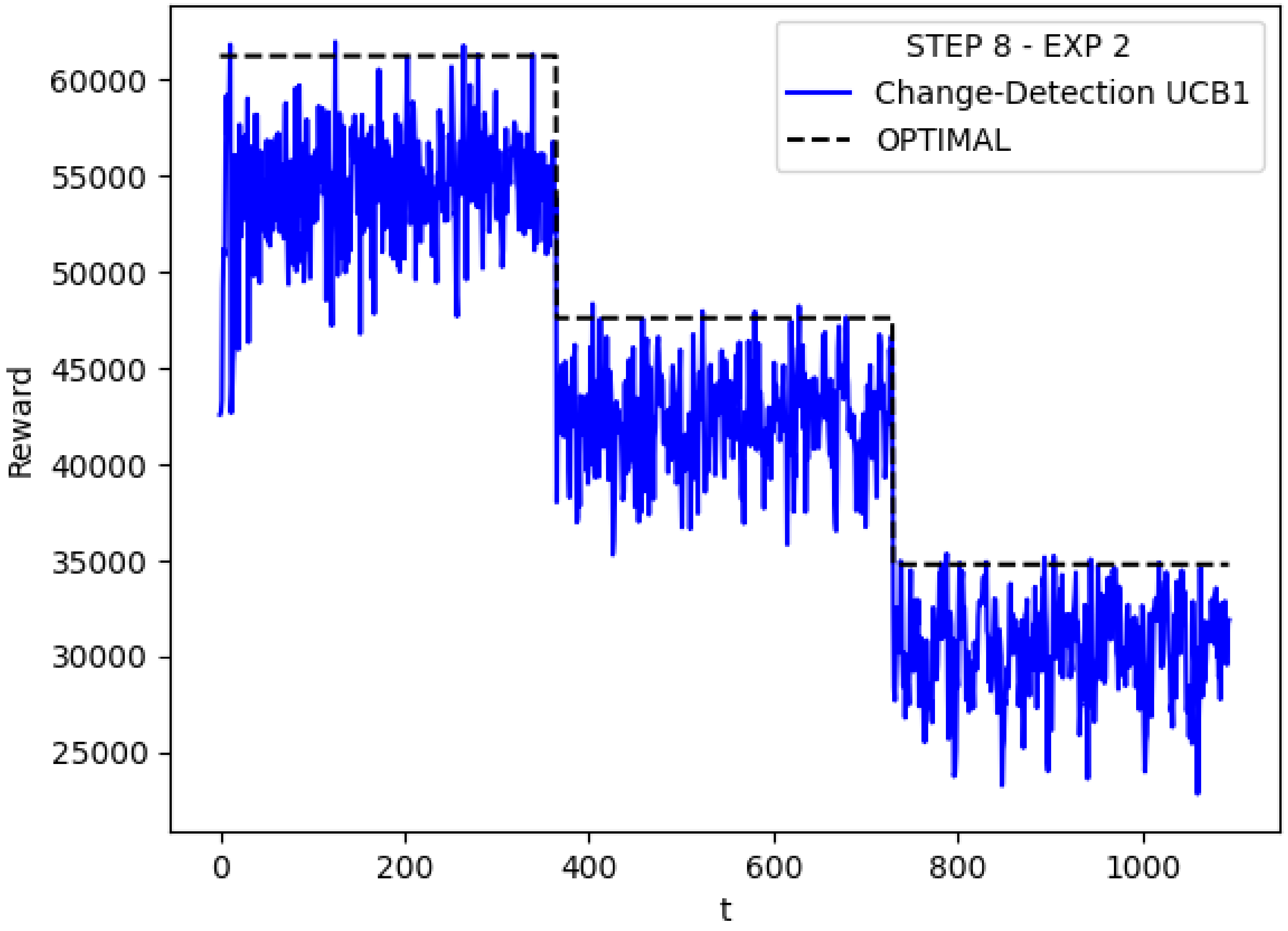




T = 2000 days (per phase)

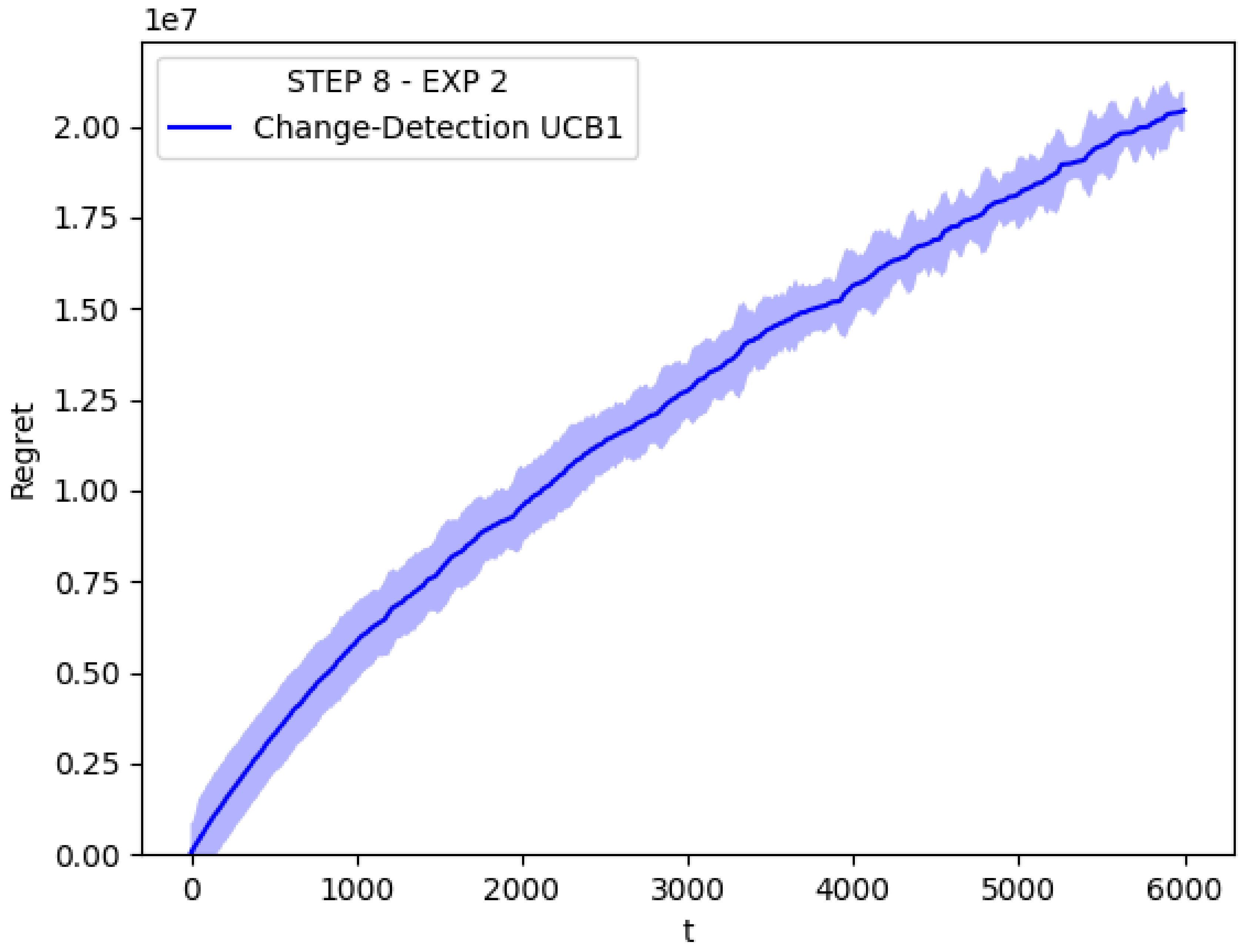
Step 8: Results (second experiment)

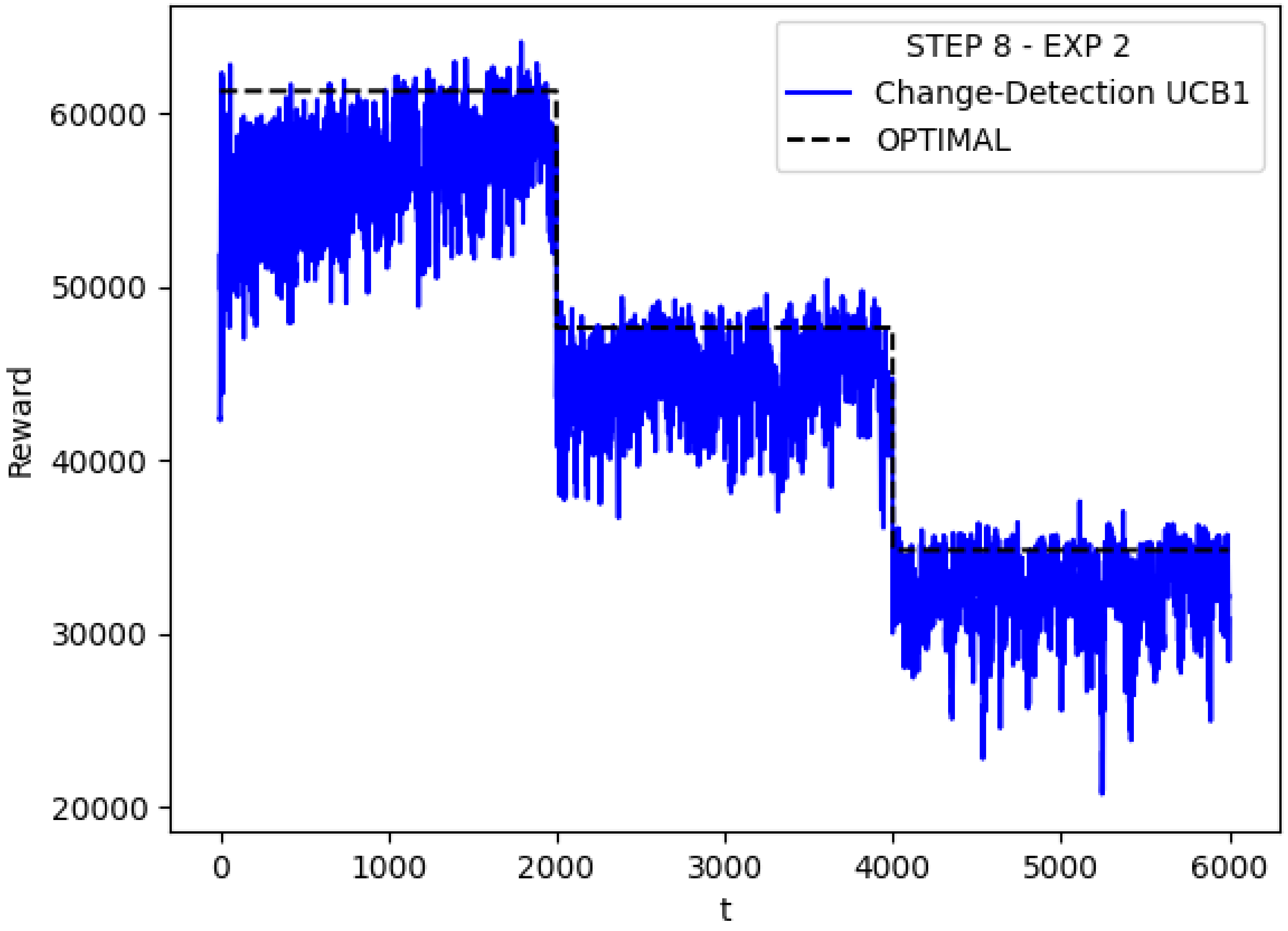




T = 365 days (per phase)

Step 8: Results (second experiment)





T = 2000 days (per phase)

Thank You!