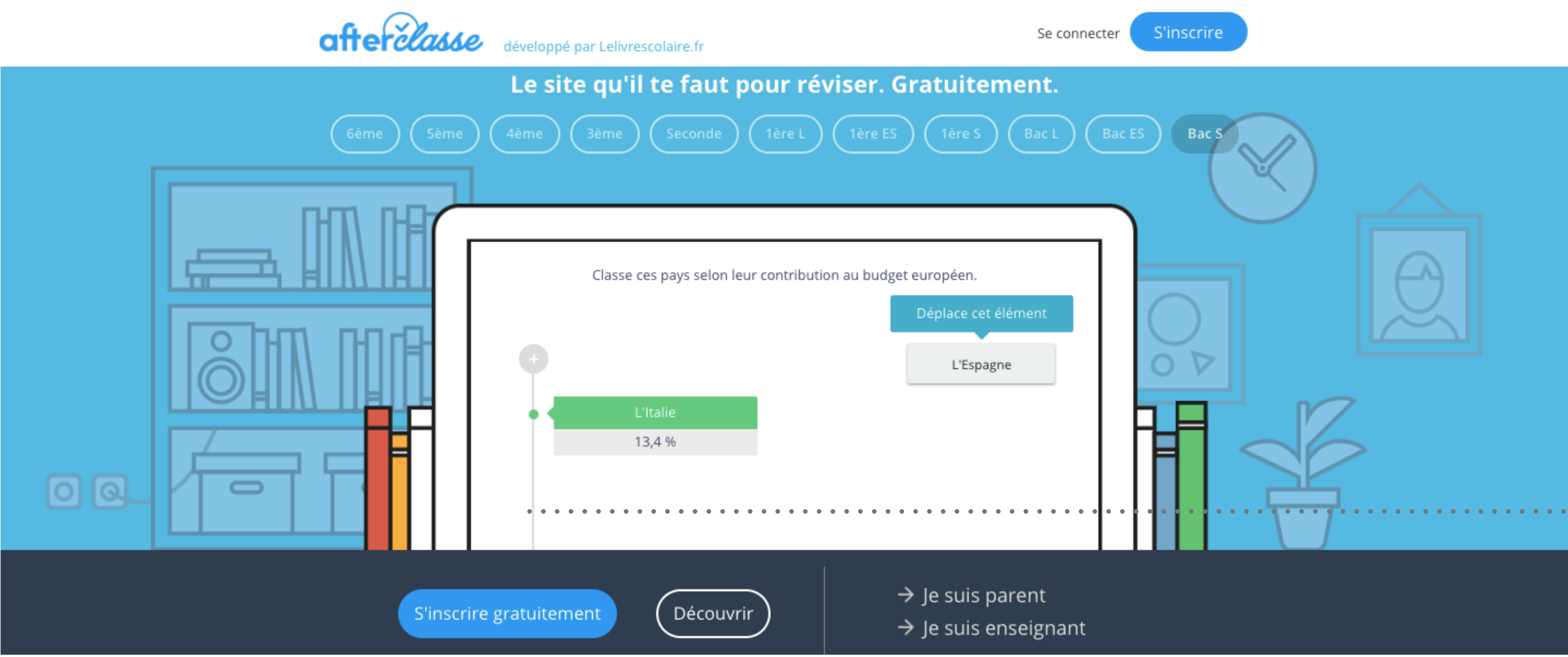


HOW DIFFICULT ARE ROTTING BANDITS?

Inria inventors for the digital world

https://www.afterclasse.fr

lelivrescolaire.fr

Julien SEZNEC

with A. Locatelli, A. Carpentier, A. Lazaric, M. Valko

SequeL @ Inria Lille — Nord Europe

WHEN BANDITS GO ROTTING …

WHEN BANDITS GO ROTTING …

2 days before the national exam

WHEN BANDITS GO ROTTING …

CHAPITRE 1

L'origine des séismes et des éruptions volcaniques

CHAPITRE 2

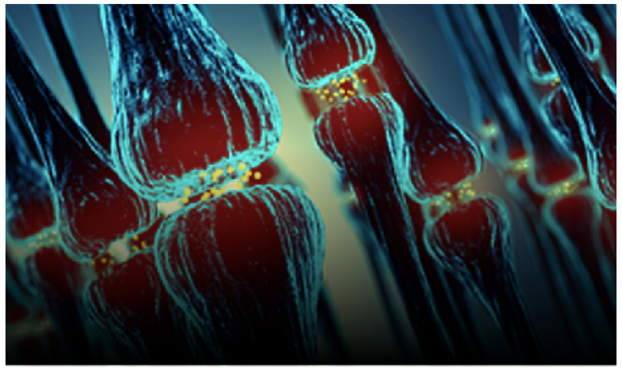
Les changements climatiques actuels et leurs conséquences

afterclasse

2 days before the national exam

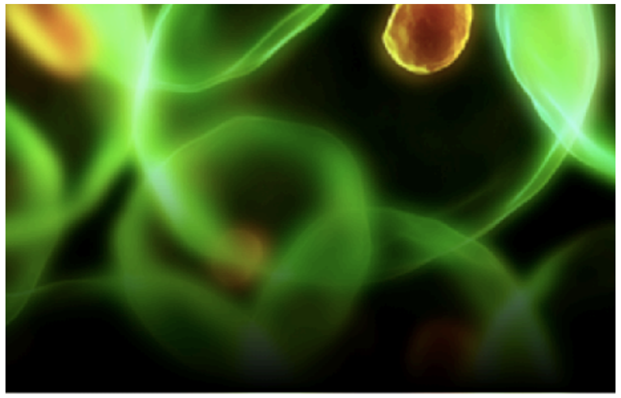
CHAPITRE 3

Les impacts des activités humaines sur l'environnement



CHAPITRE 8

Le fonctionnement du système nerveux



CHAPITRE 4

La nutrition à l'échelle cellulaire

Inria informatics mathematics

lelivrescolaire.fr

ROTTING BANDITS ARE …

Stochastic bandits …

- K arms
- At each round t, agent pulls arm i and receives a noisy reward $r_t \leftarrow \mu_i + \epsilon_t$ (ϵ_t i.i.d. ;σ-subgaussian)
- Maximize cumulative reward : $\mathbb{E}\left[\sum_{t \leq T} r_t\right]$

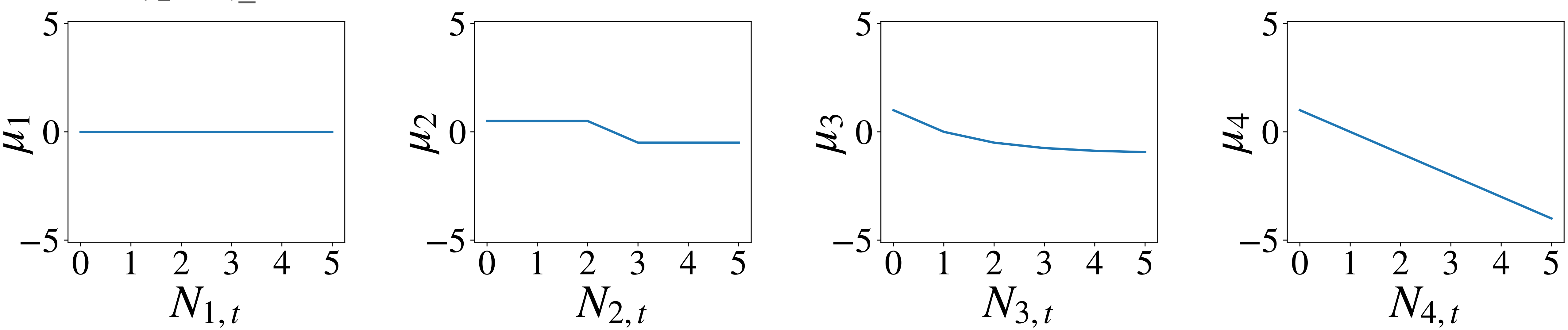
ROTTING BANDITS ARE …

Stochastic bandits …

- K arms
- At each round t, agent pulls arm i and receives a noisy reward $r_t \leftarrow \mu_i + \epsilon_t$ (ϵ_t i.i.d. ;σ-subgaussian)
- Maximize cumulative reward : $\mathbb{E}\left[\sum_{t \leq T} r_t\right]$

… with rotting arms

- $\{\mu_i\}$ are ***non-increasing*** functions of $N_{i,t}$ the ***number of pulls of arm*** i at time t
- $L \triangleq \max_{i \in K} \max_{n \leq T} \mu_i(n) - \mu_i(n+1)$



OPTIMAL ORACLE POLICY [HEIDARI, 2016]

Algorithm 1 $\mathcal{A}_0$ (Heidari et al., 2016)

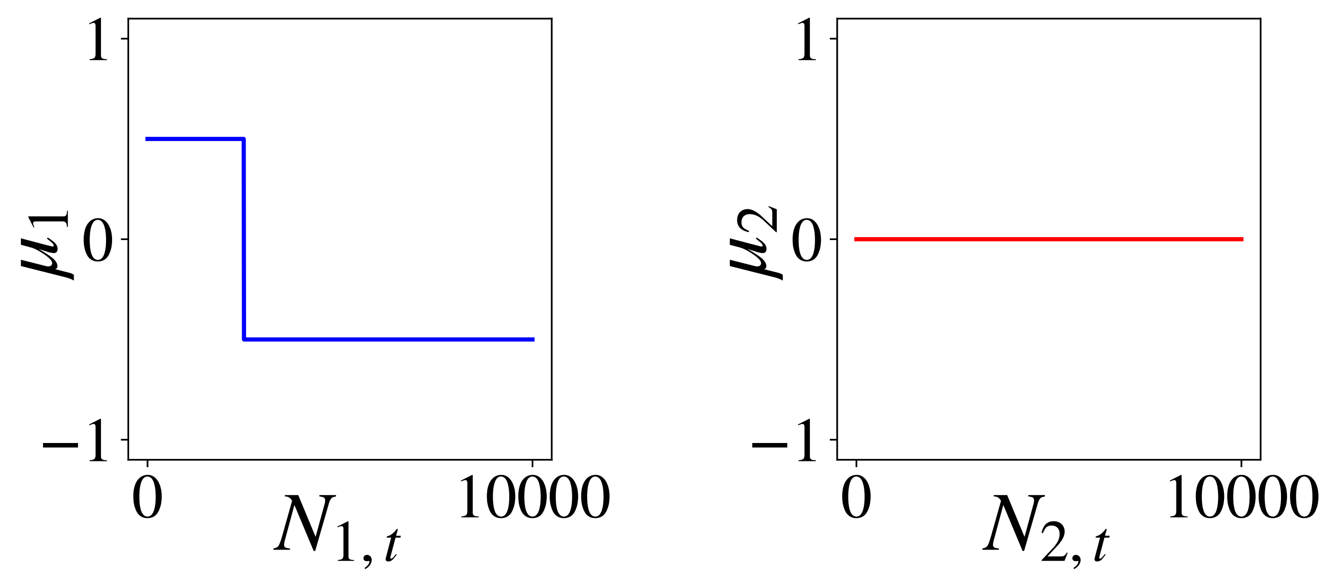
```
for t ← 1, 2, ... do
    SELECT : arg max_{i∈K} μ_i(N_{i,t})
end for
```

$$\textbf{for } t \leftarrow 1, 2, \ldots \textbf{ do} \quad \text{SELECT} : \arg\max_{i \in \mathcal{K}} \mu_i(N_{i,t}) \quad \textbf{end for}$$

OPTIMAL ORACLE POLICY [HEIDARI, 2016]

Algorithm 1 $\mathcal{A}_0$ (Heidari et al., 2016)

for $t \leftarrow 1, 2, \ldots$ **do**
    SELECT : $\arg\max_{i \in \mathcal{K}} \mu_i(N_{i,t})$
end for



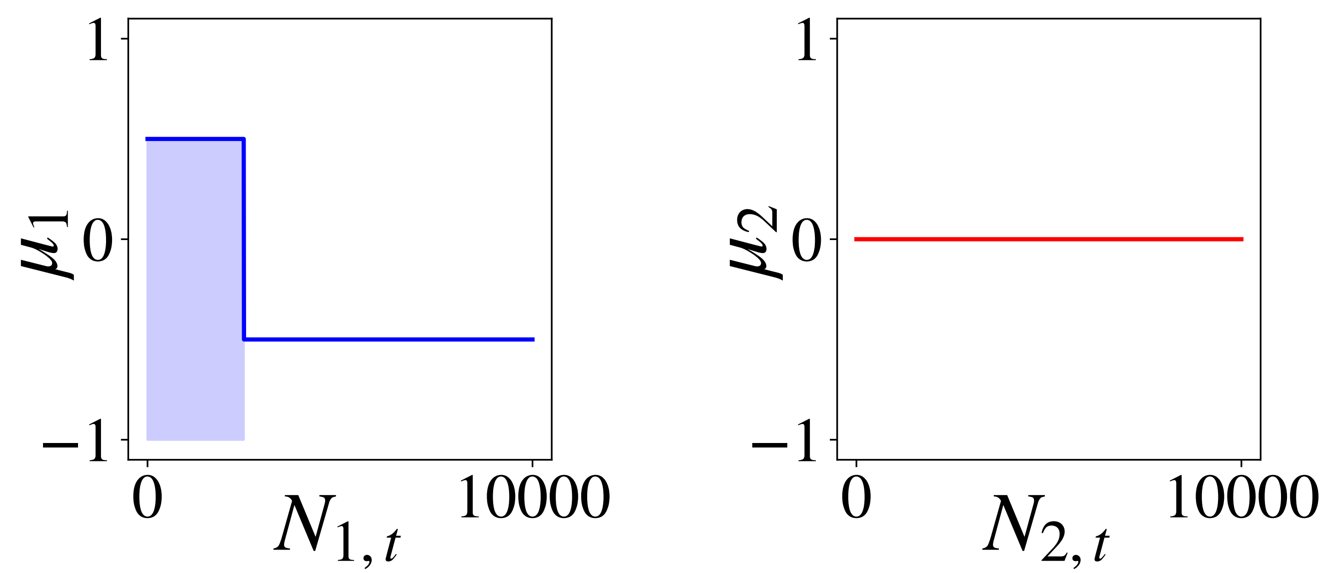
informatics mathematics

lelivrescolaire.fr

OPTIMAL ORACLE POLICY [HEIDARI, 2016]

Algorithm 1 $\mathcal{A}_0$ (Heidari et al., 2016)

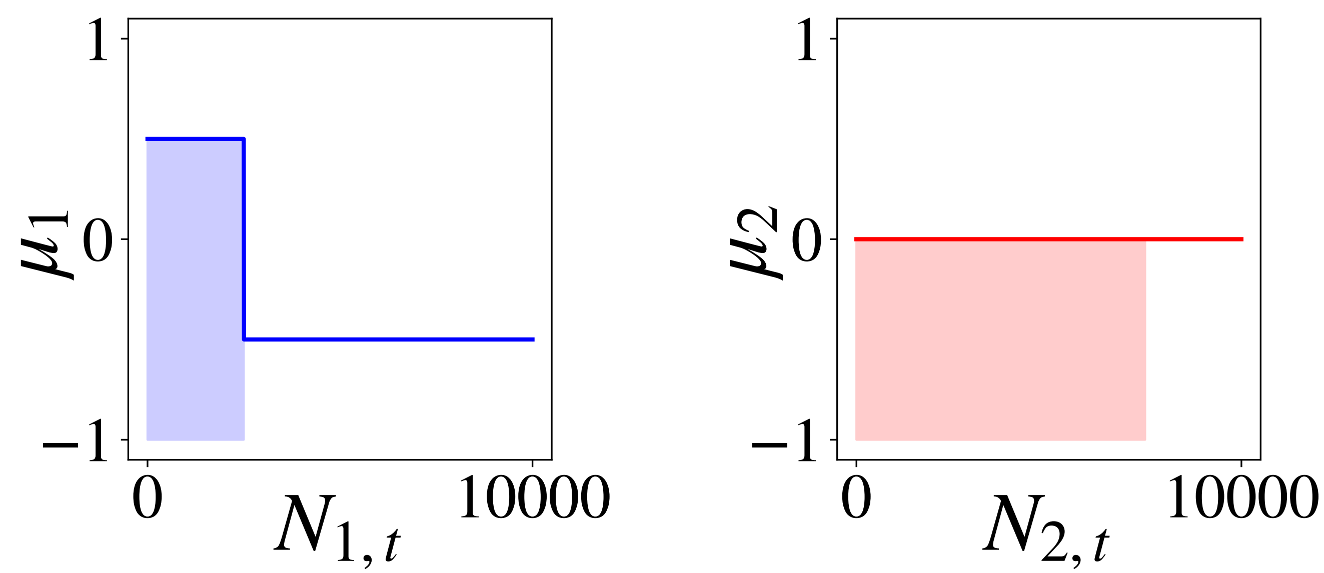
for $t \leftarrow 1, 2, \ldots$ **do**
    SELECT : $\arg\max_{i \in \mathcal{K}} \mu_i(N_{i,t})$
end for



OPTIMAL ORACLE POLICY [HEIDARI, 2016]

Algorithm 1 $\mathcal{A}_0$ (Heidari et al., 2016)

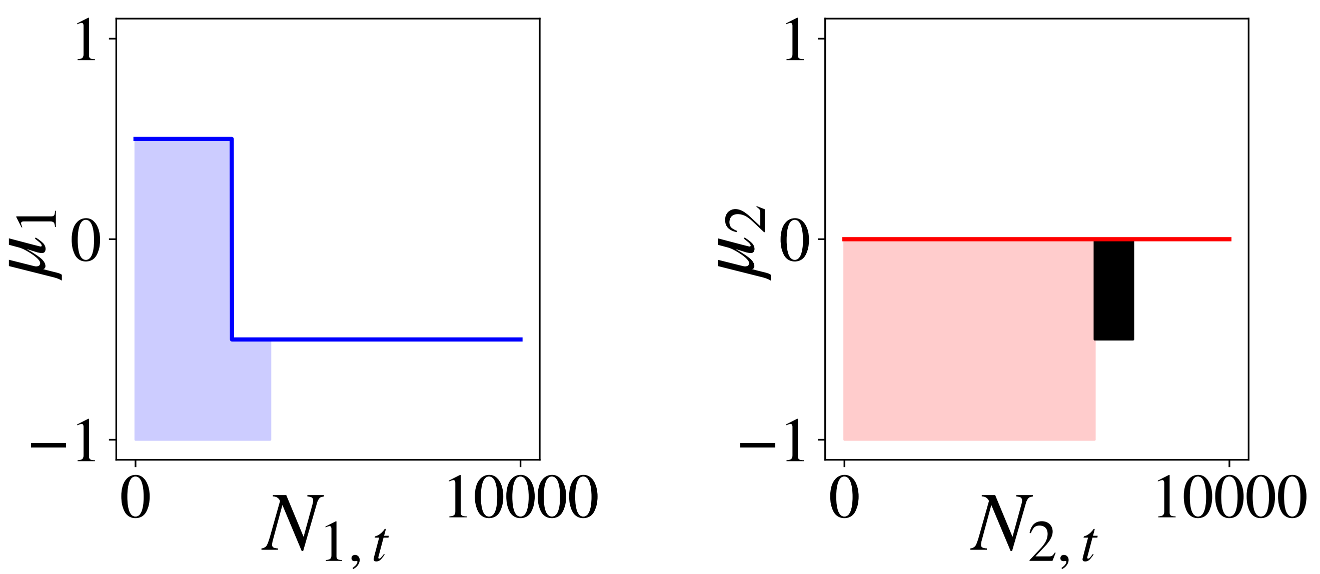
```
for t ← 1, 2, ... do
    SELECT : arg max_{i∈K} μ_i(N_{i,t})
end for
```



OPTIMAL ORACLE POLICY [HEIDARI, 2016]

Algorithm 1 $\mathcal{A}_0$ (Heidari et al., 2016)

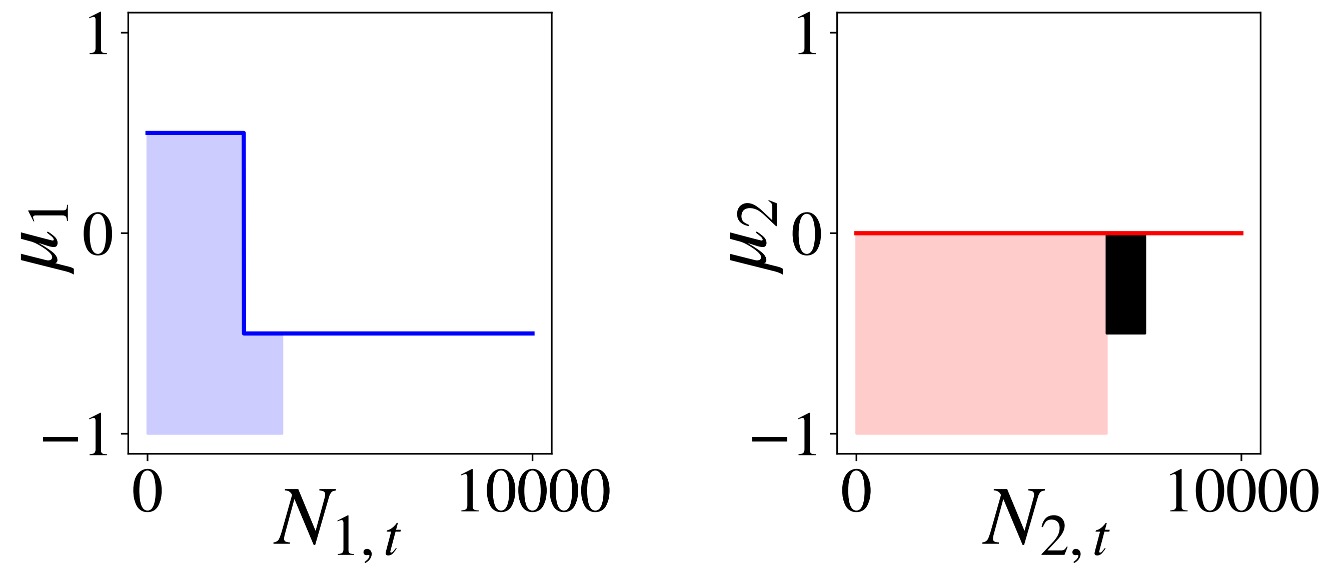
1: **for** $t \leftarrow 1, 2, \ldots$ **do**
2: SELECT : $\arg\max_{i \in \mathcal{K}} \mu_i(N_{i,t})$
3: **end for**



OPTIMAL ORACLE POLICY [HEIDARI, 2016]

Algorithm 1 $\mathcal{A}_0$ (Heidari et al., 2016)

1: **for** $t \leftarrow 1, 2, \ldots$ **do**
2: SELECT : $\arg\max_{i \in \mathcal{K}} \mu_i(N_{i,t})$
3: **end for**

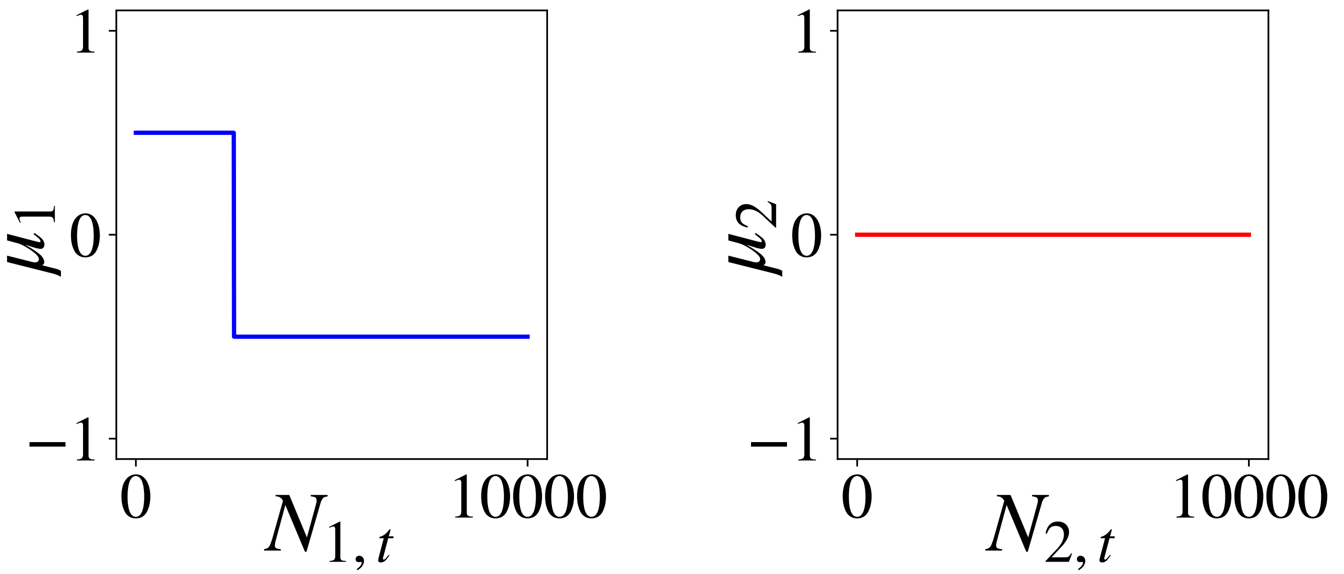


$$R_T(\pi) = \sum_{i \in \mathrm{UP}} \sum_{s=N^{\pi}_{i,T}+1}^{N^{\star}_{i,T}} \mu_i(s) - \sum_{i \in \mathrm{OP}} \sum_{s=N^{\star}_{i,T}+1}^{N^{\pi}_{i,T}} \mu_i(s)$$

NOISE-FREE BANDIT POLICY [HEIDARI, 2016]

Algorithm 2 $\mathcal{A}_2$ (Heidari et al., 2016)

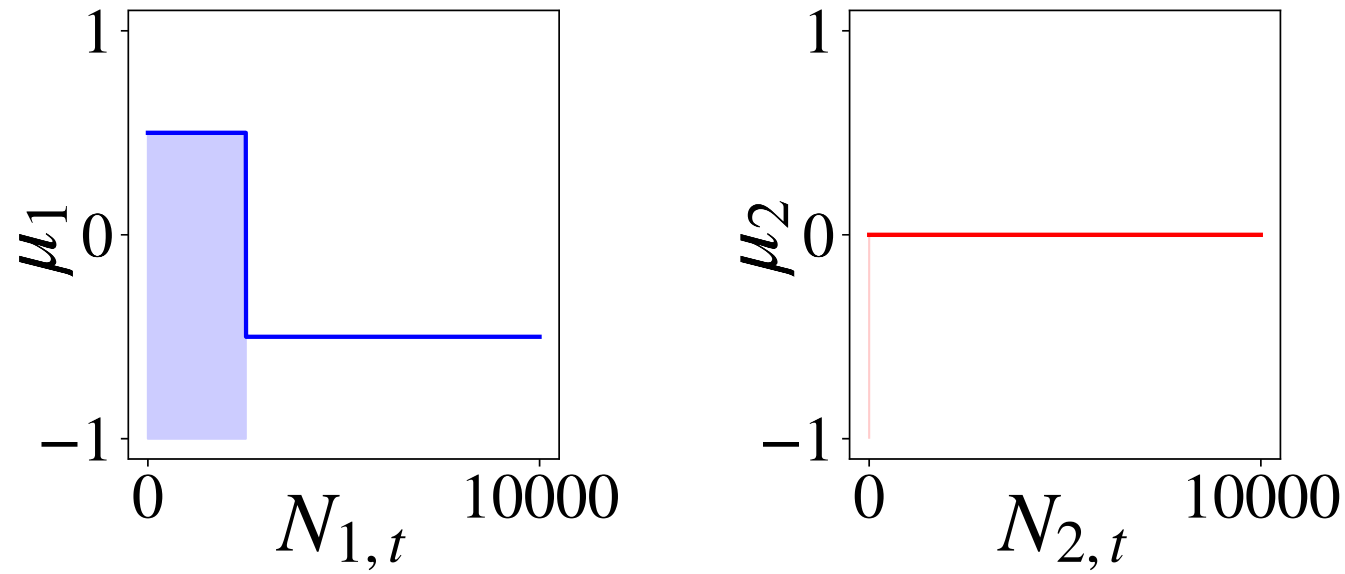
1: **for** $t \leftarrow K+1, K+2, \ldots$ **do**
2: $\quad$ SELECT : $\arg\max_{i \in \mathcal{K}} \mu_i(N_{i,t} - 1)$
3: **end for**



NOISE-FREE BANDIT POLICY [HEIDARI, 2016]

Algorithm 2 $\mathcal{A}_2$ (Heidari et al., 2016)

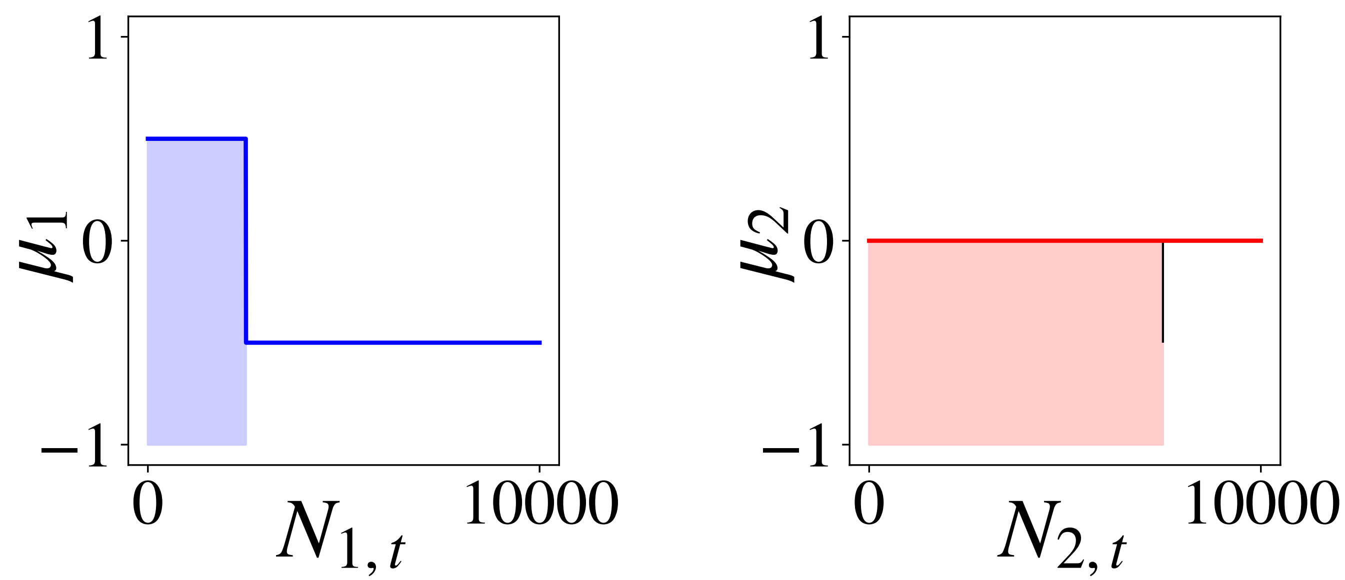
1: **for** $t \leftarrow K+1, K+2, \ldots$ **do**
2: SELECT : $\arg\max_{i \in \mathcal{K}} \mu_i(N_{i,t} - 1)$
3: **end for**



NOISE-FREE BANDIT POLICY [HEIDARI, 2016]

Algorithm 2 $\mathcal{A}_2$ (Heidari et al., 2016)

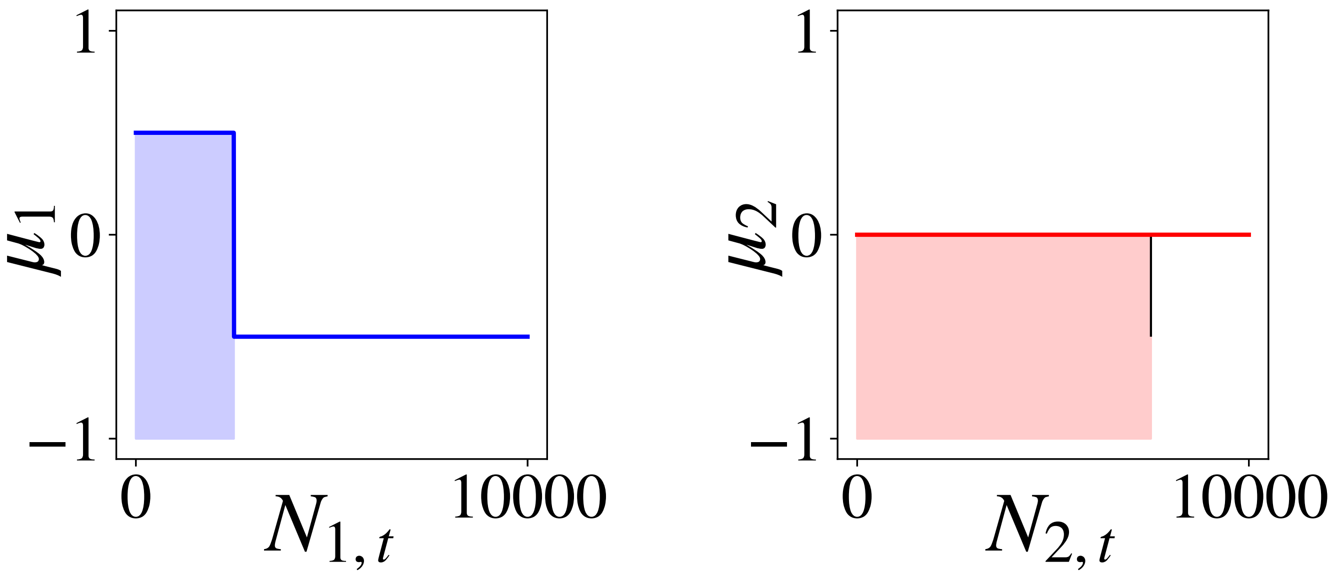
1: **for** $t \leftarrow K+1, K+2, \ldots$ **do**
2: SELECT : $\arg\max_{i \in \mathcal{K}} \mu_i(N_{i,t} - 1)$
3: **end for**



NOISE-FREE BANDIT POLICY [HEIDARI, 2016]

Algorithm 2 $\mathcal{A}_2$ (Heidari et al., 2016)

1: **for** $t \leftarrow K+1, K+2, \ldots$ **do**
2: SELECT : $\arg\max_{i \in \mathcal{K}} \mu_i(N_{i,t} - 1)$
3: **end for**



Worst-case minimax optimal rate : $R_T(\pi_{A_2}) \leq KL$

wSWA [LEVINE, 2017]

Algorithm 3 SWA (Levine et al., 2017)

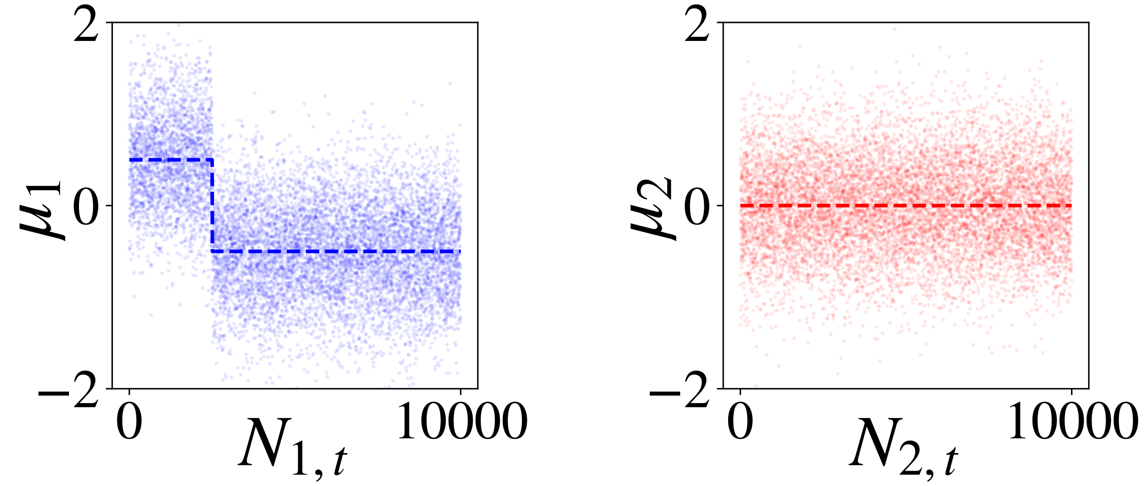
Input: K, L, T, σ

$h \leftarrow \widetilde{\mathcal{O}}\left(\left(\frac{\sigma T}{KL}\right)^{2/3}\right)$

for $t \leftarrow Kh+1, Kh+2, \ldots$ **do**

  SELECT : $\arg\max_{i \in \mathcal{K}} \widehat{\mu}_i^h(N_{i,t})$

end for

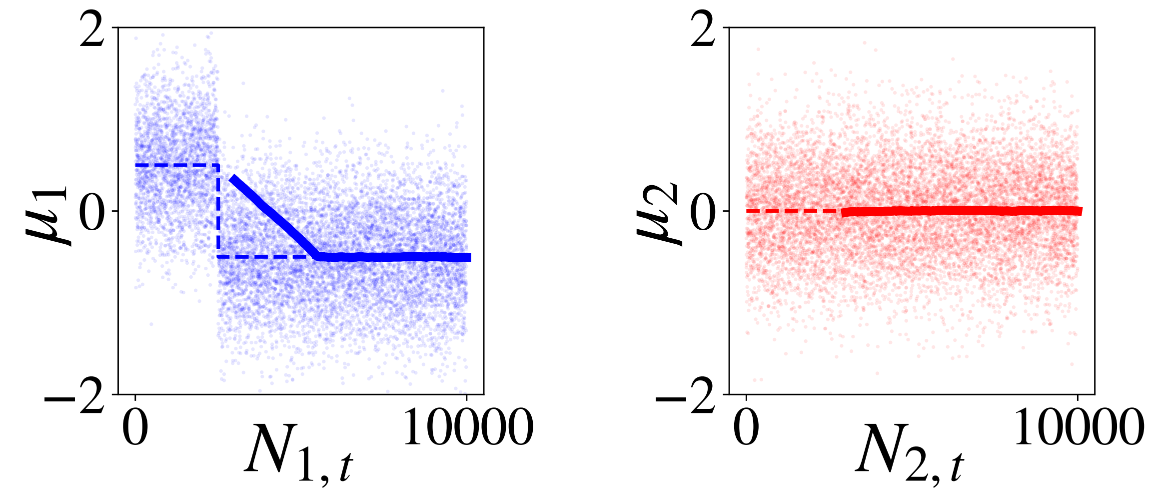


wSWA [LEVINE, 2017]

Algorithm 3 SWA (Levine et al., 2017)

Input: K, L, T, σ

$h \leftarrow \widetilde{\mathcal{O}}\left(\left(\frac{\sigma T}{KL}\right)^{2/3}\right)$
for $t \leftarrow Kh+1, Kh+2, \ldots$ **do**
  SELECT : $\arg\max_{i \in \mathcal{K}} \widehat{\mu}_i^h(N_{i,t})$
end for



h = 3000

Regret due to bias: $\widetilde{O}\left(LKh\right)$

wSWA [LEVINE, 2017]

Algorithm 3 SWA (Levine et al., 2017)

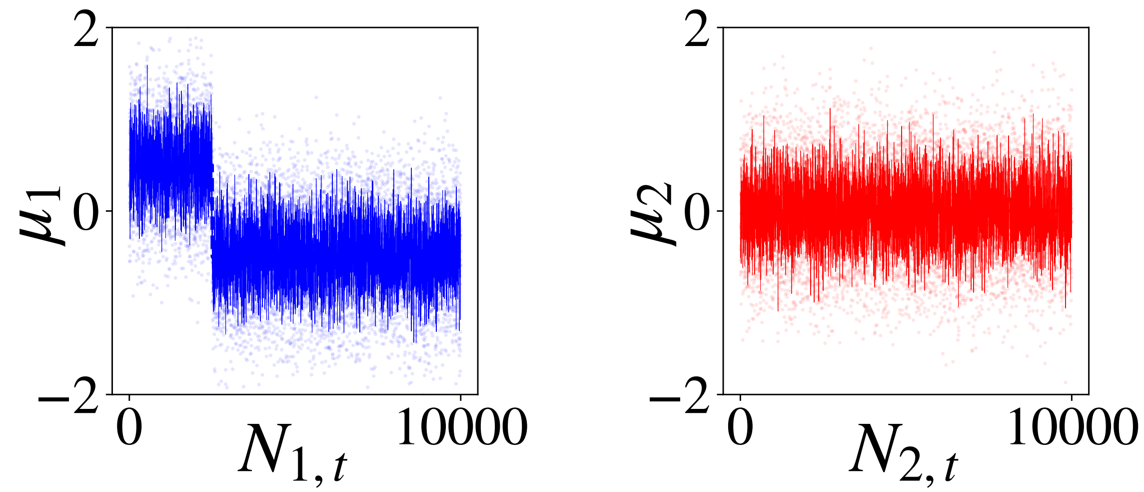
Input: K, L, T, σ

1: $h \leftarrow \widetilde{\mathcal{O}}\left(\left(\frac{\sigma T}{KL}\right)^{2/3}\right)$

2: **for** $t \leftarrow Kh+1, Kh+2, \ldots$ **do**

3: SELECT : $\arg\max_{i \in \mathcal{K}} \widehat{\mu}_i^h(N_{i,t})$

4: **end for**



h = 3

Regret due to bias: $\widetilde{O}\left(LKh\right)$

Regret due to variance : $\widetilde{O}\left(\sigma T\sqrt{h}^{-1}\right)$

wSWA [LEVINE, 2017]

Algorithm 3 SWA (Levine et al., 2017)

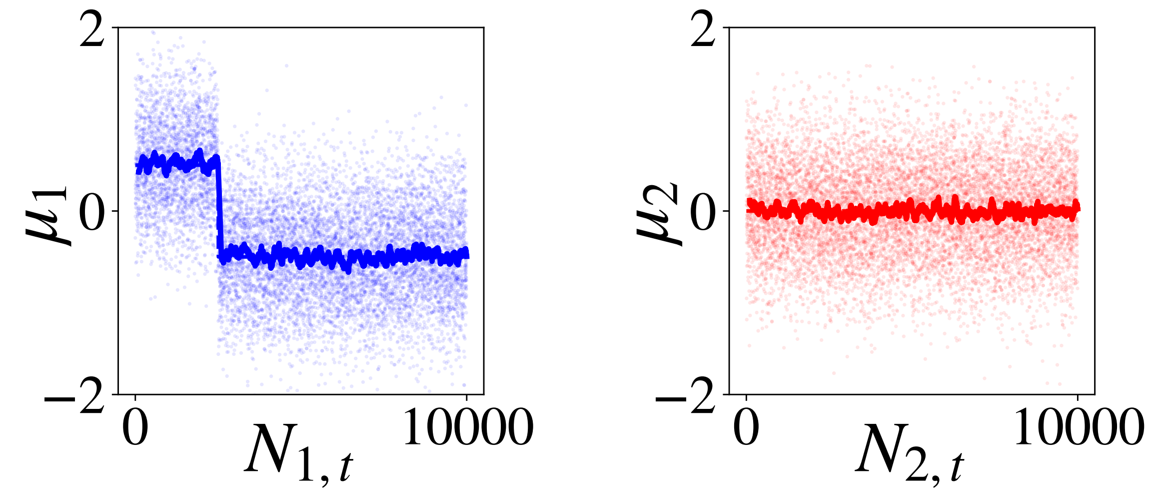
Input: K, L, T, σ

$h \leftarrow \widetilde{\mathcal{O}}((\frac{\sigma T}{KL})^{2/3})$

for $t \leftarrow Kh+1, Kh+2, \dots$ **do**

  SELECT : $\arg\max_{i \in \mathcal{K}} \widehat{\mu}_i^h(N_{i,t})$

end for



h = 100

Regret due to bias: $\widetilde{O}\left(LKh\right)$

Regret due to variance : $\widetilde{O}\left(\sigma T \sqrt{h}^{-1}\right)$

Worst case regret : $\widetilde{O}\left(K^{1/3}T^{2/3}\right)$

THE FAILURE OF wSWA

Sample	old																			last
Arm 1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
Arm 2	1	1	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	1	1	1
Arm 3	**X**	**X**	1	1	1	1	1	1	1	1	0	1	1	0	1	1	0	1	1	0

THE FAILURE OF wSWA

Sample	old																			last
Arm 1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
Arm 2	1	1	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	1	1	1
Arm 3	**X**	**X**	1	1	1	1	1	1	1	1	0	1	1	0	1	1	0	1	1	0

Won't we benefit from a data-adaptive window ?

FILTERING ON EXPANDING WINDOW AVERAGE (FEWA)

Algorithm 4 FEWA

Input: K, σ, α

for $t \leftarrow K+1, K+2, \dots$ **do**
  $\delta_t \leftarrow \frac{1}{Kt^\alpha}$
  $h \leftarrow 1$
  $\mathcal{K}_1 \leftarrow \mathcal{K}$
  do
    $\mathcal{K}_{h+1} \leftarrow \left\{ i \in \mathcal{K}_h \,|\, \widehat{\mu}_i^h(N_{i,t}) \geq \max_{j \in \mathcal{K}} \widehat{\mu}_j^h(N_{j,t}) - 2c(h, \delta_t) \right\}$
    $h \leftarrow h+1$
  while $h \leq \min_{i \in \mathcal{K}_h} N_{i,t}$
  SELECT : $\{ i \in \mathcal{K}_h \,|\, h > N_{i,t} \}$
end for

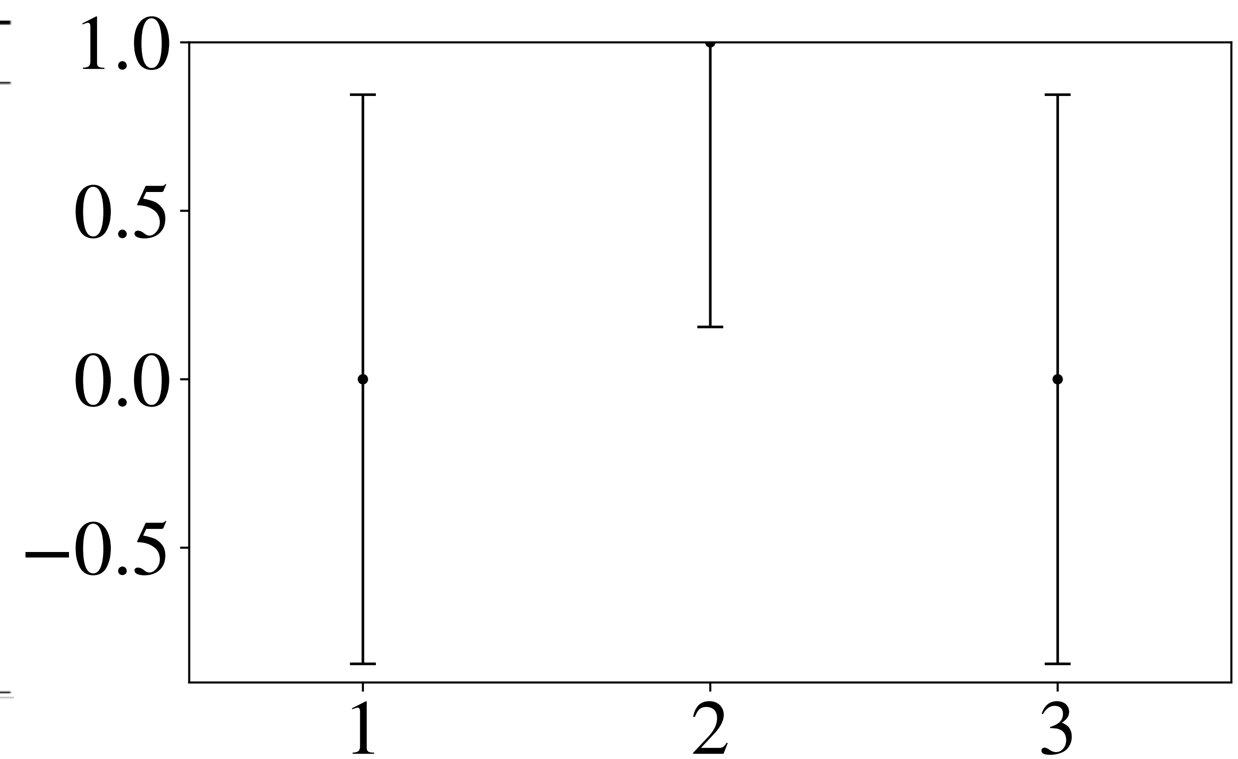
Sample	old																			last
Arm 1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
Arm 2	1	1	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	1	1	1
Arm 3	X	X	1	1	1	1	1	1	1	1	0	1	1	0	1	1	0	1	1	0

FILTERING ON EXPANDING WINDOW AVERAGE (FEWA)

Algorithm 4 FEWA

Input: K, σ, α

for $t \leftarrow K+1, K+2, \dots$ **do**
 $\delta_t \leftarrow \frac{1}{Kt^\alpha}$
 $h \leftarrow 1$
 $\mathcal{K}_1 \leftarrow \mathcal{K}$
 do
  $\mathcal{K}_{h+1} \leftarrow \{i \in \mathcal{K}_h \,|\, \widehat{\mu}_i^h(N_{i,t}) \geq \max_{j\in\mathcal{K}} \widehat{\mu}_j^h(N_{j,t}) - 2c(h,\delta_t)\}$
  $h \leftarrow h+1$
 while $h \leq \min_{i\in\mathcal{K}_h} N_{i,t}$
 SELECT : $\{i \in \mathcal{K}_h \,|\, h > N_{i,t}\}$
end for



Sample	old																			last
Arm 1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
Arm 2	1	1	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	1	1	1
Arm 3	**X**	**X**	1	1	1	1	1	1	1	1	0	1	1	0	1	1	0	1	1	0

h = 1

FILTERING ON EXPANDING WINDOW AVERAGE (FEWA)

Algorithm 4 FEWA

Input: K, σ, α

for $t \leftarrow K+1, K+2, \dots$ **do**

  $\delta_t \leftarrow \frac{1}{Kt^\alpha}$

  $h \leftarrow 1$

  $\mathcal{K}_1 \leftarrow \mathcal{K}$

  do

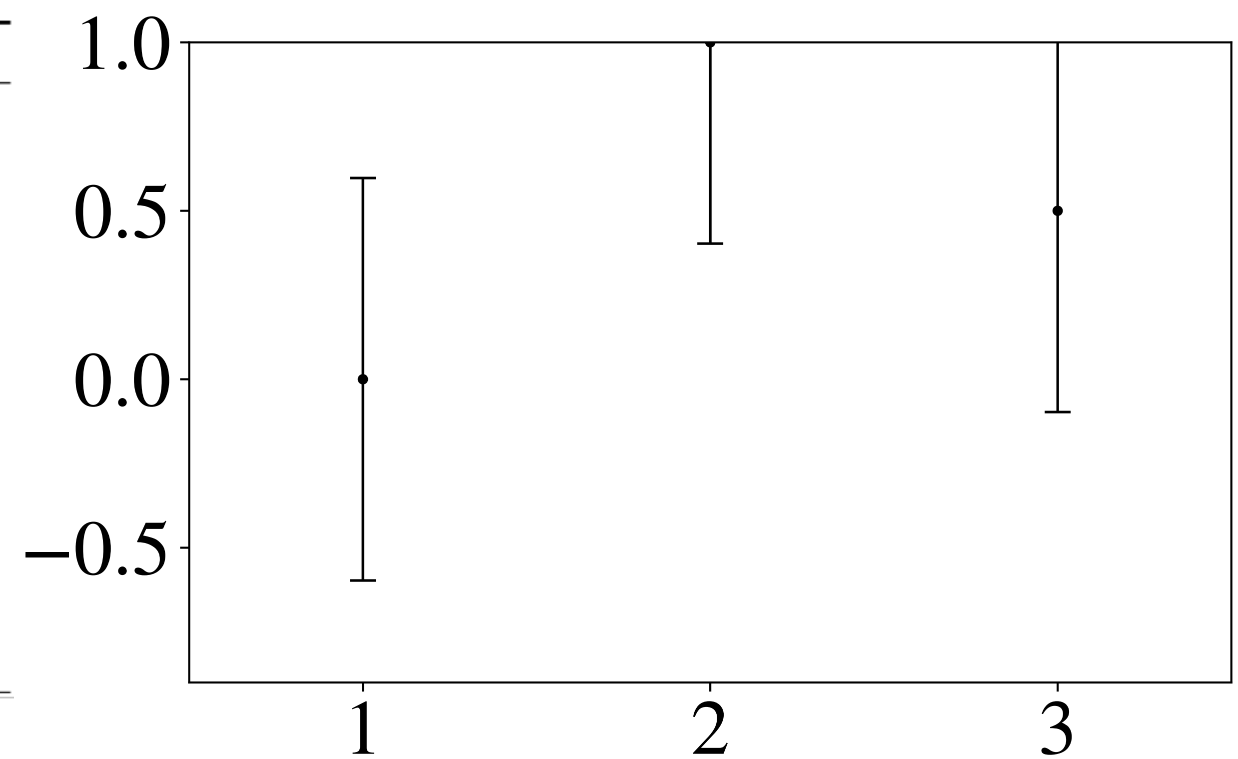
    $\mathcal{K}_{h+1} \leftarrow \{i \in \mathcal{K}_h \,|\, \widehat{\mu}_i^h(N_{i,t}) \geq \max_{j \in \mathcal{K}} \widehat{\mu}_j^h(N_{j,t}) - 2c(h, \delta_t)\}$

    $h \leftarrow h+1$

  while $h \leq \min_{i \in \mathcal{K}_h} N_{i,t}$

  SELECT : $\{i \in \mathcal{K}_h \,|\, h > N_{i,t}\}$

end for



Sample	old																		last
Arm 1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
Arm 2	1	1	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	1	1	1
Arm 3	**X**	**X**	1	1	1	1	1	1	1	1	0	1	1	0	1	1	0	1	1	0

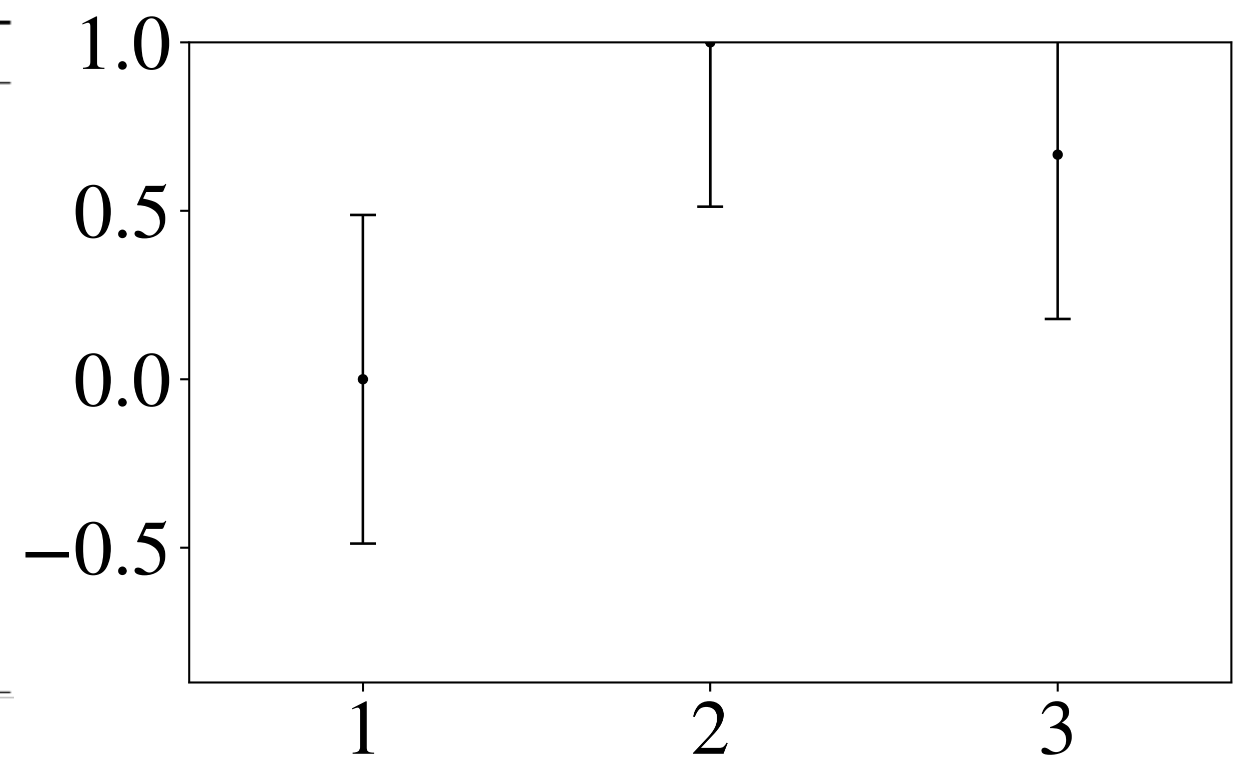
h = 2

FILTERING ON EXPANDING WINDOW AVERAGE (FEWA)

Algorithm 4 FEWA

Input: K, σ, α

for $t \leftarrow K+1, K+2, \dots$ **do**
  $\delta_t \leftarrow \frac{1}{Kt^\alpha}$
  $h \leftarrow 1$
  $\mathcal{K}_1 \leftarrow \mathcal{K}$
  do
    $\mathcal{K}_{h+1} \leftarrow \{i \in \mathcal{K}_h \,|\, \widehat{\mu}_i^h(N_{i,t}) \geq \max_{j \in \mathcal{K}} \widehat{\mu}_j^h(N_{j,t}) - 2c(h, \delta_t)\}$
    $h \leftarrow h+1$
  while $h \leq \min_{i \in \mathcal{K}_h} N_{i,t}$
  SELECT : $\{i \in \mathcal{K}_h \,|\, h > N_{i,t}\}$
end for



Sample	old																			last
Arm 1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
Arm 2	1	1	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	1	1	1
Arm 3	**X**	**X**	1	1	1	1	1	1	1	1	0	1	1	0	1	1	0	1	1	0

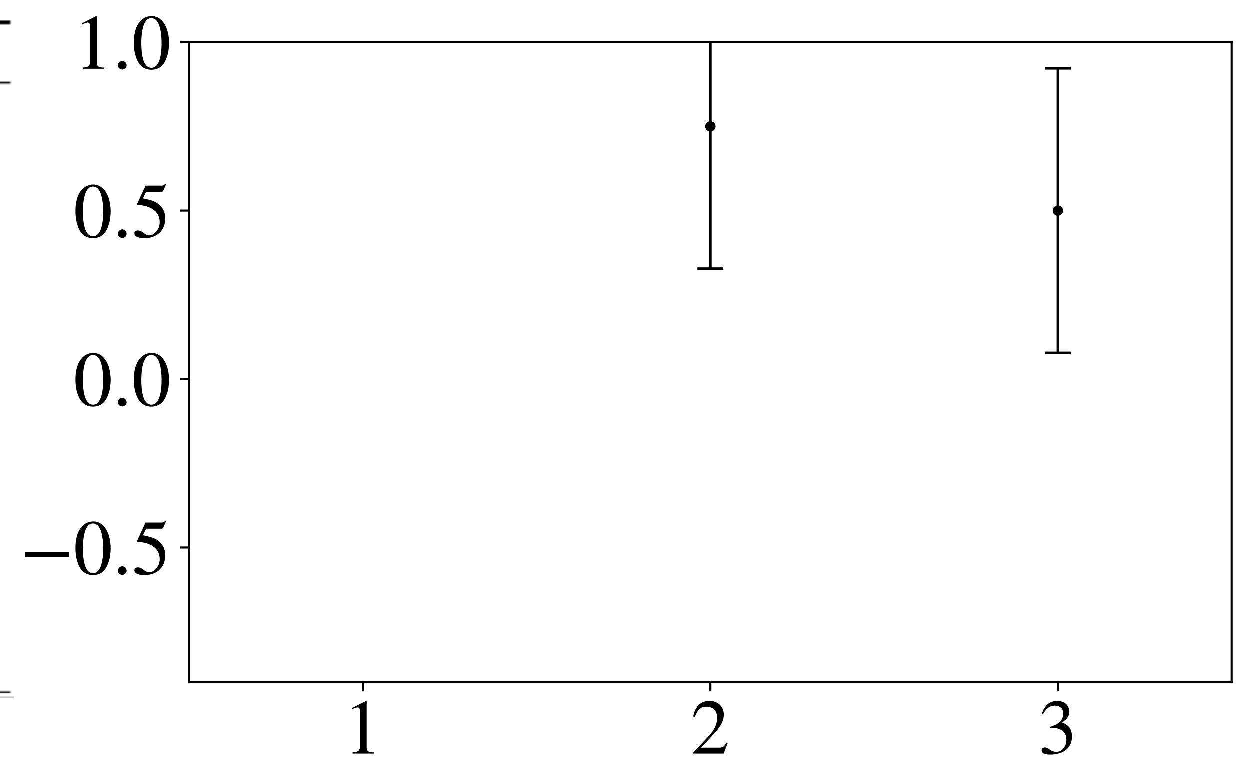
h = 3

FILTERING ON EXPANDING WINDOW AVERAGE (FEWA)

Algorithm 4 FEWA

Input: K, σ, α

for $t \leftarrow K+1, K+2, \dots$ **do**
    $\delta_t \leftarrow \frac{1}{Kt^\alpha}$
    $h \leftarrow 1$
    $\mathcal{K}_1 \leftarrow \mathcal{K}$
    do
        $\mathcal{K}_{h+1} \leftarrow \left\{ i \in \mathcal{K}_h \,|\, \widehat{\mu}_i^h(N_{i,t}) \geq \max_{j\in\mathcal{K}} \widehat{\mu}_j^h(N_{j,t}) - 2c(h,\delta_t) \right\}$
        $h \leftarrow h+1$
    while $h \leq \min_{i\in\mathcal{K}_h} N_{i,t}$
    SELECT : $\{ i \in \mathcal{K}_h \,|\, h > N_{i,t} \}$
end for



Sample	old																			last
Arm 1																				
Arm 2	1	1	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	1	1	1
Arm 3	**X**	**X**	1	1	1	1	1	1	1	1	0	1	1	0	1	1	0	1	1	0

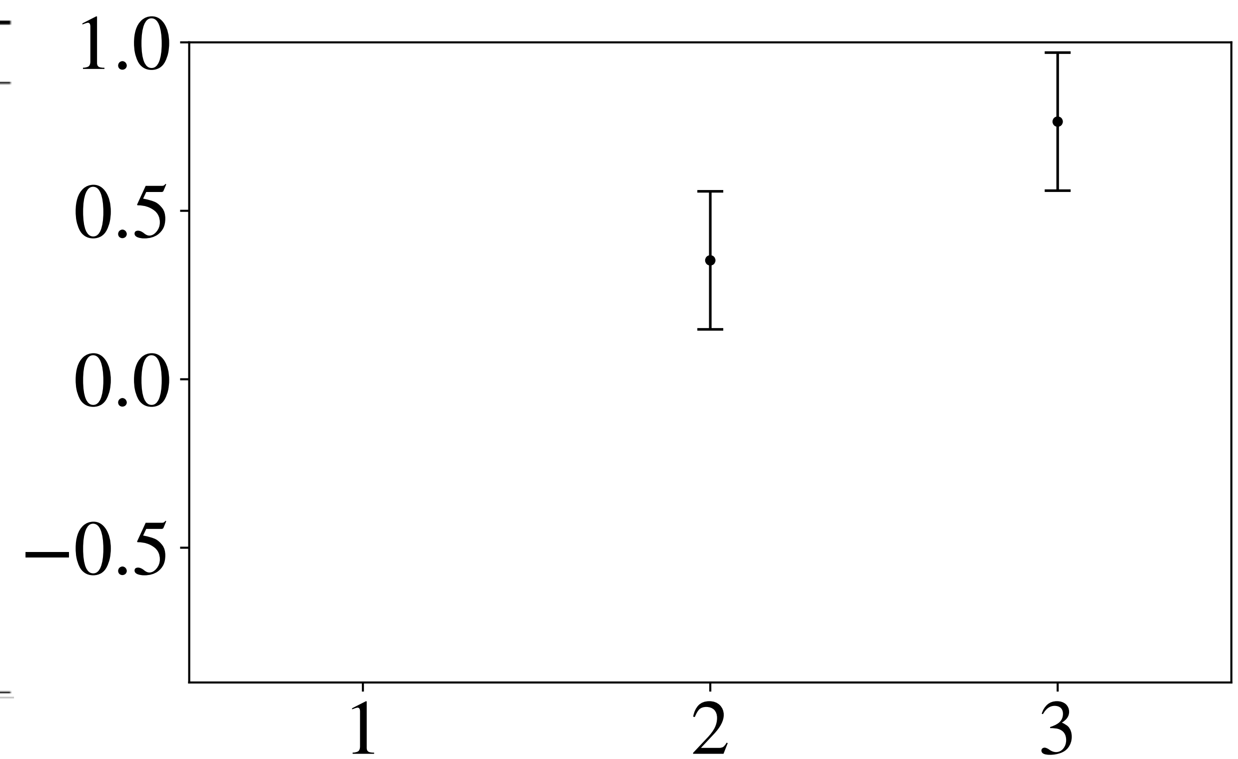
h = 4

FILTERING ON EXPANDING WINDOW AVERAGE (FEWA)

Algorithm 4 FEWA

Input: K, σ, α

for $t \leftarrow K+1, K+2, \dots$ **do**
  $\delta_t \leftarrow \frac{1}{Kt^\alpha}$
  $h \leftarrow 1$
  $\mathcal{K}_1 \leftarrow \mathcal{K}$
  do
    $\mathcal{K}_{h+1} \leftarrow \{i \in \mathcal{K}_h \,|\, \widehat{\mu}_i^h(N_{i,t}) \geq \max_{j \in \mathcal{K}} \widehat{\mu}_j^h(N_{j,t}) - 2c(h, \delta_t)\}$
    $h \leftarrow h+1$
  while $h \leq \min_{i \in \mathcal{K}_h} N_{i,t}$
  SELECT : $\{i \in \mathcal{K}_h \,|\, h > N_{i,t}\}$
end for



Sample	old																			last
Arm 1																				
Arm 2	1	1	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	1	1	1
Arm 3	**X**	**X**	1	1	1	1	1	1	1	1	0	1	1	0	1	1	0	1	1	0

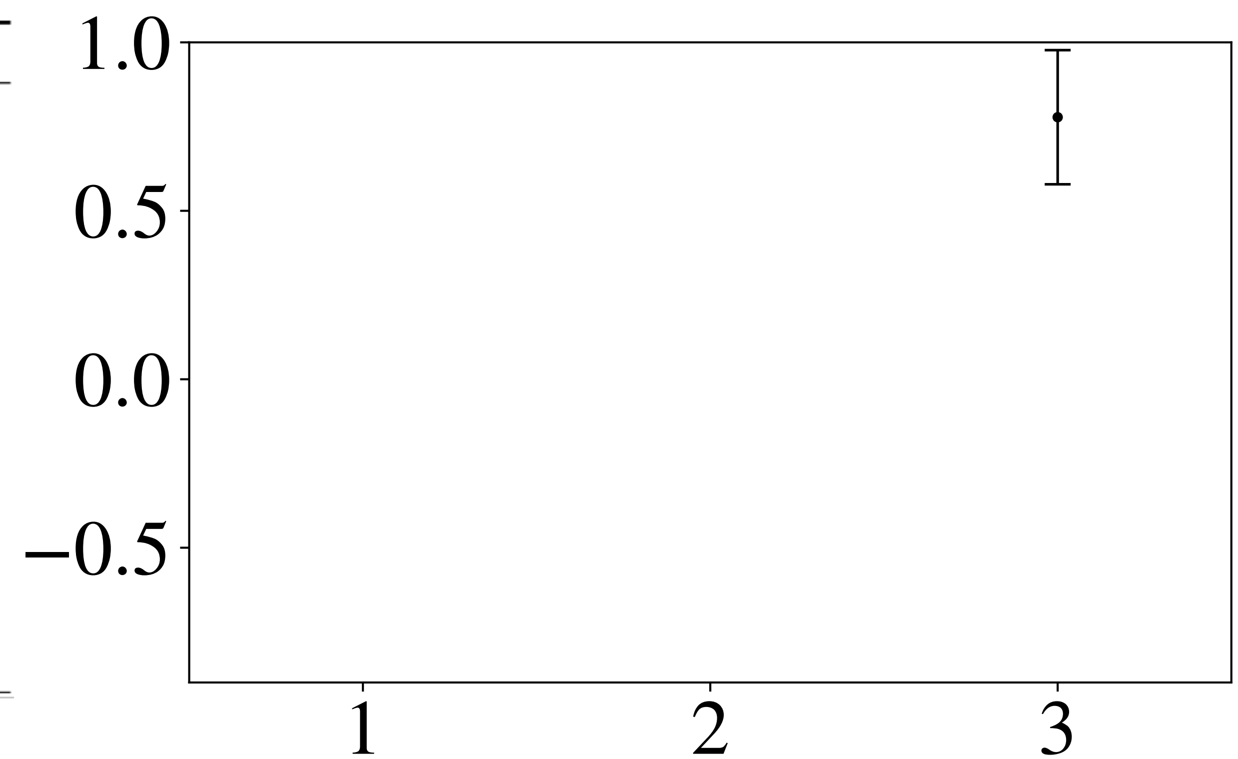
h = 17

FILTERING ON EXPANDING WINDOW AVERAGE (FEWA)

Algorithm 4 FEWA

Input: K, σ, α

for $t \leftarrow K+1, K+2, \dots$ **do**
  $\delta_t \leftarrow \frac{1}{Kt^\alpha}$
  $h \leftarrow 1$
  $\mathcal{K}_1 \leftarrow \mathcal{K}$
  do
    $\mathcal{K}_{h+1} \leftarrow \left\{ i \in \mathcal{K}_h \,|\, \widehat{\mu}_i^h(N_{i,t}) \geq \max_{j \in \mathcal{K}} \widehat{\mu}_j^h(N_{j,t}) - 2c(h, \delta_t) \right\}$
    $h \leftarrow h + 1$
  while $h \leq \min_{i \in \mathcal{K}_h} N_{i,t}$
  SELECT : $\{ i \in \mathcal{K}_h \,|\, h > N_{i,t} \}$
end for



Sample	old																			last
Arm 1																				
Arm 2																				
Arm 3	X	X	1	1	1	1	1	1	1	1	0	1	1	0	1	1	0	1	1	0

h = 18

UPPER BOUNDS

Worst-case upper bound

$$\mathbb{E}\left[R_T\left(\pi_{\mathrm{F}}\right)\right] \leq C\sigma\sqrt{KT\log(KT)} + KL$$

Comparison w/ wSWA

$$\mathbb{E}\left[R_T\left(\pi_{\mathrm{wSWA}}\right)\right] = \tilde{O}\left(L^{1/3}\sigma^{2/3}K^{1/3}T^{2/3}\right)$$

UPPER BOUNDS

Worst-case upper bound

$$\mathbb{E}\left[R_T\left(\pi_{\mathrm{F}}\right)\right] \leq C\sigma\sqrt{KT\log(KT)} + KL$$

Comparison w/ wSWA

$$\mathbb{E}\left[R_T\left(\pi_{\mathrm{wSWA}}\right)\right] = \tilde{O}\left(L^{1/3}\sigma^{2/3}K^{1/3}T^{2/3}\right)$$

Problem-dependent upper bound

$$\mathbb{E}\left[R_T\left(\pi_{\mathrm{F}}\right)\right] \leq \sum_{i\in\mathcal{K}} O\left(\frac{\log(KT)}{\Delta_{i,h^+_{i,T}-1}}\right)$$

Comparison w/ wSWA

Pure worst-case strategy

$\Delta_{i,h}$ Difference between the average of the h first overpulls of arm i and the worst reward pulled by the optimal policy

$h^+_{i,T}$ High-probability upper bound on the number of overpulls for FEWA

UPPER BOUNDS

Worst-case upper bound

$$\mathbb{E}\left[R_T\left(\pi_{\mathrm{F}}\right)\right] \leq C\sigma\sqrt{KT\log(KT)} + KL$$

Comparison w/ wSWA

$$\mathbb{E}\left[R_T\left(\pi_{\mathrm{wSWA}}\right)\right] = \tilde{O}\left(L^{1/3}\sigma^{2/3}K^{1/3}T^{2/3}\right)$$

Problem-dependent upper bound

$$\mathbb{E}\left[R_T\left(\pi_{\mathrm{F}}\right)\right] \leq \sum_{i\in\mathcal{K}} O\left(\frac{\log(KT)}{\Delta_{i,h^+_{i,T}-1}}\right)$$

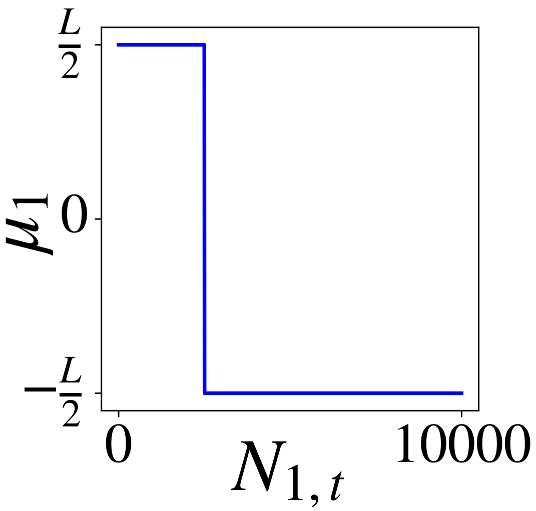
Comparison w/ wSWA

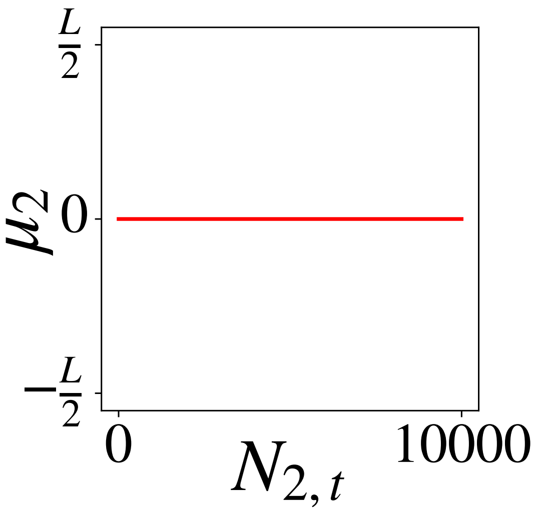
Pure worst-case strategy

$\Delta_{i,h} = \Delta_i$ on a stationner bandits problem

$\Delta_{i,h^+_{i,T}-1}$ is a problem-dependent quantity

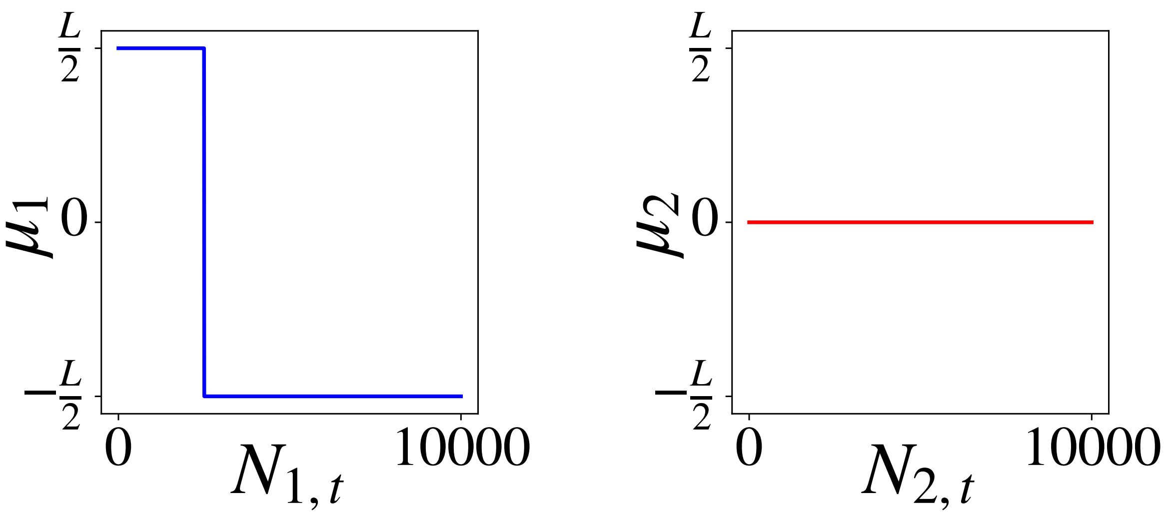
2-ARMS EXPERIMENTS



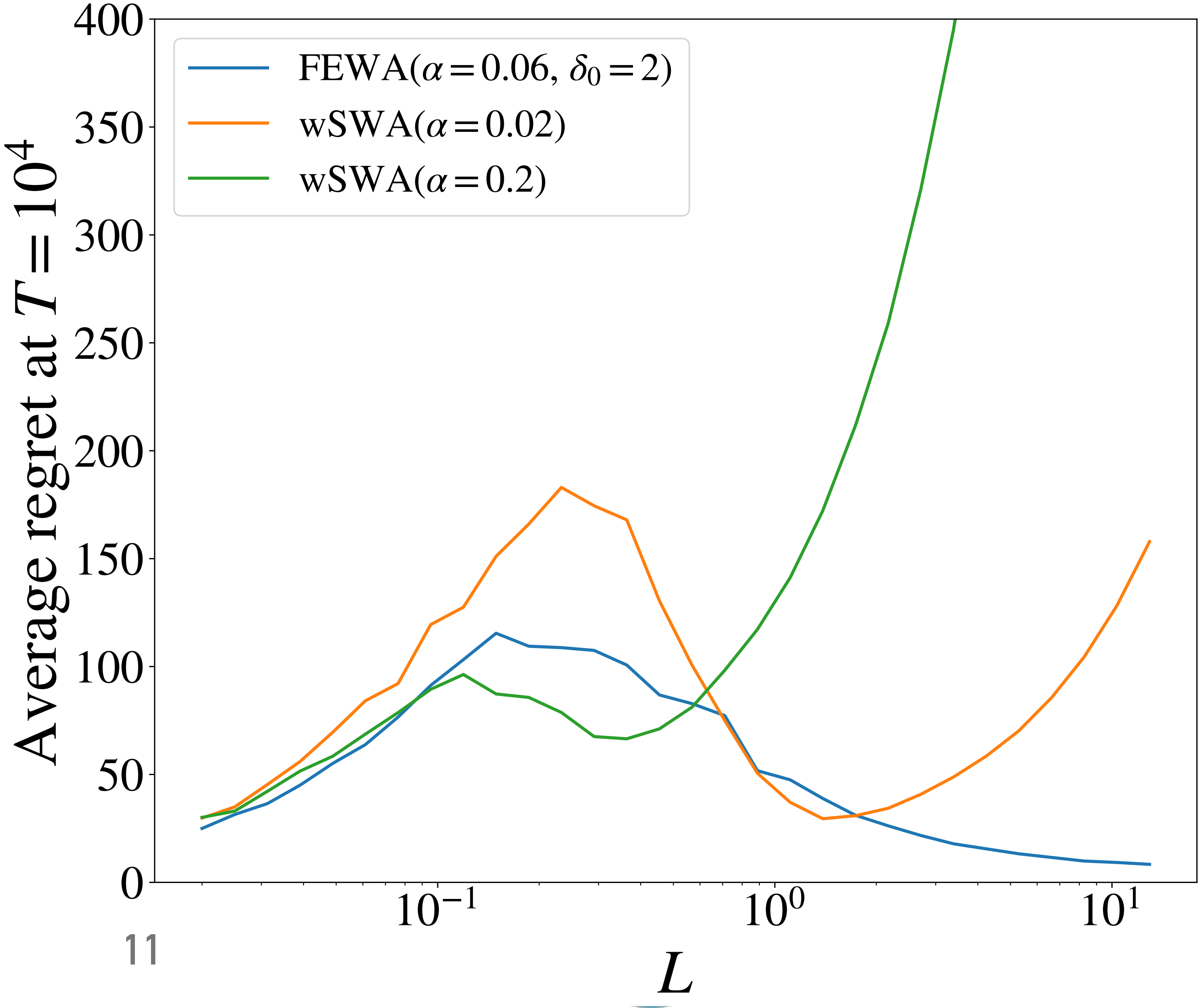


$$\sigma = 1 \; ; \; L$$

2-ARMS EXPERIMENTS



$$\sigma = 1 \; ; \; L$$



CONTRIBUTIONS

1 Rotting bandits are not harder than stochastic bandits

- ✅ $\tilde{\mathcal{O}}\left(\sqrt{KT}\right)$ worst-case bound
- ✅ $\tilde{\mathcal{O}}\left(\log\left(t\right)\right)$ problem-dependent bound

CONTRIBUTIONS

1 Rotting bandits are not harder than stochastic bandits

- ✅ $\tilde{\mathcal{O}}\left(\sqrt{KT}\right)$ worst-case bound
- ✅ $\tilde{\mathcal{O}}\left(\log(t)\right)$ problem-dependent bound

2 FEWA, a policy

- ✅ with a new data-adaptive window mechanism
- ✅ agnostic to L

CONTRIBUTIONS

1 Rotting bandits are not harder than stochastic bandits

- ✅ $\tilde{\mathcal{O}}\left(\sqrt{KT}\right)$ worst-case bound
- ✅ $\tilde{\mathcal{O}}\left(\log\left(t\right)\right)$ problem-dependent bound

2 FEWA, a policy

- ✅ with a new data-adaptive window mechanism
- ✅ agnostic to L

3 EFF-FEWA, a policy

- ✅ with FEWA's regret guarantees
- ✅ logarithmic space and time complexity