# TFG Presentation:

## *Toward reducing cumulative bias in automated decision-making systems using Multi-Armed Bandits*

**Quim De Las Heras Molins (u160402)**

quim.delasheras01@estudiant.upf.edu

Juliol 2022

Treball de Fi de Grau

Enginyeria Informàtica

Universitat Pompeu Fabra

# OVERVIEW

- Context and Motivation
- Problem Statement: bias and cumulative bias
- Solution Proposal: BC-UCB
- Results Evaluation
- Conclusions and Future Work

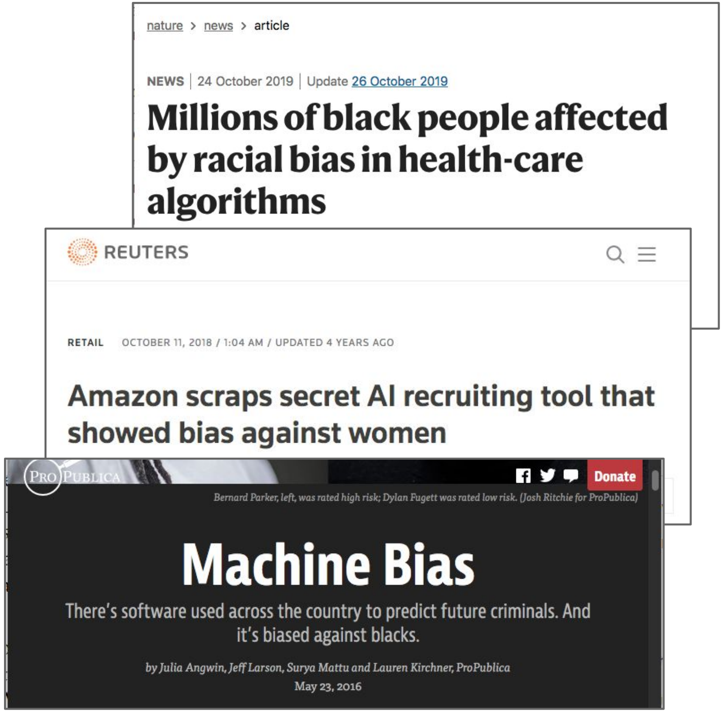
# Context and Motivation

[1] [2] [3]

- Automated Decision-Making Systems are on the rise:
  - health care
  - the labor market
  - justice
  - **mortgage lending**

- ↑ Their use becoming more extensive…

  ↑ Higher risk of harmful effects caused by bias.



# Context and Motivation

- Types of bias:
    - **Sampling Bias**
    - Negative Set Bias
    - Measurement Bias (Capture, Device or Proxy)
    - **Label Bias**
    - Confounding Bias (Omitted Variable or Proxy)

# Context and Motivation

- **Biased** AI models can have harmful **effects** on their **environment.**
- These effects can be **cumulative.**
- Discrimination against **sensitive** groups can be **perpetuated** or **escalated.**
- **Multi-Armed Bandit** problems can be used to model the relationship between an automated-decision making system and its environment.

# Context and Motivation

- For a limited amount of rounds T, each round $t \in [1,T]$ the learner will choose one of the $k$ available actions $a_t \in$ A and receive a reward $X_t \in \mathbb{R}$ sampled from distribution $P_{at}$ by the environment.

---

**Algorithm 1** - Upper Confidence Bound (UCB)

---

**Input:** $T$ and $l$

**Output:** $\sum_{t=1}^{T} X_t$

**while** $t \leq T$ **do**

Choose arm $a_t = argmax_{a \in A} \left[ \widehat{\mu_t}(a) + l \sqrt{\frac{\log(t)}{N_t(a)}} \right]$

Observe reward $X_t$ and update $\widehat{\mu_t}(a_t)$ and $N_t(a_t)$

**end while**

---

# Context and Motivation

- The aims of this project are:
  - **Modeling** of an automated decision-making system as a **Multiple Armed Bandit** problem where instances grouped based on **sensitive** variables present arbitrarily different reward distributions that end up being **unfairly optimized** and reveal bias.
  - Study of the **cumulative bias effect** that said model gives rise to when its decisions have an **effect on the environment**, which creates **feedback** loops that expand any existing discriminations **over time** in a manner consistent with real-life scenarios.
  - **Proposing solutions** to MAB problems where **sensitive** features exist within the data which consider not only the exploration-exploitation dilemma but also take into consideration **distances between distributions and arm commitment**.

# Problem Statement: bias and cumulative bias

- In our system:
  - Limited amount of resources to be distributed
  - Sensitive features within the data
- Analysis on groups based on intersectional sensitive options.
- Optimizing determinant features should not be on the detriment of sensitive ones.

# Problem Statement: bias and cumulative bias

- 717,997 samples from the database by *The Home Mortgage Disclosure Act* **(HMDA)**.
- US states of Alabama, Arkansas, Georgia, Mississippi, Louisiana and Tennessee, produced during 2020.
- 99 features; a binary target encodes whether the **loan originated or was declined**.

$$k = \prod_{i=1}^{s} o_i$$

- race = {**W**hite, **B**lack or African American}, ethnicity = {**N**ot **H**ispanic or Latino, **H**ispanic or Latino} and sex = {**M**ale, **F**emale}

# Problem Statement: bias and cumulative bias

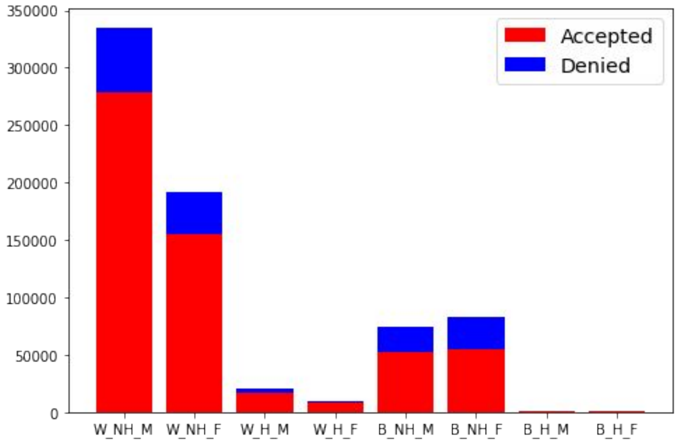



Figure 3.1: Inspection of instance distribution in HMDA database

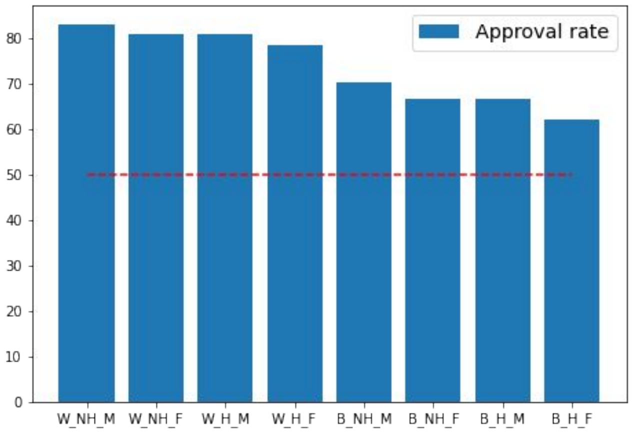



Figure 3.2: Inspection of approval rate spread in HMDA database

# Problem Statement: bias and cumulative bias

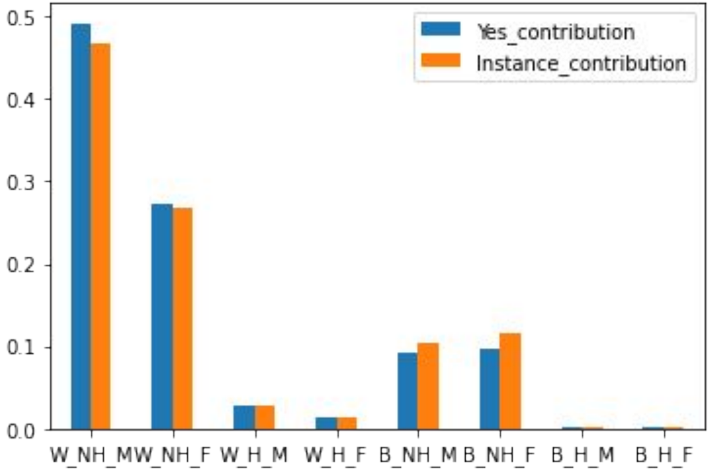



Figure 3.3: Contribution of each demographic to data and to total approvals

$p_a \in$ [0.83, 0.807, 0.808, 0.782, 0.702, 0.664, 0.666, 0.619]

$$P = (\,Bernoulli(\,p_a\,) : a \in A\,)$$

$w_a \in$ [0.467, 0.267, 0.028, 0.014, 0.103, 0.116, 0.002, 0.002]

$$N = (\,n_a \sim Binomial(\,|C|\,,\,w_a\,) : a \in A\,)$$

Stationary Environment

# Problem Statement: bias and cumulative bias

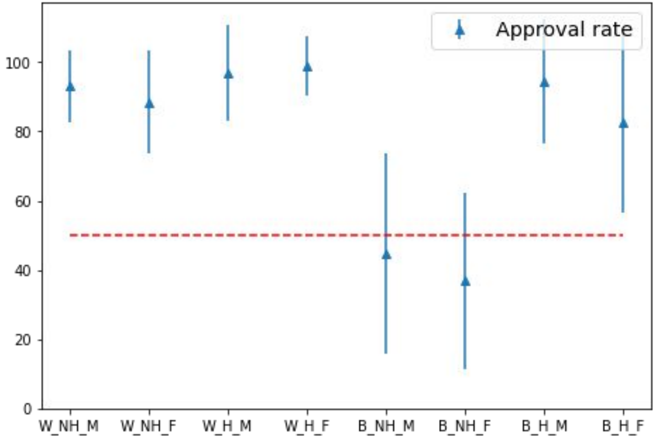



Figure 3.4: Approval rate granted by UCB per each demographic after 100 stationary executions on HMDA-based data

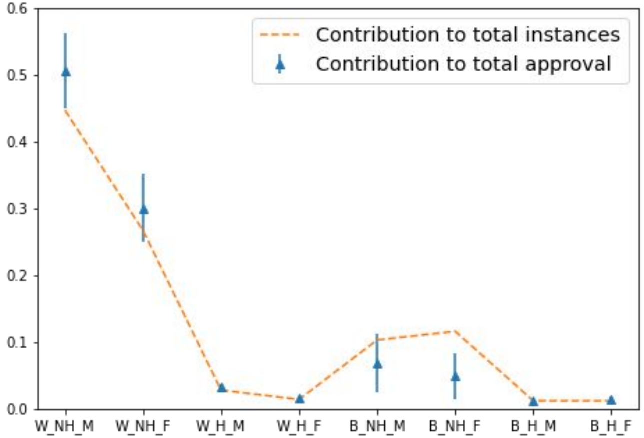



Figure 3.5: Contribution of each demographic to data and to total approvals after 100 stationary executions on HMDA-based data

# Problem Statement: bias and cumulative bias

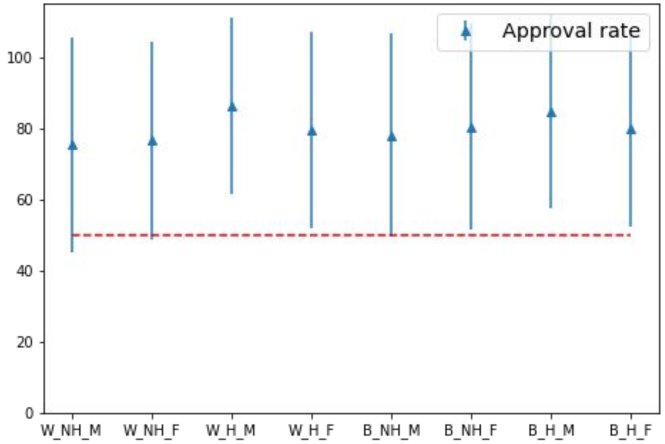



Figure 3.6: Approval rate granted by UCB per each demographic after 100 stationary executions on unbiased data

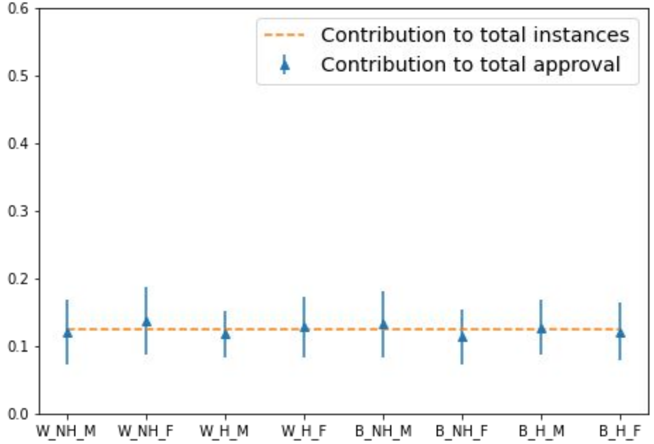



Figure 3.7: Contribution of each demographic to data and to total approvals after 100 stationary executions on unbiased data

# Non-Stationary Environment

# Problem Statement: bias and cumulative bias

$$p_a \in reward\_d_t \rightarrow P = (\,Bernoulli(p_a) : a \in A\,)$$

$$w_a \in population\_w_t \rightarrow N = (\,n_a \sim Binomial(\,|C|\,, w_a) : a \in A\,)$$

$\leftarrow$ $reward\_d_0$ , $population\_w_0$

x100

$C$ → Upper Confidence Bound (UCB) → $C'$

$|C| = 2500$

$l = 25$

$|C'| = 0.8 * |C|$

$$presence\_data = (\,|C_a| / |C| : a \in A\,)$$

$$presence\_approval = (\,|C_a \cup C'| / |C'| : a \in A)$$

$$increments = (1 + \frac{presence\_approval_a - presence\_data_a}{presence\_data_a} : a \in A)$$

$$reward\_d_{t+1} = (1 - r_{coef})reward\_d_t + r_{coef}(increments \cdot reward\_d_t)$$

$$population\_w_{t+1} = (1 - p_{coef})population\_w_t + p_{coef}(increments \cdot population\_w_t)$$

$r_{coef}$ , $p_{coef}$

(0.01,0.1)

# Problem Statement: bias and cumulative bias

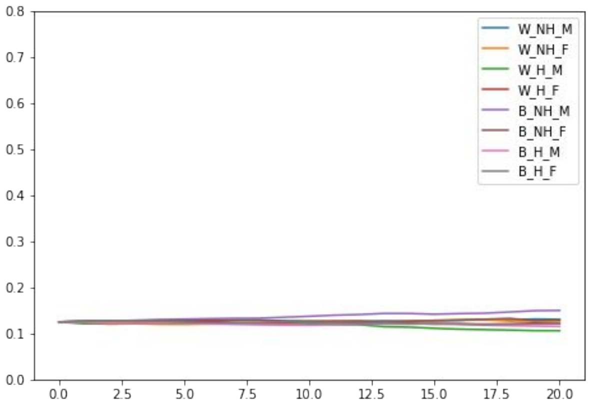



Figure 3.8: Evolution of population weights every 5 epochs in an initially unbiased environment

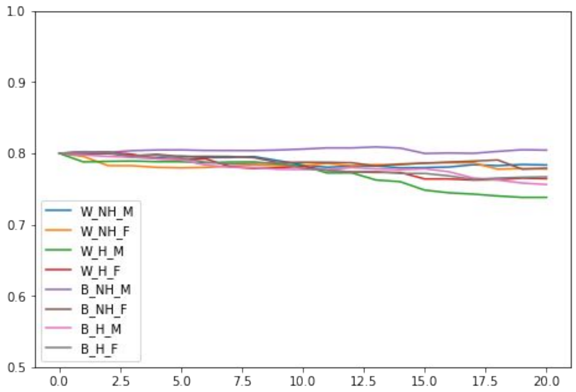



Figure 3.10: Evolution of reward estimates every 5 epochs in an initially unbiased environment

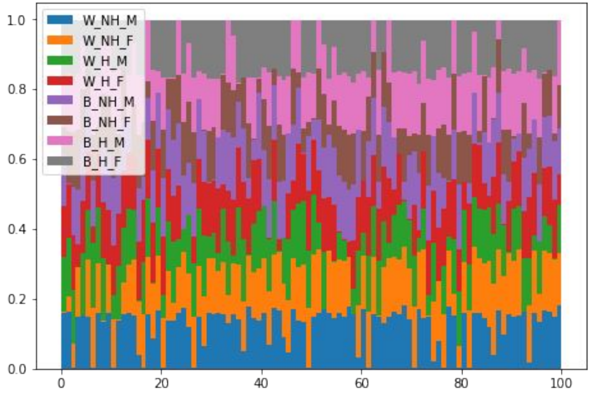



Figure 3.12: Evolution of contribution to total approvals for every epoch in an initially unbiased environment

# Problem Statement: bias and cumulative bias

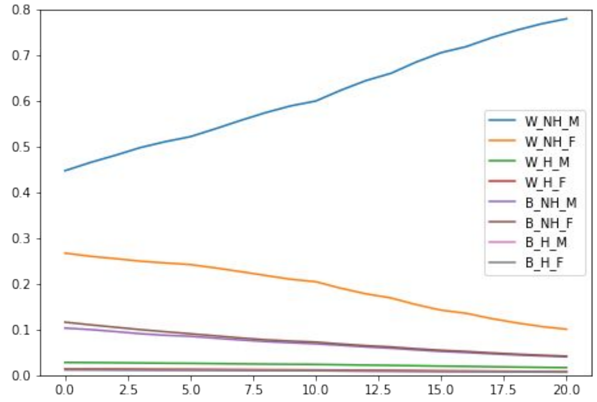



Figure 3.9: Evolution of population weights every 5 epochs in an HMDA-based environment

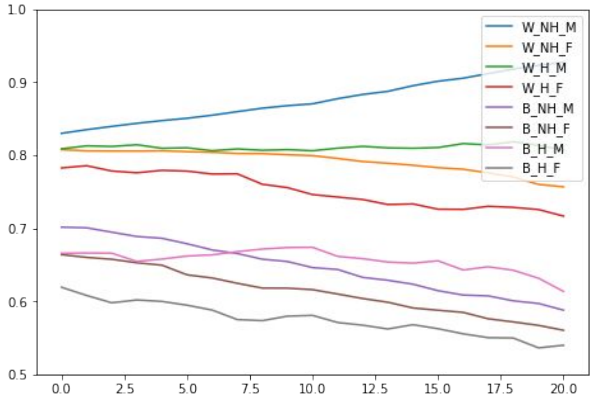



Figure 3.11: Evolution of reward estimates every 5 epochs in an HMDA-based environment

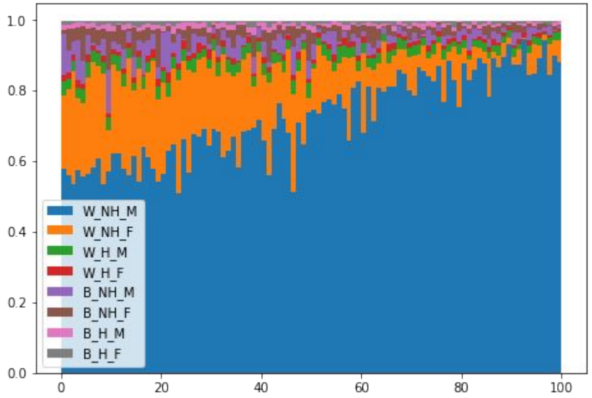



Figure 3.13: Evolution of contribution to total approvals for every epoch in an HMDA-based environment

# Solution Proposal: BC-UCB

- Formalization of the four bias metrics:

1. $$bias\_presence = \sum_{a=1}^{k} \left( \left| presence\_approval_a - presence\_data_a \right| \right)$$

2. $$bias\_rates = \sum_{a=1}^{k} \left( \left| \overline{approval\_rates} - approval\_rates_a \right| \right)$$

3. $$bias\_posteriors = \sum_{a=1}^{k} \left( \left| \overline{posteriors\_groups} - posteriors\_groups_a \right| \right)$$

4. $$bias\_rewards = \sum_{a=1}^{k} \left( \left| \overline{\widehat{\mu}}_t - \widehat{\mu}_t(a) \right| \right)$$

# Solution Proposal: BC-UCB

$$posteriors\_groups = (P(c \in C_a | c \in C') : a \in A)$$

$$posteriors\_groups = (\frac{P(c \in C' | c \in C_a) * P(c \in C_a)}{P(c \in C')} : a \in A)$$

$$posteriors\_groups = \frac{approval\_rates_a * presence\_data_a}{0.8} : a \in A)$$

# Solution Proposal: BC-UCB

- Bias Constrained UCB (BC-UCB):
  - *bias_metric* ∈ {*bias_presence, bias_rate, bias_posteriors, bias_rewards*}
  - $bias\_term_t = 100 * MinMax(bias\_metric(t + 1))$

| | Sampling step formula |
|---|---|
| UCB | $a_t = argmax_{a \in A}[\widehat{\mu}_t(a) + l\sqrt{\frac{log(t)}{N_t(a)}}]$ |
| BC-UCB | $a_t = argmax_a (1 - \theta) * [\widehat{\mu}_t(a) + l\sqrt{\frac{log(t)}{N_t(a)}}] - \theta * bias\_term_t(a)$ |

# Solution Proposal: BC-UCB

**Algorithm 2** - Bias Constrained UCB (BC-UCB)

**Input**: $T$ , $l$ , $\theta$, $bias\_metric$

**Output**: $\sum_{t=1}^{T} X_t, bias\_metric(T)$

**while** $t \leq T$ **do**

Compute $bias\_term_t = 100 * MinMax(bias\_metric(t+1))$

Compute $A' = (a \in A \text{ s.t. } N_t(a) < n_a)$

Choose arm $a_t = argmax_{a \in A'}((1-\theta) * \left[\widehat{\mu_t}(a) + l\sqrt{\frac{\log(t)}{N_t(a)}}\right] - \theta * bias\_term_t(a))$

Observe reward $X_t$ and update $\widehat{\mu_t}(a_t)$ and $N_t(a_t)$

**end while**

# Solution Proposal: BC-UCB

- Navigating the tradeoff between final total reward and bias:

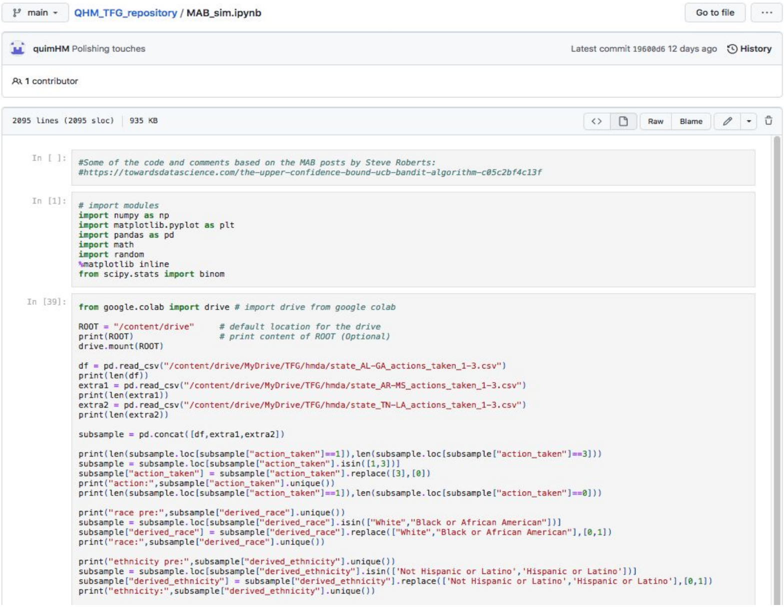
$$Frobenius(v) = v \,/ \sqrt{\left(\sum_{i=1} v_i^2\right)}$$

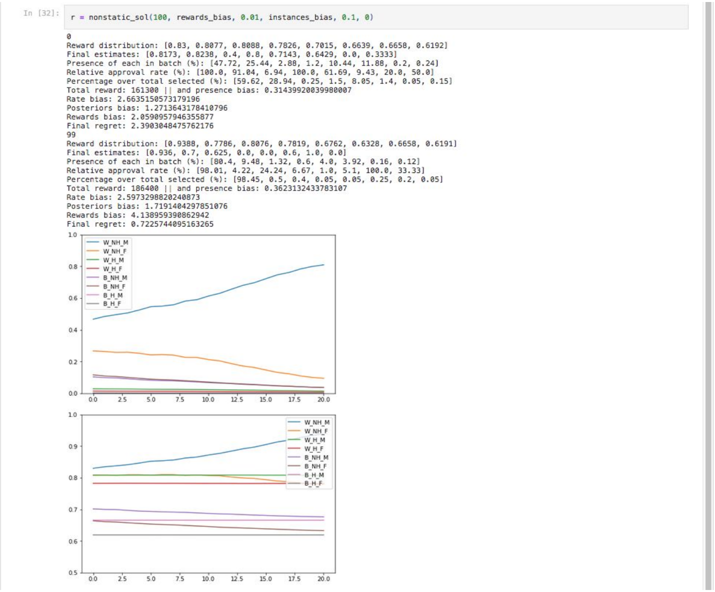
$$\varphi(frew, fbias) = Frobenius(frew) - \beta * Frobenius(fbias)$$

$$\theta^*_{bias\_metric} = argmax_{(\theta)} \left(\varphi(BC\text{-}UCB(\theta,\ bias\_metric)\right)$$

# Results Evaluation

https://github.com/quimHM/QHM_TFG_repository (Last commit: 22nd June)



Stationary Environment

# APPENDICES

## Appendix A: Application in a Stationary Environment

### a) Presence Bias

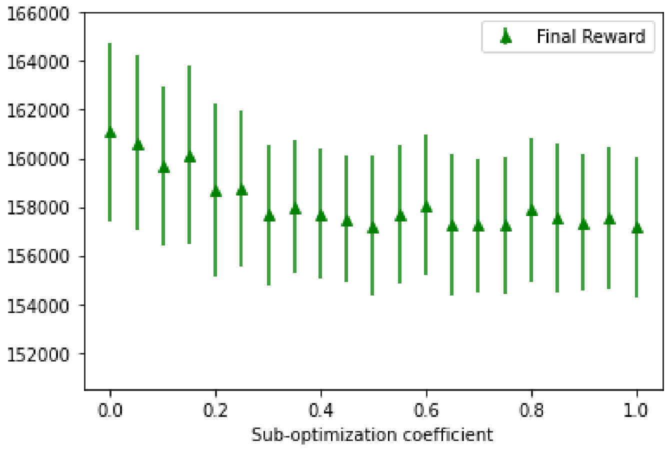



Figure A.1: Final Reward spread depending on θ when *bias_metric = bias_presence*

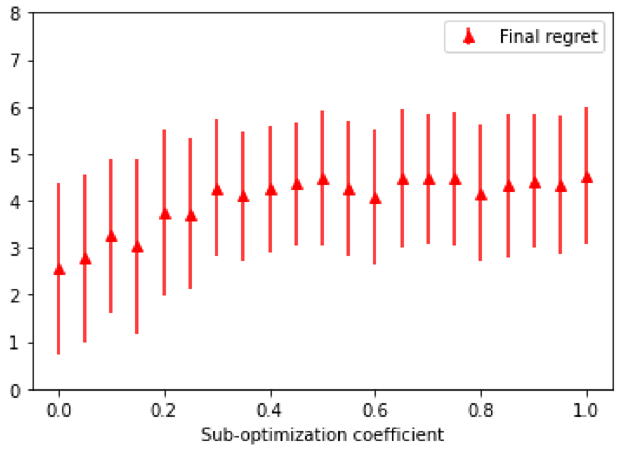



Figure A.2: Final Regret spread depending on θ when *bias_metric = bias_presence*

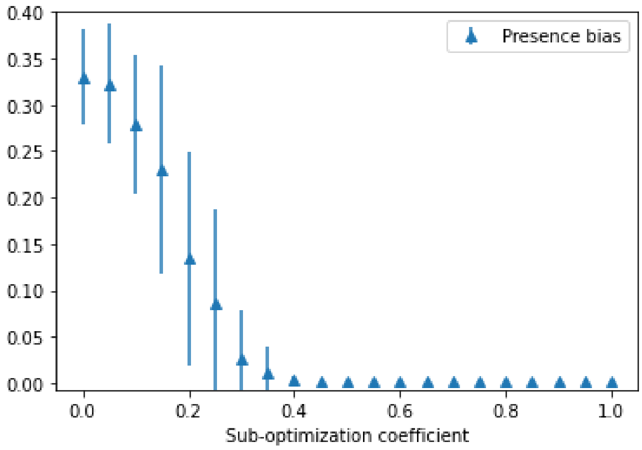



Figure A.3: Final Presence bias spread depending on θ when *bias_metric = bias_presence*

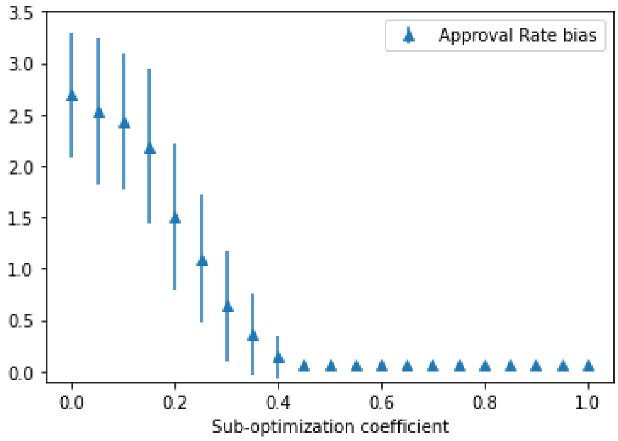



Figure A.4: Final Approval Rate bias spread depending on θ when *bias_metric = bias_presence*

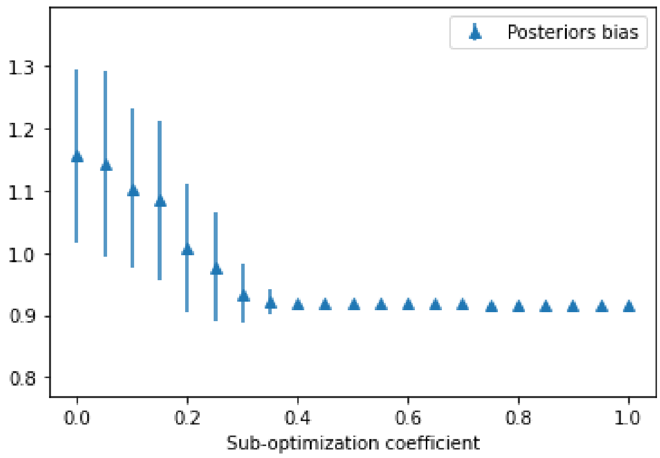



Figure A.5: Final Posteriors bias spread depending on θ when *bias_metric = bias_presence*

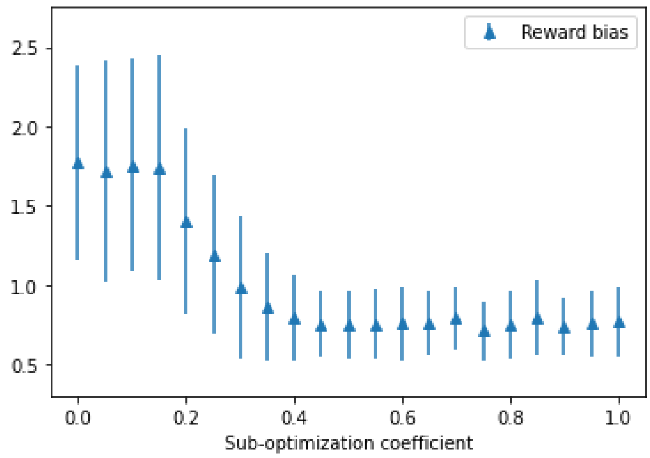



Figure A.6: Final Rewards bias spread depending on θ when *bias_metric = bias_presence*

## b) Approval Rate Bias

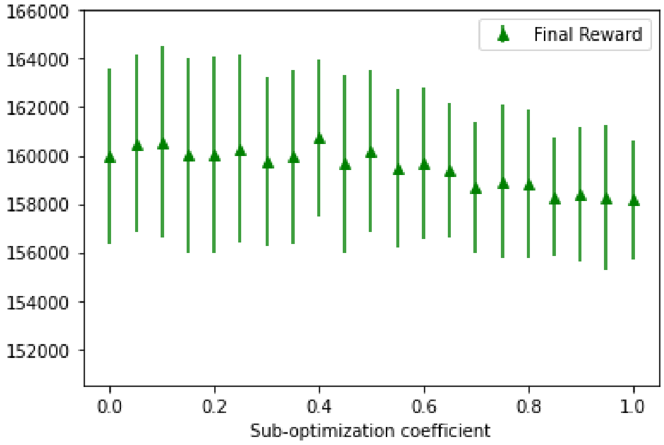



Figure A.7: Final Reward spread depending on θ when *bias_metric* = *bias_rates*

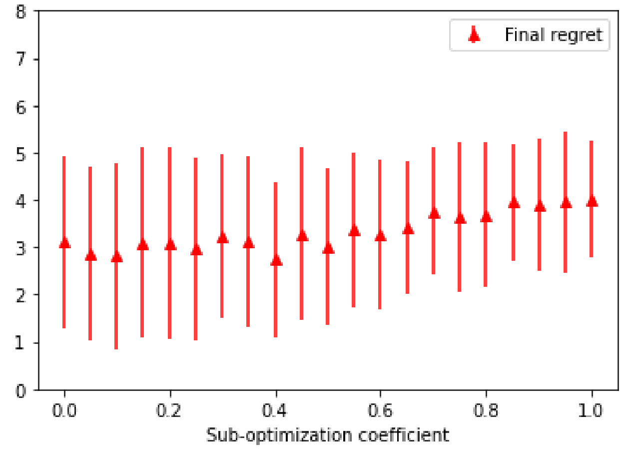



Figure A.8: Final Regret spread depending on θ when *bias_metric* = *bias_rates*

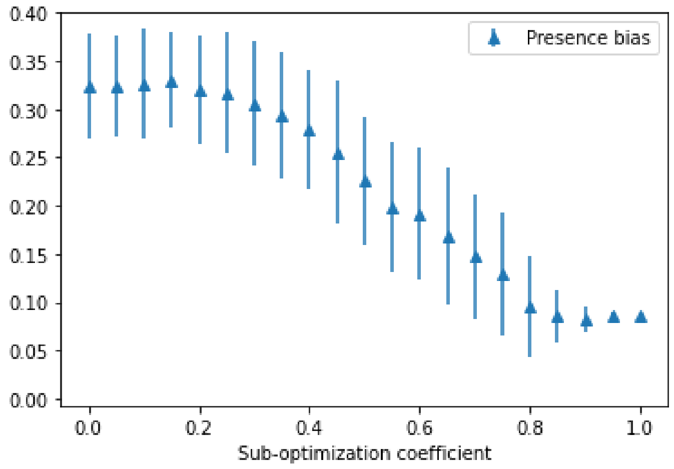



Figure A.9: Final Presence bias spread depending on θ when *bias_metric* = *bias_rates*

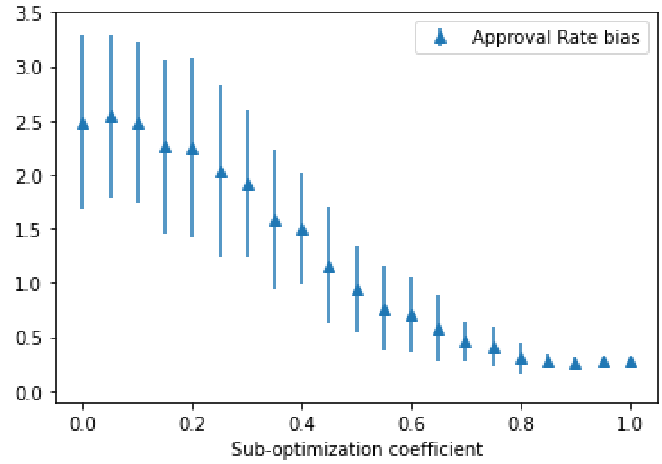



Figure A.10: Final Approval Rate bias spread depending on θ when *bias_metric* = *bias_rates*

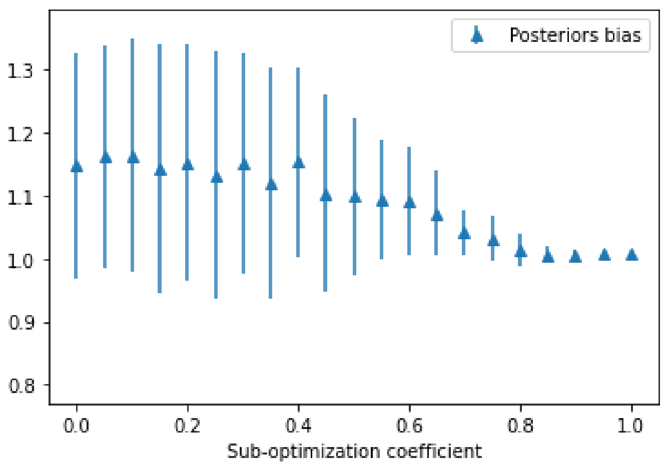



Figure A.11: Final Posteriors bias spread depending on θ when *bias_metric* = *bias_rates*

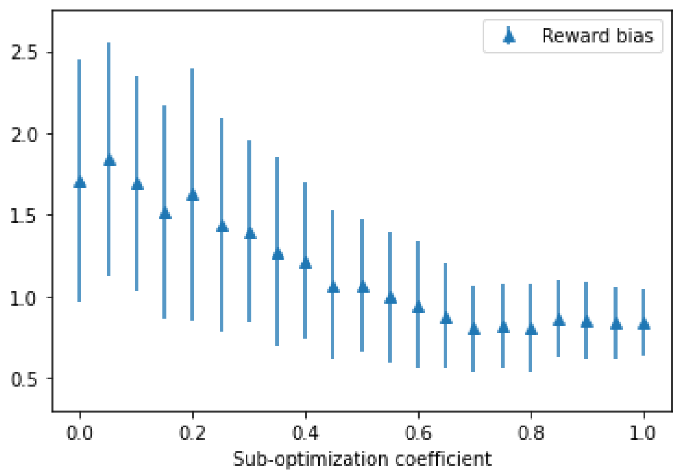



Figure A.12: Final Rewards bias spread depending on θ when *bias_metric* = *bias_rates*

## c) Posteriors Bias

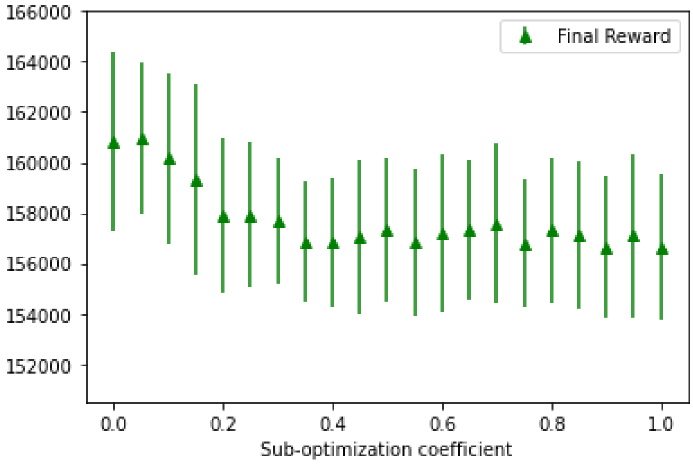



Figure A.13: Final Reward spread depending on θ when *bias_metric = bias_posteriors*

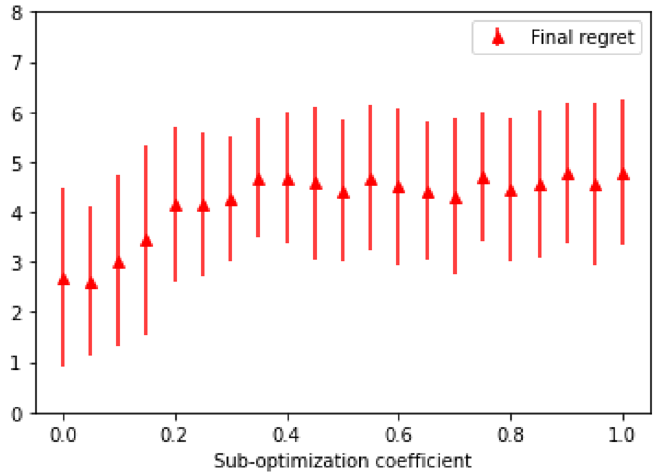



Figure A.14: Final Regret spread depending on θ when *bias_metric = bias_posteriors*

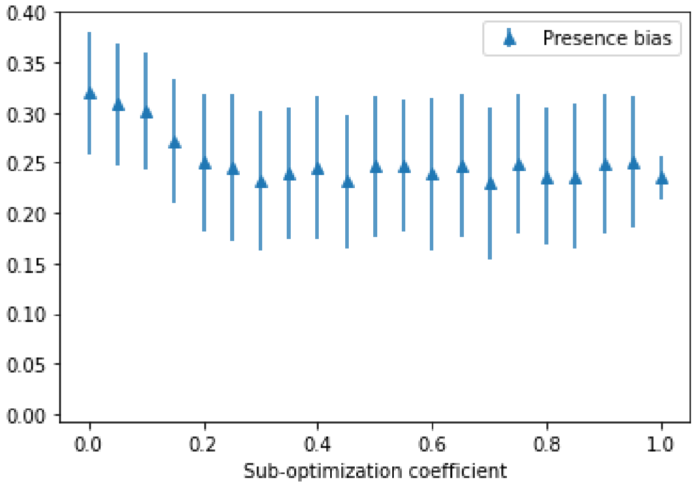



Figure A.15: Final Presence bias spread depending on θ when *bias_metric = bias_posteriors*

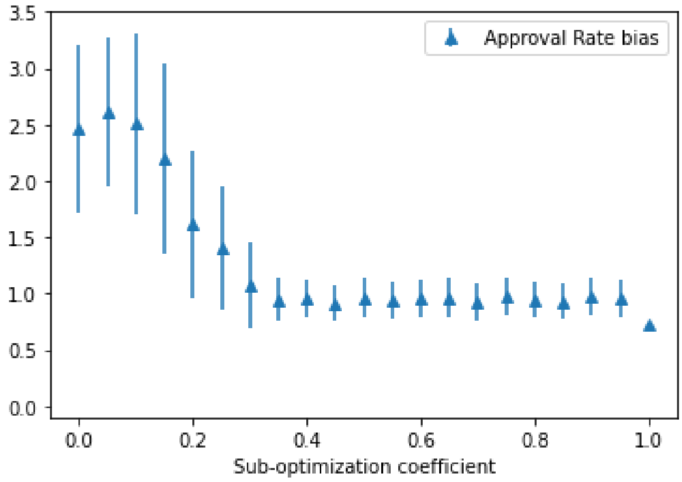



Figure A.16: Final Approval Rate bias spread depending on θ when *bias_metric = bias_posteriors*

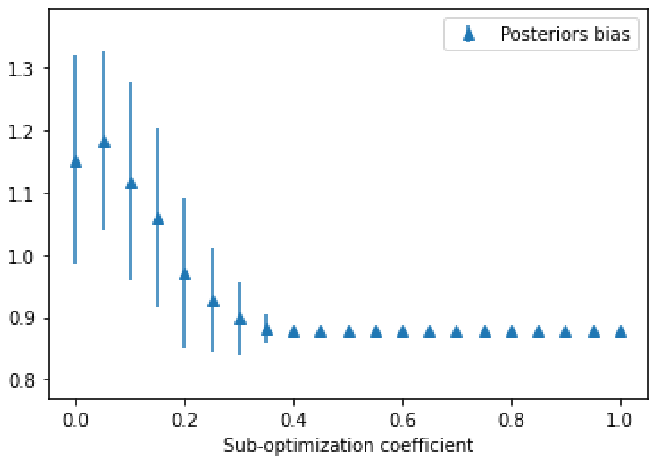



Figure A.17: Final Posteriors bias spread depending on θ when *bias_metric = bias_posteriors*

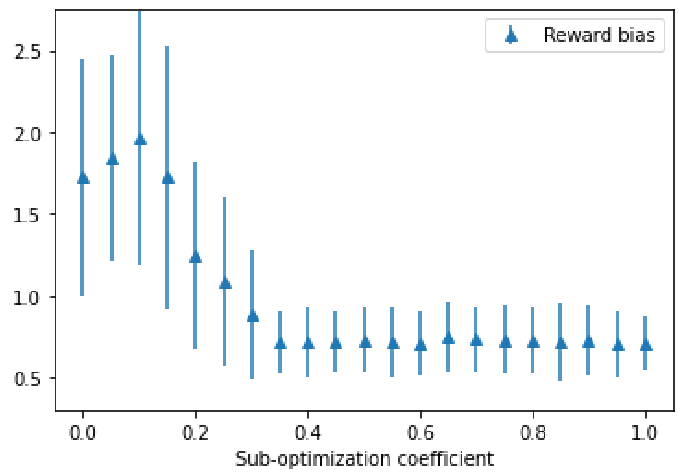



Figure A.18: Final Rewards bias spread depending on θ when *bias_metric = bias_posteriors*

d) Rewards Bias

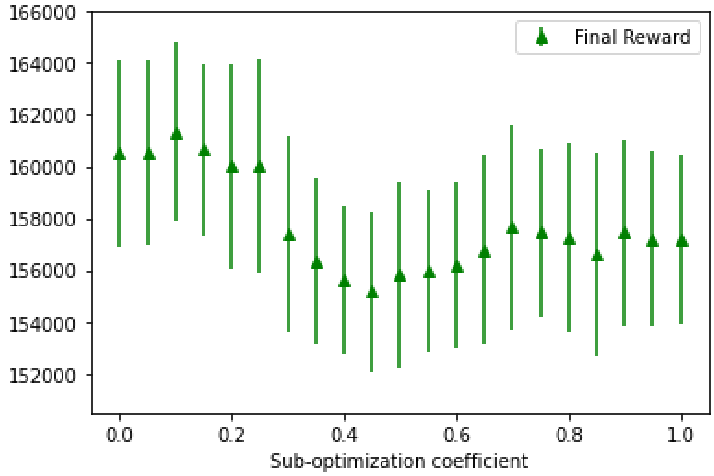



Figure A.19: Final Reward spread depending on θ when *bias_metric* = *bias_rewards*

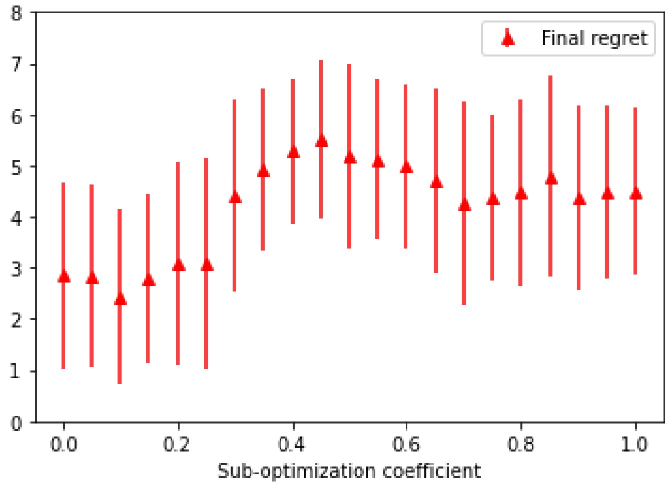



Figure A.20: Final Regret spread depending on θ when *bias_metric* = *bias_rewards*

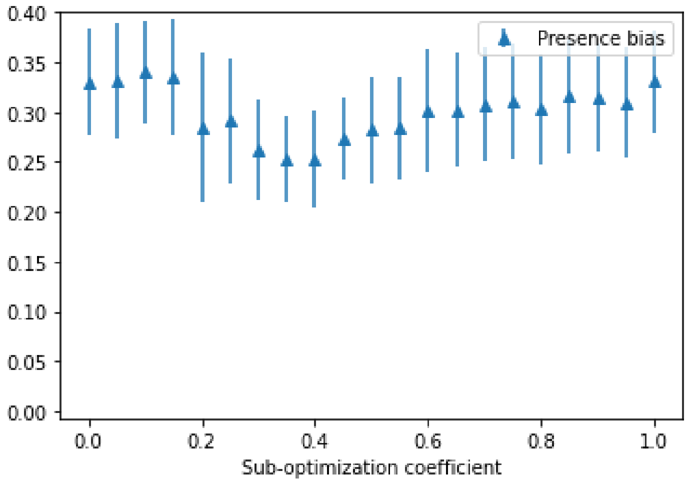



Figure A.21: Final Presence bias spread depending on θ when *bias_metric* = *bias_rewards*

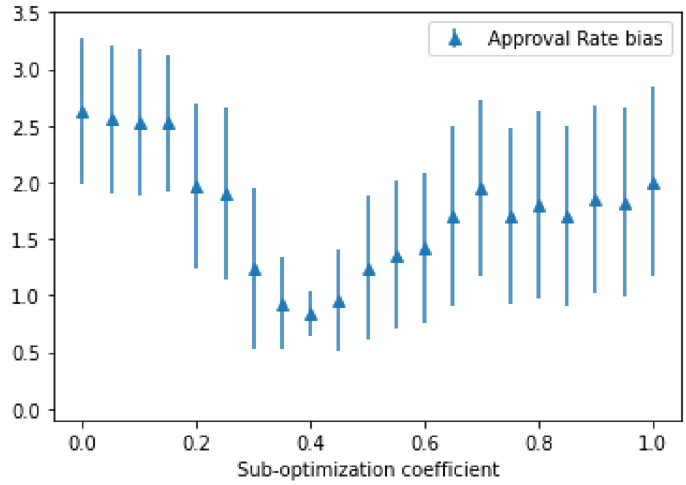



Figure A.22: Final Approval Rate bias spread depending on θ when *bias_metric* = *bias_rewards*

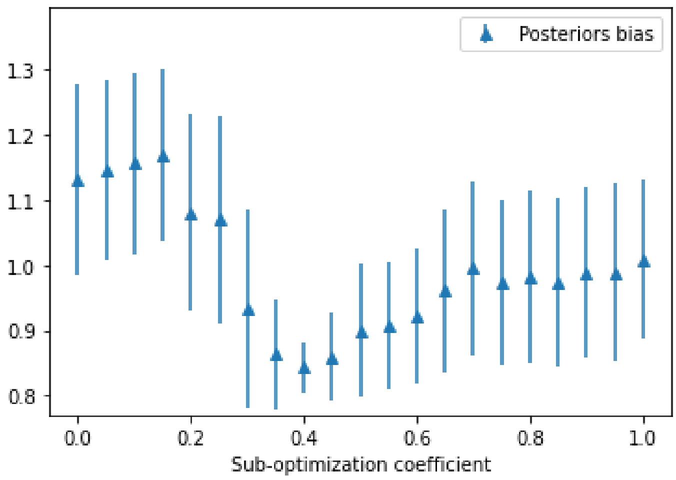



Figure A.23: Final Posteriors bias spread depending on θ when *bias_metric* = *bias_rewards*

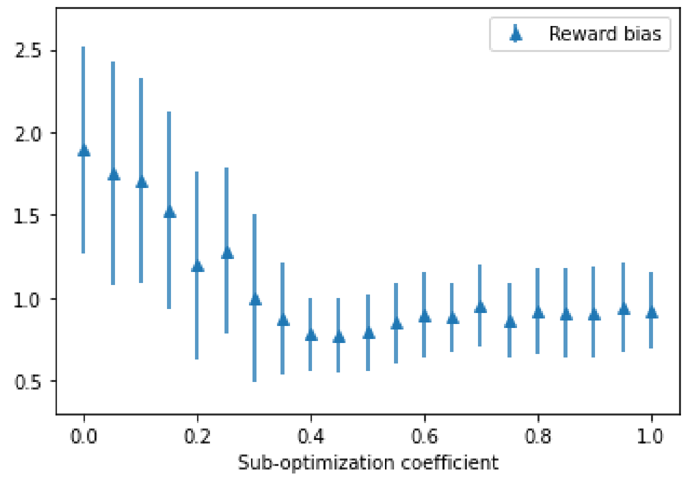



Figure A.24: Final Rewards bias spread depending on θ when *bias_metric* = *bias_rewards*

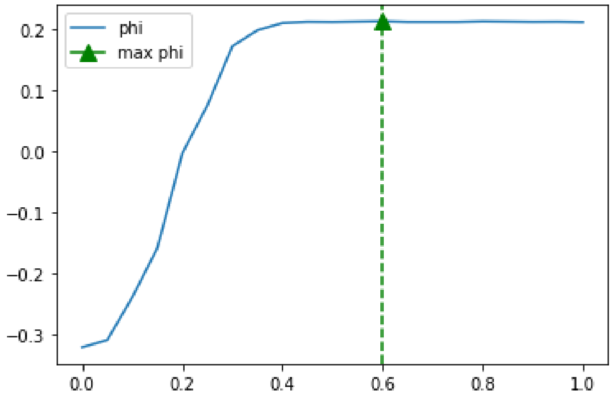



Figure 5.1: φ when *bias_metric* = *bias_presence*

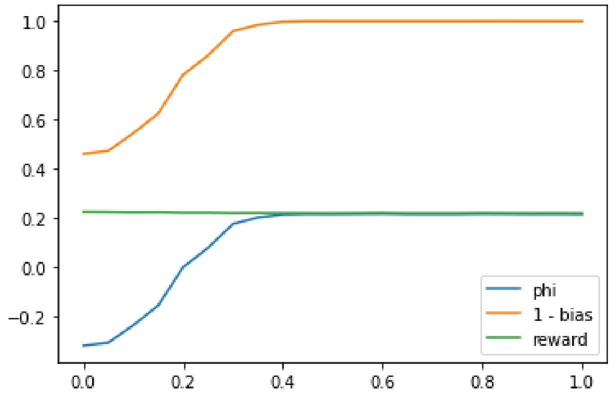



Figure 5.2: Normalized values of A.1 and A.3

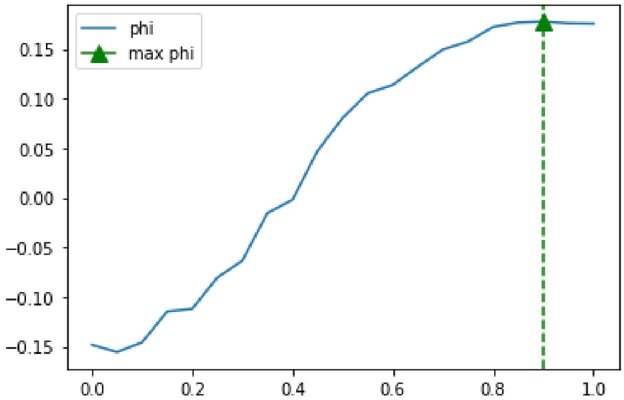



Figure 5.3: φ when *bias_metric* = *bias_rate*

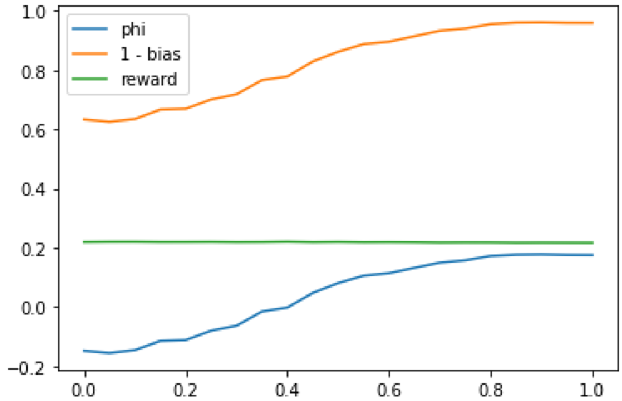



Figure 5.4: Normalized values of A.7 and A.10

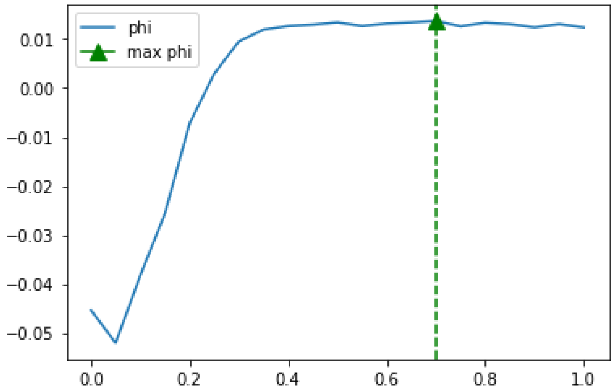



Figure 5.5: φ when *bias_metric* = *bias_posteriors*

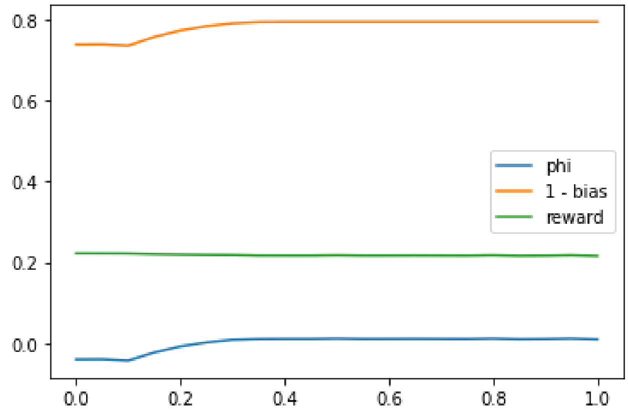



Figure 5.6: Normalized values of A.13 and A.17

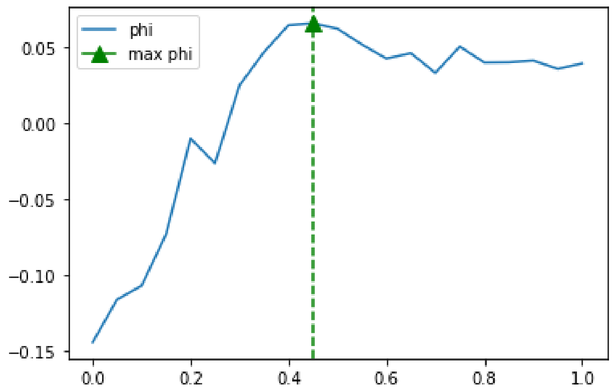



Figure 5.7: φ when *bias_metric* = *bias_rewards*

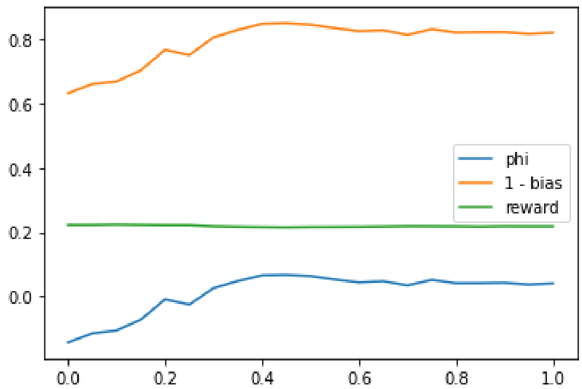



Figure 5.8: Normalized values of A.19 and A.24

# Results Evaluation

- Presence bias:

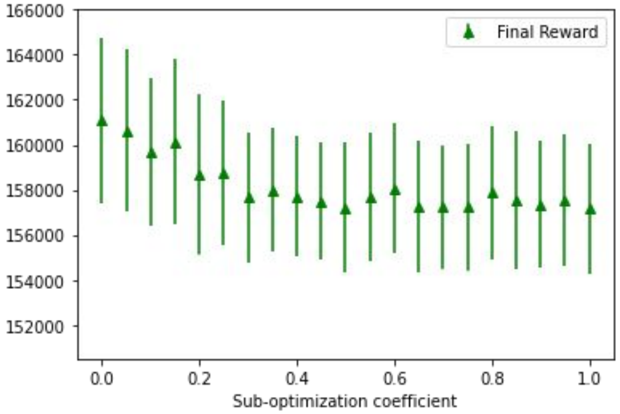



Figure A.1: Final Reward spread depending on θ when *bias_metric* = *bias_presence*

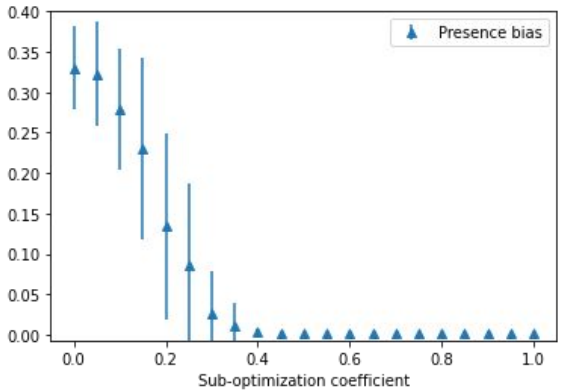



Figure A.3: Final Presence bias spread depending on θ when *bias_metric* = *bias_presence*

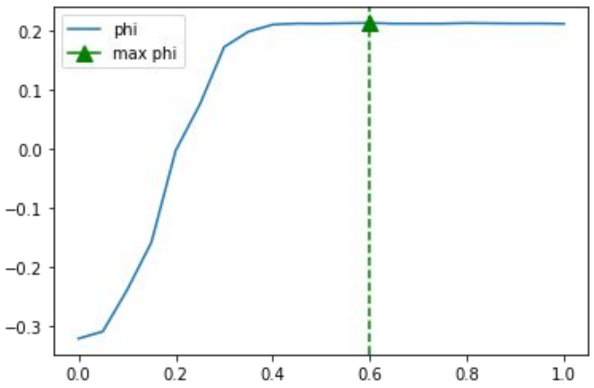



Figure 5.1: φ when *bias_metric* = *bias_presence*

# Results Evaluation

- Approval Rate bias:

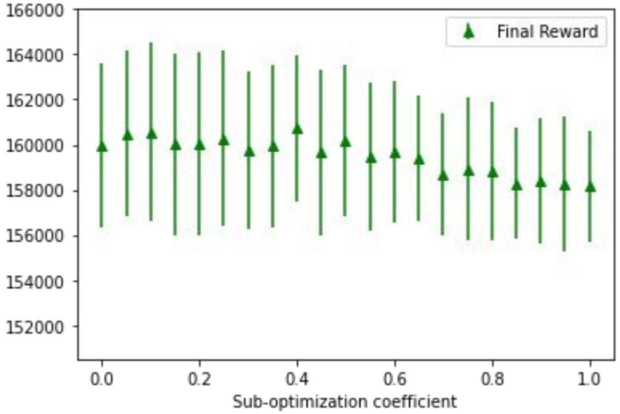



Figure A.7: Final Reward spread depending on θ when *bias_metric* = *bias_rates*

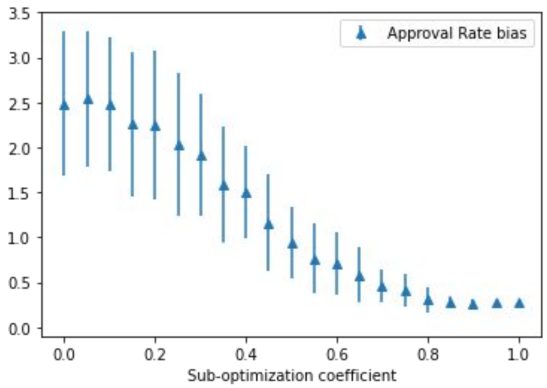



Figure A.10: Final Approval Rate bias spread depending on θ when *bias_metric* = *bias_rates*

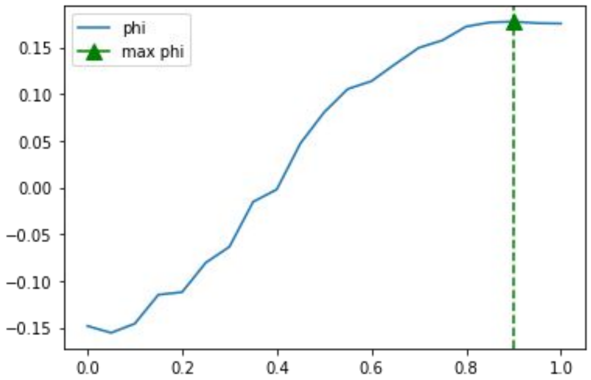



Figure 5.3: φ when *bias_metric* = *bias_rate*

# Results Evaluation

- Posteriors bias:

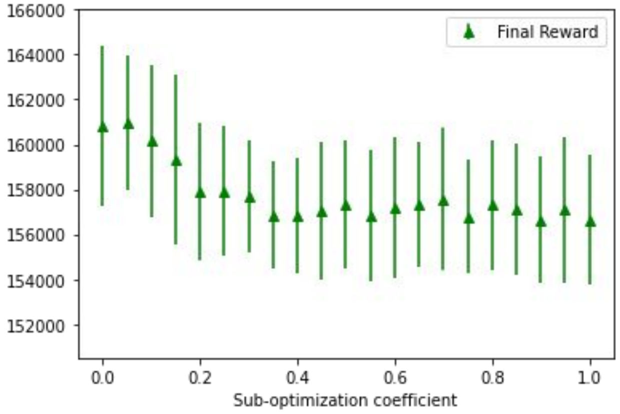



Figure A.13: Final Reward spread depending on θ when *bias_metric* = *bias_posteriors*

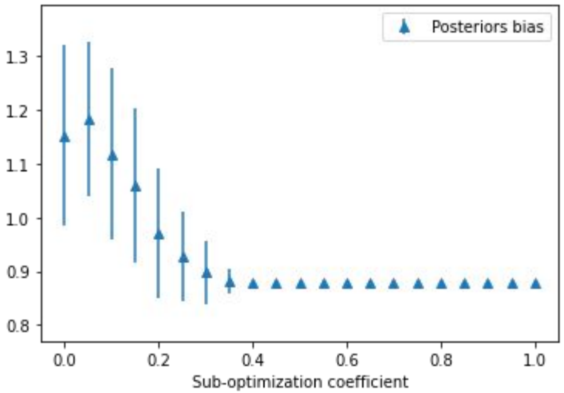



Figure A.17: Final Posteriors bias spread depending on θ when *bias_metric* = *bias_posteriors*

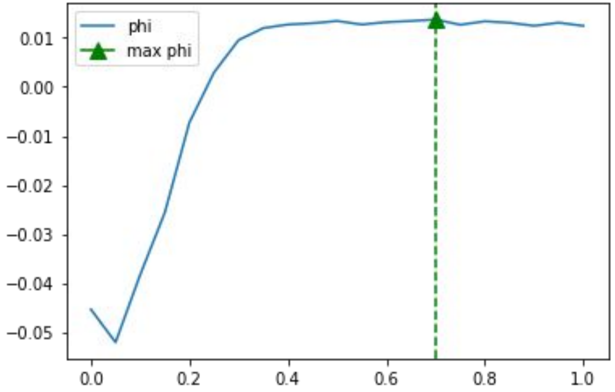



Figure 5.5: φ when *bias_metric* = *bias_posteriors*

# Results Evaluation

- Rewards bias:

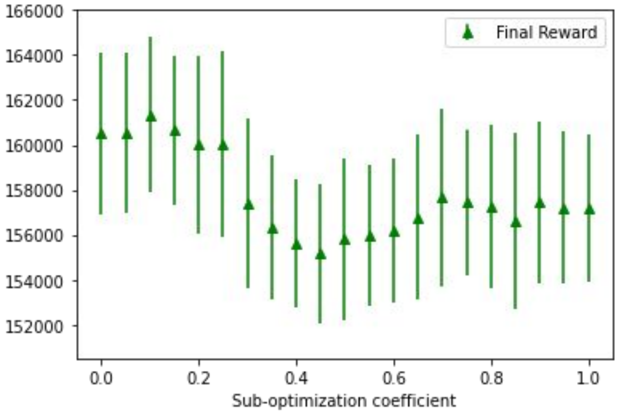



Figure A.19: Final Reward spread depending on θ when *bias_metric* = *bias_rewards*

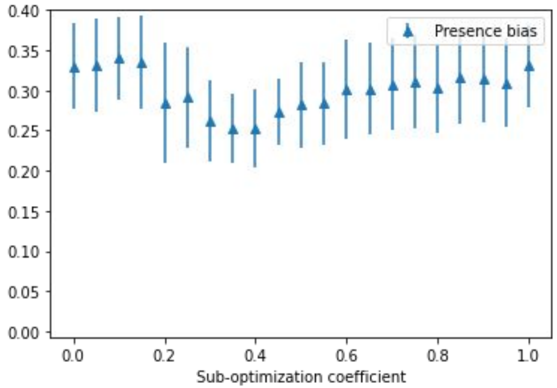



Figure A.21: Final Rewards bias spread depending on θ when *bias_metric* = *bias_rewards*

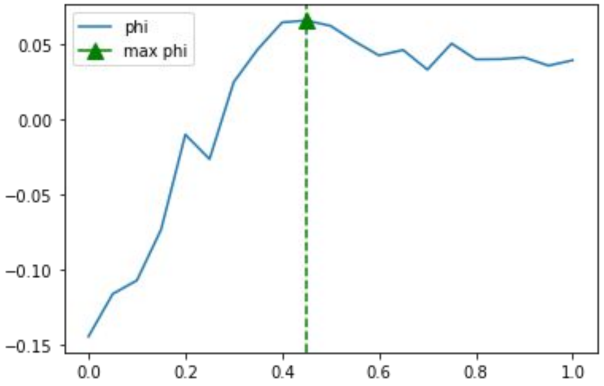


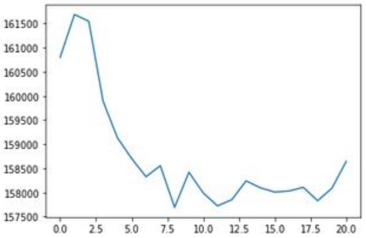

Figure 5.7: φ when *bias_metric* = *bias_rewards*

In [70]:
```python
print(yr)
print([round(i,3) for i in yr/np.sum(yr)])
print([round(i,3) for i in yr/np.linalg.norm(yr)])
print([round(i,3) for i in minmax(yr)])
```

```
[160800.0, 161686.0, 161546.0, 159890.0, 159134.0, 158702.0, 158326.0, 158554.0, 157692.0, 158418.0, 157990.0, 157722.0, 157850.0, 15824
0.0, 158096.0, 158008.0, 158030.0, 158106.0, 157828.0, 158088.0, 158644.0]
[0.048, 0.049, 0.048, 0.048, 0.048, 0.048, 0.047, 0.048, 0.047, 0.048, 0.047, 0.047, 0.047, 0.047, 0.047, 0.047, 0.047, 0.047, 0.047, 0.0
47, 0.048]
[0.221, 0.222, 0.222, 0.22, 0.219, 0.218, 0.218, 0.218, 0.217, 0.218, 0.217, 0.217, 0.217, 0.218, 0.217, 0.217, 0.217, 0.217, 0.217, 0.21
7, 0.218]
[0.778, 1.0, 0.965, 0.55, 0.361, 0.253, 0.159, 0.216, 0.0, 0.182, 0.075, 0.008, 0.04, 0.137, 0.101, 0.079, 0.085, 0.104, 0.034, 0.099, 0.
238]
```

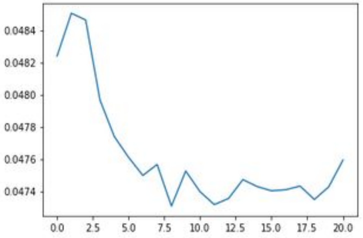
In [71]:
```python
plt.plot(yr)
```

Out[71]: [<matplotlib.lines.Line2D at 0x7fd4a5247fd0>]



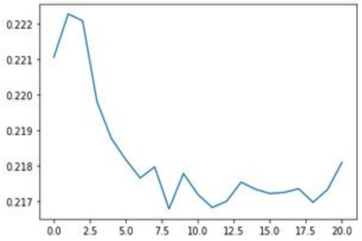

In [72]:
```python
plt.plot(yr/np.sum(yr))
```

Out[72]: [<matplotlib.lines.Line2D at 0x7fd4a4b68110>]



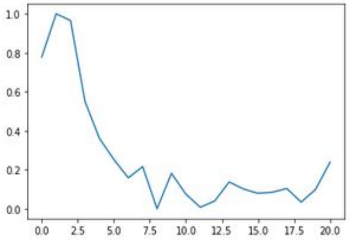

In [73]:
```python
plt.plot(yr/np.linalg.norm(yr))
```

Out[73]: [<matplotlib.lines.Line2D at 0x7fd4ad52e490>]




In [74]:
```python
plt.plot(minmax(yr))
```

Out[74]: [<matplotlib.lines.Line2D at 0x7fd4a5071090>]



# Non-Stationary Environment

## Appendix B: Application in a Non-Stationary Environment

### a) Presence Bias (θ=0.6)

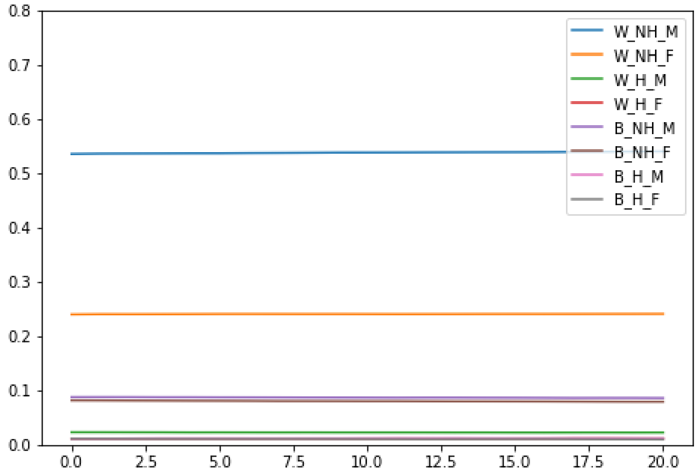



Figure B.1: Evolution of population weights every 5 epochs in a Presence-constrained environment

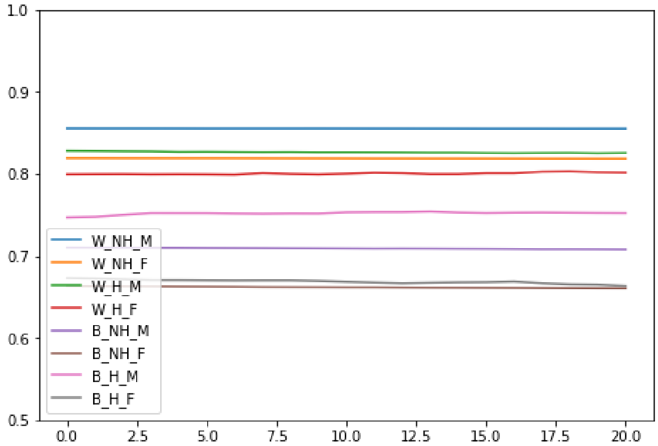



Figure B.2: Evolution of reward estimates every 5 epochs in a Presence-constrained environment

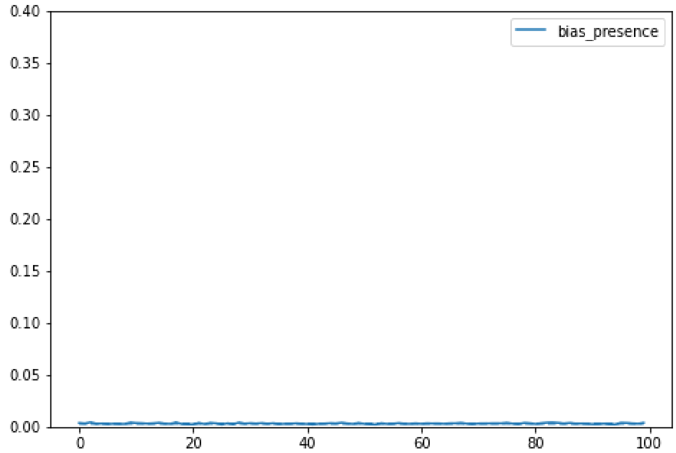



Figure B.3: Evolution of final Presence bias for every epoch in a Presence-constrained environment

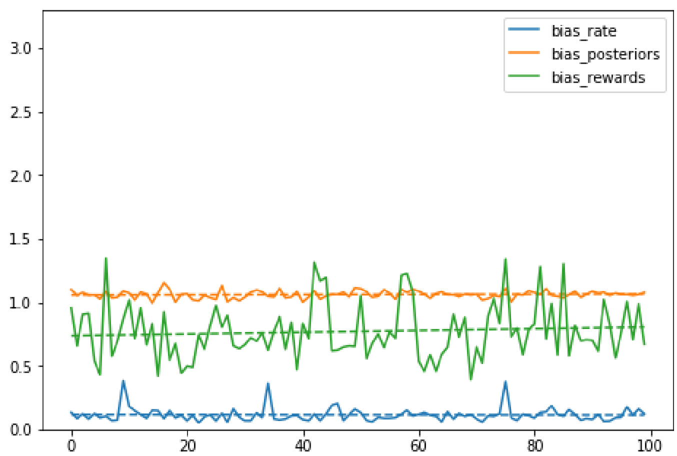



Figure B.4: Evolution of final Rate, Posteriors and Rewards biases in a Presence-constrained environment

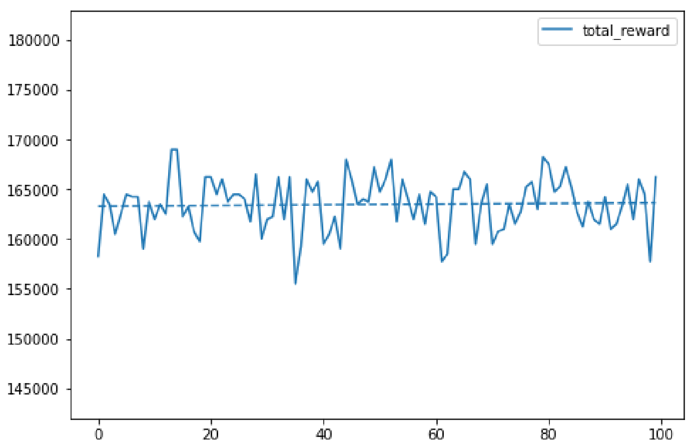



Figure B.5: Evolution of final Total Reward for every epoch in a Presence-constrained environment

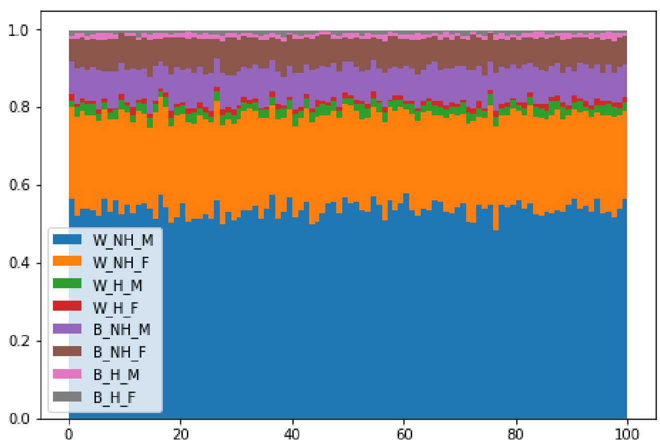



Figure B.6: Evolution of contribution to total approvals for every epoch in a Presence-constrained environment

# b) Approval Rate Bias (θ=0.9)

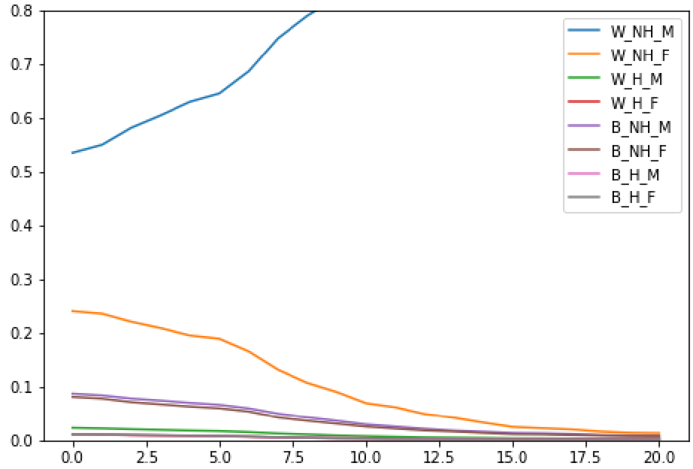



Figure B.7: Evolution of population weights every 5 epochs in an Approval Rate-constrained environment

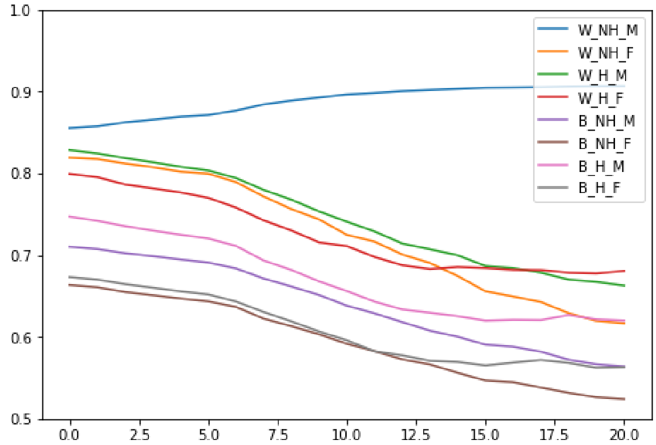



Figure B.8: Evolution of reward estimates every 5 epochs in a Approval Rate-constrained environment

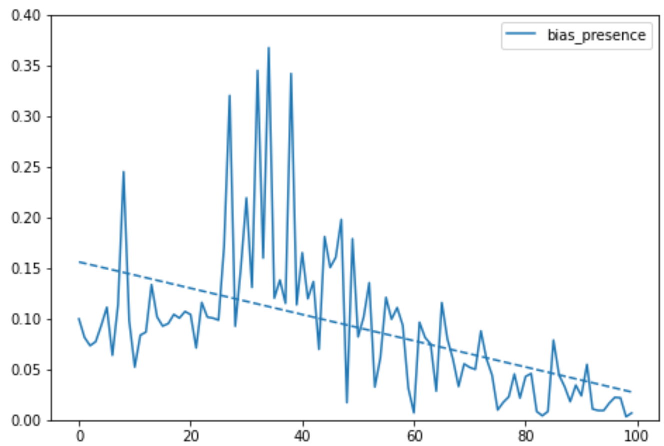



Figure B.9: Evolution of final Presence bias for every epoch in a Approval Rate-constrained environment

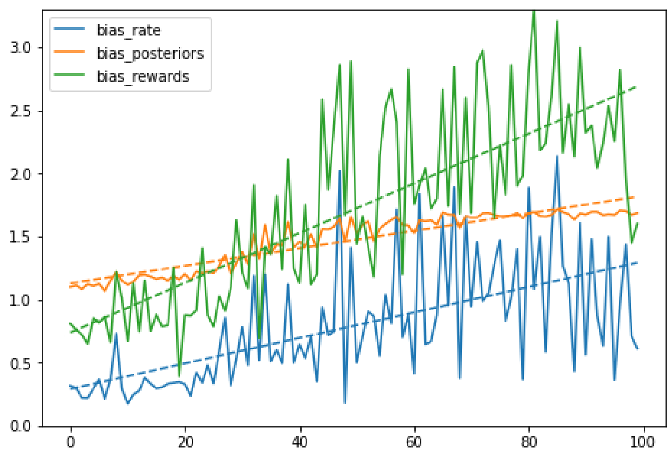



Figure B.10: Evolution of final Rate, Posteriors and Rewards biases in a Approval Rate-constrained environment

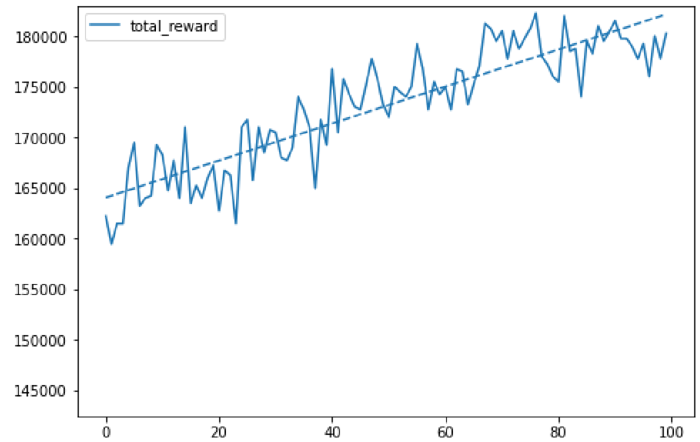



Figure B.11: Evolution of final Total Reward for every epoch in a Approval Rate-constrained environment

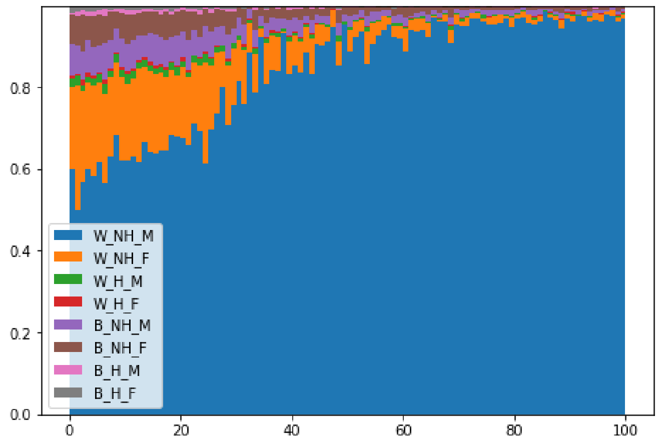



Figure B.12: Evolution of contribution to total approvals for every epoch in a Approval Rate-constrained environment

c) Posteriors Bias (θ=0.7)

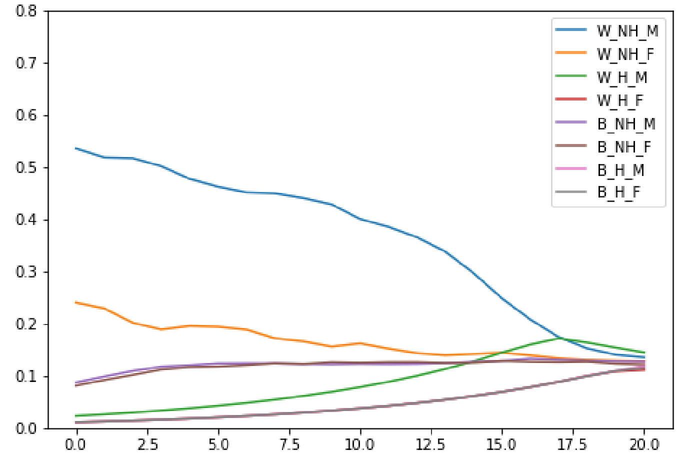



Figure B.13: Evolution of population weights every 5 epochs in a Posteriors-constrained environment

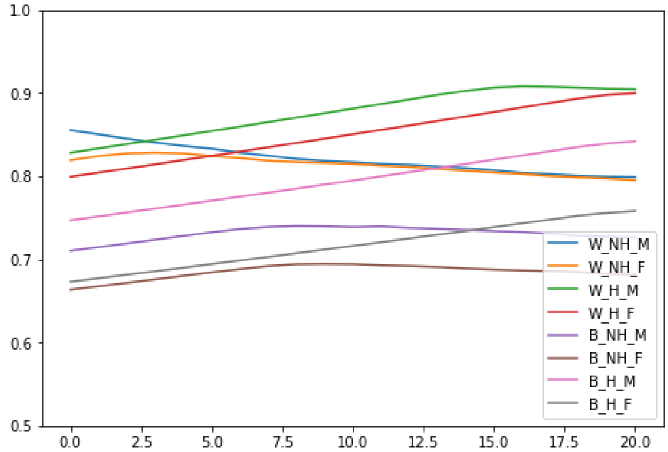



Figure B.14: Evolution of reward estimates every 5 epochs in a Posteriors-constrained environment

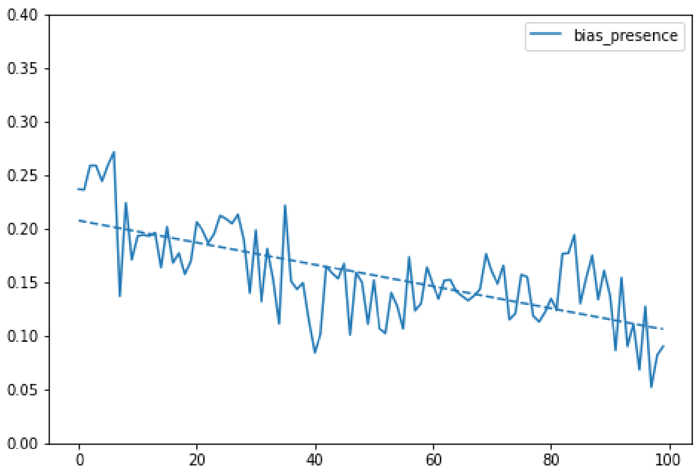



Figure B.15: Evolution of final Presence bias for every epoch in a Posteriors-constrained environment

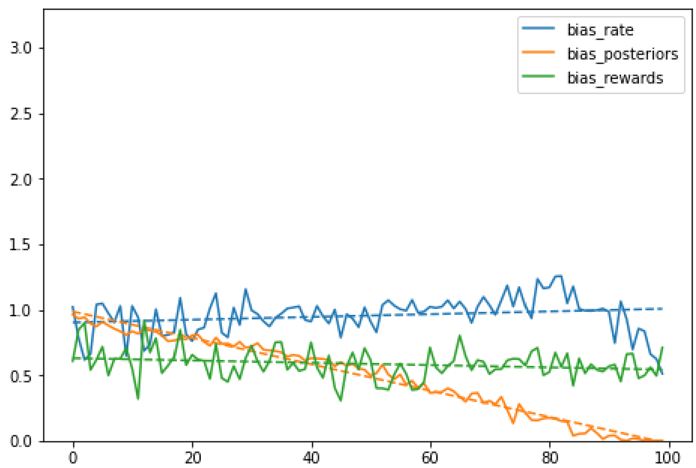



Figure B.16: Evolution of final Rate, Posteriors and Rewards biases in a Posteriors-constrained environment

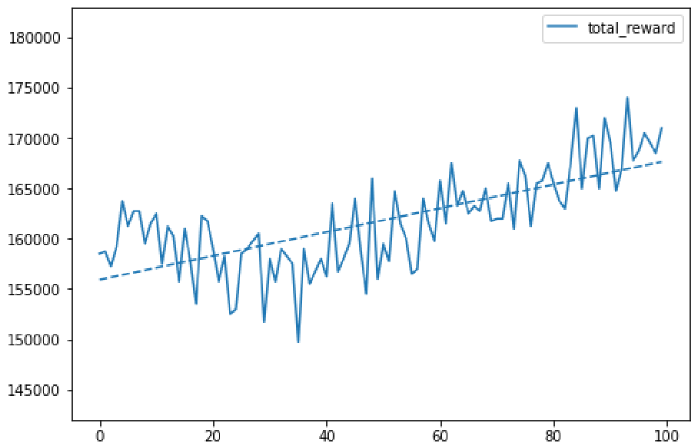



Figure B.17: Evolution of final Total Reward for every epoch in a Posteriors-constrained environment

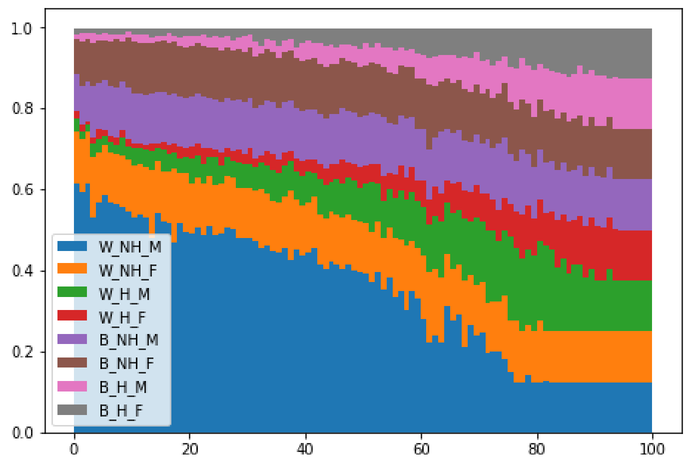



Figure B.18: Evolution of contribution to total approvals for every epoch in a Posteriors-constrained environment

d) Rewards Bias (θ=0.45)

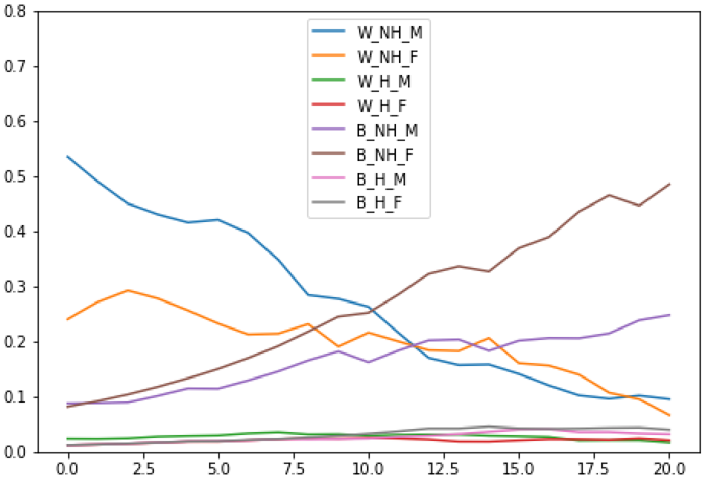



Figure B.19: Evolution of population weights every 5 epochs in a Rewards-constrained environment

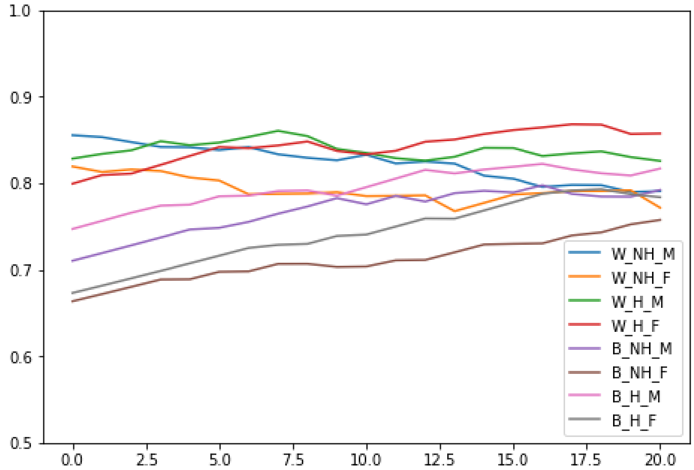



Figure B.20: Evolution of reward estimates every 5 epochs in a Rewards-constrained environment

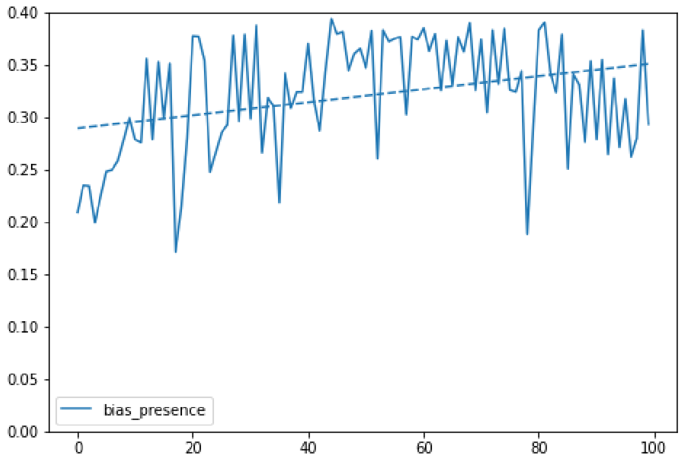



Figure B.21: Evolution of final Presence bias for every epoch in a Rewards-constrained environment

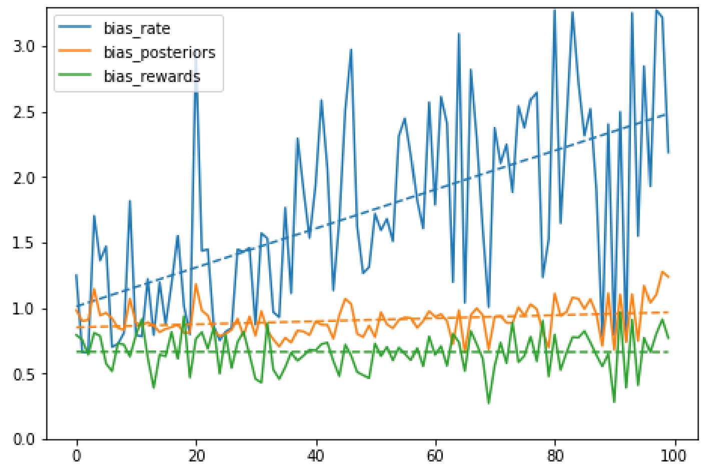



Figure B.22: Evolution of final Rate, Posteriors and Rewards biases in a Rewards-constrained environment

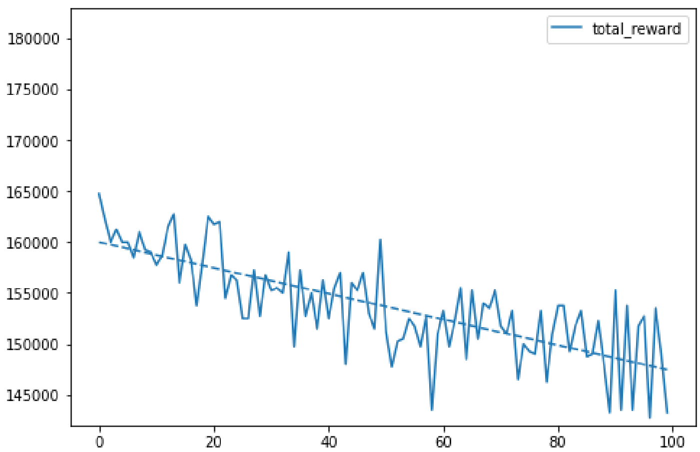



Figure B.23: Evolution of final Total Reward for every epoch in a Rewards-constrained environment

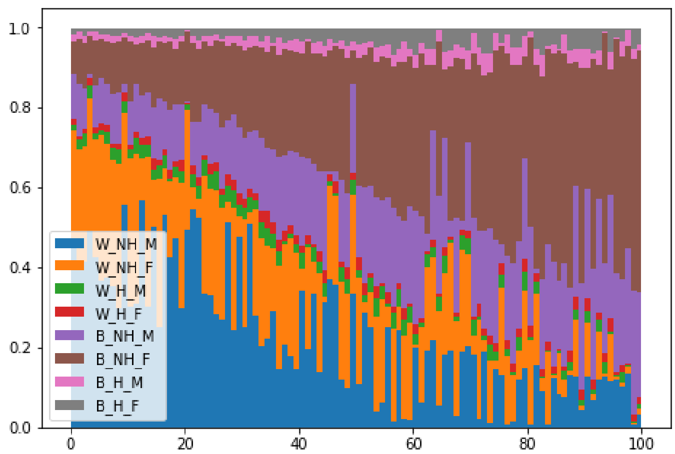



Figure B.24: Evolution of contribution to total approvals for every epoch in a Rewards-constrained environment

$$approval\_rates = \left( \left| C_a \cup C' \right| / \left| C_a \right| : a \in A \right)$$

$$bias\_rates = \sum_{a=1}^{k} \left( \left| \overline{approval\_rates} - approval\_rates_a \right| \right)$$

$$bias\_rates = \sum_{a=1}^{k} \left( \left| max(approval\_rates) - approval\_rates_a \right| \right)$$

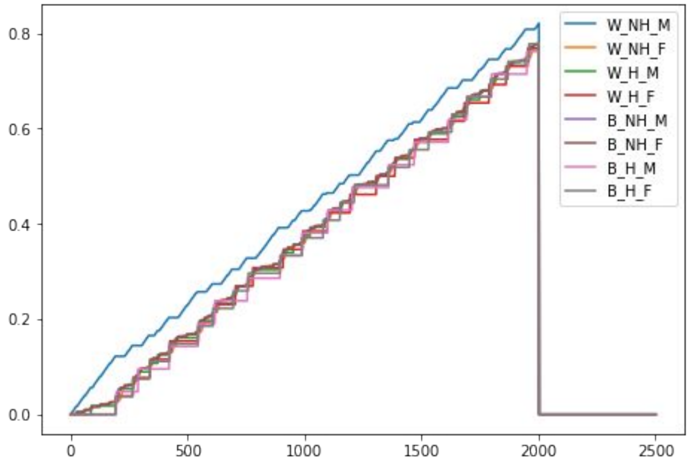



Figure 5.9: Evolution of Approval Rates within a single execution using the faulty Approval Rate constraint as-is

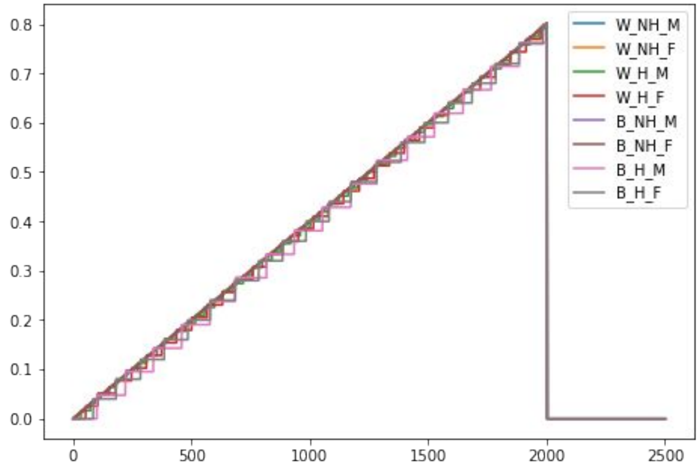



Figure 5.10: Evolution of Approval Rates within a single execution using the corrected Approval Rate constraint

$$approval\_rates = \left( \left| C_a \cup C' \right| / \left| C_a \right| : a \in A \right)$$

$$bias\_rates = \sum_{a=1}^{k} \left( \left| \overline{approval\_rates} - approval\_rates_a \right| \right)$$

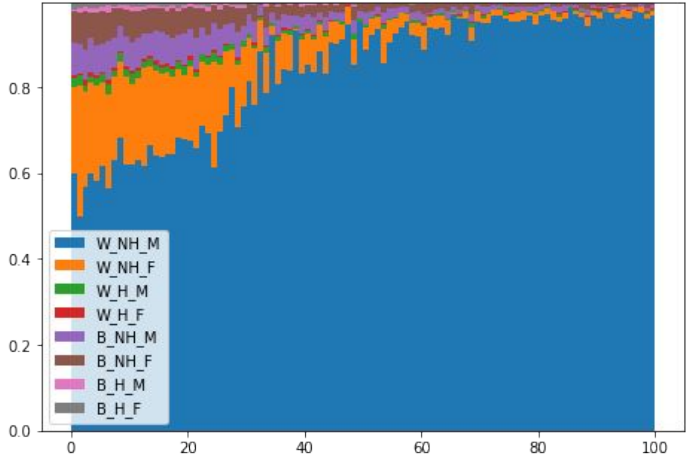



Figure B.12: Evolution of contribution to total approvals for every epoch in a Approval Rate-constrained environment

$$bias\_rates = \sum_{a=1}^{k} \left( \left| max(approval\_rates) - approval\_rates_a \right| \right)$$

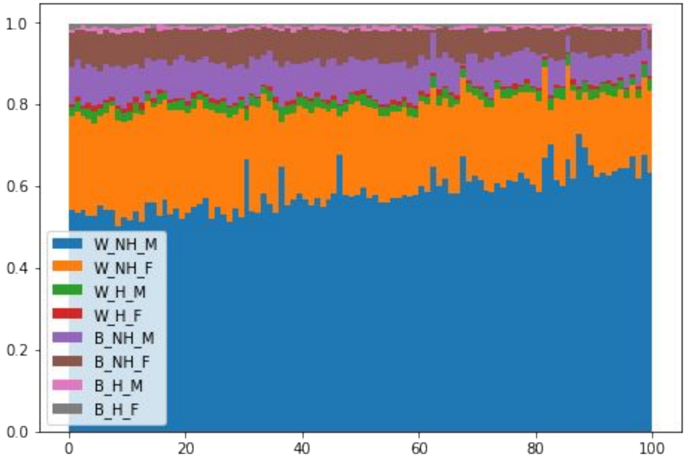



Figure 5.12: Evolution of contribution to total approvals for every epoch using the corrected Approval Rate constraint

# Conclusions and Future Work

- We can summarize what was achieved in this work as the following:
  - Studied and **identified** indications of **bias** in the **original HMDA** dataset.
  - Showcased the **harmful effect** of those underlying biases on a **live environment** when a MAB model computed with a **UCB** algorithm is applied to it sequentially over a **sustained period of time**.
  - Developed **4 bias metrics** adequate to the scope of our simulation.
  - **Examined the effect** of using these metrics in a new version of the UCB algorithm, **BC-UCB**, both in an **isolated** environment as well as in the same **live** environment that changed according to its performance.

# Conclusions and Future Work

- The present work could be expanded in the following directions:
  - Adaptation to domains of work other than mortgage lending
  - Expansion of the bias metrics used, either in definition or formalization
  - Development of more adequate $\varphi$ formula, and deliberate usage of $\beta$
  - More thorough transformation processes from real data to simulated one
  - Refinement of the environment's modelization
  - Refinement of the (MAB) problem formulation

# IMAGE REFERENCES


1. Ledford, Heidi. “Millions of Black People Affected by Racial Bias in Health-Care Algorithms.” *Nature*, vol. 574, no. 7780, 2019, pp. 608–609, doi:10.1038/d41586-019-03228-6.

2. Dastin, Jeffrey. “Amazon Scraps Secret AI Recruiting Tool That Showed Bias against Women.” *Reuters*, 2018, https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G.

3. Angwin, Julia, et al. “Machine Bias.” *ProPublica*, 2016, https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

# Thank you for your attention

**Quim De Las Heras Molins (u160402)**

quim.delasheras01@estudiant.upf.edu

Juliol 2022

Treball de Fi de Grau

Enginyeria Informàtica

Universitat Pompeu Fabra