### The Multi-Armed Bandit Problem

- **K arms**, each characterized by a distribution \( \mu_i \) of mean \( \mu_i \) and variance \( \sigma^2_i \).
- Given an arbitrary sequence of \( n \) arms, \( X_n = (I_1, I_2, \ldots, I_n) \) with \( I_i \sim \text{i.i.d.} \) and \( I_i \in [1, K] \).
- Given a weight parameter \( w \in (0, 1) \), \( f(w; \lambda) = w \lambda - (1 - w) \).

### The Forcing Balance Algorithm

#### Intuition
- **Forcing** = accurate \( \hat{\mu} \) and \( \hat{\lambda} \).
- **Tracking** = accurate \( \lambda \).
- Vanishing forcing \( \sqrt{\text{Rescaled regret}} \) when \( \lambda \to \lambda^* \).

### Theoretical Guarantees

#### Lemma 3.
For any allocation \( \lambda \in D_K \) and any \( c \in [K] \), \( |\lambda - \lambda^*| \leq \frac{\sqrt{2}}{n} \sum_{i=1}^{K} \lambda_i \max_i \lambda_i \).

#### Lemma 4.
For any allocation \( \lambda \in D_K \), \( |f(\lambda^*) - f(\lambda)| \leq \frac{\sqrt{2}}{n} \sum_{i=1}^{K} \lambda_i \max_i \lambda_i \).

#### Lemma 5.
Let \( \hat{\lambda} \) be such that \( |\hat{\lambda} - \lambda| \leq \frac{\epsilon}{2 \lambda^*} \). Then for any allocation \( \lambda \in D_K \), \( f(\lambda^*) - f(\hat{\lambda}) \leq \frac{\sqrt{2}}{n} \lambda^* \).

#### Assumption 1.
Let \( \lambda_{\text{min}} = \min_i \lambda_i \). We assume that \( \lambda_{\text{min}} \geq \lambda_{\text{min}}^* \) (i.e., \( \lambda^* \in D_K \)).

#### Theorem.
Under Assumptions 1, Forcing Balance with a parameter \( \eta \leq 21 \) and a simplex \( \Delta_K \) restricted to \( \hat{\lambda} \) succeeds. Alerts to \( R_n(\hat{\lambda}) \) with

### Educational Experiment

#### The setting.
- \( K = 61 \) arms, \( w = 0.9 \) (i.e., favor rewards over errors).
- Parameter \( q = 1, \lambda_{\text{min}} = 0 \). Arm 4 has the largest variance and it should be pulled the most to minimize \( c \).
- Arm 5 has the largest reward and it should be pulled the most to maximize \( \mu \).
- The optimal allocation \( \lambda^* \) is unbalanced towards arm 5 and a bit on arm 6.

#### The results.
- Varying \( w \) from 0.01 to 0.09.
- For \( w = 0 \), the minimization of \( c \) induces an optimal allocation with \( \lambda_{\text{min}} = 0.12 \) and \( \lambda_{\text{max}} = 0.20 \).
- For \( w = 0.95 \), the maximization of \( \mu \) induces an optimal allocation with \( \lambda_{\text{min}}^* = 0.0848 \) and \( \lambda_{\text{max}}^* = 0.9268 \).
- **Forcing Balance** is more effective in approaching the performance of \( \lambda^* \) for small values of \( w \). For \( w = 0 \), \( \lambda_{\text{min}} = 0.097 \), while for \( w = 0.95 \), \( \lambda_{\text{min}}^* = 0.004 \).

#### References


#### Notes
- **Problem**.
- **Control-theoretic analysis**.
- **Optimal allocation with**.